Amazon Simple Queue Service (SQS)

1. What Is 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 supports both **standard** and **FIFO queues**.

- SQS uses **pull based** (polling) not push based

- Amazon SQS is a distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component where a queue is a temporary repository for messages that are awaiting processing.

- Messages can contain up to 256 KB of text in any format such as json, xml, etc.

- User can access Amazon SQS from their VPC using **VPC endpoint**.

Scenario-1:

Suppose the user wants to upload a photo and wants to convert into Meme. User uploads a photo on a website and website might store a photo in s3. As soon as it finished uploads, it triggers a Lambda function. Lambda analyzes the data about this particular image to SQS, and this data can be "what the top of the meme should say", "what the bottom of the meme should say", the location of the S3 bucket, etc. The data sits inside the SQS as a message. An EC2 instance looks at the message and performs its job. An EC2 instance creates a Meme and stores it in S3 bucket. Once the EC2 instance completed its job, it moves back to the SQS. The best thing is that if you lose your EC2 instance, then also you would not lose the job as the job sits inside the S3 bucket.

Scenario-2:

Suppose the user wants to look for a package holiday and wants to look at the best possible flight. A User types a query in a browser, it then hits the EC2 instance. An EC2 instance looks "What the user is looking for?", it then puts the message in a queue to the SQS. An EC2 instance pulls queue. An EC2 instance continuously pulling the queue and looking for the jobs to do. Once it gets the job, it then processes it. It interrogates the Airline service to get all the best possible flights. It sends the result to the web server, and the web server sends back the result to the user. A User then selects the best flight according to his or her budget.

If we didn't have SQS, then what happened?

A web server passes the information to an application server and then application server queried an Airline service. If an Application server crashes, then a user loses its query. One of the great thing about SQS is that data is queued in the SQS even if the application server crashes, the message in the queue is marked as an invisible in a timeout interval window. When the timeout runs out, message reappears in the queue; then a new EC2 instance can use this message to perform its job. Therefore, we can say that **SQS removes the application server dependency**.

Benefits of SQS:

- **Security**: You control who can send messages to and receive messages from an Amazon SQS queue.

- Support Server-Side Encryption **(SSE)**

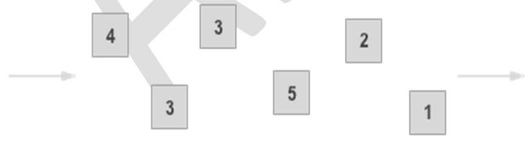
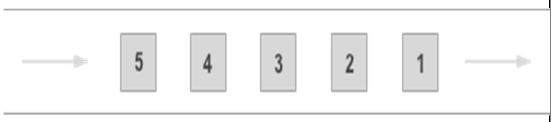
- **Durability**: SQS store message on multiple servers.

- **Availability**: Amazon SQS uses redundant infrastructure to provide highly-concurrent access to messages and high availability for producing and consuming messages.

- **Scalability**: SQS can process each buffered request independently, scaling transparently to handle any load increases or spikes without any provisioning instructions.

- **Reliability**: SQS locks your messages during processing, so that multiple producers can send and multiple consumers can receive messages at the same time.

Types of Queue:

Standard	FIFO
Available in all regions	Available only US East(NV & Ohio), US West (Oregon), EU (Ireland), Asia Pacific (Sydney & Tokyo) regions.
Unlimited Throughput – Standard queues support a nearly unlimited number of transactions per second (TPS)	**High Throughput** By default, FIFO queues support up to 300 messages per second with batching
At-Least-Once Delivery – A message is delivered at least once, but occasionally more than one copy of a message is delivered.	**Exactly-Once Processing** – A message is delivered once and remains available until a consumer processes and deletes it. Duplicates aren't introduced into the queue.
Best-Effort Ordering – Occasionally, messages might be delivered in an order different from which they were sent.	**First-In-First-Out Delivery** – The order in which messages are sent and received is strictly preserved.
	

- You can include structured metadata (such as timestamps, geospatial data, signatures, and identifiers) with messages using **message attributes**.

- **Message timers** let you specify an initial invisibility period for a message added to a queue. The default (minimum) invisibility period for a message is 0 seconds. The maximum is 15 minutes.

- SQS doesn't automatically delete a message after receiving it for you, in case you don't successfully receive the message.

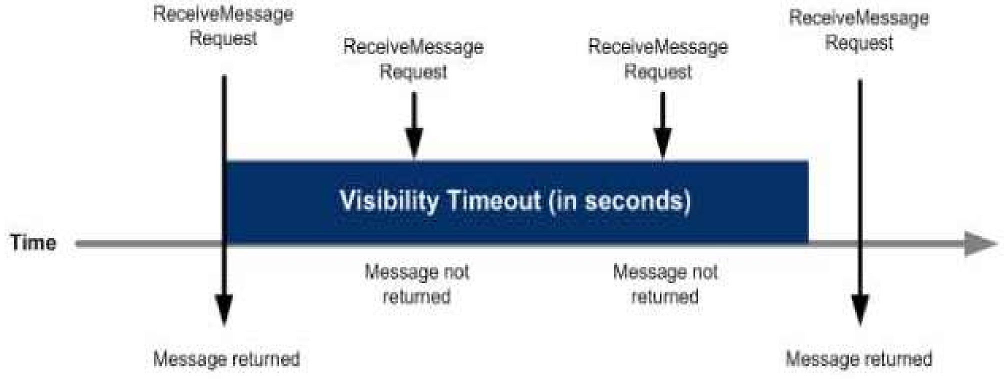
- You can subscribe one or more SQS queues to an Amazon SNS topic from a list of topics available for the selected queue.

