

## 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 (99.999999999%, 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 (99.999999999%) 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 Amazon S3 'Upload' page for a bucket named 'rajiv2021'. The 'Files and folders' section shows one file, 'cat2.jpeg', with a size of 39.7 KB. The 'Destination' section shows the path 's3://rajiv2021'. The 'Permissions' section is expanded, and the 'Properties' section is highlighted with a red box. The 'Storage class' section shows a table with the 'Standard' class selected, indicated by a blue circle and a red box.

Storage class	Designed for	Availability Zones	Min storage duration
<input checked="" type="radio"/> Standard	Frequently accessed data (more than once a month) with milliseconds access	≥ 3	-

## Edit the storage class

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

The screenshot shows the 'Storage class' edit page. The 'Storage class' section shows the 'Standard' class. An 'Edit' button is visible in the top right corner.

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	Price class
<input type="radio"/> Standard	Frequently accessed data (more than once a month) with milliseconds access	≥ 3	-	-
<input type="radio"/> Intelligent-Tiering	Data with changing or unknown access patterns	≥ 3	-	-
<input checked="" type="radio"/> Standard-IA	Infrequently accessed data (once a month) with milliseconds access	≥ 3	30 days	1
<input type="radio"/> One Zone-IA	Recreateable, infrequently accessed data (once a month) stored in a single Availability Zone with milliseconds access	1	30 days	1
<input type="radio"/> Glacier Instant Retrieval	Long-lived archive data accessed once a quarter with instant retrieval in milliseconds	≥ 3	90 days	1
<input type="radio"/> Glacier Flexible Retrieval (formerly Glacier)	Long-lived archive data accessed once a year with retrieval of minutes to hours	≥ 3	90 days	-
<input type="radio"/> Glacier Deep Archive	Long-lived archive data accessed less than once a year with retrieval of hours	≥ 3	180 days	-
<input type="radio"/> Reduced redundancy	Noncritical, frequently accessed data with milliseconds access (not recommended as S3 Standard is more cost effective)	≥ 3	-	-

**Specified objects**

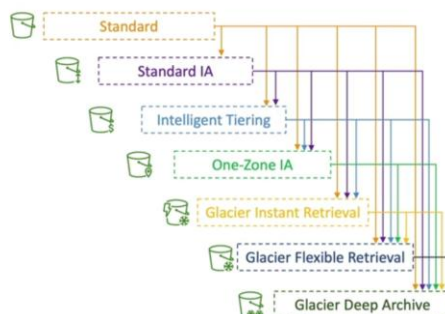
Find objects by name  < 1 >

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

Objects | Properties | Permissions | Metrics | **Management** | Access Points

**Lifecycle rules (0)**

Use lifecycle rules to define actions you want Amazon S3 to take during an object's lifetime such as transitioning objects to another storage class, archiving them, or deleting them after a specified period of time. Learn more

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.						
<div>Create lifecycle rule</div>						

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

## Create lifecycle rule [Info](#)

### Lifecycle rule configuration

#### Lifecycle rule name

s3-lifecycle-rule-1

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

Standard-IA	30	Remove
Intelligent-Tiering	60	Remove
Glacier Instant Retrieval	90	Remove
<div>Add transition</div>		

### Review transition and expiration actions

#### Current version actions

Day 0

- Objects uploaded



Day 30

- Objects move to Standard-IA



Day 60

- Objects move to Intelligent-Tiering



Day 90

- Objects move to Glacier Instant Retrieval

#### Noncurrent versions actions

Day 0

No actions defined.

