

Simple Storage Service -S3

S3 use cases

- Backup and storage
- Disaster Recovery
- Archive
- Hybrid Cloud storage
- Application hosting
- Media hosting
- Data lakes & big data analytics
- Software delivery
- Static website

S3 -Buckets

- Amazon S3 allows people to store objects (files) in "buckets" (directories)
- Buckets must have a globally unique name (across all regions all accounts)
- Buckets are defined at the region level
- S3 looks like a global service but buckets are created in a region
- Naming convention
 - No uppercase, No underscore
 - 3-63 characters long
 - Not an IP
 - Must start with lowercase letter or number
 - Must NOT start with the prefix xn-
 - Must NOT end with the suffix -s3alias



Availability: up time 90% means in 10 days it will be down 1 day. 100% availability means it will never down.

Durability: data persistent 90% durability means if we upload 10 files 9 files will be keep safe 1 file will be damage by any reason.

S3 -Objects

- Objects (files) have a Key
- The [key](#) is the FULL path:
 - s3://my-bucket/[my_file.txt](#)
 - s3://my-bucket/[my_folder1/another_folder/my_file.txt](#)
- The key is composed of [prefix](#) + [object name](#)
 - s3://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 ("/")



S3 – Objects (cont.)

- Object values are the content of the body:
 - Max. Object Size is 5TB (5000GB)
 - 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)

S3- Security

- User-Based
 - IAM Policies – which API calls should be allowed for a specific user from IAM
- Resource-Based
 - Bucket Policies – bucket wide rules from the S3 console - allows cross account
 - Object Access Control List (ACL) – finer grain (can be disabled)
 - Bucket Access Control List (ACL) – less common (can be disabled)
- Note: an IAM principal can access an S3 object if
 - The user IAM permissions ALLOW it OR the resource policy ALLOWS it
 - AND there's no explicit DENY
- Encryption: encrypt objects in Amazon S3 using encryption keys

S3 Bucket Policies

- JSON based policies
 - Resources: buckets and objects
 - Effect: Allow / Deny
 - Actions: Set of API to Allow or 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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket/*"
      ]
    }
  ]
}
```

Example: Public Access – Use Bucket Policy



Example: User Access to S3 – IAM permission



Example: EC2 instance access – Use IAM role



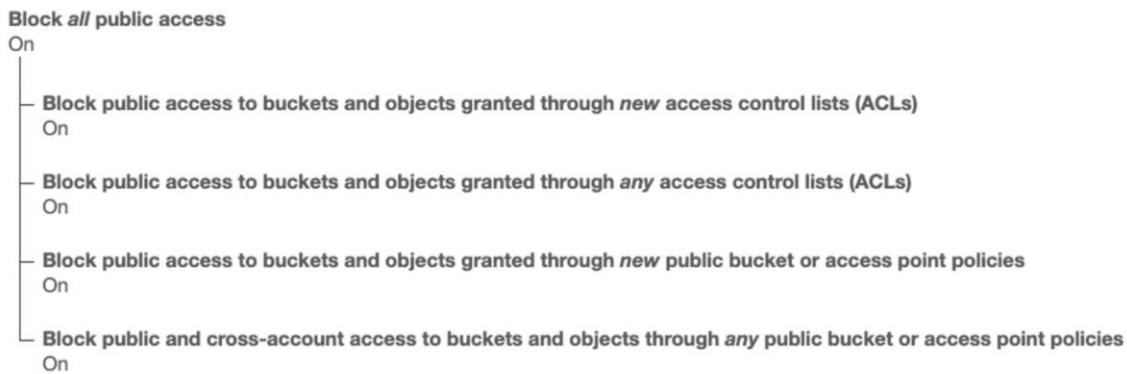
Cross Account Access – Use bucket policy



Cross Account access Using bucket policy



Bucket setting for Block public Access



- These settings were created to prevent company data leaks
- If you know your bucket should never be public, leave these on
- Can be set at the account level

Lab:1

Create a bucket

upload a image to the bucket

check

Create a bucket

Upload a image to the bucket

Change the permissions

Amazon S3 > Buckets > rajiv2021

rajiv2021 [Info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory [\[?\]](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more \[?\]](#)

[Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	cat.jpeg	jpeg	June 13, 2023, 12:04:20 (UTC+06:00)	4.7 KB	Standard

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 \[?\]](#)

[Edit](#)

Block **all** public access

Off

► Individual Block Public Access settings for this bucket

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](#)

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](#)

[Edit](#)

Block all public access

Off

► Individual Block Public Access settings for this bucket

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

[Edit](#) [Delete](#)

No policy to display.

[Copy](#)

Amazon S3 > Buckets > rajiv2021 > Edit bucket policy

Edit bucket policy [Info](#)

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

[Policy examples](#) [Policy generator](#)

Bucket ARN

arn:aws:s3:::rajiv2021

Policy

1 | [Edit statement](#)

Select a statement

Select an existing statement in the policy or add a new statement.

[+ Add new statement](#)

AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products information about creating policies, see [key concepts in Using AWS Identity and Access Management](#). Here are [sample policies](#).

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an [IAM Policy](#), an [S3 Bucket Policy](#), an [SNS Policy](#), and an [SQS Queue Policy](#).

Select Type of Policy S3 Bucket Policy ▼

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See [a description of elements](#) that you can use in statements.

Effect Allow Deny

Principal Use a comma to separate multiple values.

AWS Service All Services ('*')

Actions All Actions ('*')

Amazon Resource Name (ARN)
BucketName}://\${KeyName}.

Actions
<input checked="" type="checkbox"/> GetObject <input type="checkbox"/> GetObjectAcl <input type="checkbox"/> GetObjectAttributes <input type="checkbox"/> GetObjectLegalHold <input type="checkbox"/> GetObjectRetention <input type="checkbox"/> GetObjectTagging <input type="checkbox"/> GetObjectTorrent <input type="checkbox"/> GetObjectVersion

AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products. For more information about creating policies, see [key concepts in Using AWS Identity and Access Management](#). Here are [sample policies](#).

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an [IAM Policy](#), an [S3 Bucket Policy](#), an [SI Policy](#), and an [SQS Queue Policy](#).

Select Type of Policy S3 Bucket Policy ▾

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See [a description of elements](#) that you can use in statements.

Effect Allow Deny

Principal

Use a comma to separate multiple values.

AWS Service All Services (*)

Actions All Actions (*)

Amazon Resource Name (ARN)

ARN should follow the following format: arn:aws:s3:::\${BucketName}/\${KeyName}.
Use a comma to separate multiple values.

Add Conditions (Optional)

Add Statement

Effect Allow Deny

Principal

Use a comma to separate multiple values.

AWS Service All Services (*)

Actions All Actions (*)

Amazon Resource Name (ARN)

ARN should follow the following format: arn:aws:s3:::\${BucketName}/\${KeyName}.
Use a comma to separate multiple values.

Add Conditions (Optional)

Add Statement

1

You added the following statements. Click the button below to Generate a policy.

Principal(s)	Effect	Action	Resource	Conditions
• *	Allow	• s3:GetObject	arn:aws:s3:::rajiv2021/*	None

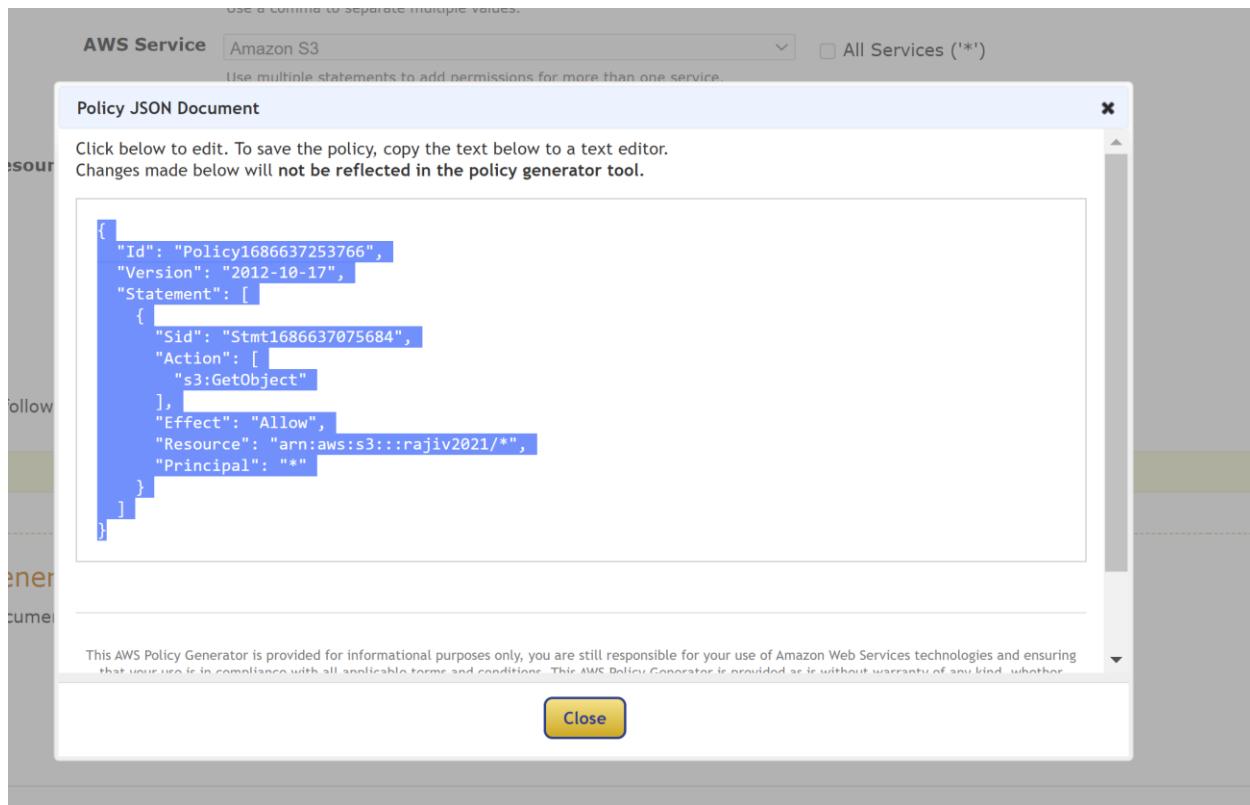
Step 3: Generate Policy

A *policy* is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements.

Generate Policy

Start Over

2



Amazon S3 > Buckets > rajiv2021 > Edit bucket policy

Edit bucket policy Info

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

[Policy examples](#)

[Policy generator](#)

Bucket ARN

arn:aws:s3:::rajiv2021

Policy