- You can configure an existing SQS queue to trigger an AWS Lambda function when new messages arrive in a queue.
    - Your queue and Lambda function must be in the same AWS Region.
    - FIFO queues don't support Lambda function triggers.
    - You can associate only one queue with one or more Lambda functions.
    - You can't associate an encrypted queue that uses an AWS managed Customer Master Key for SQS with a Lambda function in a different AWS account.

- You can delete all the messages in your queue by **purging** them.

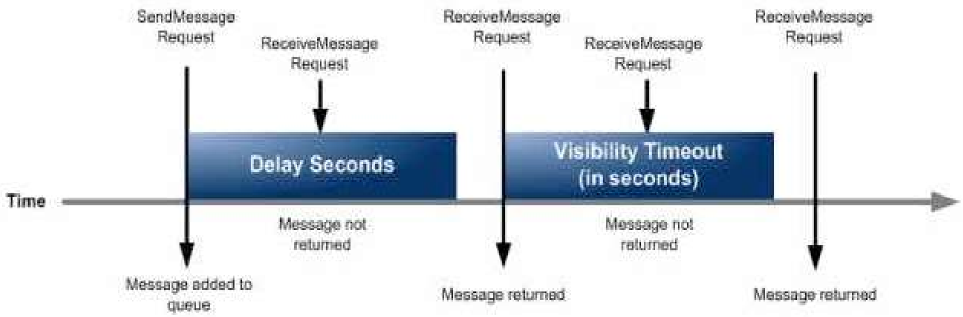
- **Long polling** helps reduce the cost by eliminating the number of empty responses and false empty responses. While the regular **short polling** returns immediately, even if the message queue being polled is empty, long polling doesn't return a response until a message arrives in the message queue, or the long poll times out.

- To prevent other consumers from processing a message redundantly, SQS sets a **visibility timeout**, a period of time SQS prevents other consumers from receiving and processing the message. The default visibility timeout for a message is 30 seconds. The minimum is 0 seconds. The maximum is 12 hours.



- SQS supports **dead-letter queues**, which other queues can target for messages that can't be processed successfully.

- **Delay queues** let you postpone the delivery of new messages to a queue for a number of seconds.



2. Basic SQS Architecture

- Main Parts:
  - The components of your distributed system.
  - The queue
  - The messages
- Standard Queues
  - Default queue type.
  - Makes a best effort to preserve the order of messages.
  - Stores copies of your messages on multiple servers for redundancy and high availability.
  - Consumes messages using **short polling** (default) – take a subset of SQS servers (based on a weighted random distribution) and returns messages from only those servers.
- FIFO Queues
  - The order in which messages are sent and received is strictly preserved and a message is delivered once and remains available until a consumer processes and deletes it.
  - Duplicates aren’t introduced into the queue.
  - FIFO queues support **message groups** that allow multiple ordered message groups within a single queue.
- When you create a new queue, you must specify a queue name unique for your AWS account and region. This becomes your queue url. https://sqs.region.amazonaws.com/accountnumber/queuename
- Each message receives a system-assigned message ID for identifying messages.

- Every time you receive a message from a queue, you receive a receipt handle for that message.
- You can use cost allocation tags to organize your AWS bill to reflect your own cost structure.
- Send, receive, or delete messages in batches of up to 10 messages or 256KB.

Best Practices

- Extend the message's visibility timeout to the maximum time it takes to process and delete the message. If you don't know how long it takes to process a message, as long as your consumer still works on the message, keep extending the visibility timeout.
- Using the appropriate polling mode.
- Configure a dead-letter queue to capture problematic messages.
- To avoid inconsistent message processing by standard queues, avoid setting the number of maximum receives to 1 when you configure a dead-letter queue.
- Don't create reply queues per message. Instead, create reply queues on startup, per producer, and use a correlation ID message attribute to map replies to requests. Don't let your producers share reply queues.
- Reduce cost by batching message actions.

Monitoring, Logging, and Automating

- Monitor SQS queues using CloudWatch
- Log SQS API Calls Using AWS CloudTrail
- Automate notifications from AWS Services to SQS using CloudWatch Events

Security

- Use IAM for user authentication.
- SQS has its own resource-based permissions system that uses policies written in the same language used for IAM policies.
- Protect data using Server-Side Encryption and AWS KMS.

