



Simple Queue Service

Amazon Simple Queue Service (Amazon SQS) offers a secure, durable, and available hosted queue that lets you integrate and decouple distributed software systems and components. Amazon SQS offers common constructs such as dead-letter queues and cost allocation tags. It provides a generic web services API that you can access using any programming language that the AWS SDK supports.



Benefits of using Amazon SQS

1. **Security** – You control who can send messages to and receive messages from an Amazon SQS queue. You can choose to transmit sensitive data by protecting the contents of messages in queues by using default Amazon SQS managed server-side encryption (SSE), or by using custom SSE keys managed in AWS Key Management Service (AWS KMS).
2. **Durability** – For the safety of your messages, Amazon SQS stores them on multiple servers. Standard queues support at-least-once message delivery, and FIFO queues support exactly-once message processing and high-throughput mode.
3. **Availability** – Amazon SQS uses redundant infrastructure to provide highly-concurrent access to messages and high availability for producing and consuming messages.
4. **Scalability** – Amazon SQS can process each buffered request independently, scaling transparently to handle any load increases or spikes without any provisioning instructions.
5. **Reliability** – Amazon SQS locks your messages during processing, so that multiple producers can send and multiple consumers can receive messages at the same time.
6. **Customization** – Your queues don't have to be exactly alike—for example, you can set a default delay on a queue. You can store the contents of messages larger than 256 KB using Amazon Simple Storage Service (Amazon S3) or Amazon DynamoDB, with Amazon SQS holding a pointer to the Amazon S3 object, or you can split a large message into smaller messages.



To begin with the Lab:

1. Login to AWS Console and navigate to SQS. Choose this service accordingly.



Simple Queue Service ☆
SQS Managed Message Queues

2. Now on the dashboard of SQS click on Create Queue.

Application integration

Amazon SQS

A message queuing service

Amazon SQS provides queues for high-throughput, system-to-system messaging. You can use queues to decouple heavyweight processes and to buffer and batch work. Amazon SQS stores messages until microservices and serverless applications process them.

Get started

Learn how to use Amazon SQS by creating a queue, sending a message to the queue, and receiving and processing the message.

Create queue

Pricing (US)

You can get started with Amazon SQS for free. All customers can make 1 million Amazon SQS requests for free each month. Some applications might be able to operate within this Free Tier limit.

Cost calculator

Documentation

[Developer guide](#) [API reference](#)

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

```

graph LR
    Producers[Producers: Messages sent from applications, microservices, and other AWS services] --> SQS[Amazon SQS]
    SQS --> Queue[SQS Queue: Queues store messages and wait for consumers to poll]
    Queue --> Consumers[Consumers: Messages are processed by applications, Lambda functions, EC2 instances, and other AWS services]
  
```

3. So, now you just need to select standard and give it a name. then just create your 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

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

4. Here your SQS has been created. Now click on Send and receive messages.

ⓘ Queue demo-sqs-queue created successfully
You can now send and receive messages.

Amazon SQS > Queues > demo-sqs-queue

demo-sqs-queue

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

Details Info	
Name	demo-sqs-queue
Type	Standard
Encryption	Amazon SQS key (SSE-SQS)
URL	https://sqs.ap-south-1.amazonaws.com/878893308172/demo-sqs-queue
ARN	arn:aws:sqs:ap-south-1:878893308172:demo-sqs-queue
Dead-letter queue	=

[More](#)

5. Now you need to write some text in message body and click on Send message.

Send and receive messages

Send messages to and receive messages from a queue.

Send message Info

Message body
Enter the message to send to the queue.
This is the test message.

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

▶ Message attributes - Optional Info

Clear content **Send message**

6. After that if you scroll down a little you will see receive messages section and here you can see that you have 1 message available.
7. Now you need to click on Poll for messages.

Receive messages Info

Messages available **1** Polling duration 30 Maximum message count 10 Polling progress 0 receives/second

Messages (0)

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

Poll for messages

8. You will see that a message with some ID has pop up click on this ID.

Receive messages Info

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

Messages (1)

ID	Sent	Size	Receive count
9d546ce1-a660-4268-9a86-58592411ca6d	2024-02-13T09:50:05.30	26 bytes	1

9. And you will be able to read the message.

Message: 9d546ce1-a660-4268-9a86-58592411ca6d X

Body **Attributes** **Details**

This is the test message.

Done