```
1  {  
2   "Id": "Policy1686637282654",  
3   "Version": "2012-10-17",  
4   "Statement": [  
5     {  
6       "Sid": "Stmt1686637075684",  
7       "Action": [  
8         "s3:GetObject"  
9       ],  
10      "Effect": "Allow",  
11      "Resource": "arn:aws:s3:::rajiv2021/*",  
12      "Principal": "*"  
13    }  
14  ]  
15 }
```

Edit statement

Select a statement

Select an existing statement in the list or [add a new statement](#).

[+ Add new statement](#)

13 }
 14]
 15 |

+ Add new statement

JSON Ln 15, Col 1

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0

Preview external access

Cancel **Save changes**

Now we can browse the object as a public from the following URL

<https://rajiv2021.s3.amazonaws.com/cat.jpeg>

Amazon S3 – Static Website Hosting

- S3 can host static websites and have them accessible on the Internet
- The website URL will be (depending on the region)
 - <http://bucket-name.s3-website-us-west-2.amazonaws.com>
 - OR
 - <http://bucket-name.s3-website.amazonaws.com>
- If you get a 403 Forbidden error; make sure the bucket policy allows public reads!



Lab: Create a static website

Objects Properties Permissions Metrics Management Access Points

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
cat.jpeg	jpeg	June 13, 2023, 12:04:20 (UTC+06:00)	4.7 KB	Standard

Now scroll down the page and you get the option

The screenshot shows the 'Edit static website hosting' configuration page for an S3 bucket named 'rajiv2021'. At the top, there's a header with 'Static website hosting' and a red-bordered 'Edit' button. Below the header, it says 'Static website hosting' is 'Disabled'. The main section is titled 'Edit static website hosting' with a 'Info' link. It contains the following fields:

- Static website hosting**: A note to use the bucket for a website or redirect requests, with a 'Learn more' link.
- Static website hosting** status: Radio buttons for 'Disable' and 'Enable', with 'Enable' selected and highlighted by a red box.
- Hosting type**: Radio buttons for 'Host a static website' (selected) and 'Redirect requests for an object', each with a 'Learn more' link.
- Index document**: A note to specify the home or default page, with a 'Using Amazon S3 Block Public Access' link.
- Error document - optional**: A note to specify the page returned for errors, with a 'Using Amazon S3 Block Public Access' link.
- Redirection rules - optional**: A note to automatically redirect webpage requests for specific content, with a 'Learn more' link.

At the bottom right, there are 'Cancel' and 'Save changes' buttons, with 'Save changes' also highlighted by a red box.

Now go to the object and upload the index.html file

Amazon S3 > Buckets > rajiv2021 > 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 (1 Total, 79.0 B)		Remove	Add files	Add folder
All files and folders in this table will be uploaded.				
<input type="text"/> Find by name < 1 >				
<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	index.html	-	text/html	79.0 B

Destination

Destination
s3://rajiv2021

▶ **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](#)

now go to the properties and scroll down and you will get the s3 website url

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting
Enabled

Hosting type
Bucket hosting

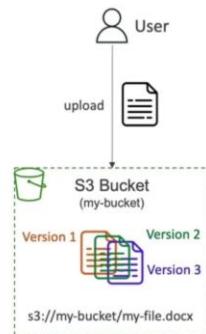
Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#)

<http://rajiv2021.s3-website-us-east-1.amazonaws.com>

now go to the browser and pest the URL and you will get the website.

Amazon S3 - Versioning

- You can version your files in Amazon S3
- It is enabled at the bucket level
- Same key overwrite will change 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
- Notes:
 - Any file that is not versioned prior to enabling versioning will have version “null”
 - Suspending versioning does not delete the previous versions



Lab: how to enable versioning

First enable the versioning

Screenshot of the AWS S3 Bucket Properties page showing the 'Bucket overview' section. The 'Properties' tab is selected. Key details shown include:

- AWS Region: US East (N. Virginia) us-east-1
- Amazon Resource Name (ARN): arn:aws:s3:::rajiv2021
- Creation date: June 13, 2023, 11:59:51 (UTC+06:00)

The 'Bucket Versioning' section shows that versioning is disabled. There is a link to 'Edit' settings.

Screenshot of the 'Edit Bucket Versioning' dialog box. The 'Bucket Versioning' section contains the following information:

Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Suspend
This suspends the creation of object versions for all operations but preserves any existing object versions.

Enable

Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)

Disabled

Buttons at the bottom: **Cancel** and **Save changes**

Now edit the index file and upload it and then check.

The screenshot shows the AWS S3 Objects page with three items:

Name	Type	Version ID	Last modified	Size	Storage class
cat.jpeg	jpeg	null	June 15, 2023, 12:04:20 (UTC+06:00)	4.7 KB	Standard
index.html	html	hrNCBBAIElkAYvc1d3N1HPintvhM	June 13, 2023, 13:11:32 (UTC+06:00)	145.0 B	Standard
index.html	html	null	June 13, 2023, 13:09:56 (UTC+06:00)	139.0 B	Standard

NB: Version ID null means its uploaded before enable versioning.

Now we see the current updated page now it we want back to previous one page then delete the current one version and then brows it

The screenshot shows the AWS S3 Objects page with three items:

Name	Type	Version ID	Last modified	Size	Storage class
cat.jpeg	jpeg	null	June 13, 2023, 12:04:20 (UTC+06:00)	4.7 KB	Standard
index.html	html	hrNCBBAIElkAYvc1d3N1HPintvhM	June 13, 2023, 13:11:32 (UTC+06:00)	145.0 B	Standard
index.html	html	null	June 13, 2023, 13:09:56 (UTC+06:00)	139.0 B	Standard

Now if we

browse then we get the previous version file

Now we delete any image example cat.jpeg

The screenshot shows the AWS S3 Objects page with two items:

Name	Type	Last-modified	Size
cat.jpeg	jpeg	November 3, 2023, 20:38:39 (UTC+06:00)	1.2 MB
cat2.jpeg	jpeg	November 8, 2023, 10:23:45 (UTC+06:00)	39.7 KB

after delete the cat.jpeg image then we can see the page is not showing.

now back to image again we need to delete the delete marker then we can see the image is showing again

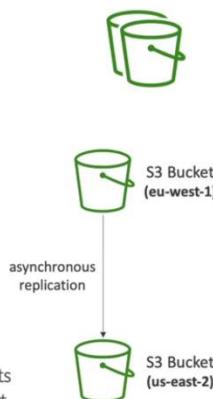
The screenshot shows the AWS S3 Objects page with three items:

Name	Type	Version ID	Last modified	Size
cat.jpeg	jpeg	Delete marker	EZnI0JT_Hi3Lb7.StDhgZOUKJPNFuEWj	November 8, 2023, 10:58:33 (UTC+06:00)
cat.jpeg	jpeg	null		November 3, 2023, 20:38:39 (UTC+06:00)
cat2.jpeg	jpeg	null		November 8, 2023, 10:23:45 (UTC+06:00)

NB: Force refresh shift+R+reload or refresh the page

Amazon S3 - Replication (CRR and SRR)

- Must enable Versioning in source and destination buckets
- Cross-Region Replication (CRR)
- Same-Region Replication (SRR)
- Buckets can be in different AWS accounts
- Copying is asynchronous
- Must give proper IAM permissions to S3
- Use cases:
 - CRR – compliance, lower latency access, replication across accounts
 - SRR – log aggregation, live replication between production and test accounts



Amazon S3 – Replication (Notes)

- After you enable Replication, only new objects are replicated
- Optionally, you can replicate existing objects using S3 Batch Replication
 - Replicates existing objects and objects that failed replication
- For DELETE operations
 - Can replicate delete markers from source to target (optional setting)
 - Deletions with a version ID are not replicated (to avoid malicious deletes)
- There is no “chaining” of replication
 - If bucket 1 has replication into bucket 2, which has replication into bucket 3
 - Then objects created in bucket 1 are not replicated to bucket 3

Lab- Replication

Create 2 buckets with versioning.

- 1.rajiv2023-original
- 2.rajiv-2023-replica

Now upload a image to rajiv-2023-orinal

Now enable the replication in rajiv2023-original bucket

The screenshot shows the AWS S3 console for the 'rajiv2023-orginal' bucket. The 'Management' tab is selected. In the 'Objects' section, there is one item: 'cat.jpeg' (Type: jpeg, Last modified: June 15, 2023, 19:44:45 (UTC+06:00), Size: 4.7 KB, Storage class: Standard). The 'Actions' menu is open, and the 'Replicate' option is highlighted with a red box. Other options visible in the Actions menu include Copy S3 URI, Copy URL, Download, Open, Delete, Create folder, and Upload.

Replication rules (0)

Use replication rules to define options you want Amazon S3 to apply during replication such as server-side encryption, replica ownership, transitioning replicas to another storage class, and more. [Learn more](#)

View details Edit rule Delete Actions ▾

Replication rule name	Status	Destination bucket	Destination Region	Priority	Scope	Storage class	Replica owner	Replication Time Control	KMS-encrypted
No replication rules									

You don't have any rules in the replication configuration.

Replication rule name

rajiv-2023-original-rule-1

Up to 255 characters. In order to be able to use CloudWatch metrics to monitor the progress of your replication rule, the replication rule name must only contain English characters.

Status

Choose whether the rule will be enabled or disabled when created.

- Enabled
 Disabled

Priority

The priority value resolves conflicts that occur when an object is eligible for replication under multiple rules to the same destination. The rule is added to the configuration at the highest priority and the priority can be changed on the replication rules table.

0

Source bucket

Source bucket name

rajiv2023-orginal

Source Region

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

Choose a rule scope

- Limit the scope of this rule using one or more filters
 Apply to all objects in the bucket

Destination

Destination

You can replicate objects across buckets in different AWS Regions (Cross-Region Replication) or you can replicate objects across buckets in the same AWS Region (Same-Region Replication). You can also specify a different bucket for each rule in the configuration. [Learn more](#) or see [Amazon S3 pricing](#)

- Choose a bucket in this account
 Specify a bucket in another account

Bucket name

Choose the bucket that will receive replicated objects.

rajiv2023-replica

Destination Region