Pricing

- You are charged per 1 million SQS requests. Price depends on the type of queue being used. Requests include:
    - API Actions
    - FIFO Requests

  - A single request of 1 to 10 messages, up to a maximum total payload of 256 KB
  - Each 64 KB chunk of a payload is billed as 1 request
  - Interaction with Amazon S3
  - Interaction with AWS KMS
- Data transfer out of SQS per TB/month after consuming 1 GB for that month.

About SQS Long Polling:

- Long polling helps reduce your cost of using Amazon SQS by reducing the number of empty responses when there are no messages available to return in reply to a ReceiveMessage request sent to an Amazon SQS queue and eliminating false empty responses when messages are available in the queue but aren't included in the response.

- Long polling reduces the number of empty responses by allowing Amazon SQS to wait until a message is available in the queue before sending a response. Unless the connection times out, the response to the ReceiveMessage request contains at least one of the available messages, up to the maximum number of messages specified in the ReceiveMessage action.

- Long polling eliminates false empty responses by querying all (rather than a limited number) of the servers. Long polling returns messages as soon any message becomes available.

3. Getting Started with Amazon SQS

Step 1: Create a Queue

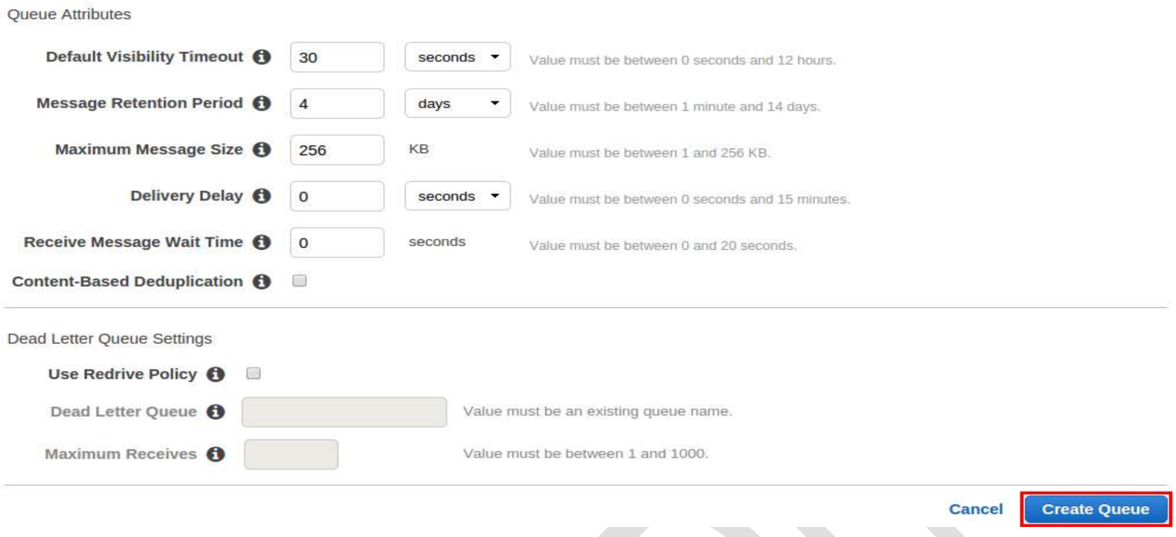
The first and most common Amazon SQS task is creating queues.

1. Sign in to the Amazon SQS console.
2. Choose **Create New Queue.**
3. On the **Create New Queue** page, ensure that you're in the correct region and then type the **Queue Name**.

Note: The name of a FIFO queue must end with the **.fifo** suffix.

4. **Standard** is selected by default. Choose **FIFO**.
5. To create your queue with the default parameters, choose **Quick-Create Queue**.

   Your new queue is created and selected in the queue list.



Queue Attributes

Default Visibility Timeout: 30 seconds — Value must be between 0 seconds and 12 hours.

Message Retention Period: 4 days — Value must be between 1 minute and 14 days.

Maximum Message Size: 256 KB — Value must be between 1 and 256 KB.

Delivery Delay: 0 seconds — Value must be between 0 seconds and 15 minutes.

Receive Message Wait Time: 0 seconds — Value must be between 0 and 20 seconds.

Content-Based Deduplication ☐

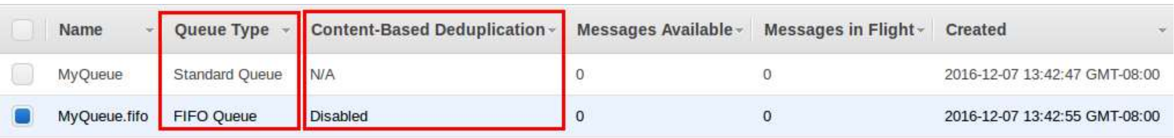
Dead Letter Queue Settings

Use Redrive Policy ☐

Dead Letter Queue — Value must be an existing queue name.

