

SQS (Simple Queue Service):

In aws console go to sqs

The screenshot shows the AWS SQS homepage. At the top right, it says "Account ID: 2355-". Below that is a "Get started" box with a "Create queue" button. To the left, there's a diagram titled "How it works" showing a flow from "Producers" to a central queue with an orange arrow, and then to "Consumers".

Click on create queue set as standard type and give name and leave all the settings as default.

The screenshot shows the "Create queue" wizard. In the "Details" step, the "Type" section is selected, showing the "Standard" option (which is highlighted with a blue border). It describes "At-least-once delivery, message ordering isn't preserved" and lists "At-least once delivery" and "Best-effort ordering" as options. The "FIFO" option is also shown but not selected. A note below states "You can't change the queue type after you create a queue." In the "Name" section, the queue is named "MyTestQueue". The "Configuration" section is partially visible at the bottom.

Go to your queue and click on send and receive messages.

The screenshot shows the AWS SQS Queue Details page for a queue named 'MyTestQueue'. The 'Send and receive messages' button is highlighted with a red arrow. The page includes details like Name (MyTestQueue), Type (Standard), ARN (arn:aws:sqs:us-east-1:235351028455:MyTestQueue), and Dead-letter queue (-). Navigation tabs at the bottom include Queue policies, Monitoring, SNS subscriptions, Lambda triggers, EventBridge Pipes, Dead-letter queue, Tagging, Encryption, and Dead-letter queue redri.

Type the message in message body.

The screenshot shows the 'Send and receive messages' page for the 'MyTestQueue'. The 'Send message' section is visible, showing a message body input field containing 'Hello from AWS SQS!'. Other fields include Message group ID (optional), Delivery delay (info), and Message attributes (optional). Buttons for Clear content and Send message are at the bottom.

If you want to read the message scroll down you will see receive messages and click on poll messages

AWS | Search [Alt+S] | United States (N. Virginia) | Amazon SQS > Queues > MyTestQueue > Send and receive messages

Message attributes - optional [Info](#)

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

In receive messages you will see this messages.

AWS | Search [Alt+S] | Account ID: 2353-3102-042 | United States (N. Virginia) | mujah | Amazon SQS > Queues > MyTestQueue > Send and receive messages

Message attributes - optional [Info](#)

Receive messages [Info](#)

Messages available: 1 Polling duration: 30 Maximum message count: 10 Polling progress: 2 receives/second

Messages (2)

ID	Sent	Size	Receive count
6dfc1189-740a-42a0-8111-a8661c05ae43	2025-10-29T16:19+05:30	20 bytes	2
9ea11d61-d41f-4ad9-b4f2-c6769237e6a3	2025-10-29T16:20+05:30	20 bytes	1

MQS (Message Queue Service)

Go to aws console and click as amazon mq and click on create broker.

aws | Search [Alt+S] | United States (N. Virginia) | Account ID: [REDACTED]

Application Integration

Amazon MQ

Fully managed open source message broker service

Amazon MQ is a managed message broker service for Apache ActiveMQ and RabbitMQ that makes it easy to set up and operate message brokers in the cloud, so you can migrate your messaging and applications without rewriting code.

Create brokers

Create message brokers in the cloud.

Get started

Benefits

- Accelerate migration**
- Offload operations**

With Amazon MQ, you pay only for what you use. There are no minimum fees or upfront commitments. You pay for the time your message broker instance runs, the storage you use monthly, and standard data transfer fees.

[Learn more](#)

Select rabbit mq

aws | Search [Alt+S] | United States (N. Virginia) | Account ID: 2353-5102-8455 | mujheed

Amazon MQ > Brokers > Create RabbitMQ brokers

Step 1
 Select broker engine
 Step 2
 Step 3
 Step 4
 Review and create

Select broker engine Info

Broker engine types

Select broker engine
A broker engine is a type of message broker that runs on Amazon MQ.

Apache ActiveMQ RabbitMQ

[Cancel](#) [Next](#)

aws | Search [Alt+S] | United States (N. Virginia) | Account ID: 2353-5102-8455 | mujheed

Amazon MQ > Brokers > Create RabbitMQ brokers

Step 1
 Select broker engine
 Select deployment mode
 Step 3
 Step 4
 Review and create