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

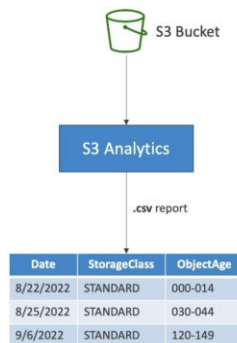
## Amazon S3 – Life cycle rules (Scenario-2)

- 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.**

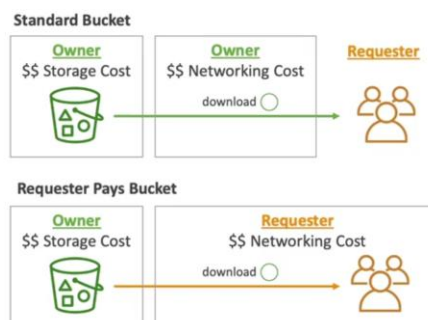
## Amazon S3 Analytics – Storage Class Analysis

- 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



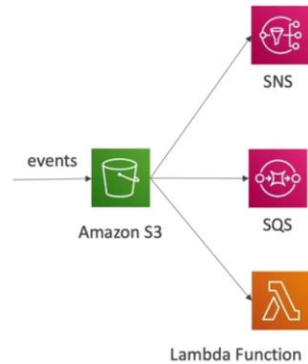
## S3 Requester pay.

- 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)

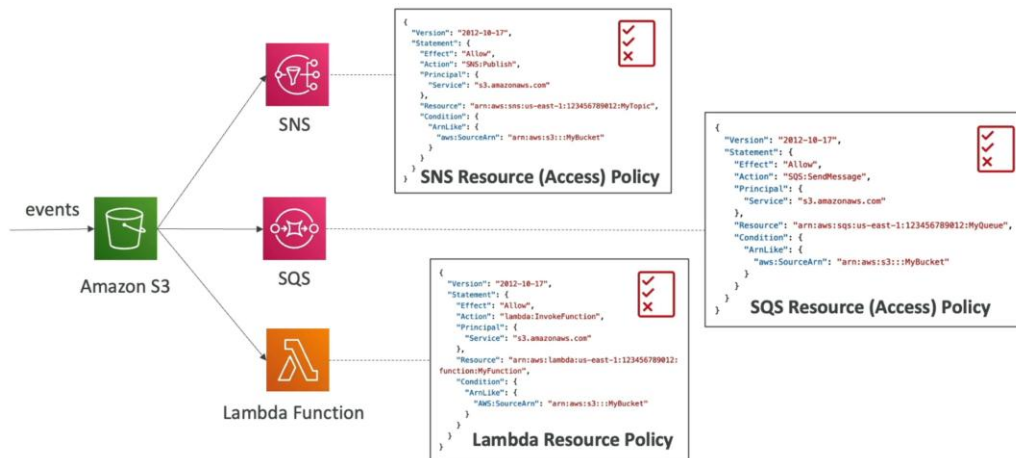


## S3 Event Notifications

- 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.

**Event notifications (0)** [Edit](#) [Delete](#) [Create event notification](#)

Send a notification when specific events occur in your bucket. [Learn more](#)

Name	Event types	Filters	Destination type	Destination
No event notifications				
Choose <a href="#">Create event notification</a> to be notified when a specific event occurs.				
<a href="#">Create event notification</a>				

**Amazon EventBridge**

[Edit](#)

Send notifications to Amazon EventBridge for all events in this bucket

Off

[Amazon S3](#) > [Buckets](#) > [rajiv-s3-sqs](#) > [Edit Amazon EventBridge](#)

**Edit Amazon EventBridge** [Info](#)

**Amazon EventBridge**

For additional capabilities, use Amazon EventBridge to build event-driven applications at scale using S3 event notifications. [Learn more](#) or see [EventBridge pricing](#)

Send notifications to Amazon EventBridge for all events in this bucket

☐ Off

☒ On

[Cancel](#) [Save changes](#)

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

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

**Event notifications (0)** [Edit](#) [Delete](#) [Create event notification](#)

Send a notification when specific events occur in your bucket. [Learn more](#)

Name	Event types	Filters	Destination type	Destination
No event notifications				
Choose <a href="#">Create event notification</a> to be notified when a specific event occurs.				
<a href="#">Create event notification</a>				

before doing anything create a SQS

give a SQS name and rest of thing keep same and click create que.