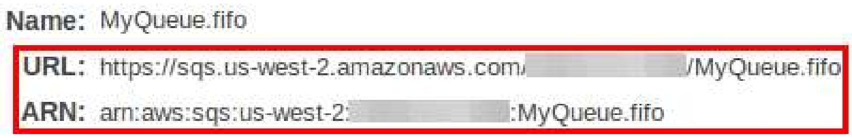
Maximum Receives — Value must be between 1 and 1000.

- The **Queue Type** column helps you distinguish standard queues from FIFO queues at a glance. For a FIFO queue, the **Content-Based Deduplication** column displays whether you have enabled exactly-once processing.



☐	Name	Queue Type	Content-Based Deduplication	Messages Available	Messages in Flight	Created
☐	MyQueue	Standard Queue	N/A	0	0	2016-12-07 13:42:47 GMT-08:00
☑	MyQueue.fifo	FIFO Queue	Disabled	0	0	2016-12-07 13:42:55 GMT-08:00

Your queue's **Name**, **URL**, and **ARN** are displayed on the **Details** tab.



Name: MyQueue.fifo

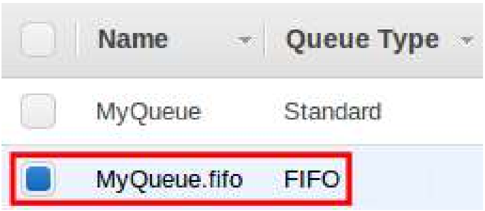
URL: https://sqs.us-west-2.amazonaws.com/[redacted]/MyQueue.fifo

ARN: arn:aws:sqs:us-west-2:[redacted]:MyQueue.fifo

Step 2: Send a Message

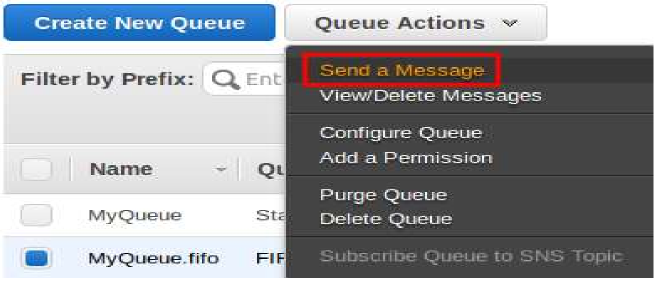
After you create your queue, you can send a message to it. The following example shows sending a message to an existing queue.

1. From the queue list, select the queue that you've created.

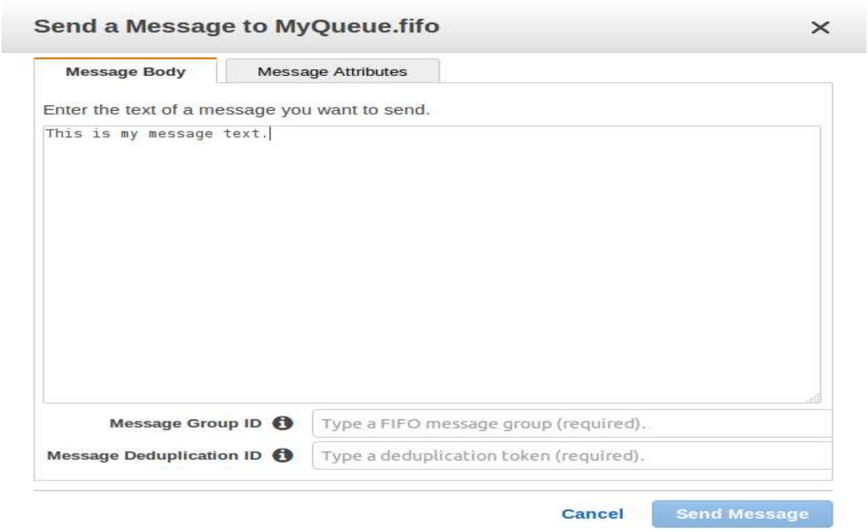


☐	Name	Queue Type
☐	MyQueue	Standard
☑	MyQueue.fifo	FIFO

2. From **Queue Actions**, select **Send a Message**.



- The **Send a Message to *QueueName*** dialog box is displayed.
- The following example shows the **Message Group ID** and **Message Deduplication ID** parameters specific to FIFO queues (content-based deduplication is disabled).



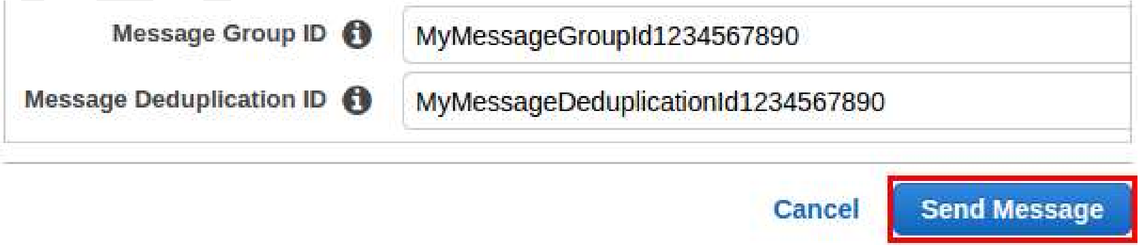
Enter the text of a message you want to send.

```
This is my message text.
```

Message Group ID: Type a FIFO message group (required).