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

IAM role

- Choose from existing IAM roles
- Enter IAM role ARN

IAM role

Create new role



View

Encryption

Server-side encryption protects data at rest.

- Replicate objects encrypted with AWS KMS

You can replicate objects that are encrypted with AWS Key Management Service (AWS KMS) keys.

Destination storage class

Amazon S3 offers a range of storage classes designed for different use cases. [Learn more](#) or see [Amazon S3 pricing](#)

- Change the storage class for the replicated objects

Additional replication options

- Replication Time Control (RTC)

Replication Time Control replicates 99.99% of new objects within 15 minutes and includes replication metrics. Additional fees will apply. [Learn more](#)

- Replication metrics

With replication metrics, you can monitor the total number and size of objects that are pending replication, and the maximum replication time to the destination Region. You can also view and diagnose replication failures. CloudWatch metrics fees apply. [Learn more](#) or see [Amazon CloudWatch pricing](#)

- Delete marker replication

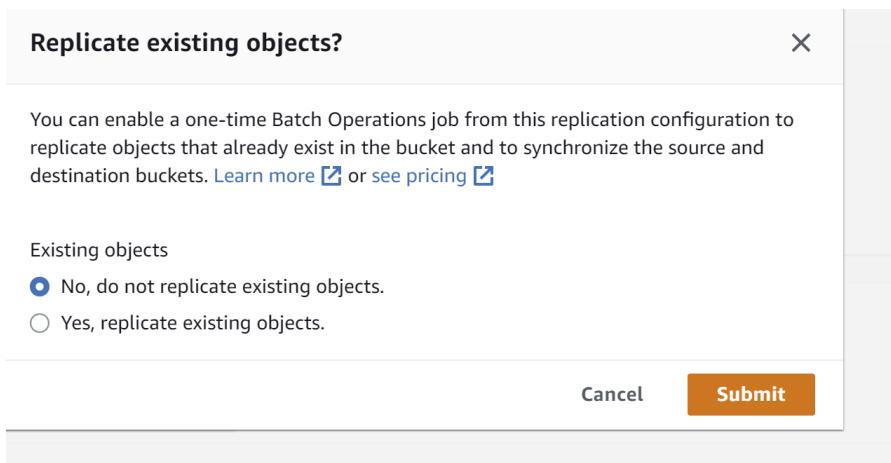
Delete markers created by S3 delete operations will be replicated. Delete markers created by lifecycle rules are not replicated. [Learn more](#)

- Replica modification sync

Replicate metadata changes made to replicas in this bucket to the destination bucket. [Learn more](#)

Cancel

Save



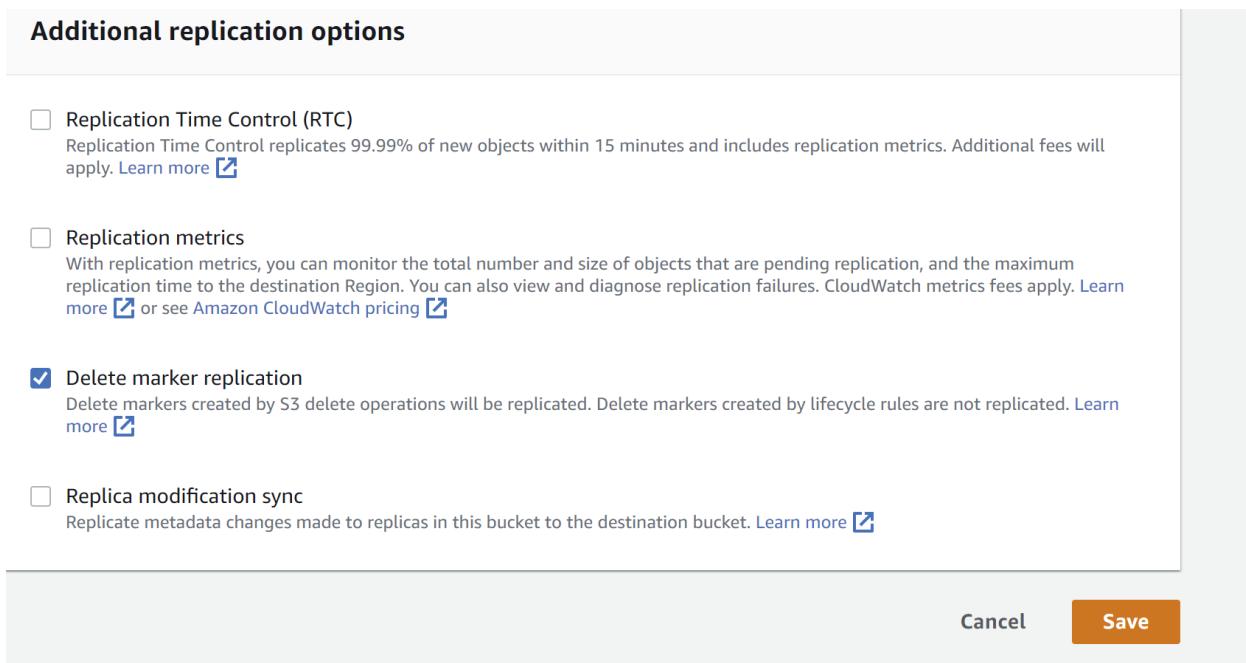
Now upload a picture in rajiv-2023-original

And now go to the rajiv-2023-replica bucket and wait for few seconds and refresh the page and we can see the image is now showing here.

Enable the delete market

select the bucket >Management>select the rule>click edit rule>scroll down >and select the Delete marker replication and save it

Now if we delete any image from the original file the delete marker will replica to the replica bucket



NB: When we delete any image, it will not copy to the replica bucket it only copy delete marker.
if we delete the image from original file permanently it will not delete the file from replica

Objects (4)							
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket . For others to access your objects, you'll need to explicitly grant them permissions. Learn more here							
	Name	Type	Version ID	Last modified	Size	Storage class	Actions
<input type="checkbox"/>	cat.jpeg	jpeg	Yet05XAgpPqck3UngU9Fhb7Xtr0LMeO	June 15, 2023, 20:01:02 (UTC+06:00)	4.7 KB	Standard	Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload
<input type="checkbox"/>	↳ cat.jpeg	jpeg	S72HFvZj4_6_USMkmJXF3_P_sYpUDdy7	June 15, 2023, 19:44:45 (UTC+06:00)	4.7 KB	Standard	Show versions
<input checked="" type="checkbox"/>	↳ cat1.jpeg	Delete marker	QjKdanYHbVOL6i27BcDyd4Eb6lzojY.f	June 15, 2023, 20:03:53 (UTC+06:00)	0 B	-	Show versions
<input type="checkbox"/>	↳ cat1.jpeg	jpeg	vGEr94NQPbq\$98WN.Q9fHKyDOVtb8WoN	June 15, 2023, 19:57:39 (UTC+06:00)	1.2 MB	Standard	Show versions

S3 Storage Class

- Amazon S3 Standard - General Purpose
- Amazon S3 Standard-Infrequent Access (IA)
- Amazon S3 One Zone-Infrequent Access
- Amazon S3 Glacier Instant Retrieval
- Amazon S3 Glacier Flexible Retrieval
- Amazon S3 Glacier Deep Archive
- Amazon S3 Intelligent Tiering
- Can move between classes manually or using S3 Lifecycle configurations

S3 Durability Ad Availability

- Durability:
 - High durability (99.99999999%, 11 9's) of objects across multiple AZ
 - If you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years
 - Same for all storage classes
- Availability:
 - Measures how readily available a service is
 - Varies depending on storage class
 - Example: S3 standard has 99.99% availability = not available 53 minutes a year

S3 Standard – General

- 99.99% Availability
- Used for frequently accessed data
- Low latency and high throughput
- Sustain 2 concurrent facility failures
- Use Cases: Big Data analytics, mobile & gaming applications, content distribution...

S3 Storage Classes – Infrequent Access

- For data that is less frequently accessed, but requires rapid access when needed
- Lower cost than S3 Standard
- Amazon S3 Standard-Infrequent Access (S3 Standard-IA)
 - 99.9% Availability
 - Use cases: Disaster Recovery, backups
- Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA)
 - High durability (99.99999999%) in a single AZ; data lost when AZ is destroyed
 - 99.5% Availability
 - Use Cases: Storing secondary backup copies of on-premise data, or data you can recreate



S3 Glacier Storage Classes

- Low-cost object storage meant for archiving / backup
- Pricing: price for storage + object retrieval cost
- Amazon S3 Glacier Instant Retrieval
 - Millisecond retrieval, great for data accessed once a quarter
 - Minimum storage duration of 90 days
- Amazon S3 Glacier Flexible Retrieval (formerly Amazon S3 Glacier):
 - Expedited (1 to 5 minutes), Standard (3 to 5 hours), Bulk (5 to 12 hours) – free
 - Minimum storage duration of 90 days
- Amazon S3 Glacier Deep Archive – for long term storage:
 - Standard (12 hours), Bulk (48 hours)
 - Minimum storage duration of 180 days



S3 Intelligent – Tiering

- Small monthly monitoring and auto-tiering fee
- Moves objects automatically between Access Tiers based on usage
- There are no retrieval charges in S3 Intelligent-Tiering
- Frequent Access tier (automatic): default tier
- Infrequent Access tier (automatic): objects not accessed for 30 days
- Archive Instant Access tier (automatic): objects not accessed for 90 days
- Archive Access tier (optional): configurable from 90 days to 700+ days
- Deep Archive Access tier (optional): config. from 180 days to 700+ days

S3 Storage Classes Comparison

	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Glacier Deep Archive
Durability	99.999999999% == (11 9's)						
Availability	99.99%	99.9%	99.9%	99.5%	99.9%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99%	99.9%	99.9%
Availability Zones	>= 3	>= 3	>= 3	1	>= 3	>= 3	>= 3
Min. Storage Duration Charge	None	None	30 Days	30 Days	90 Days	90 Days	180 Days
Min. Billable Object Size	None	None	128 KB	128 KB	128 KB	40 KB	40 KB
Retrieval Fee	None	None	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved

S3 Storage Classes – Price Comparison Example: us-east-1

	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Glacier Deep Archive
Storage Cost (per GB per month)	\$0.023	\$0.0025 - \$0.023	%0.0125	\$0.01	\$0.004	\$0.0036	\$0.00099
Retrieval Cost (per 1000 request)	GET: \$0.0004 POST: \$0.005	GET: \$0.0004 POST: \$0.005	GET: \$0.001 POST: \$0.01	GET: \$0.001 POST: \$0.01	GET: \$0.01 POST: \$0.02	GET: \$0.0004 POST: \$0.03 Expedited: \$10 Standard: \$0.05 Bulk: free	GET: \$0.0004 POST: \$0.05 Standard: \$0.10 Bulk: \$0.025
Retrieval Time	Instantaneous					Expedited (1 – 5 mins) Standard (3 – 5 hours) Bulk (5 – 12 hours)	Standard (12 hours) Bulk (48 hours)
Monitoring Cost (per 1000 objects)		\$0.0025					

Pricing URL:

<https://aws.amazon.com/s3/pricing/>

Storage class details

<https://aws.amazon.com/s3/storage-classes/>

Lab: storage class

Create a bucket and then upload image

or

Choose storage class when upload an image

The screenshot shows the AWS S3 'Upload' interface. At the top, it says 'Amazon S3 > Buckets > rajiv2021 > Upload'. Below that is a 'Upload' button and an 'Info' link. A note states: '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'.

The main area has a large input field with the placeholder 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Below it, a table lists 'Files and folders (1 Total, 39.7 KB)'. The table has columns: Name, Folder, Type, and Size. It shows one entry: 'cat2.jpeg' (image/jpeg, 39.7 KB). There are 'Remove', 'Add files', and 'Add folder' buttons above the table.

Below the table is a 'Destination' section. It shows the current destination as 's3://rajiv2021'. Under 'Destination details', it says 'Bucket settings that impact new objects stored in the specified destination.'

At the bottom, there are sections for 'Permissions' and 'Properties'. The 'Properties' section is highlighted with a red box. It says 'Specify storage class, encryption settings, tags, and more...' and includes a 'Storage class' dropdown.

In the 'Storage class' dropdown, the 'Standard' option is selected and highlighted with a red box. The table below it shows details: 'Designed for' is 'Frequently accessed data (more than once a month) with milliseconds access', 'Availability Zones' is '≥ 3', and 'Min storage duration' is '-'.

Edit the storage class

click the image>scroll down>click Edit button of Storage class

The screenshot shows the 'Edit' button for a storage class. The 'Storage class' section at the top says 'Amazon S3 offers a range of storage classes designed for different use cases. Learn more' and 'see Amazon S3 pricing'.

Below it, a table shows the current storage class settings:

Storage class	Designed for	Availability Zones	Min storage duration
Standard	Frequently accessed data (more than once a month) with milliseconds access	≥ 3	-

Now select any class and click save changes.

Storage class

Amazon S3 offers a range of storage classes designed for different use cases. Learn more [\[\]](#) or see Amazon S3 pricing [\[\]](#)

Storage class	Designed for	Availability Zones	Min storage duration	Cost
Standard	Frequently accessed data (more than once a month) with milliseconds access	≥ 3	-	-
Intelligent-Tiering	Data with changing or unknown access patterns	≥ 3	-	-
Standard-IA	Infrequently accessed data (once a month) with milliseconds access	≥ 3	30 days	1
One Zone-IA	Recreatable, infrequently accessed data (once a month) stored in a single Availability Zone with milliseconds access	1	30 days	1
Glacier Instant Retrieval	Long-lived archive data accessed once a quarter with instant retrieval in milliseconds	≥ 3	90 days	1
Glacier Flexible Retrieval (formerly Glacier)	Long-lived archive data accessed once a year with retrieval of minutes to hours	≥ 3	90 days	-
Glacier Deep Archive	Long-lived archive data accessed less than once a year with retrieval of hours	≥ 3	180 days	-
Reduced redundancy	Noncritical, frequently accessed data with milliseconds access (not recommended as S3 Standard is more cost effective)	≥ 3	-	-