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

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

**Name**  
demo-sqs-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  
Should be between 0 seconds and 12 hours.  
Seconds

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

**Delivery delay** info  
0  
Seconds

**Maximum message size** info  
256  
KB

Amazon SQS > Queues > demo-sqs-1

## demo-sqs-1

EditDeletePurgeSend and receive messagesStart DLQ redrive

### Details

info

<b>Name</b> demo-sqs-1	<b>Type</b> Standard	<b>ARN</b> arn:aws:sqs:us-east-1:999838272208:demo-sqs-1
<b>Encryption</b> Amazon SQS key (SSE-SQS)	<b>URL</b> https://sqs.us-east-1.amazonaws.com/999838272208/demo-sqs-1	<b>Dead-letter queue</b> -
<a href="#">► More</a>		

SNS subscriptionsLambda triggersDead-letter queueMonitoringTaggingAccess policyEncryptionDead-letter queue redrive tasks

### Access policy (Permissions)

info

Edit

Define who can access your queue.

```
{
  "Version": "2012-10-17",
  "Id": "_default_policy_id",
  "Statement": [
    {
      "Sid": "_owner_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::999838272208:root"
      },
      "Action": "SQS:*",
      "Resource": "arn:aws:sqs:us-east-1:999838272208:demo-sqs-1"
    }
  ]
}
```

Now go to the event creation page and refresh it and give the name and select sqs and select the created sqs

## 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

Event name can contain up to 255 characters.

#### Prefix - *optional*

Limit the notifications to objects with key starting with specified characters.

#### Suffix - *optional*

Limit the notifications to objects with key ending with specified characters.

### 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

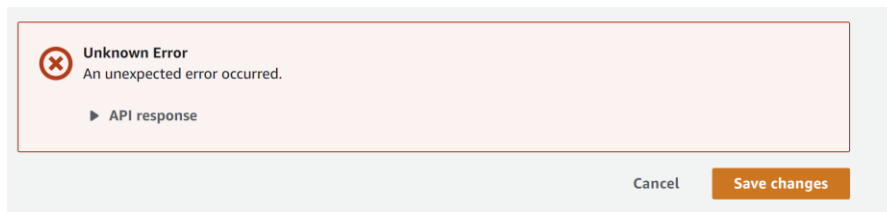
- ☒ Choose from your SQS queues
- ☐ Enter SQS queue ARN

#### SQS queue

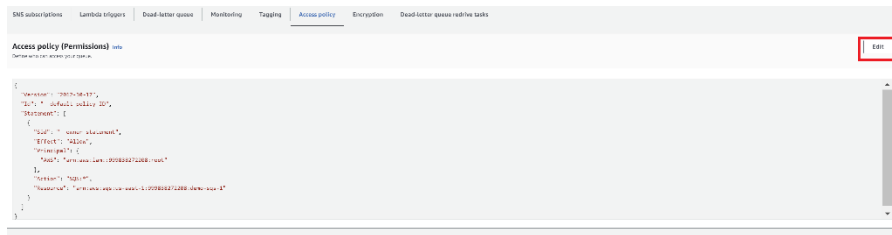
Cancel

Save changes

Click save changes and you will get an error



For fix this need to edit the access policy



Scroll down and click 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 (\*\*)

Use multiple statements to add permissions for more than one service.

Actions 1 Action(s) Selected ☐ All Actions (\*\*)

Amazon Resource Name (ARN) \$(Region):\${Account}:\${QueueName}

☐ ListQueueTags  
☐ ListQueues  
☐ PurgeQueue  
☐ ReceiveMessage  
☐ RemovePermission  
☒ SendMessage  
☐ SetQueueAttributes

valid. You must enter a valid ARN.

### Step 3: Generate Policy

now go to the policy page and copy the arn and paste it here



## 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 Endpoi 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

Use multiple statements to add permissions for more than one service.

☐ All Services ("\*")

Actions 1 Action(s) Selected ☐ All Actions ("\*")

Amazon Resource Name (ARN) arn:aws:sqs:us-east-1:9998

ARN should follow the following format: arn:aws:sqs:(Region):(Account):(QueueName).  
Use a comma to separate multiple values.