Message Deduplication ID: Type a deduplication token (required).

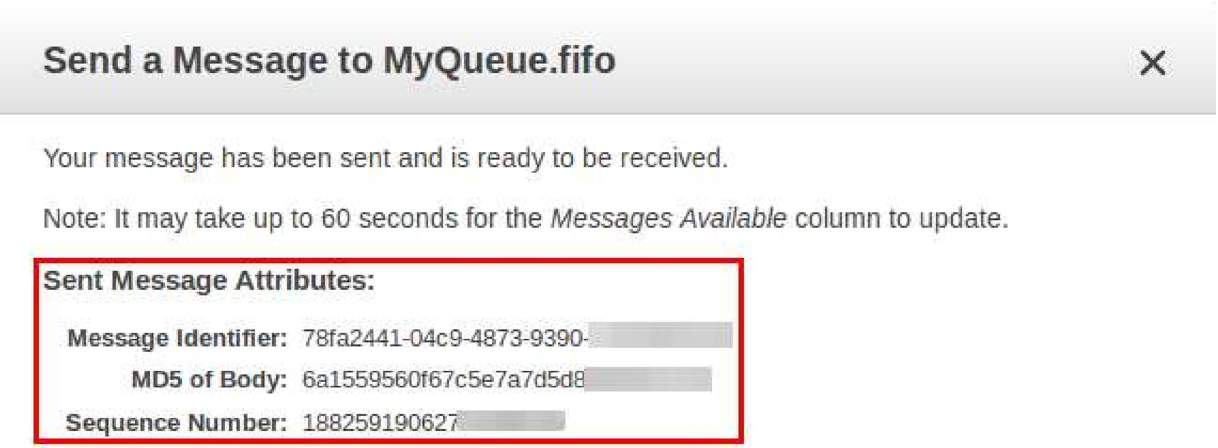
3. To send a message to a FIFO queue, type the **Message Body**, the **Message Group ID** MyMessageGroupId1234567890, and the **Message Deduplication ID** MyMessageDeduplicationId1234567890, and then choose **Send Message**. For more information, see FIFO Delivery Logic.



Message Group ID: MyMessageGroupId1234567890

Message Deduplication ID: MyMessageDeduplicationId1234567890

- Your message is sent and the **Send a Message to *QueueName*** dialog box is displayed, showing the attributes of the sent message.
- The following example shows the **Sequence Number** attribute specific to FIFO queues.


Send a Message to MyQueue.fifo

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

Note: It may take up to 60 seconds for the *Messages Available* column to update.

Sent Message Attributes:

Message Identifier: 78fa2441-04c9-4873-9390-[redacted]

MD5 of Body: 6a1559560f67c5e7a7d5d8[redacted]

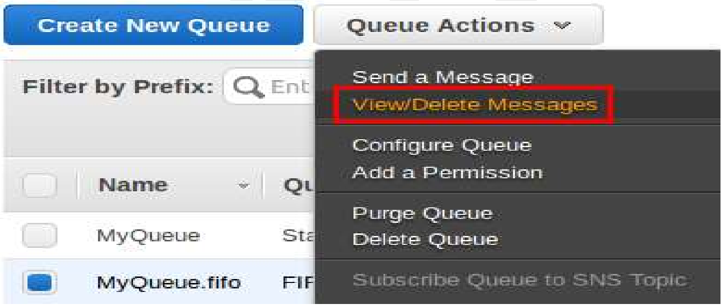
Sequence Number: 188259190627[redacted]

4. Choose **Close**.

Step 3: Receive and Delete Your Message

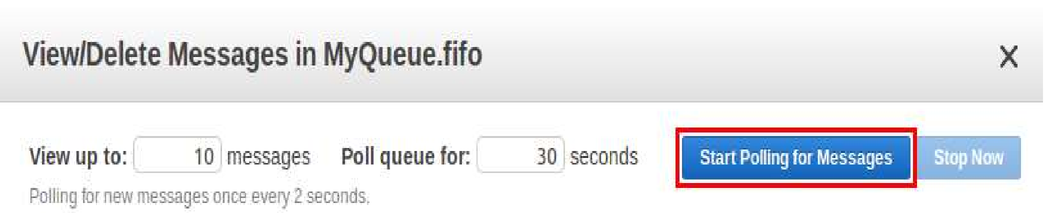
After you send a message into a queue, you can consume it (retrieve it from the queue). When you request a message from a queue, you can't specify which message to get. Instead, you specify the maximum number of messages (up to 10) that you want to get.

1. From the queue list, select the queue that you have created.
2. From **Queue Actions**, select **View/Delete Messages**.



- The **View/Delete Messages in *QueueName*** dialog box is displayed.

