

Simple Queue Service (SQS)

Amazon Simple Queue Service (SQS) is a fully managed message queuing service provided by Amazon Web Services (AWS). It enables decoupling of the components of a cloud application by allowing them to communicate asynchronously. SQS acts as a mediator for different parts of a distributed system, allowing messages to be sent between various components without direct connections between them.

Here are some key features and concepts related to Amazon SQS:

1. Message Queues:

- SQS uses message queues to transmit any volume of data, at any level of throughput, without the need for additional infrastructure.
- Messages can be up to 256 KB in size.

2. Queue Types:

- **Standard Queue:** Offers a high throughput, best-effort delivery, and messages are delivered at least once. However, occasional duplication of messages is possible.
- **FIFO Queue (First-In-First-Out):** Guarantees that messages are processed exactly once and in the order they are sent.

3. Message Lifecycle:

- Messages are sent to a queue by a producer and then retrieved and processed by a consumer.
- Once a message is successfully processed, it can be deleted from the queue.
- If a message is not deleted, it will eventually become visible again for retrieval after a specified visibility timeout period.

4. Visibility Timeout:

- When a message is retrieved from the queue, it becomes invisible to other consumers for a specified duration called the visibility timeout.
- This allows the consumer enough time to process the message without worrying about other consumers processing the same message.

5. Scaling:

- SQS automatically scales to handle the volume of messages without requiring intervention.
- It can be used to decouple the components of a cloud application, allowing each component to scale independently.

6. Retention Period:

- Messages are retained in the queue for a specified duration (the retention period) from the time of message enqueue.

7. Dead-Letter Queues:

- Provides a mechanism to capture and store messages that cannot be processed successfully after a certain number of attempts.
- Helps in troubleshooting and debugging.

8. Integration with Other AWS Services:

- SQS can be easily integrated with other AWS services such as AWS Lambda, Amazon EC2, Amazon S3, and more.

9. Security and Access Control:

- SQS provides fine-grained access control using AWS Identity and Access Management (IAM).
- Messages can be encrypted in transit using HTTPS and at rest using AWS Key Management Service (KMS).

Amazon SQS is a reliable, scalable, and fully managed service that facilitates the building of distributed and decoupled systems in the AWS cloud environment. It is commonly used in scenarios such as task scheduling, event notification, and handling asynchronous workflows in applications.

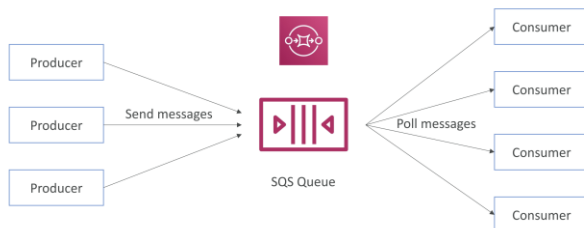
Basic information

- When we start deploying multiple applications, they will inevitably need to communicate with one another
- There are two patterns of application communication



- Synchronous between applications can be problematic if there are sudden spikes of traffic
- What if you need to suddenly encode 1000 videos but usually it's 10?
- In that case, it's better to **decouple** your applications,
 - using SQS: queue model
 - using SNS: pub/sub model
 - using Kinesis: real-time streaming model
- These services can scale independently from our application!

What is Queue:



SQS- Standard Queue

- Oldest offering (over 10 years old)
- Fully managed service, used to **decouple applications**
- Attributes:
 - Unlimited throughput, unlimited number of messages in queue
 - Default retention of messages: 4 days, maximum of 14 days
 - Low latency (<10 ms on publish and receive)
 - Limitation of 256KB per message sent
- Can have duplicate messages (at least once delivery, occasionally)
- Can have out of order messages (best effort ordering)

SQS- Producing Message

- Produced to SQS using the SDK (SendMessage API)
- The message is **persisted** in SQS until a consumer deletes it
- Message retention: default 4 days, up to 14 days

- Example: send an order to be processed

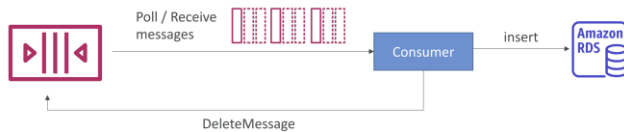
- Order id
- Customer id
- Any attributes you want



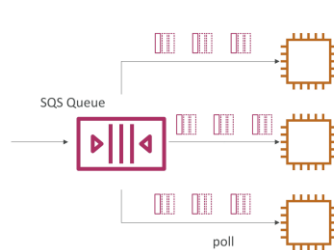
- SQS standard: unlimited throughput

SQS- Consuming Message

- Consumers (running on EC2 instances, servers, or AWS Lambda)...
- Poll SQS for messages (receive up to 10 messages at a time)
- Process the messages (example: insert the message into an RDS database)
- Delete the messages using the DeleteMessage API