Add Conditions (Optional)

Add Statement

you copy this from previos page

#### Access policy [Info](#)

Define who can access your queue.

```
1 {
2   "Version": "2012-10-17",
3   "Id": "__default_policy_10",
4   "Statement": [
5     {
6       "Sid": "__owner_statement",
7       "Effect": "Allow",
8       "Principal": {
9         "AWS": "arn:aws:iam::999838272208:root"
10      },
11       "Action": "SQS:*",
12       "Resource": "arn:aws:sqs:us-east-1:999838272208:demo-sqs-1"
13     }
14   ]
15 }
```

[Policy generator](#)

## After clicking Add Statement

Effect ☒ Allow ☐ Deny

Principal \*

Use a comma to separate multiple values.

AWS Service Amazon SQS

Use multiple statements to add permissions for more than one service.

☐ All Services ("\*")

Actions -- Select Actions -- ☐ All Actions ("\*")

Amazon Resource Name (ARN) \*

ARN should follow the following format: arn:aws:sqs:(Region):(Account):(QueueName).  
Use a comma to separate multiple values.

Add Conditions (Optional)

Add Statement

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

Principal(s)	Effect	Action	Resource	Conditions
* *	Allow	* sqs: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.

Generate Policy

Start Over

Use a comma to separate multiple values.

AWS Service: Amazon SQS ☐ All Services (\*)

Use multiple statements to add permissions for more than one service.

**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": [
        "sqs: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

Close

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

**Access policy** [info](#)

Define who can access your queue.

```

1 {
2   "Id": "Policy1686897956937",
3   "Version": "2012-10-17",
4   "Statement": [
5     {
6       "Sid": "Stmt1686897917347",
7       "Action": [
8         "sqs:SendMessage"
9       ],
10      "Effect": "Allow",
11      "Resource": "arn:aws:sqs:us-east-1:999838272208:demo-sqs-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.

You can add up to 50 tags.

Cancel

Now go to the S3 page and click save changes and now you will not get the error

### 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**

demo-sqs-1

**Unknown Error**  
An unexpected error occurred.

API response

Cancel

Save changes

Now go to the sqs page and click send and receive messages then scroll down

Amazon SQS > Queues > demo-sqs-1

demo-sqs-1

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

**Details** info

Name	Type	ARN
demo-sqs-1	Standard	arn:aws:sqs:us-east-1:999838272208:demo-sqs-1
Encryption	URL	Dead-letter queue
Amazon SQS key (SSE-SQS)	https://sqs.us-east-1.amazonaws.com/999838272208/demo-sqs-1	-

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": [
    {
      "Sid": "Stmt1668807917547",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "sqs:SendMessage",
      "Resource": "arn:aws:sqs:us-east-1:999838272208:demo-sqs-1"
    }
  ]
}

```

Edit

**Send message** info

Clear content Send message

**Message body**  
Enter the message to send to the queue.  
Enter message

**Delivery delay** info  
0 Seconds  
Should be between 0 seconds and 15 minutes.

Message attributes - Optional info

**Receive messages** info

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

Messages available	Polling duration	Maximum message count	Polling progress
1	30	10	0%

Messages (0)

Search messages

View details Delete

ID	Sent	Size	Receive count
No messages. To view messages in the queue, poll for messages.			