Specified objects

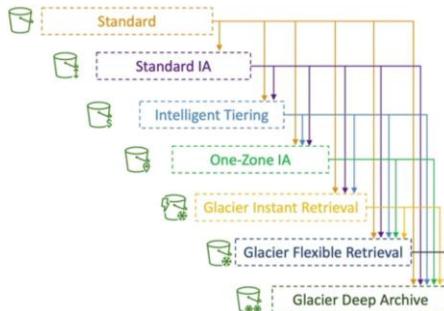
Find objects by name: < 1 >

Name	Type	Last modified	Size	Storage class
cat.jpeg	jpeg	June 15, 2023, 18:11:17 (UTC+06:00)	4.7 KB	Standard

Cancel **Save changes**

Amazon S3 – Moving between Storage Classes

- You can transition objects between storage classes
- For infrequently accessed object, move them to Standard IA
- For archive objects that you don't need fast access to, move them to Glacier or Glacier Deep Archive
- Moving objects can be automated using a Lifecycle Rules



Amazon S3 – Life cycle rule

- Transition Actions – configure objects to transition to another storage class
 - Move objects to Standard IA class 60 days after creation
 - Move to Glacier for archiving after 6 months
- Expiration actions – configure objects to expire (delete) after some time
 - Access log files can be set to delete after a 365 days
 - Can be used to delete old versions of files (if versioning is enabled)
 - Can be used to delete incomplete Multi-Part uploads
- Rules can be created for a certain prefix (example: s3://mybucket/mp3/*)
- Rules can be created for certain objects Tags (example: Department: Finance)

LAB: Create life cycle rule:

Select or get in the bucket > Management>create lifecycle rule:

Amazon S3 > Buckets > rajiv2021

rajiv2021 Policy available

Objects Properties Permissions Metrics **Management** Access Points

Lifecycle rules [0] Use lifecycle rules to define actions you want Amazon S3 to take during or after certain events such as transitioning objects to another storage class, moving them or deleting them after a specified period of time.

View details Edit Delete Actions Create lifecycle rule

Lifecycle rule name	Status	Scope	Current version actions	Noncurrent versions actions	Expired object delete markers	Incomplete multipart uploads
No lifecycle rules						

There are no lifecycle rules for this bucket.

Create lifecycle rule

Amazon S3 > Buckets > rajiv2021 > Lifecycle configuration > Create lifecycle rule

Create lifecycle rule Info

Lifecycle rule configuration

Lifecycle rule name Up to 255 characters

Choose a rule scope Limit the scope of this rule using one or more filters Apply to all objects in the bucket

⚠️ Apply to all objects in the bucket
If you want the rule to apply to specific objects, you must use a filter to identify those objects. Choose "Limit the scope of this rule using one or more filters". [Learn more](#)

I acknowledge that this rule will apply to all objects in the bucket.

Lifecycle rule actions

Choose the actions you want this rule to perform. Per-request fees apply. [Learn more](#) or see [Amazon S3 pricing](#)

Move current versions of objects between storage classes Move noncurrent versions of objects between storage classes Expire current versions of objects Permanently delete noncurrent versions of objects Delete expired object delete markers or incomplete multipart uploads

These actions are not supported when filtering by object tags or object size.

Transition current versions of objects between storage classes

Choose transitions to move current versions of objects between storage classes based on your use case scenario and performance access requirements. These transitions start from when the objects are created and are consecutively applied. [Learn more](#)

Choose storage class transitions Days after object creation

Standard-IA	30	<input type="button"/> Remove
Intelligent-Tiering	60	<input type="button"/> Remove
Glacier Instant Retrieval	90	<input type="button"/> Remove

Add transition

Review transition and expiration actions

Current version actions	Noncurrent versions actions
Day 0 • Objects uploaded	Day 0 No actions defined.
↓	
Day 30 • Objects move to Standard-IA	
↓	
Day 60 • Objects move to Intelligent-Tiering	
↓	
Day 90 • Objects move to Glacier Instant Retrieval	

Cancel **Create rule**

Amazon S3 – Life cycle rules (Scenario-1)

- Your application on EC2 creates images thumbnails after profile photos are uploaded to Amazon S3. These thumbnails can be easily recreated, and only need to be kept for 60 days. The source images should be able to be immediately retrieved for these 60 days, and afterwards, the user can wait up to 6 hours. How would you design this?
- S3 source images can be on Standard, with a lifecycle configuration to transition them to Glacier after 60 days
- S3 thumbnails can be on One-Zone IA, with a lifecycle configuration to expire them (delete them) after 60 days

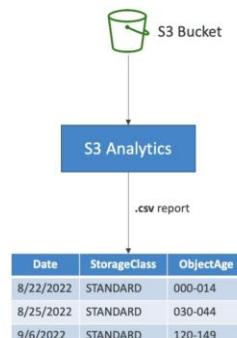
Amazon S3 – Life cycle rules (Scenario-2)

- A rule in your company states that you should be able to recover your deleted S3 objects immediately for 30 days, although this may happen rarely. After this time, and for up to 365 days, deleted objects should be recoverable within 48 hours.
- Enable S3 Versioning in order to have object versions, so that "deleted objects" are in fact hidden by a "delete marker" and can be recovered
- Transition the "noncurrent versions" of the object to Standard IA
- Transition afterwards the "noncurrent versions" to Glacier Deep Archive

NB: non-current version means when we delete any object that object is save with a delete marker this deleted objects are non-current version.

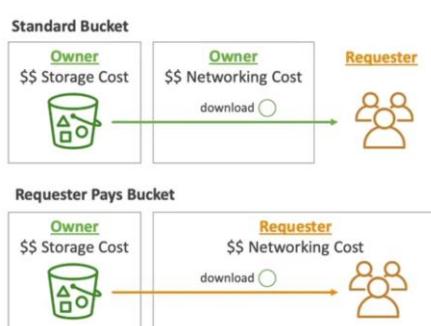
Amazon S3 Analytics – Storage Class Analysis

- Help you decide when to transition objects to the right storage class
- Recommendations for Standard and Standard IA
 - Does NOT work for One-Zone IA or Glacier
- Report is updated daily
- 24 to 48 hours to start seeing data analysis



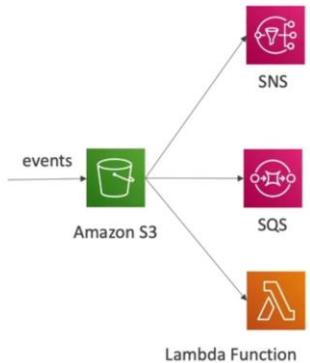
S3 Requester pay.

- In general, bucket owners pay for all Amazon S3 storage and data transfer costs associated with their bucket
- With Requester Pays buckets, the requester instead of the bucket owner pays the cost of the request and the data download from the bucket
- Helpful when you want to share large datasets with other accounts
- The requester must be authenticated in AWS (cannot be anonymous)

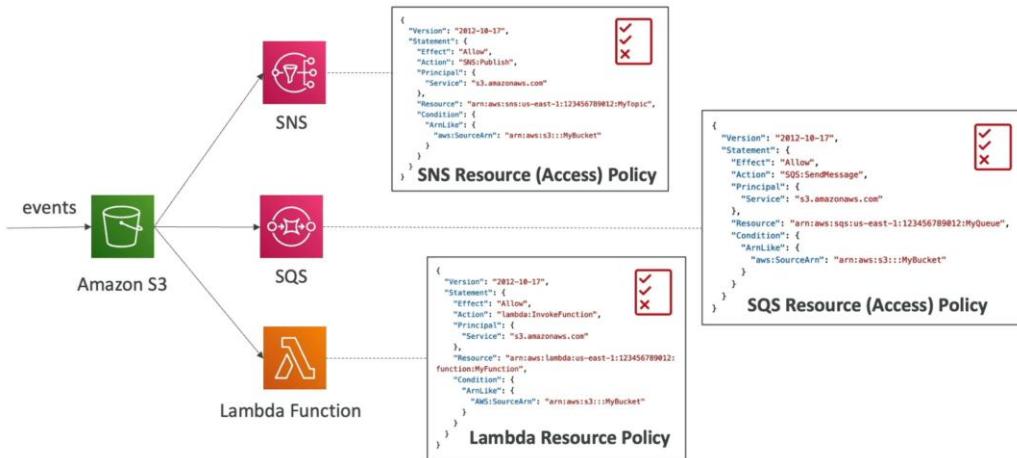


S3 Event Notifications

- S3:ObjectCreated, S3:ObjectRemoved, S3:ObjectRestore, S3:Replication...
- Object name filtering possible (*.jpg)
- Use case: generate thumbnails of images uploaded to S3
- Can create as many "S3 events" as desired
- S3 event notifications typically deliver events in seconds but can sometimes take a minute or longer



S3 Event Notifications – IAM Permissions



S3 Event Notifications with Amazon EventBridge

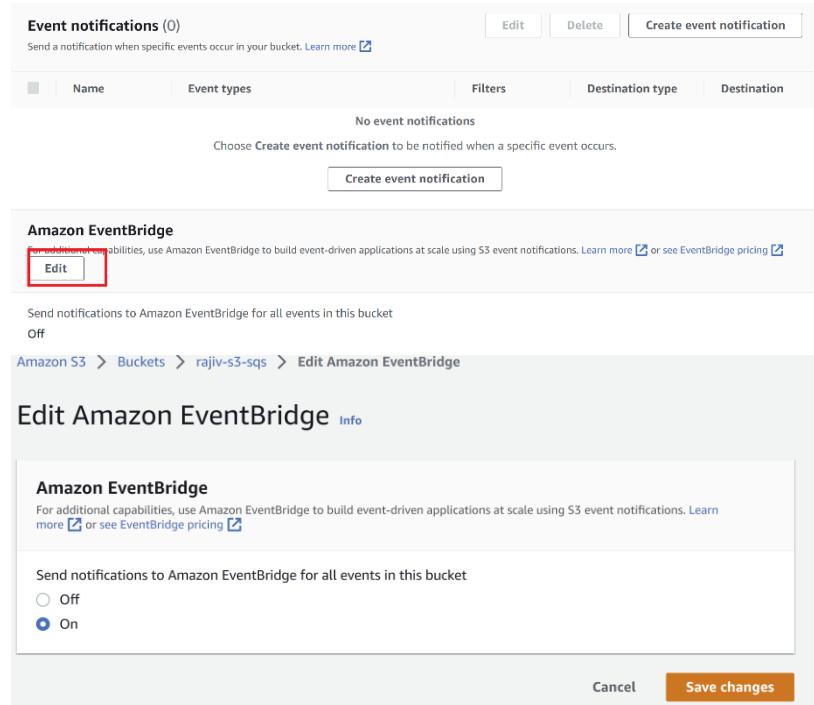


- Advanced filtering options with JSON rules (metadata, object size, name...)
- Multiple Destinations – ex Step Functions, Kinesis Streams / Firehose...
- EventBridge Capabilities – Archive, Replay Events, Reliable delivery

Lab: s3 event notification

Create a bucket

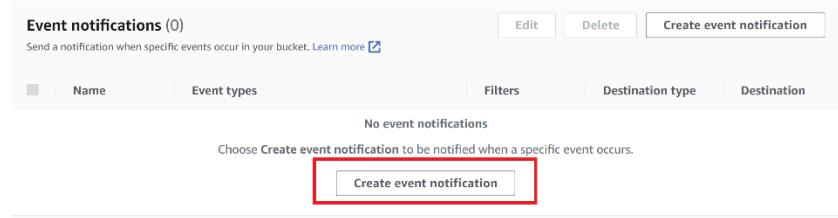
select the bucket>properties>scroll down>edit amazon event bridge>select ON>click save changes.



The screenshot shows the 'Event notifications (0)' section of the S3 bucket properties. It includes tabs for 'Name', 'Event types', 'Filters', 'Destination type', and 'Destination'. A 'Create event notification' button is at the top right. Below it, a message says 'No event notifications' and 'Choose Create event notification to be notified when a specific event occurs.' A 'Create event notification' button is also here. The 'Amazon EventBridge' section follows, with a note about using it for event-driven applications. The 'Edit' button is highlighted with a red box. At the bottom, there's a note about sending notifications to EventBridge and a radio button for 'On' (which is selected). Finally, 'Cancel' and 'Save changes' buttons are at the bottom right.

Event bridge is complicated so we will only see event notifications.

select the bucket>properties>scroll down>click event notification



This screenshot is identical to the one above, showing the 'Event notifications (0)' section of the S3 bucket properties. The 'Create event notification' button is highlighted with a red box.

keep on the create notification page

now create a SQS



The screenshot shows the Amazon SQS service landing page. It features a dark header with 'Amazon SQS' and a 'Get started' button. Below is a main section with the heading 'Amazon SQS' and 'A message queuing service'. It describes Amazon SQS as a provider for high-throughput, system-to-system messaging. A 'Create queue' button is prominently displayed at the bottom right of this section, also highlighted with a red box.

Amazon SQS > Queues > Create queue

Create queue

Details

Type
Choose the queue type for your application or cloud infrastructure.

Standard Info At-least-once delivery; message ordering isn't preserved

- At-least once delivery
- Best-effort ordering

FIFO Info First-in-first-out delivery; message ordering is preserved

- First-in-first-out delivery
- Exactly-once processing

Note: You can't change the queue type after you create a queue.

Name
demo-sq-1

A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (_).

Configuration Info Set the maximum message size, visibility to other consumers, and message retention.

Visibility timeout **Info** 30 Seconds Should be between 0 seconds and 12 hours.

Message retention period **Info** 4 Days Should be between 1 minute and 14 days.

Delivery delay **Info** 0 Seconds Maximum message size **Info** 256 KB

Dead-letter queue - Optional Info Send undeliverable messages to a dead-letter queue.

Set this queue to receive undeliverable messages.

Disabled
 Enabled

Tags - Optional Info A tag is a label assigned to an AWS resource. Use tags to search and filter your resources or track your AWS costs.

Key Value - optional
Q Enter key Q Enter value Remove

Add new tag
You can add up to 49 more tags.

Cancel **Create queue**

After clicking the create queue then click the access policy

Amazon SQS > Queues > demo-sq-1

demo-sq-1

Details Info

Name demo-sq-1 Type Standard ARN arn:aws:sqs:us-east-1:999938272208:demo-sq-1

Encryption Amazon SQS key (SQS-SQS)
More

SNS subscriptions Lambda triggers Dead-letter queue Monitoring Tagging **Access policy** Encryption Dead-letter queue redrive tasks

Access policy (Permissions) Info Define who can access your queue