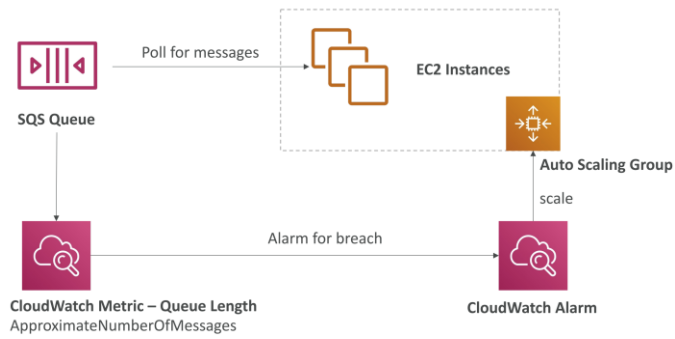


SQS - Multiple Ec2 Instances Consumers

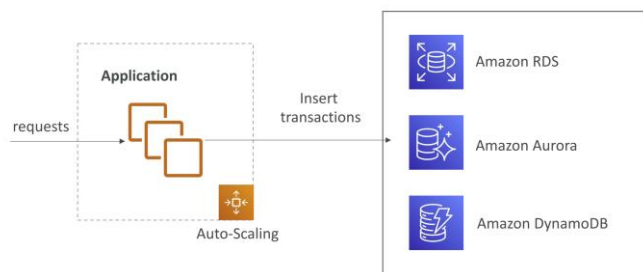


- Consumers receive and process messages in parallel
- At least once delivery
- Best-effort message ordering
- Consumers delete messages after processing them
- We can scale consumers horizontally to improve throughput of processing

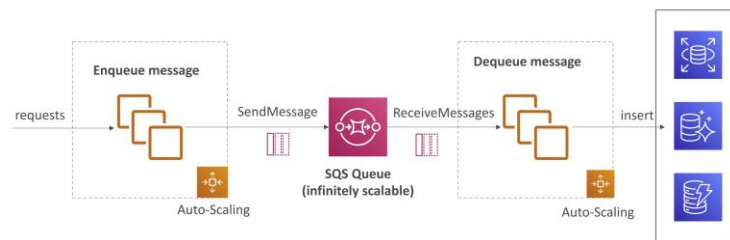
SQS – with Auto scaling Group (ASG)



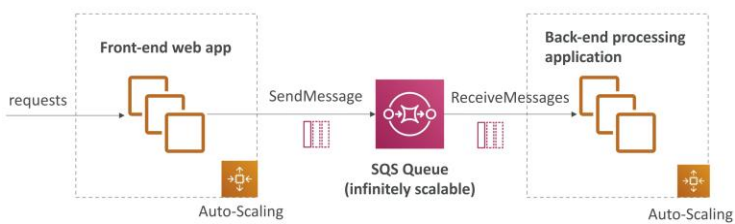
SQS – if the load is too big, some transactions may be lost



SQS – as a buffer to database writes



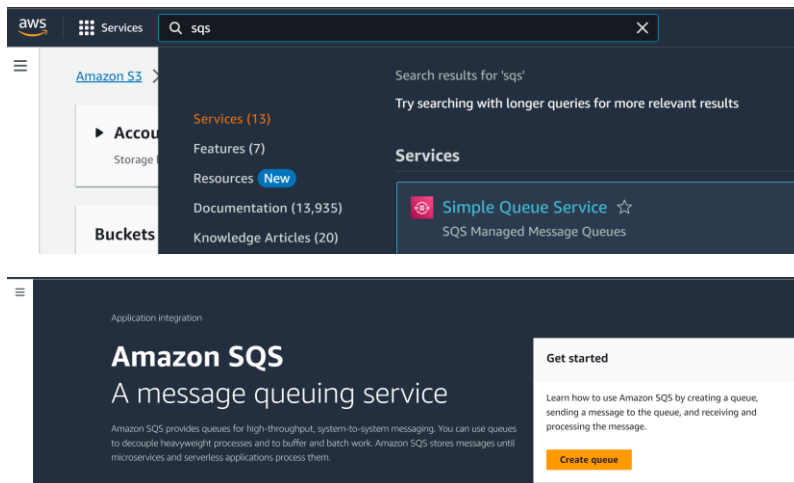
SQS – to decouple between applications tiers:



SQS – Security

- **Encryption:**
 - In-flight encryption using HTTPS API
 - At-rest encryption using KMS keys
 - Client-side encryption if the client wants to perform encryption/decryption itself
- **Access Controls:** IAM policies to regulate access to the SQS API
- **SQS Access Policies** (similar to S3 bucket policies)
 - Useful for cross-account access to SQS queues
 - Useful for allowing other services (SNS, S3...) to write to an SQS queue

Lab: create SQS then send and pull messages





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

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

Encryption [Info](#)

Amazon SQS provides in-transit encryption by default. To add at-rest encryption to your queue, enable server-side encryption.