Poll for messages

Now click the message and you can see the message after read click the done and delete the message

Receive messages [info](#) [Edit poll settings](#) [Stop polling](#) [Poll for messages](#)

Messages available: 1    Polling duration: 30    Maximum message count: 10    Polling progress: 1 message(s) received

Messages (1) [View details](#) [Delete](#)

<input checked="" type="checkbox"/>	ID	Sent	Size	Receive count
<input checked="" type="checkbox"/>	455dc614-8ada-40ca-b839-24dab1baeb20	Jun 16, 2023, 12:50:40 GMT+6	223 bytes	1

Message: 455dc614-8ada-40ca-b839-24dab1baeb20 [×](#)

[Body](#) [Attributes](#) [Details](#)

```
{ "Service": "Amazon S3", "Event": "s3:TestEvent", "Time": "2023-06-16T06:50:39.960Z", "Bucket": "rajiv-s3-sqs", "RequestId": "GTTVVPBGVHAKGS8", "HostId": "oNTQkTbJtm4V39aN9jRgGjAjL3YVntZF0o3hLoh6OOgk8dZUuGpba+XWj5eMKLGArk1U1F+Fe4=" }
```

[Done](#)

Receive messages [info](#) [Edit poll settings](#) [Stop polling](#) [Poll for messages](#)

Messages available: 1    Polling duration: 30    Maximum message count: 10    Polling progress: 1 message(s) received

Messages (1) [View details](#) [Delete](#)

<input checked="" type="checkbox"/>	ID	Sent	Size	Receive count
<input checked="" type="checkbox"/>	455dc614-8ada-40ca-b839-24dab1baeb20	Jun 16, 2023, 12:50:40 GMT+6	223 bytes	1

Now go to s3 page and upload an image then go to sql notification page and pull the message

Amazon S3 [buckets](#) [rajiv-s3-sqs](#)

rajiv-s3-sqs [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 [to get a list of all objects in your bucket](#).

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	cat.jpg	jpg	June 16, 2023, 12:58:50 (UTC+06:00)	4.7 KB	Standard

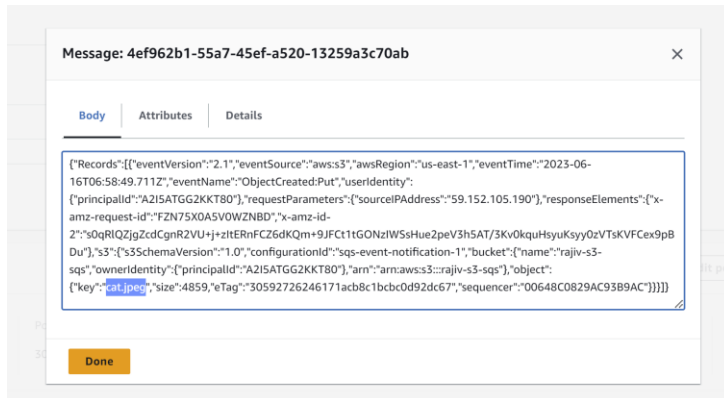
Receive messages [info](#) [Edit poll settings](#) [Stop polling](#) [Poll for messages](#)

Messages available: 0    Polling duration: 30    Maximum message count: 10    Polling progress: 1 message(s) received

Messages (1) [View details](#) [Delete](#)

<input type="checkbox"/>	ID	Sent	Size	Receive count
<input type="checkbox"/>	4a9f92b1-35a7-43af-a120-13239a3c70ab	Jun 16, 2023, 12:58:50 GMT+6	737 bytes	1





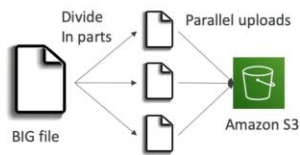
After reading the message delete it

### 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

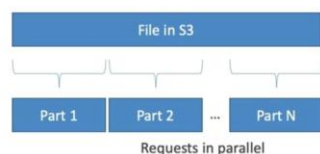
- Multi-Part upload:
  - recommended for files > 100MB, must use for files > 5GB
  - Can help parallelize uploads (speed up transfers)
- S3 Transfer Acceleration
  - Increase transfer speed by transferring file to an AWS edge location which will forward the data to the S3 bucket in the target region
  - Compatible with multi-part upload



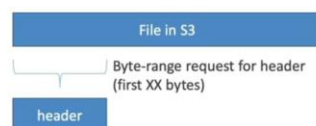
### S3 performance – S3 Byte-Range Fetches