```
[{"version": "2012-10-17", "statement": [{"principal": "arn:aws:iam::000000000000:root", "action": "SQS:SendMessage", "resource": "arn:aws:sqs:us-east-1:999938272208:demo-sq-1"}]}
```

Access policy [Info](#)
Define who can access your queue.

```

1 {
2     "Version": "2012-10-17",
3     "Id": "default_policy_id",
4     "Statement": [
5         {
6             "Sid": "AllowOwner",
7             "Effect": "Allow",
8             "Principal": "*",
9             "Action": "SQS:*",
10            "Resource": "arn:aws:sqs:us-east-1:999838272298:demo-ss-1"
11        }
12    ]
13 }
14 }
15 }
```

[Policy generator](#)



AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [key concepts in Using AWS Identity and Access Management](#). Here are sample policies.

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an [IAM Policy](#), an [S3 Bucket Policy](#), an [SNS Topic Policy](#), and an [SQS Queue Policy](#).

Select Type of Policy [SQS Queue Policy](#)

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a [description of elements](#) that you can use in statements.

Effect Allow Deny

Principal

Use a comma to separate multiple values.

AWS Service [Amazon SQS](#) All Services (*)

Actions [1 Action\(s\) Selected](#) All Actions (*)

Amazon Resource Name (ARN) \${Region}:\${Account}:\${QueueName}.
Valid. You must enter a valid ARN.

ListQueueTags
ListQueues
PurgeQueue
ReceiveMessage
RemovePermission
 SendMessage
SetQueueAttributes

Step 3: Generate Policy



AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [key concepts in Using AWS Identity and Access Management](#). Here are [sample policies](#).

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an [IAM Policy](#), an [S3 Bucket Policy](#), an [SNS Topic Policy](#), a [VPC Endpoint Policy](#), and an [SQS Queue Policy](#).

Select Type of Policy [SQS Queue Policy](#)

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a [description of elements](#) that you can use in statements.

Effect Allow Deny

Principal

Use a comma to separate multiple values.

AWS Service [Amazon SQS](#) All Services (*)

Actions [1 Action\(s\) Selected](#) All Actions (*)

Amazon Resource Name (ARN) arn:aws:sqs:us-east-1:9998
you copy this from previous page

ARN should follow the following format: arn:aws:sqs:\${Region}:\${Account}:\${QueueName}.

Add Conditions (Optional)

[Add Statement](#)

Access policy [Info](#)

Define who can access your queue.

```

1 { "Version": "2012-10-17",
2  "Id": "..._default_policy_ID",
3  "Statement": [
4    {
5      "Sid": "..._owner_statement",
6      "Effect": "Allow",
7      "Principal": {
8        "AWS": "arn:aws:iam::999838272208:root"
9      },
10     "Action": "SQS:*",
11     "Resource": "arn:aws:sqs:us-east-1:999838272208:demo-sqs-1"
12   }
13 ]
14 }
15 
```

[Policy generator](#) [Edit JSON](#)

After clicking Add Statement

Effect Allow Deny

Principal

AWS Service All Services (*)

Actions All Actions (*)

Amazon Resource Name (ARN)

Add Conditions (Optional)

You added the following statements. Click the button below to Generate a policy.

Principal(s)	Effect	Action	Resource	Conditions
*	Allow	sq:SendMessage	arn:aws:sqs:us-east-1:999838272208:demo-sqs-1	None

Step 3: Generate Policy

A *policy* is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements.

Use a comma to separate multiple values.

AWS Service All Services (*)

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": "Policy1686897956937",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1686897917347",
      "Action": [
        "sq:SendMessage"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:sqs:us-east-1:999838272208:demo-sqs-1",
      "Principal": "*"
    }
  ]
}
```

This AWS Policy Generator is provided for informational purposes only. You are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as-is without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the

(Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as-is without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the

Now copy this and paste it in access policy which is in previous page and save it

```

Access policy info
Define who can access your queue.

1 [ "Effect": "Allow",
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "SQSService"
6     },
7     {
8       "Action": [
9         "SQS:SendMessage"
10      ],
11      "Resource": "arn:aws:queue:region-east-1:999999999999:demo-1q-1",
12      "Principal": "*"
13    }
14  ]
15 }

Policy generator \[?\]

Redrive allow policy - Optional info
Identify which source queues can use this queue as the dead-letter queue.

Select which source queues can use this queue as the dead-letter queue.
 Disabled
 Enabled

Dead-letter queue - Optional info
Send undeliverable messages to a dead-letter queue.

Set this queue to receive undeliverable messages.
 Disabled
 Enabled

Tags - Optional info
A tag is a label assigned to an AWS resource. Use tags to search and filter your resources or track your AWS costs.

No tags associated with this queue.
Add new tag
You can add up to five tags.
Cancel Save

```

Now go back to event notification page and a event notification

Amazon S3 > Buckets > rajiv-s3-sqs > Create event notification

Create event notification [info](#)

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

General configuration

Event name
sq-s3-event-notification-1
Event name can contain up to 255 characters.