Server-side encryption

- ☐ Disabled
- ☒ Enabled

Encryption key type

☒ **Amazon SQS key (SSE-SQS)**

An encryption key that Amazon SQS creates, manages, and uses for you.

☐ **AWS Key Management Service key (SSE-KMS)**

An encryption key protected by AWS Key Management Service (AWS KMS).

Access policy [Info](#)

Define who can access your queue.

Choose method

☒ Basic

Use simple criteria to define a basic access policy.

☐ Advanced

Use a JSON object to define an advanced access policy.

Define who can send messages to the queue

☒ Only the queue owner

Only the owner of the queue can send messages to the queue.

☐ Only the specified AWS accounts, IAM users and roles

Only the specified AWS account IDs, IAM users and roles can send messages to the queue.

Define who can receive messages from the queue

☒ Only the queue owner

Only the owner of the queue can receive messages from the queue.

☐ Only the specified AWS accounts, IAM users and roles

Only the specified AWS account IDs, IAM users and roles can receive messages from the queue.

JSON (read-only)

```
{
  "Version": "2012-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__owner_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "999838272208"
      },
      "Action": [
        "SQS:*"
      ],
      "Resource": "arn:aws:sqs:us-east-1:999838272208:sqs-11-nov"
    }
  ]
}
```

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.

Key

Value - *optional*

You can add 49 more tags.

Cancel

Create queue

After creating SQS

Amazon SQS > Queues > sqs-11-nov

sqs-11-nov

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

Details Info

Name	Type	ARN
sq-11-nov	Standard	arn:aws:sqs:us-east-1:999838272208:sqs-11-nov
Encryption	URL	Dead-letter queue
Amazon SQS key (SSE-SQS)	https://sqs.us-east-1.amazonaws.com/999838272208/sqs-11-nov	-

► More

After click send and receive message we the following screen now we write a message and send it now if go down and we can see the message count 1 and if we wait some time we can see the count is increase because as long as we read and delete the message it will sending. Now pull and delete the message now again pull the message and we can see now no new message is coming.

Amazon SQS > Queues > sqs-11-nov > Send and receive messages

Send and receive messages

Send messages to and receive messages from a queue.

Send message Info

Clear content **Send message**

Message body
Enter the message to send to the queue.

hello rajiv

Delivery delay Info

0 Seconds ▼

Should be between 0 seconds and 15 minutes.

► Message attributes - Optional Info

After click send message

Send and receive messages

Send messages to and receive messages from a queue.

Send message [Info](#)

[Clear content](#)

[Send message](#)

✔ Your message has been sent and is ready to be received.

[View details](#)

✕

Message body

Enter the message to send to the queue.

hello rajiv

Delivery delay [Info](#)

0

Seconds

Should be between 0 seconds and 15 minutes.

► [Message attributes - Optional](#) [Info](#)

Now scroll down for getting the message and also pull 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

0 receives/second 0%

Messages (0)

[View details](#)

[Delete](#)

Q Search messages

< 1 > ⚙

☐

ID

▼

Sent

▲

Size

▼

Receive count

▼

No messages. To view messages in the queue, poll for messages.

[Poll for messages](#)

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)

[View details](#)

[Delete](#)