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

Can be used to speed up downloads

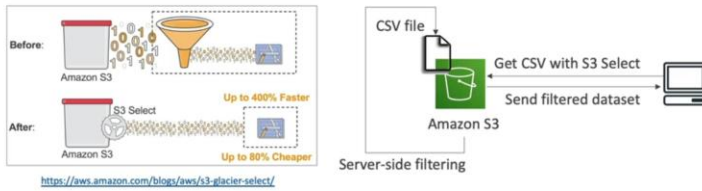


Can be used to retrieve only partial data (for example the head of a file)



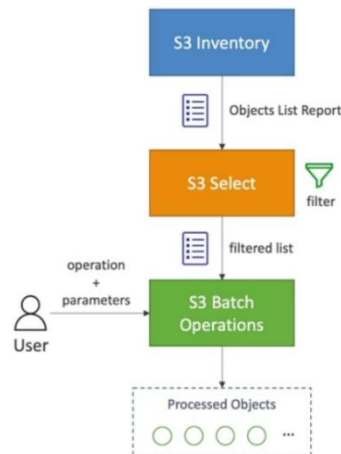
## S3 Select and Glacier Select

- 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

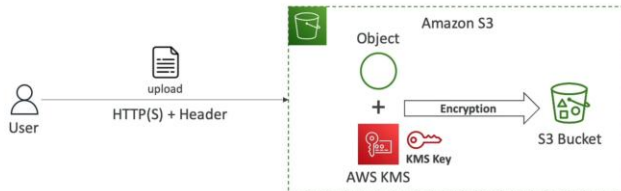
## S3 Encryption – SSE-S3

- 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"



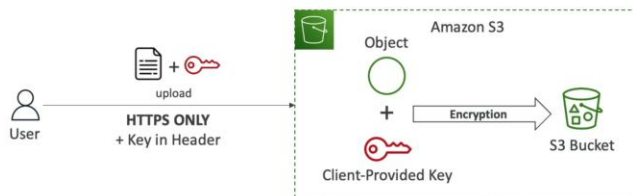
### SSE-KMS Limitation

- 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



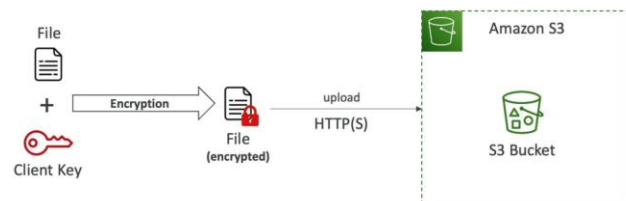
## S3 Encryption – SSE-C

- 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 be 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