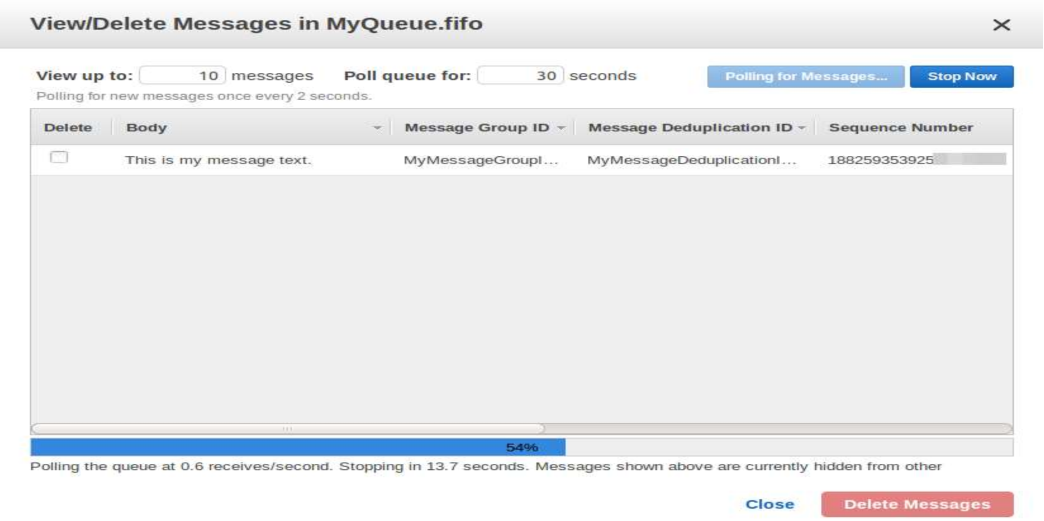
3. Choose **Start Polling for messages.**


View/Delete Messages in MyQueue.fifo

View up to: 10 messages Poll queue for: 30 seconds

Polling for new messages once every 2 seconds.

- Amazon SQS begins to poll the messages in the queue. The dialog box displays a message from the queue. A progress bar at the bottom of the dialog box displays the status of the message's visibility timeout.
- The following example shows the **Message Group ID**, **Message Deduplication ID**, and **Sequence Number** columns specific to FIFO queues.



View/Delete Messages in MyQueue.fifo

View up to: 10 messages Poll queue for: 30 seconds

Polling for new messages once every 2 seconds.

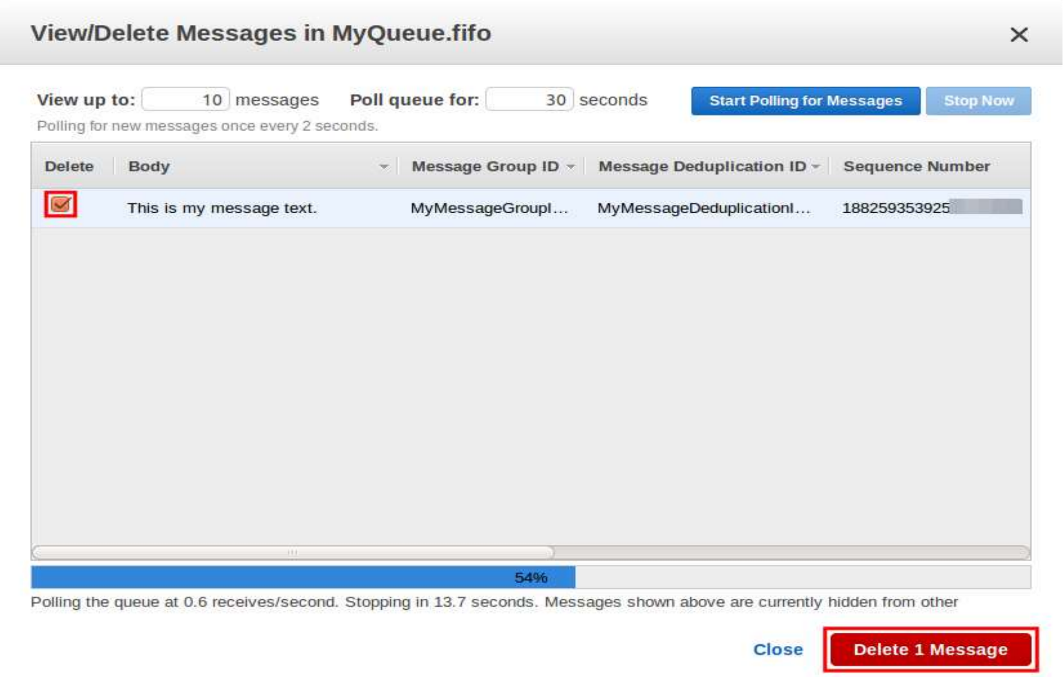
Delete	Body	Message Group ID	Message Deduplication ID	Sequence Number
□	This is my message text.	MyMessageGroupI...	MyMessageDeduplicationI...	188259353925[redacted]

54%

Polling the queue at 0.6 receives/second. Stopping in 13.7 seconds. Messages shown above are currently hidden from other

Note: When the progress bar is filled in, the visibility timeout expires and the message becomes visible to consumers.