Q Search messages

< 1 > ⚙

☐

ID

▼

Sent

▲

Size

▼

Receive count

▼

☐

55542f70-36b8-40c1-aa9c-e88b801d59ef

2023-11-11T22:44+06:00

11 bytes

1

After click the message

Message: 55542f70-36b8-40c1-aa9c-e88b801d59ef

×

Body

Attributes

Details

hello rajiv

⌵

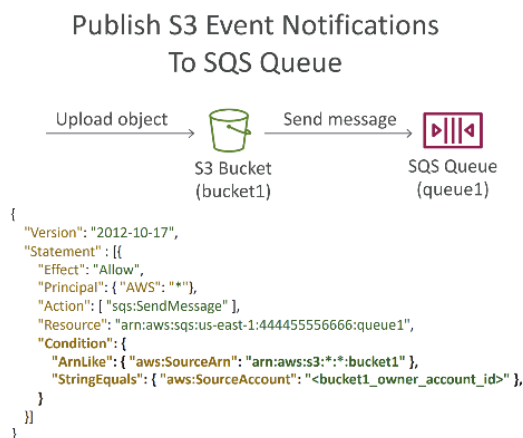
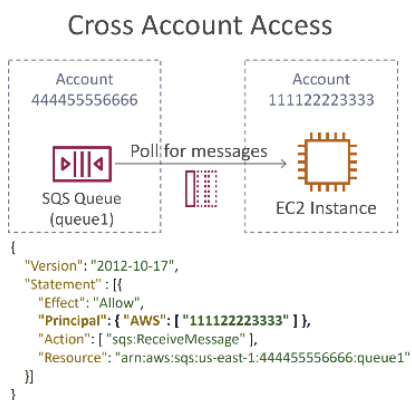
Done

⌵

After that if you did not delete the message and refresh the page it will send message again

So delete the message then it will not send any message again.

SQS-Queue Access policy



Lab: SQS access policy

Create one bucket

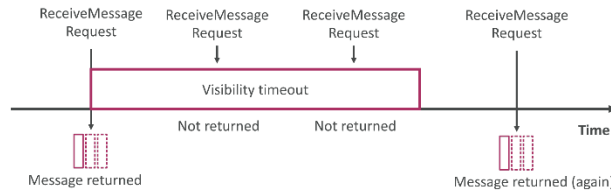
Create a SQS and create a access policy to allow that bucket to send messages to this SQS

now if we upload or download any image to this bucket it will send a message to this SQS

We already see the lab in S3 class

SQS – Message visibility Timeout

- After a message is polled by a consumer, it becomes **invisible** to other consumers
- By default, the "message visibility timeout" is **30 seconds**
- That means the message has 30 seconds to be processed
- After the message visibility timeout is over, the message is "visible" in SQS



- If a message is not processed within the visibility timeout, it will be processed **twice**
- A consumer could call the **ChangeMessageVisibility** API to get more time
- If visibility timeout is high (hours), and consumer crashes, re-processing will take time
- If visibility timeout is too low (seconds), we may get duplicates

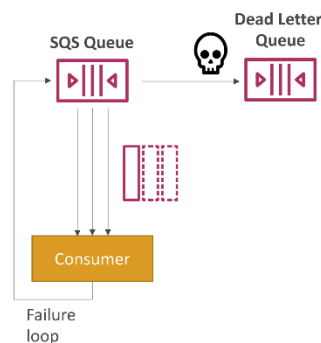
Lab: SQS visibility timeout

Create an SQS and open the SQS in two windows now send a message from the first window and click pull message and go to the second window click pull message and we can see the message is not showing now go to the first window and click the stop pulling and go back to second window and we can see the message is show here now.

If we want to change the visibility time out edit the SQS visibility timeout and by default it set 30 seconds.

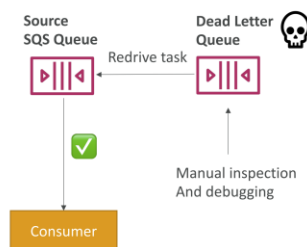
SQS – Dead Letter Queue (DLQ)

- If a consumer fails to process a message within the Visibility Timeout... the message goes back to the queue!
- We can set a threshold of how many times a message can go back to the queue
- After the **MaximumReceives** threshold is exceeded, the message goes into a Dead Letter Queue (DLQ)
- Useful for debugging!
- DLQ of a FIFO queue must also be a FIFO queue
- DLQ of a Standard queue must also be a Standard queue
- Make sure to process the messages in the DLQ before they expire:
 - Good to set a retention of 14 days in the DLQ



SQS DLQ – Redrive to Source

- Feature to help consume messages in the DLQ to understand what is wrong with them
- When our code is fixed, we can redrive the messages from the DLQ back into the source queue (or any other queue) in batches without writing custom code



LAB: Dead Letter Queue (DLQ)

First create SQS (sqs-dlq-save) with 14 days retention period as shown below image and don't change anything just create the SQS.

Amazon SQS × [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

sqs-dlq-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](#)

14 Days

Should be between 1 minute and 14 days.

Delivery delay [info](#)

0 Seconds

Should be between 0 seconds and 15 minutes.

Maximum message size [info](#)

256 KB

Should be between 1 KB and 256 KB.

Receive message wait time [info](#)

0 Seconds

Should be between 0 and 20 seconds.

Access policy [info](#)

Define who can access your queue.

Choose method

☒ **Basic**
Use simple criteria to define a basic access policy.

☐ **Advanced**
Use a JSON object to define an advanced access policy.