Select deployment mode Info

Deployment mode

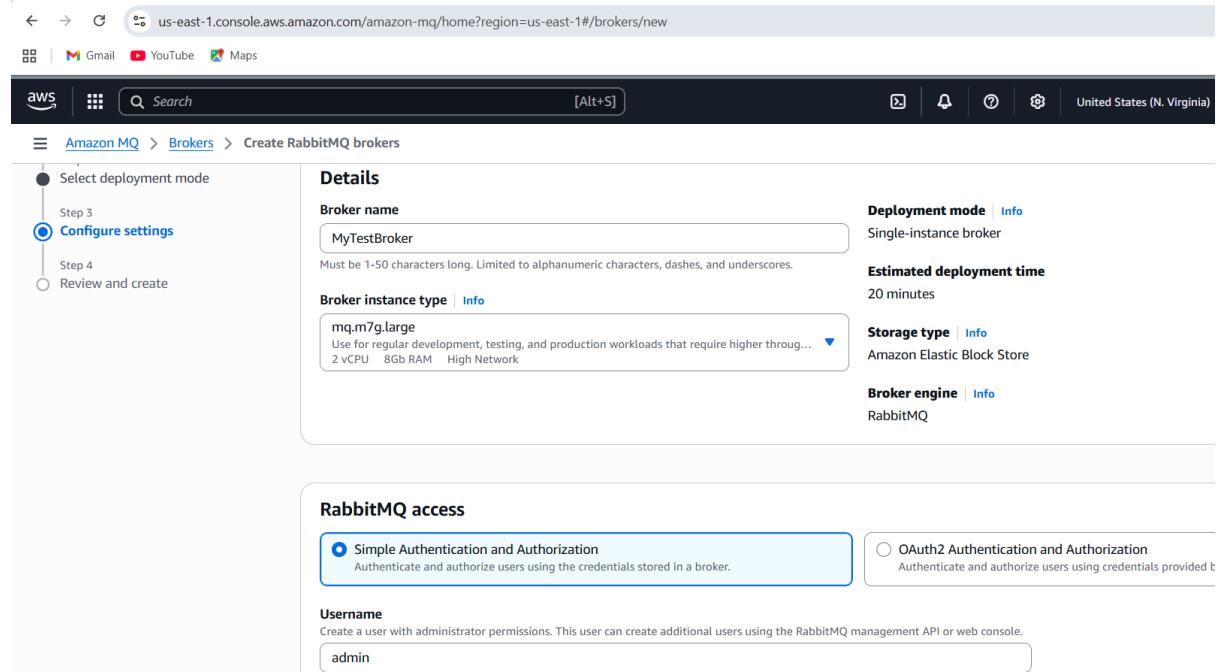
Select deployment mode

Single-instance broker
One broker in one Availability Zone. Use for development and testing, or for smaller workloads.

Cluster deployment
Three single-instance brokers connected in a mesh network across multiple Availability Zones. Use for production workloads that require high availability and message durability.

[Cancel](#) [Previous](#) [Next](#)

Give the name of broker and instance type and select simple authentication and give username and password and click on create.

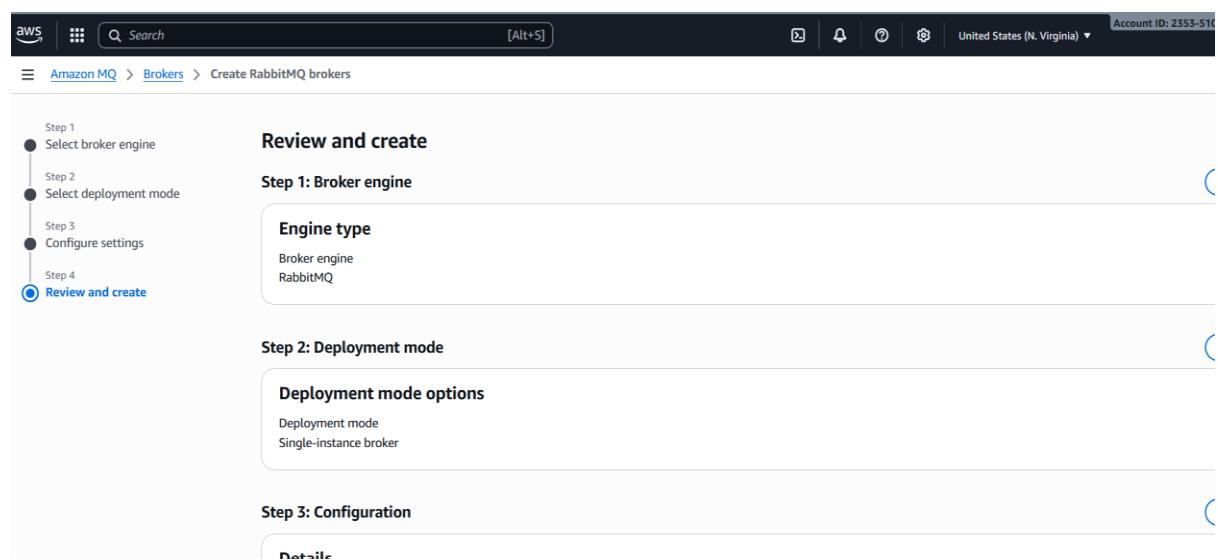


The screenshot shows the 'Configure settings' step of the wizard. It includes fields for 'Broker name' (MyTestBroker), 'Broker instance type' (mq.m7g.large), 'Deployment mode' (Single-instance broker), 'Storage type' (Amazon Elastic Block Store), and 'Broker engine' (RabbitMQ). The 'Simple Authentication and Authorization' option is selected under 'RabbitMQ access'.

RabbitMQ access

- Simple Authentication and Authorization
Authenticate and authorize users using the credentials stored in a broker.
- OAuth2 Authentication and Authorization
Authenticate and authorize users using credentials provided by an external provider.

Username
Create a user with administrator permissions. This user can create additional users using the RabbitMQ management API or web console.
admin



The screenshot shows the 'Review and create' step of the wizard. It displays the summary of the configuration:

- Step 1: Select broker engine (Broker engine: RabbitMQ)
- Step 2: Select deployment mode (Deployment mode: Single-instance broker)
- Step 3: Configure settings
- Step 4: Review and create (selected)

Step 1: Broker engine

Engine type
Broker engine
RabbitMQ

Step 2: Deployment mode

Deployment mode options
Deployment mode
Single-instance broker

Step 3: Configuration

Details