Prefix - optional
Limit the notifications to objects with key starting with specified characters.
images/

Suffix - optional
Limit the notifications to objects with key ending with specified characters.
.jpg

Event types

Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

Object creation

- All object create events
s3:ObjectCreated:*
- Put
s3:ObjectCreated:Put
- Post
s3:ObjectCreated:Post
- Copy
s3:ObjectCreated:Copy

Scroll down

Destination

Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. [Learn more](#)

Destination
Choose a destination to publish the event. [Learn more](#)

- Lambda function**
Run a Lambda function script based on S3 events.
- SNS topic**
Fanout messages to systems for parallel processing or directly to people.
- SQS queue**
Send notifications to an SQS queue to be read by a server.

Specify SQS queue

- Choose from your SQS queues**
- Enter SQS queue ARN**

SQS queue

Choose SQS queue

demo-sqs-1

Cancel **Save changes**

After create event notification go to SQS page click send and receive message button

Amazon SQS > Queues > demo-sqs1

demo-sqs1

Edit Delete Purge **Send and receive messages** Start DLQ receive

Details [info](#)

Name	demo-sqs1	Type	Standard
Encryption	Amazon SQS key (SSSE-SQS)	URL	https://sqs.us-east-1.amazonaws.com/999938272208/demo-sqs1
		ARN	arn:aws:sqs:us-east-1:999938272208:demo-sqs1
		Dead-letter queue	-

More

Scroll down and pull the message as shown below

Receive messages [info](#)

Edit poll settings Stop polling **Poll for messages**

Messages available: 1 Polling duration: 30 Maximum message count: 10 Polling progress: 1 receives/second

Messages (1)

Search messages

ID	Sent	Size	Receive count
19f72c9ab-22e2-40df-b16c-006a365facd8	2023-11-10T19:18+06:00	776 bytes	1

Now select the message and delete it

Now upload an image to that bucket where we create the event notification

now go to the SQS page and click pull for message

Now click the message id

The screenshot shows the 'Receive messages' interface with the following details:

- Messages available: 1
- Polling duration: 30
- Maximum message count: 10
- Polling progress: 1 receives/second
- Message ID: 19f2c9ab-22e2-404f-b16c-006a365facd8 (highlighted with a red box)
- Message Sent: 2023-11-10T19:18:06.000Z
- Message Size: 76 bytes
- Message Receive count: 1

After clicking the id we get the message where is show the upload image details.

The screenshot shows the 'Message details' view for message ID 19f2c9ab-22e2-404f-b16c-006a365facd8. The message body contains the following JSON payload:

```
{
  "Records": [
    {
      "eventVersion": "2.1",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-2",
      "eventTime": "2023-11-10T13:18:15.474Z",
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
        "principalId": "A215ATGG2KKT80"
      },
      "requestParameters": {
        "sourceIPAddress": "103.146.55.127"
      },
      "responseElements": {
        "x-amz-request-id": "EXQTDAB5T438H2PJ",
        "x-amz-id-2": "J6M9XMHuKtVfk3gCrpVq8PZR6p6l9P2sikh5C4E4yHad2Zt+hNDE7AJdub7+D3rqCN6O5iCvruhQyTSMSCqDkX+mIgq5b",
        "s3": {
          "configurationId": "event-notification-10-nov",
          "bucket": {
            "name": "rajiv-event-notification-10-nov"
          },
          "ownerIdentity": {
            "principalId": "A215ATGG2KKT80"
          },
          "arn": "arn:aws:s3:::rajiv-event-notification-10-nov",
          "object": {
            "key": "cat.jpg",
            "size": 27361,
            "eTag": "e9d35b0fdd065d2c437e7171a1fb54fd",
            "sequencer": "00654E2D976FC2D646"
          }
        }
      }
    }
  ]
}
```

A yellow 'Done' button is visible at the bottom left.

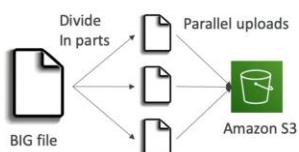
After reading the message delete it

S3- Baseline performance

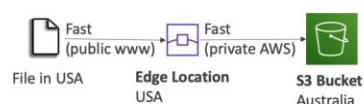
- Amazon S3 automatically scales to high request rates, latency 100-200 ms
- Your application can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per prefix in a bucket.
- There are no limits to the number of prefixes in a bucket.
- Example (object path => prefix):
 - bucket/folder1/sub1/file => /folder1/sub1/
 - bucket/folder1/sub2/file => /folder1/sub2/
 - bucket/1/file => /1/
 - bucket/2/file => /2/
- If you spread reads across all four prefixes evenly, you can achieve 22,000 requests per second for GET and HEAD

S3 performance

- Multi-Part upload:**
 - recommended for files > 100MB, must use for files > 5GB
 - Can help parallelize uploads (speed up transfers)



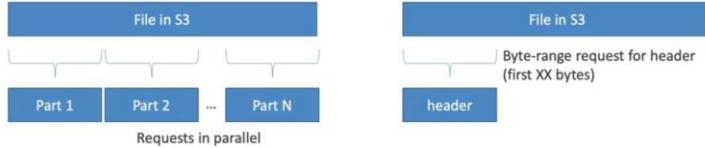
- S3 Transfer Acceleration**
 - Increase transfer speed by transferring file to an AWS edge location which will forward the data to the S3 bucket in the target region
 - Compatible with multi-part upload



S3 performance – S3 Byte-Range Fetches

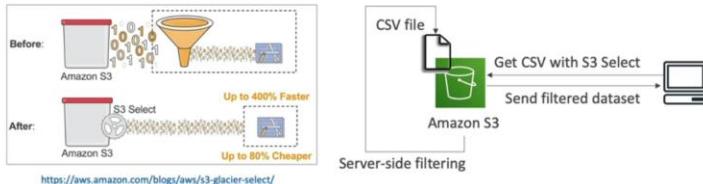
- Parallelize GETs by requesting specific byte ranges
- Better resilience in case of failures

Can be used to speed up downloads



S3 Select and Glacier Select

- Retrieve less data using SQL by performing server-side filtering
- Can filter by rows & columns (simple SQL statements)
- Less network transfer; less CPU cost client-side



S3 Batch Operations

- Perform bulk operations on existing S3 objects with a single request, example:
 - Modify object metadata & properties
 - Copy objects between S3 buckets
 - Encrypt un-encrypted objects
 - Modify ACLs, tags
 - Restore objects from S3 Glacier
 - Invoke Lambda function to perform custom action on each object
- A job consists of a list of objects, the action to perform, and optional parameters
- S3 Batch Operations manages retries, tracks progress, sends completion notifications, generate reports ...
- You can use S3 Inventory to get object list and use S3 Select to filter your objects

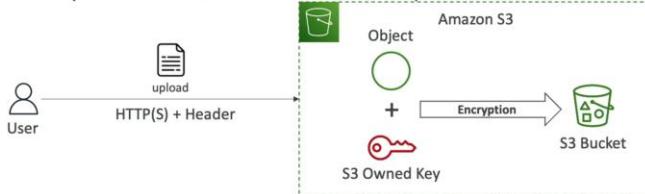


S3 – Object Encryption

- You can encrypt objects in S3 buckets using one of 4 methods
- Server-Side Encryption (SSE)
 - Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3) – Enabled by Default
 - Encrypts S3 objects using keys handled, managed, and owned by AWS
 - Server-Side Encryption with KMS Keys stored in AWS KMS (SSE-KMS)
 - Leverage AWS Key Management Service (AWS KMS) to manage encryption keys
 - Server-Side Encryption with Customer-Provided Keys (SSE-C)
 - When you want to manage your own encryption keys
- Client-Side Encryption
- It's important to understand which ones are for which situation for the exam

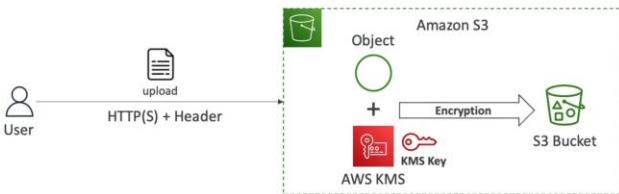
S3 Encryption – SSE-S3

- Encryption using keys handled, managed, and owned by AWS
- Object is encrypted server-side
- Encryption type is AES-256
- Must set header "x-amz-server-side-encryption": "AES256"
- Enabled by default for new buckets & new objects



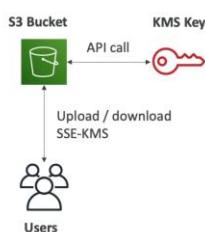
S3 Encryption – SSE-KMS

- Encryption using keys handled and managed by AWS KMS (Key Management Service)
- KMS advantages: user control + audit key usage using CloudTrail
- Object is encrypted server side
- Must set header "x-amz-server-side-encryption": "aws:kms"



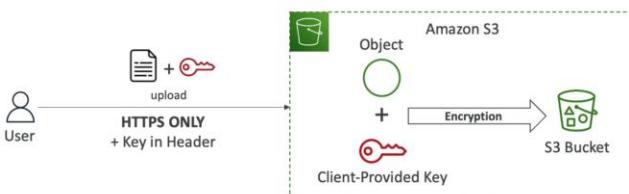
SSE-KMS Limitation

- If you use SSE-KMS, you may be impacted by the KMS limits
- When you upload, it calls the GenerateDataKey KMS API
- When you download, it calls the Decrypt KMS API
- Count towards the KMS quota per second (5500, 10000, 30000 req/s based on region)
- You can request a quota increase using the Service Quotas Console



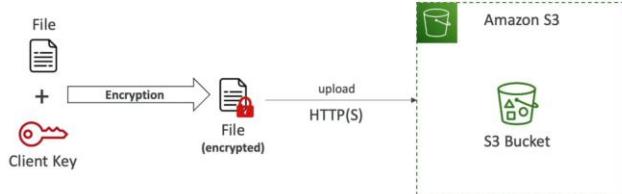
S3 Encryption – SSE-C

- Server-Side Encryption using keys fully managed by the customer outside of AWS
- 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



Amazon S3 Encryption – Client- Side Encryption

- Use client libraries such as Amazon S3 Client-Side Encryption Library
- Clients must encrypt data themselves before sending to Amazon S3
- Clients must decrypt data themselves when retrieving from Amazon S3
- Customer fully manages the keys and encryption cycle

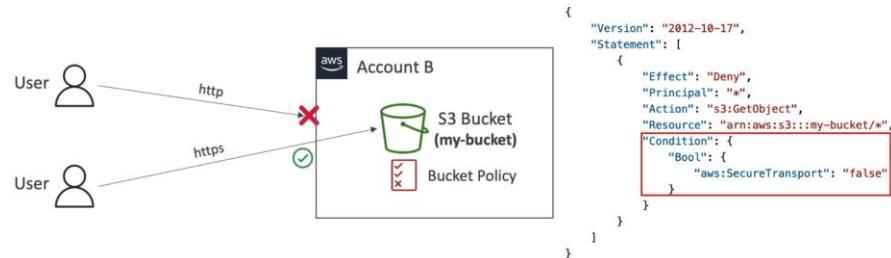


Amazon S3 Encryption in transit (SSL/TSL)