Define who can send messages to the queue

☒ **Only the queue owner**
Only the owner of the queue can send messages to the queue.

☐ **Only the specified AWS accounts, IAM users and roles**
Only the specified AWS account IDs, IAM users and roles can send messages to the queue.

Define who can receive messages from the queue

☒ **Only the queue owner**
Only the owner of the queue can receive messages from the queue.

☐ **Only the specified AWS accounts, IAM users and roles**
Only the specified AWS account IDs, IAM users and roles can receive messages from the queue.

JSON (read-only)

```
{
  "Id": "_default_policy_ID",
  "Statement": [
    {
      "Sid": "_owner_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "999838272288"
      },
      "Action": [
        "SQS:*"
      ],
      "Resource": "arn:aws:sqs:us-east-1:999838272288:sqs-dlq-1"
    }
  ]
}
```

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.

Key	Value - optional	
<input type="text" value="Enter key"/>	<input type="text" value="Enter value"/>	<input type="button" value="Remove"/>
<input type="button" value="Add new tag"/>		

You can add 49 more tags.

Cancel

Now create another SQS (sqs-main) and edit it set visibility timeout 5 and scroll down and enable dead letter que and chose where we save the dead letter queue and we will the SQS just we create before

which is SQS (sqs-dlq-save) and maximum receive 3 as shown below image

Amazon SQS

Queue sqs-dlq-1 created successfully
You can now send and receive messages.

Queues

Amazon SQS > Queues > sqs-dlq-1

sqs-dlq-1

Edit Delete Purge Send and receive messages Start DLQ redrive

Details Info

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

► More

Amazon SQS

Amazon SQS > Queues > sqs-dlq-1 > Edit

Edit sqs-dlq-1

Details

Name	Type
sqs-dlq-1	Standard

Configuration Info

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

Visibility timeout Info

5 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

Should be between 0 seconds and 15 minutes.

Maximum message size Info

256 KB

Should be between 1 KB and 256 KB.

Receive message wait time Info

0 Seconds

Should be between 0 and 20 seconds.

Encryption Info

Amazon SQS provides in-transit encryption by default. To add at-rest encryption to your queue, enable server-side encryption.

Server-side encryption

☐ Disabled

☒ Enabled

Encryption key type

☒ Amazon SQS key (SSE-SQS)

An encryption key that Amazon SQS creates, manages, and uses for you.

☐ AWS Key Management Service key (SSE-KMS)

An encryption key protected by AWS Key Management Service (AWS KMS).

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": "__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:sqs-dlq-1"
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

Choose queue
arn:aws:sqs:us-east-1:999838272208:SQS-dlq-save

Maximum receives
3
Should be between 1 and 1000

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 50 more tags.

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

Now go to the SQS (sqs-dlq-save) and click send and pull message and then scroll down and click pull message and then go back to main SQS (sqs-main) and send a message and pull the message and wait for 3 messages is come and now stop pull message and click the pull message again we see now no message is coming.

Amazon SQS > [Queues](#) > sqs-dlq-1

sqs-dlq-1

[Edit](#) [Delete](#) [Purge](#) [Send and receive messages](#) [Start DLQ redrive](#)

Details [Info](#)

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

[More](#)

Send and receive messages

Send messages to and receive messages from a queue.

Send message [Info](#)

[Clear content](#) [Send message](#)

Message body
Enter the message to send to the queue.

Delivery delay [Info](#)
 Seconds
Should be between 0 seconds and 15 minutes.

[▶ Message attributes - Optional](#) [Info](#)

Scroll down

Receive messages [Info](#)

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

Messages available	Polling duration	Maximum message count	Polling progress
1	30	10	<div><div></div>0%</div> 0 receives/second

Messages (0) [View details](#) [Delete](#)

IDSentSizeReceive count

No messages. To view messages in the queue, poll for messages.

[Poll for messages](#)

Now click stop polling and we see the receive count

Receive messages [Info](#)

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

Messages available	Polling duration	Maximum message count	Polling progress
0	30	10	<div><div></div>0.1 receives/second</div>

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

IDSentSizeReceive count

☐

[7a6640a0-0763-4cad-bafd-6250c59d698f](#)

2023-11-15T23:16+06:00

11 bytes

3

Now poll for message again we can the message is gone

Receive messages [Info](#)

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

Messages available	Polling duration	Maximum message count	Polling progress
0	30	10	<div><div></div>0 receives/second</div>

Messages (0) [View details](#) [Delete](#)

IDSentSizeReceive count

No messages. To view messages in the queue, poll for messages.