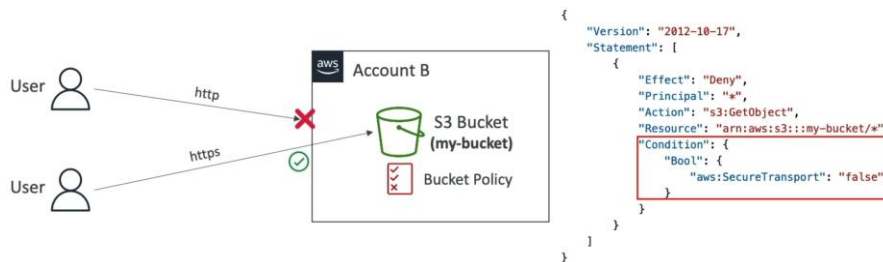


## Amazon S3 Encryption in transit (SSL/TSL)

- 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

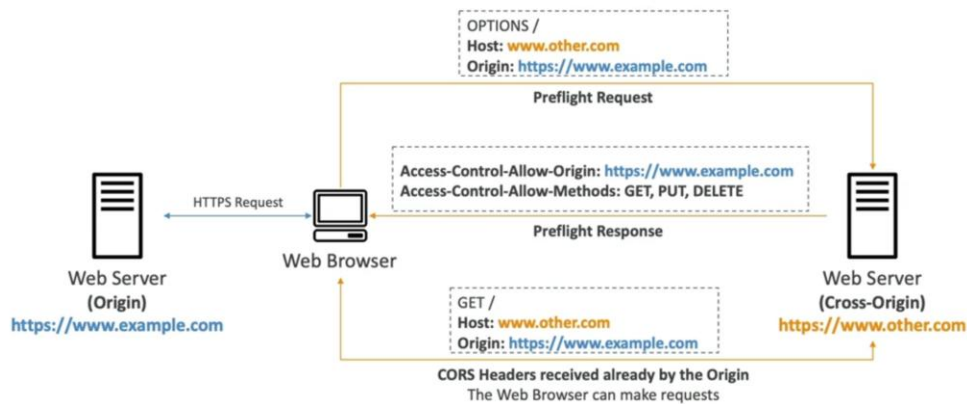
- SSE-S3 encryption is automatically applied to new objects stored in S3 bucket
- Optionally, you can “force encryption” using a bucket policy and refuse any API call to PUT an S3 object without encryption headers (SSE-KMS or SSE-C)

```
{
  "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"
        }
      }
    }
  ]
}
```

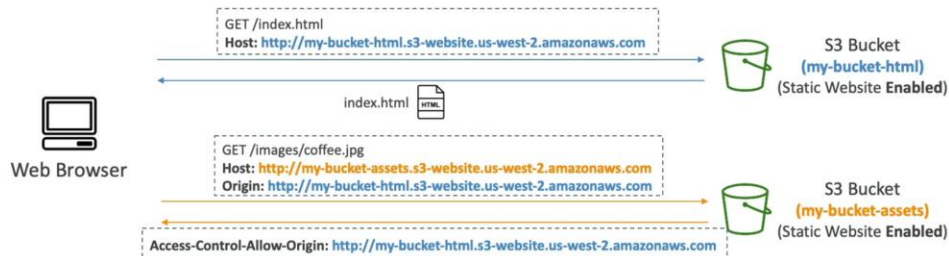
## CORS (Cross Origin Resource Sharing)

- 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

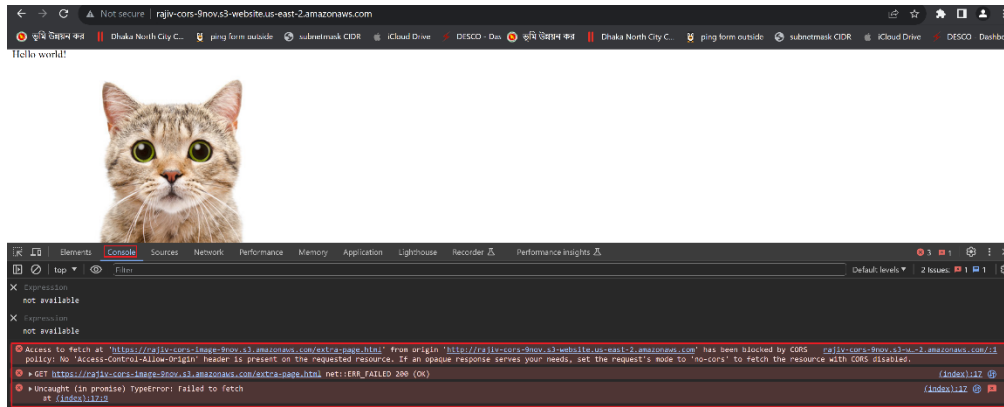
Step:3

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 2<sup>nd</sup> 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

### Cross-origin resource sharing (CORS)

The CORS configuration, written in JSON, defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. [Learn more](#)

[Edit](#)

Amazon S3 > Buckets > rajiv-cors-image-9nov > Edit cross-origin resource sharing (CORS)

### Edit cross-origin resource sharing (CORS) [info](#)

**Cross-origin resource sharing (CORS)**  
The CORS configuration, written in JSON, defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. [Learn more](#)

```

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 ]

```

make sure dont put any / here

JSON Ln 10, Col 71 0 Errors 0 Warnings 0

[Cancel](#) [Save changes](#)

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