- Encryption in flight is also called SSL/TLS
- Amazon S3 exposes two endpoints:
 - HTTP Endpoint – non encrypted
 - HTTPS Endpoint – encryption in flight
- HTTPS is recommended
- HTTPS is mandatory for SSE-C
- Most clients would use the HTTPS endpoint by default



Amazon S3 – Force Encryption in Transit aws: Secure Transport



Amazon S3 – Default Encryption vs Bucket Policies

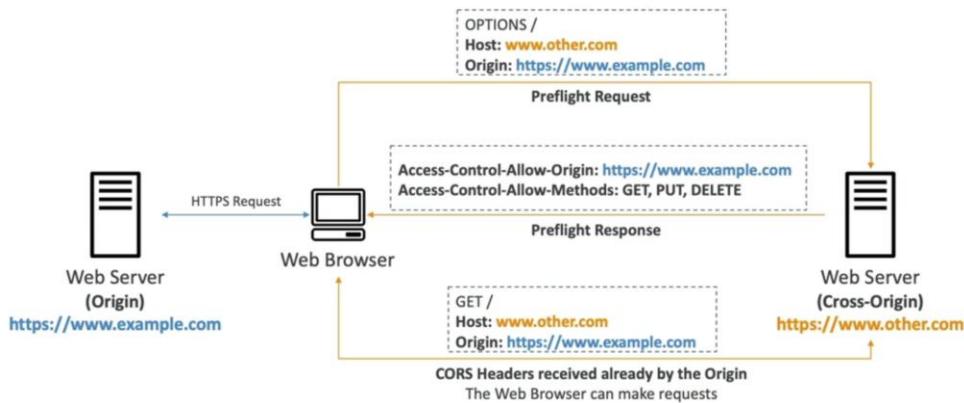
- SSE-S3 encryption is automatically applied to new objects stored in S3 bucket
- Optionally, you can "force encryption" using a bucket policy and refuse any API call to PUT an S3 object without encryption headers (SSE-KMS or SSE-C)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "s3:PutObject",
      "Principal": "*",
      "Resource": "arn:aws:s3:::my-bucket/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "aws:kms"
        }
      }
    }
  ]
}

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "s3:PutObject",
      "Principal": "*",
      "Resource": "arn:aws:s3:::my-bucket/*",
      "Condition": {
        "Null": {
          "s3:x-amz-server-side-encryption-customer-algorithm": "true"
        }
      }
    }
  ]
}
```

CORS (Cross Origin Resource Sharing)

- Cross-Origin Resource Sharing (CORS)
- Origin = scheme (protocol) + host (domain) + port
 - example: <https://www.example.com> (implied port is 443 for HTTPS, 80 for HTTP)
- Web Browser based mechanism to allow requests to other origins while visiting the main origin
- Same origin: <http://example.com/app1> & <http://example.com/app2>
- Different origins: <http://www.example.com> & <http://other.example.com>
- The requests won't be fulfilled unless the other origin allows for the requests, using CORS Headers (example: Access-Control-Allow-Origin)



CORS use in S3

- If a client makes a cross-origin request on our S3 bucket, we need to enable the correct CORS headers
- It's a popular exam question
- You can allow for a specific origin or for * (all origins)



Lab: CORS

Step: 1

first create a bucket with public enable then enable static web hosting and also add the Json script to access the bucket publicly

upload 3 files index.html,extra-page.html,cat.jpg

now brows the static web page- extra-page bottom portion is coming

Step:2

Now, create another bucket with public enable then enable static web hosting and json script for access publicly upload the extra-page.html and check the URL that this page is showing the page

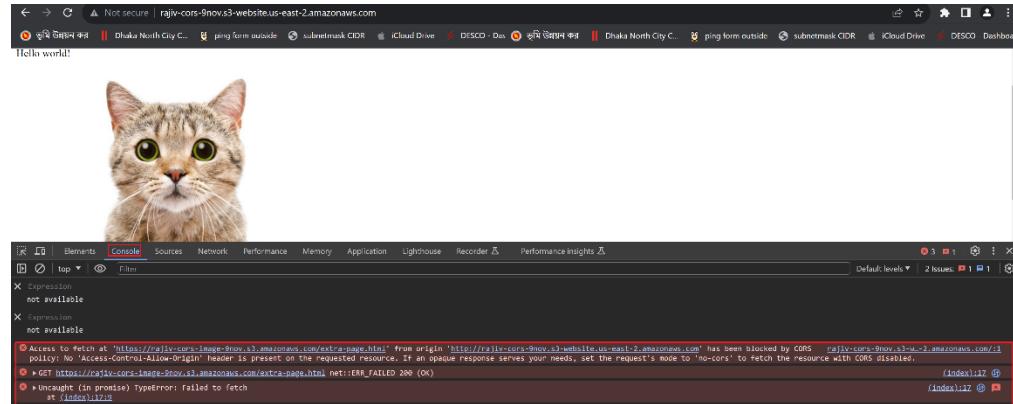
Step:3

now go to the previous bucket and delete the extra-page.html

now brose the bucket one URL and we can see the bottom extra page option is not coming

edit the index page add the extra page URL (<https://rajiv-cors-image-9nov.s3.amazonaws.com/extra-page.html>) to the index file which we get from image bucket. Now upload the index file

now brows the first bucket webpage with developer toll and we can see there is error come



Step:4

now in the 2nd bucket add the cross Json script and save

select the image bucket >permission >scroll down > click the edit and then pest the script and save changes as shown below

A screenshot of the AWS S3 'Edit cross-origin resource sharing (CORS)' configuration page. The page title is 'Edit cross-origin resource sharing (CORS)'. A red box highlights the JSON configuration area. The JSON code is as follows:1 [
2 {
3 "AllowedHeaders": [
4 "Authorization"
5],
6 "AllowedMethods": [
7 "GET"
8],
9 "AllowedOrigins": [
10 "http://rajiv-cors-9nov.s3-website.us-east-2.amazonaws.com"
11],
12 "ExposeHeaders": [],
13 "MaxAgeSeconds": 3600
14}
15]A red arrow points from the text 'make sure dont put any / here' to the double slash in the 'AllowedOrigins' field. At the bottom of the editor, there are 'Cancel' and 'Save changes' buttons.

now browse the page and we can see page is showing properly and don't get any error in the console.

Amazon S3 MFA Delete

- MFA (Multi-Factor Authentication) – force users to generate a code on a device (usually a mobile phone or hardware) before doing important operations on S3
- MFA will be required to:
 - Permanently delete an object version
 - Suspend Versioning on the bucket
- MFA won't be required to:
 - Enable Versioning
 - List deleted versions
- To use MFA Delete, Versioning must be enabled on the bucket
- Only the bucket owner (root account) can enable/disable MFA Delete



Lab: MFA enable and disable on a bucket

First create a bucket with enable versioning.

Select bucket >properties >click edit bucket versioning > then you can see MFA is disable.

A screenshot of the AWS Bucket Versioning settings page. At the top, there is a heading "Bucket Versioning" with an "Edit" button. Below it, a section titled "Bucket Versioning" has a status of "Enabled". A red box highlights this status. Underneath, there is a sub-section titled "Multi-factor authentication (MFA) delete" with a note: "An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API." A link "Learn more" is provided. At the bottom of this section, the status is shown as "Disabled", also highlighted by a red box.

Click user at the right top corner > Security credentials> and you can see your device name.

A screenshot of the AWS IAM Security Credentials page. In the top navigation bar, under "Identity and Access Management (IAM)", the "Security credentials" tab is selected. A red box highlights this tab. The main content area shows "My security credentials" with a note: "The root user has access to all AWS resources in this account, and we recommend following best practices. To learn more about the types of AWS credentials and how they're used, see AWS Security Reference." Below this, there is an "Account details" section and a "Multi-factor authentication (MFA)" section. The "Multi-factor authentication (MFA)" section shows a table with one row: "Device type" (Virtual), "Identifier" (Your mfa device name will be shown here), "Certifications" (Not Applicable), and "Created on" (2014-09-09). A red box highlights the "Identifier" field.

Click user at the right top corner > Security credentials> click access key

A screenshot of the AWS IAM Access Keys page. In the top navigation bar, under "Identity and Access Management (IAM)", the "Access keys" tab is selected. A red box highlights this tab. The main content area shows a table with one row: "Access key ID" (12345678901234567890), "Created on" (141 days ago), "Access key last used" (141 days ago), "Region last used" (us-east-1), "Service last used" (s3), and "Status" (Inactive). An "Actions" dropdown menu is open, and a "Create access key" button is highlighted with a red box.

now go to the aws cli and run the following command

```
# aws configure --profile root-mfa-delete-demo
```

```
#put the access key
```

```
#put the secret access key
```

```
#us-east-1
```

```
enter
```

Now we are login as a root by access key and try the following command

```
#aws s3 ls  
#aws s3 --profile root-mfa-delete-demo  
#aws s3api put-bucket-versioning --bucket you_bucket_name --versioning-configuration  
Status=Enabled,MFADelete=Enabled --mfa "arn-of-mfa-device mfa-code" --profile root-mfa-delete-demo
```

Now go to the aws console and you can see the MFA is enable now

The screenshot shows the 'Edit Bucket Versioning' page for the 'rajiv-2023-mfa-enable' bucket. At the top, there's a note: 'Bucket Versioning can't be suspended because Multi-factor authentication (MFA) delete is enabled for this bucket.' Below this, there are two radio button options: 'Suspend' (selected) and 'Enable'. Under 'Multi-factor authentication (MFA) delete', it says 'Enabled' with a red box around it. At the bottom are 'Cancel' and 'Save changes' buttons.

Upload a picture in this bucket

delete the image

now enable the version

now delete the image again and you will get an error message and not able to delete it

The screenshot shows the 'Delete objects' page for the same bucket. A red box highlights an error message: 'You can't delete object versions because Multi-factor authentication (MFA) delete is enabled for this bucket. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. Learn more'.

Disable the MFA

Go to the aws cli