[Poll for messages](#)

Now go to DLQ SQS where our message is saved pull for message and click the message and we see the message

Amazon SQS > Queues

Queues (2)

Search queues by prefix

	Name	Type	Created	Messages available	Messages in flight	Encryption	Content-based deduplication
<input type="radio"/>	sqs-dlq-1	Standard	2023-11-15T16:59+06:00	0	0	Amazon SQS key (SSE-SQS)	-
<input type="radio"/>	SQS-dlq-save	Standard	2023-11-15T17:04+06:00	1	0	Amazon SQS key (SSE-SQS)	-

Amazon SQS > Queues > SQS-dlq-save

SQS-dlq-save

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

Details Info

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

► More

Scroll down

Receive messages Info

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

Messages available: 1 Polling duration: 30 Maximum message count: 10 Polling progress: 0% (0 receives/second)

Messages (0)

Search messages

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

Poll for messages

Receive messages Info

Edit poll settings Stop polling Poll for messages

Messages available: 1 Polling duration: 30 Maximum message count: 10 Polling progress: 23% (1 receives/second)

Messages (1)

Search messages

ID	Sent	Size	Receive count
7a6640a0-0763-4cad-bafd-6250c59d698f	2023-11-15T23:16+06:00	11 bytes	4

Now click the message and we can see that message

Message: 7a6640a0-0763-4cad-bafd-6250c59d698f



Body

Attributes

Details

hello rajiv

Done

Now retrieve my DQL message

Go to DLQ SQS

[Amazon SQS](#) > [Queues](#) > [SQS-dlq-save](#) > Send and receive messages

Send and receive messages

Send messages to and receive messages from a queue.

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](#)

[Amazon SQS](#) > [Queues](#) > [SQS-dlq-save](#)

SQS-dlq-save

[Edit](#)

[Delete](#)

[Purge](#)

[Send and receive messages](#)

[Start DLQ redrive](#)

Details [Info](#)

Name

📄 SQS-dlq-save

Type

Standard

ARN

📄 arn:aws:sqs:us-east-1:999838272208:SQS-dlq-save

Encryption

Amazon SQS key (SSE-SQS)

URL

📄 <https://sqs.us-east-1.amazonaws.com/999838272208/SQS-dlq-save>

Dead-letter queue

-

► [More](#)

Amazon SQS > Queues > [SQS-dlq-save](#) > DLQ redrive

Dead-letter queue redrive [Info](#)

Redrive messages from this dead-letter queue to the selected destination queue

Redrive configuration

Message destination [Info](#)

☒ **Redrive to source queue(s)**
Redrive messages to their respective source queues

☐ **Redrive to a custom destination**
Redrive messages to the selected destination queue

Select an existing queue

Select a queue

Velocity control settings [Info](#)

☒ **System optimized**
Redrive messages with SQS optimized maximum number of messages per second

☐ **Custom max velocity**
Redrive messages with a custom maximum rate of messages per second. Maximum allowed rate is 500 messages per second

Messages per second

Inspect messages - Optional

[Info](#)

Edit poll settings

Stop polling

Poll for messages

Messages available	Polling duration	Maximum message count	Polling progress
1	30	10	<div>0 receives/second</div> 0%

Messages (0)

Q Search messages

< 1 > ⚙

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

Poll for messages

Cancel

DLQ redrive

After clicking the DLQ go back to main SQS and pull the message and we see the message is showing now.

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

Dead-letter queue redrive status [Info](#)

View the status of the most recent redrive task for this queue

Date started
2023-11-15T23:34+06:00

Redrive destination
Source queue(s)

Percent processed
100%

Status

🟢 Successfully completed

Cancel redrive

Now go back to main SQS

Amazon SQS > Queues > sqs-dlq-1

sqs-dlq-1

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

Details info

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

► More

Click pull messages and we can see the message is retrieve

Receive messages info

Edit poll settings Stop polling Poll for messages

Messages available: 1 Polling duration: 30 Maximum message count: 10 Polling progress: 0.2 receives/second 30%

Messages (1)

Search messages

ID	Sent	Size	Receive count
28e8ad20-3dc9-44dd-b261-36ab7a346683	2023-11-15T23:34+06:00	11 bytes	2

SQS – Delay Queue

- Delay a message (consumers don't see it immediately) up to 15 minutes
- Default is 0 seconds (message is available right away)
- Can set a default at queue level
- Can override the default on send using the `DelaySeconds` parameter



LAB: Delay Queue

Create SQS with 10 sec delay and after creating the delay and send a message and pull the message then we can see we need to wait 10 sec to get the message.

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

delayqueue

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

10 Seconds

Should be between 0 seconds and 15 minutes.