4. *Before* the visibility timeout expires, select the message that you want to delete and then choose **Delete *1* Message**.



View/Delete Messages in MyQueue.fifo

View up to: 10 messages Poll queue for: 30 seconds

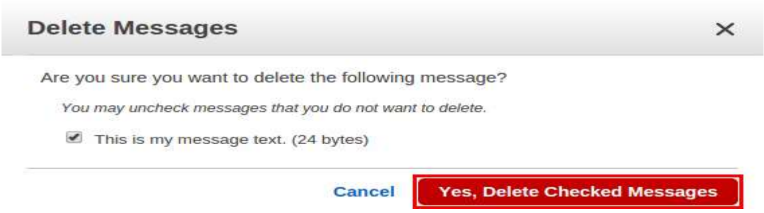
Polling for new messages once every 2 seconds.

Delete	Body	Message Group ID	Message Deduplication ID	Sequence Number
☑	This is my message text.	MyMessageGroupI...	MyMessageDeduplicationI...	188259353925[redacted]

54%

Polling the queue at 0.6 receives/second. Stopping in 13.7 seconds. Messages shown above are currently hidden from other

5. In the **Delete Messages** dialog box, confirm that the message you want to delete is checked and choose **Yes, Delete Checked Messages**.



Delete Messages

Are you sure you want to delete the following message?

You may uncheck messages that you do not want to delete.

☑ This is my message text. (24 bytes)

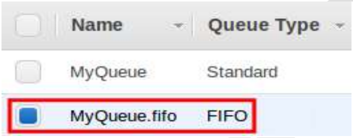
The selected message is deleted.

6. Select **Close**.

Step 4: Delete Your Queue

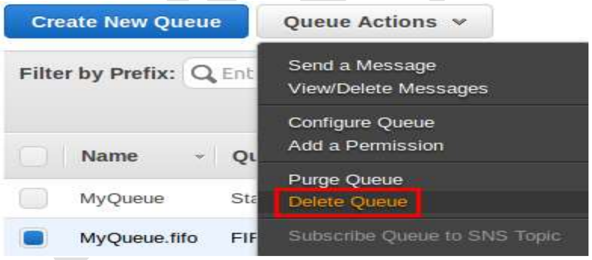
If you don't use an Amazon SQS queue (and don't foresee using it in the near future), it is a best practice to delete it from Amazon SQS. In this tutorial you'll learn how to delete a queue.

1. From the queue list, select the queue that you have created.

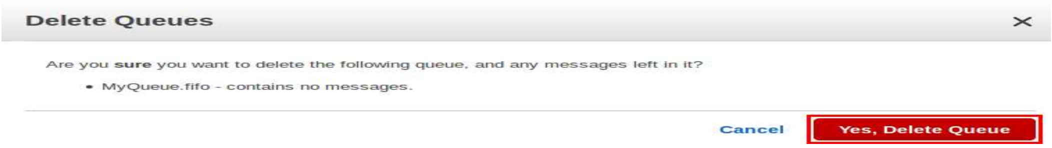


□	Name	Queue Type
□	MyQueue	Standard
■	MyQueue.fifo	FIFO

2. From **Queue Actions**, select **Delete Queue**.



➢ The **Delete Queues** dialog box is displayed.



Delete Queues

Are you sure you want to delete the following queue, and any messages left in it?

- MyQueue.fifo - contains no messages.

3. Choose **Yes, Delete Queue**.

The queue is deleted.

4. Creating an Amazon SQS Queue with Server-Side Encryption (SSE)

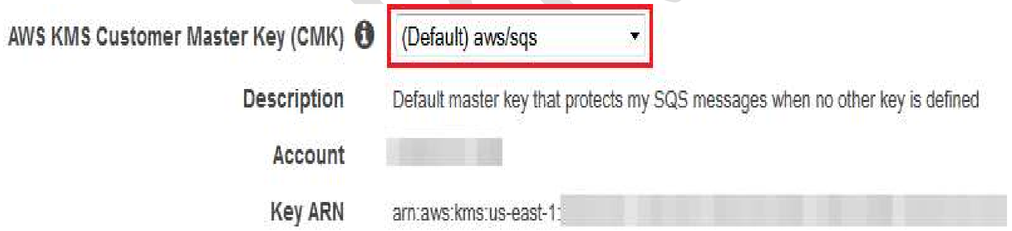
- You can enable SSE for a queue to protect its data.

1. Sign in to the Amazon SQS console.
2. Choose **Create New Queue.**
3. On the **Create New Queue** page, ensure that you're in the correct region and then type the **Queue Name**.
4. **Standard** is selected by default. Choose **FIFO**.
5. Choose **Configure Queue**, and then choose **Use SSE**.
6. Specify the customer master key (CMK) ID. For more information, see Key Terms.

Important

If you aren't the owner of the CMK, or if you log in with an account that doesn't have the `kms:ListAliases` and `kms:DescribeKey` permissions, you won't be able to view information about the CMK on the Amazon SQS console. Ask the owner of the CMK to grant you these permissions.