```
# aws s3api put-bucket-versioning --bucket you_bucket_name --versioning-configuration  
Status=Enabled,MFADelete=Disabled --mfa "arn-of-mfa-device mfa-code" --profile root-mfa-delete-demo
```

Now go to aws console and try to delete the image and you able to delete the image now and go to the versioning and you can see the MFA is disable now

The screenshot shows the 'Edit Bucket Versioning' page for a bucket named 'rajuv-2023-mfa-enable'. The 'Bucket Versioning' section has 'Suspend' selected. Below it, the 'Multi-factor authentication (MFA) delete' section shows 'Disabled' (which is highlighted with a red box). At the bottom are 'Cancel' and 'Save changes' buttons.

NB: after done the test delete the root accesses key

When you do the test make sure your root account already have enable MFA otherwise you don't get the arn and MFA code.

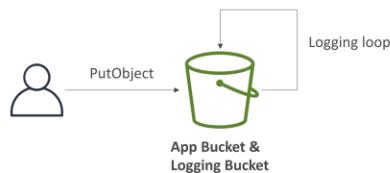
S3 – Access Log

- For audit purpose, you may want to log all access to S3 buckets
- Any request made to S3, from any account, authorized or denied, will be logged into another S3 bucket
- That data can be analyzed using data analysis tools...
- The target logging bucket must be in the same AWS region



S3 – Access Log – important issue

- Do not set your logging bucket to be the monitored bucket
- It will create a logging loop, and your bucket will grow exponentially



Access log format URL

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/LogFormat.html>

Lab: S3 – Access Log

create 1 bucket for example: arina-2023

create another bucket for example: arina-2023-log

now go to the arina-2023 > properties > scroll down and edit server access log > enable and select which bucket the log will be stored

Now do some activity like upload a file to arina-2023 and that activity will be logged in arina-2023-log

Server access logging

Log requests for access to your bucket. [Learn more](#)

Server access logging

Disabled

Amazon S3 > Buckets > rajiv-9-nov > Edit server access logging

Edit server access logging [Info](#)

Server access logging

Log requests for access to your bucket. [Learn more](#)

Server access logging

Disable

Enable

⚠ Bucket policy will be updated
When you enable server access logging, the S3 console automatically updates your bucket policy to include access to the S3 log delivery group.

Target bucket

s3://rajiv-9-nov-log [Browse S3](#)

Format: s3://bucket/prefix

Cancel **Save changes**

Now do some activity in your main bucket for example see your object or upload any file then go to your log bucket and you can see your log and **it will take some time around 1 hour to copy the log of that bucket**. We will get the log as below image.

Objects (177)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

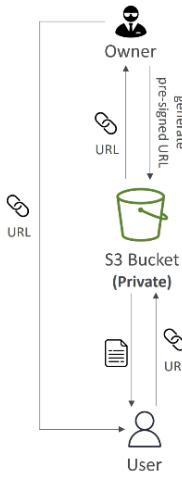
Actions [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

Find objects by prefix

Name	Type	Last modified	Size	Storage class
2023-06-21-05-17-01-1E2642ECA0984BEE	-	June 21, 2023, 11:17:02 (UTC+06:00)	656.0 B	Standard
2023-06-21-05-18-44-73D787F7792C281	-	June 21, 2023, 11:18:45 (UTC+06:00)	681.0 B	Standard
2023-06-21-05-19-50-D3EEBF99A267B457	-	June 21, 2023, 11:19:51 (UTC+06:00)	646.0 B	Standard
2023-06-21-05-21-53-41D1F7A961559822	-	June 21, 2023, 11:21:54 (UTC+06:00)	630.0 B	Standard
2023-06-21-05-22-20-96C96B61542CE2D54	-	June 21, 2023, 11:22:09 (UTC+06:00)	649.0 B	Standard
2023-06-21-05-22-57-073EBB5B09E9595CD5	-	June 21, 2023, 11:22:58 (UTC+06:00)	630.0 B	Standard
2023-06-21-05-24-16-F65F306D0ACEED58	-	June 21, 2023, 11:24:17 (UTC+06:00)	630.0 B	Standard
2023-06-21-05-27-23-0BEF47A935C6279	-	June 21, 2023, 11:27:24 (UTC+06:00)	630.0 B	Standard
2023-06-21-05-32-04-A3D1F13A8BC75SD0	-	June 21, 2023, 11:32:05 (UTC+06:00)	1.9 KB	Standard
2023-06-21-05-34-18-C4788477ADC5657F	-	June 21, 2023, 11:34:19 (UTC+06:00)	635.0 B	Standard
2023-06-21-05-35-22-6E299B37F3561BD	-	June 21, 2023, 11:35:24 (UTC+06:00)	698.0 B	Standard
2023-06-21-05-35-34-8913DEC67F6EBD79	-	June 21, 2023, 11:35:35 (UTC+06:00)	631.0 B	Standard

S3 – Preassigned URL

- Generate pre-signed URLs using the S3 Console, AWS CLI or SDK
- URL Expiration
 - S3 Console – 1 min up to 720 mins (12 hours)
 - AWS CLI – configure expiration with `--expires-in` parameter in seconds (default 3600 secs, max. 604800 secs ~ 168 hours)
- Users given a pre-signed URL inherit the permissions of the user that generated the URL for GET / PUT
- Examples:
 - Allow only logged-in users to download a premium video from your S3 bucket
 - Allow an ever-changing list of users to download files by generating URLs dynamically
 - Allow temporarily a user to upload a file to a precise location in your S3 bucket



LAB: S3 – Preassigned URL

Amazon S3 > Buckets > rajiv-9-nov

rajiv-9-nov [Info](#)

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket.

permissions [Learn more](#)

Copy S3 URI Copy URL Download Open Delete

Find objects by prefix

Name	Type	Last modified	Size	Storage class
<input checked="" type="checkbox"/> cat.jpg	jpg	November 9, 2023, 10:51:25 (UTC+06:00)	26.7 KB	Standard

Share "cat.jpg" with a presigned URL

Presigned URLs are used to grant access to an object for a limited time. [Learn more](#)

Anyone can access the object with this presigned URL until it expires, even if the bucket and object are private.

Time interval until the presigned URL expires

Using the S3 console, you can share an object with a presigned URL for up to 12 hours or until your session expires. To create a presigned URL with a longer time interval, use the AWS CLI or AWS SDK. Time intervals for presigned URLs can be restricted by your IAM policy.

- Minutes
 Hours

Number of minutes

1

Must be a whole number between 1 and 720.

After you create the presigned URL, it's automatically copied to your clipboard.

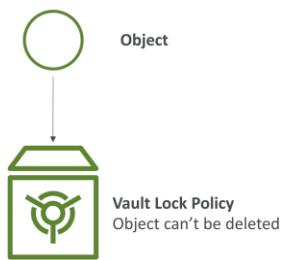
Cancel

Create presigned URL

NB: we create the preassigned URL for 1 min so after 1 min the user will not be able to access the file

S3 – Glacier Vault Lock

- Adopt a WORM (Write Once Read Many) model
- Create a Vault Lock Policy
- Lock the policy for future edits (can no longer be changed or deleted)
- Helpful for compliance and data retention



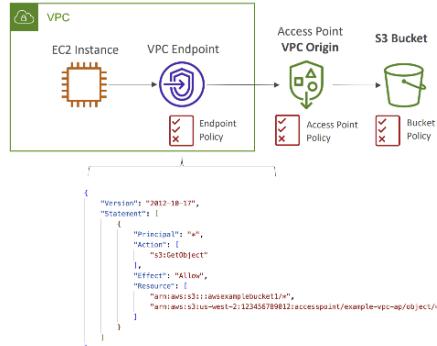
S3 – Object Lock (Versioning must be enabled)

- Adopt a WORM (Write Once Read Many) model
- Block an object version deletion for a specified amount of time
- **Retention mode - Compliance:**
 - Object versions can't be overwritten or deleted by any user, including the root user
 - Objects retention modes can't be changed, and retention periods can't be shortened
- **Retention mode - Governance:**
 - Most users can't overwrite or delete an object version or alter its lock settings
 - Some users have special permissions to change the retention or delete the object
- **Retention Period:** protect the object for a fixed period, it can be extended
- **Legal Hold:**
 - protect the object indefinitely, independent from retention period
 - can be freely placed and removed using the s3:PutObjectLegalHold IAM permission

S3 - Access point



- Access Points simplify security management for S3 Buckets
- Each Access Point has:
 - its own DNS name (Internet Origin or VPC Origin)
 - an access point policy (similar to bucket policy) – manage security at scale
- We can define the access point to be accessible only from within the VPC
- You must create a VPC Endpoint to access the Access Point (Gateway or Interface Endpoint)
- The VPC Endpoint Policy must allow access to the target bucket and Access Point



From Web console

Amazon S3 > Buckets > rajiv-9-nov

rajiv-9-nov [Info](#)

Objects Properties Permissions Metrics Management [Access Points](#)

Access Points (0)
Amazon S3 Access Points simplify managing data access at scale for shared datasets in S3. Access points are named network endpoints that are attached to buckets that you can use to perform S3 object operations. An Access Point alias provides the same functionality as an Access Point ARN and can be substituted for use anywhere an S3 bucket name is normally used for data access. [Learn more](#)

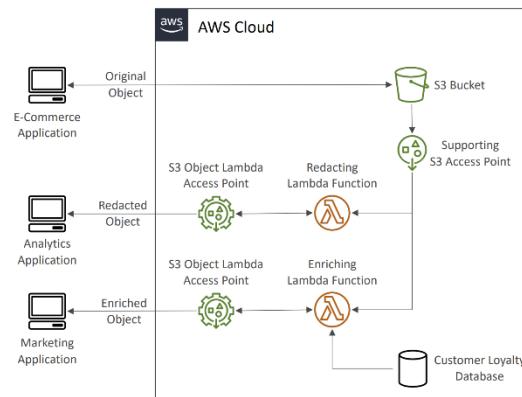
[Copy Access Point alias](#) [Copy ARN](#) [Edit policy](#) [Delete](#) [Create access point](#)

[Search for Access Points by name](#)

Name	Network origin	Access	Bucket owner account ID	Access Point alias
No Access Points	You don't have any access points for this bucket			
Create access point				

S3 – Object Access Lambda

- Use AWS Lambda Functions to change the object before it is retrieved by the caller application
- Only one S3 bucket is needed, on top of which we create S3 Access Point and S3 Object Lambda Access Points.
- Use Cases:
 - Redacting personally identifiable information for analytics or non-production environments.
 - Converting across data formats, such as converting XML to JSON.
 - Resizing and watermarking images on the fly using caller-specific details, such as the user who requested the object.



Lab: create, delete, copy and move data from local file to s3 bucket by using aws cli

show aws user list

```
#aws iam list-users
```

create bucket

```
#aws s3 mb s3://rajiv2011
```

See object from any specific bucket

```
# aws s3 ls s3://rajiv-11-nov
```

Delete/remove bucket

```
#aws s3 rb s3://bucket-name
```

Delete/remove bucket by forcing

```
#aws s3 rb s3://bucket-name --force
```

remove object from bucket

```
#aws s3 rm s3://bucket-name/example/filename.txt
```

remove all object from bucket

```
#aws s3 rm s3://bucket-name/example --recursive
```

move object from bucket to bucket

```
#aws s3 mv <source> <target> [--options]
```

example: aws s3 mv s3://bucket-name/example s3://my-bucket/

move file from your local pc to bucket

```
#aws s3 mv filename.txt s3://bucket-name
```

move file from bucket to your local pc

```
#aws s3 mv s3://bucket-name/filename.txt ./
```

copy from bucket to bucket

```
#aws s3 cp <source> <target> [--options]
```

example: aws s3 cp s3://bucket-name/example s3://my-bucket/

Ec2 instance describe

```
aws ec2 describe-instances --instance-ids i-0cee6d42c5748941c | more
```

NB: When we create a bucket from web it will be private

When we create a bucket from command line it will be public