Maximum message size info

256 KB

Should be between 1 KB and 256 KB.

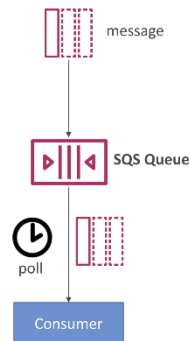
Receive message wait time info

0 Seconds

Should be between 0 and 30 seconds.

SQS - Long polling

- When a consumer requests messages from the queue, it can optionally "wait" for messages to arrive if there are none in the queue
- This is called Long Polling
- LongPolling decreases the number of API calls made to SQS while increasing the efficiency and reducing latency of your application
- The wait time can be between 1 sec to 20 sec (20 sec preferable)
- Long Polling is preferable to Short Polling
- Long polling can be enabled at the queue level or at the API level using `WaitTimeSeconds`



Lab: Long polling

when create a SQS we set receive message wait time 20 sec and click pull the message queue and now within this time if the producer sends any message then user don't need to wait the message will come right way.

So, after set Received message wait time scroll down and click the pull message button now scroll up and send a message and we can see the message comes right way.

Amazon SQS > Queues > delayqueue > Edit

Edit delayqueue

Details

Name: delayqueue Type: Standard

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

Should be between 0 seconds and 15 minutes.

Maximum message size info

256 KB

Should be between 1 KB and 256 KB.

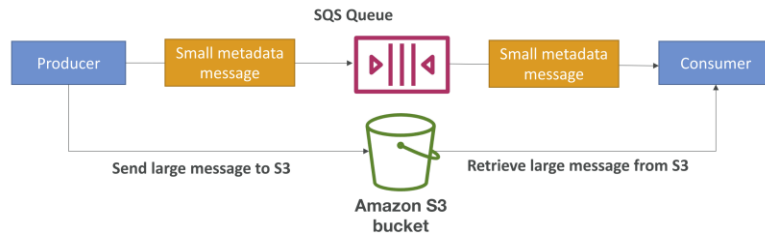
Receive message wait time info

20 Seconds

Should be between 0 and 30 seconds.

SQS - Extended client

- Message size limit is 256KB, how to send large messages, e.g. 1GB?
- Using the SQS Extended Client (Java Library)



SQS- Important things need to be:

- `CreateQueue` (`MessageRetentionPeriod`), `DeleteQueue`
 - `PurgeQueue`: delete all the messages in queue
 - `SendMessage` (`DelaySeconds`), `ReceiveMessage`, `DeleteMessage`
 - `MaxNumberOfMessages`: default 1, max 10 (for `ReceiveMessage` API)
 - `ReceiveMessageWaitTimeSeconds`: Long Polling
 - `ChangeMessageVisibility`: change the message timeout
- Batch APIs for `SendMessage`, `DeleteMessage`, `ChangeMessageVisibility` helps decrease your costs

SQS – FIFO Queue

- FIFO = First In First Out (ordering of messages in the queue)



- Limited throughput: 300 msg/s without batching, 3000 msg/s with
- Exactly-once send capability (by removing duplicates)
- Messages are processed in order by the consumer

LAB: SQS - FIFO Queue

Create a FIFO Queue and keep default settings now send 4 messages rajiv1, rajiv2, rajiv3, rajiv4 now pull the messages and check the messages and start from the top one and we can see which message we send first that one come first. Here we send rajiv1 first so we can see rajiv1 message come first.

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

my-fifo-1.fifo

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

Configuration

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

Visibility timeout

30 Seconds

Should be between 0 seconds and 12 hours.

Message retention period

4 Days

Should be between 1 minute and 14 days.

Delivery delay

0 Seconds

Should be between 0 seconds and 15 minutes.

Maximum message size

256 KB

Should be between 1 KB and 256 KB.

Receive message wait time

0 Seconds

Should be between 0 and 20 seconds.

Tags - Optional

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

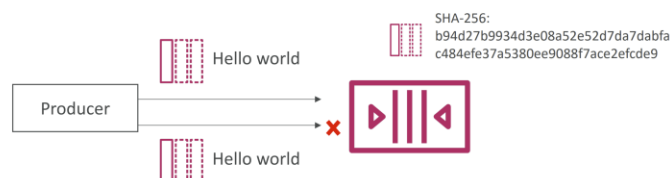
Key	Value - optional	
<input type="text" value="Enter key"/>	<input type="text" value="Enter value"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

After creating the queue send 4 messages rajiv1, rajiv2, rajiv3, rajiv4 and pull the message and check

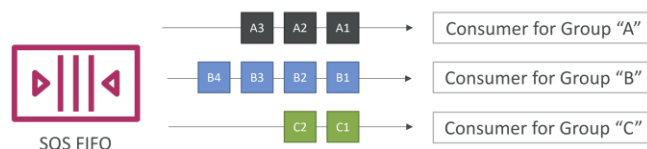
SQS- FIFO – Deduplication

- De-duplication interval is 5 minutes
- Two de-duplication methods:
 - Content-based deduplication: will do a SHA-256 hash of the message body
 - Explicitly provide a Message Deduplication ID



SQS- FIFO – Message Grouping

- If you specify the same value of **MessageGroupID** in an SQS FIFO queue, you can only have one consumer, and all the messages are in order
- To get ordering at the level of a subset of messages, specify different values for **MessageGroupID**
 - Messages that share a common Message Group ID will be in order within the group
 - Each Group ID can have a different consumer (parallel processing!)
 - Ordering across groups is not guaranteed



Lab: FIFO – Deduplication

Edit the fifo SQS and enable the content-based deduplication as shown blow

Amazon SQS > Queues > 1.fifo > Edit

Edit 1.fifo

Details

Name	1.fifo	Type	FIFO
------	--------	------	------

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

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

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

Receive message wait time info
Should be between 0 and 20 seconds.
0 Seconds

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

Maximum message size info
Should be between 1 KB and 256 KB.
256 KB

FIFO queue settings

☒ Content-based deduplication
When content-based deduplication is enabled, the message deduplication ID is optional.

☐ High throughput FIFO queue (recommended) info
Configure your FIFO queue for maximum throughput.

☒ High throughput for FIFO queues supports a higher number of transactions per second, per API.

Deduplication scope info
Specify the scope of deduplication for a FIFO queue.
☒ Queue
Deduplicate all messages in a queue.
☐ Message group
Deduplicate all messages that belong to the same message group.

FIFO throughput limit info
Specify how to apply the throughput limit on FIFO queue.
☒ Per queue
Apply the throughput limit at the queue level.
☐ Per message group ID
Apply the throughput limit at the message group level.

After clicking the send and receive message

Amazon SQS > Queues > 1.fifo

1.fifo info Edit Delete Purge Send and receive messages Start SQS redrive

Details info

Name	1.fifo	Type	FIFO
ARN	arn:aws:sqs:us-east-1:123456789012:1.fifo		
Queue URL	https://sqs.us-east-1.amazonaws.com/123456789012/1.fifo		
Created at	2023-01-01T12:34:56.789Z		

[More](#)

Now create and send a message we can see the message is going only one times even though we click send message button several times.

Amazon SQS > Queues > 1.fifo > Send and receive messages

Send and receive messages

Send messages to and receive messages from a queue.

Send message info Clear content Send message

☒ Your message has been sent and is ready to be received. View details ×

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

Message group ID
The tag that specifies that a message belongs to a specific message group.
demo

Message deduplication ID - Optional
The token used for deduplication of messages within the deduplication interval.
Enter message deduplication id

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 receives/second

we enable the deduplication so now if we click the send message many times the message will not go because same name id mesgae already send

Now if we change the message body and send the message we can see message is going

Amazon SQS > Queues > 1.fifo > Send and receive messages

Send and receive messages

Send messages to and receive messages from a queue.

Send message [info](#)

[Clear content](#) [Send message](#)

✔ Your message has been sent and is ready to be received.

[View details](#) ✕

Message body

Enter the message to send to the queue.

rajiv1

Message group ID

The tag that specifies that a message belongs to a specific message group.

demo

Message deduplication ID - Optional

The token used for deduplication of messages within the deduplication interval.

Enter message deduplication id

[▶ Message attributes - Optional](#) [info](#)

Receive messages [info](#)

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

Messages available

2

Polling duration

30

Maximum message count

10

Polling progress

0 received/second

0%

Now if we add our own duplication id and send message then the message is going because now the id is different but now if I try to send the message again it will not go because the id 123 already have the same message

Amazon SQS > Queues > 1.fifo > Send and receive messages

Send and receive messages

Send messages to and receive messages from a queue.

Send message [info](#)

[Clear content](#) [Send message](#)

✔ Your message has been sent and is ready to be received.

[View details](#) ✕

Message body

Enter the message to send to the queue.

rajiv1

Message group ID

The tag that specifies that a message belongs to a specific message group.

demo

Message deduplication ID - Optional

The token used for deduplication of messages within the deduplication interval.

123

[▶ Message attributes - Optional](#) [info](#)

Receive messages [info](#)

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

Messages available

3

Polling duration

30

Maximum message count

10

Polling progress

0 received/second

0%