- The AWS managed CMK for Amazon SQS is selected by default.



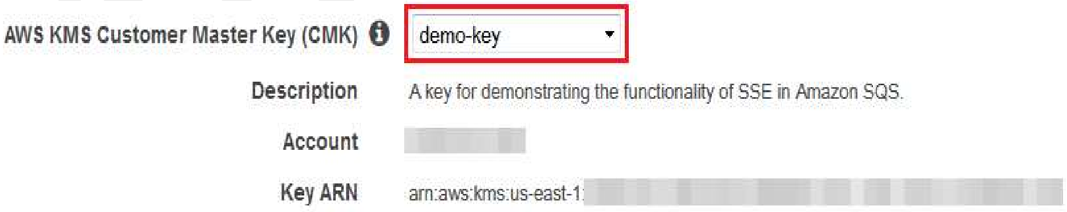
AWS KMS Customer Master Key (CMK): (Default) aws/sqs

Description: Default master key that protects my SQS messages when no other key is defined

Account: [redacted]

Key ARN: arn:aws:kms:us-east-1:[redacted]

- To use a custom CMK from your AWS account, select it from the list.



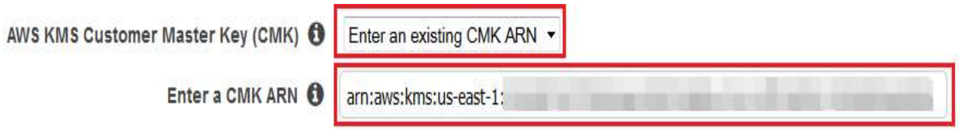
AWS KMS Customer Master Key (CMK): demo-key

Description: A key for demonstrating the functionality of SSE in Amazon SQS.

Account: [redacted]

Key ARN: arn:aws:kms:us-east-1:[redacted]

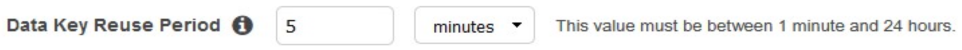
- To use a custom CMK ARN from your AWS account or from another AWS account, select **Enter an existing CMK ARN**from the list and type or copy the CMK.



AWS KMS Customer Master Key (CMK): Enter an existing CMK ARN

Enter a CMK ARN: arn:aws:kms:us-east-1:[redacted]

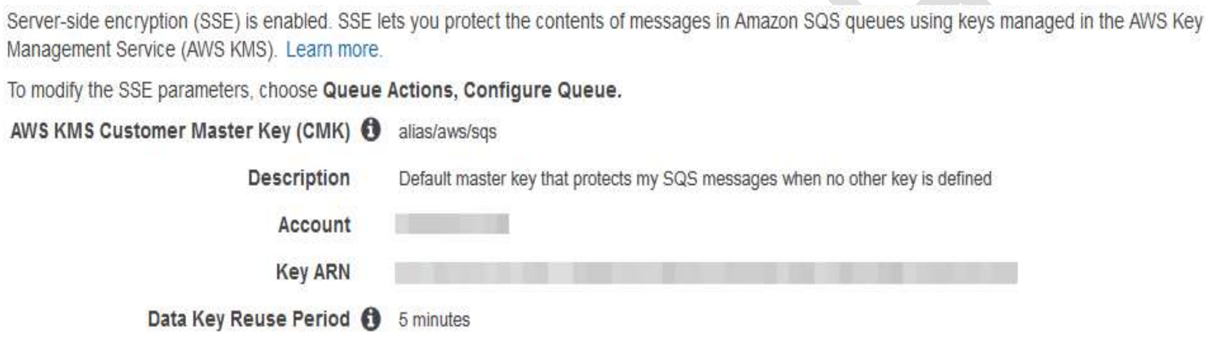
7. (Optional) For **Data key reuse period**, specify a value between 1 minute and 24 hours. The default is 5 minutes.



Data Key Reuse Period: 5 minutes — This value must be between 1 minute and 24 hours.

8. Choose **Create Queue**.

   Your new queue is created with SSE. The encryption status, alias of the CMK, **Description**, **Account**, **Key ARN**, and the **Data Key Reuse Period** are displayed on the **Encryption** tab.



Server-side encryption (SSE) is enabled. SSE lets you protect the contents of messages in Amazon SQS queues using keys managed in the AWS Key Management Service (AWS KMS). Learn more.

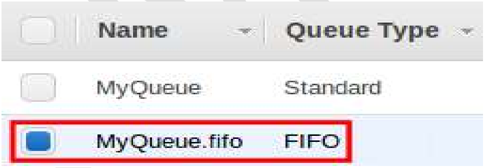
To modify the SSE parameters, choose **Queue Actions, Configure Queue.**

AWS KMS Customer Master Key (CMK)	alias/aws/sqs
Description	Default master key that protects my SQS messages when no other key is defined
Account	[redacted]
Key ARN	[redacted]
Data Key Reuse Period	5 minutes

5. Adding Permissions to an Amazon SQS Queue

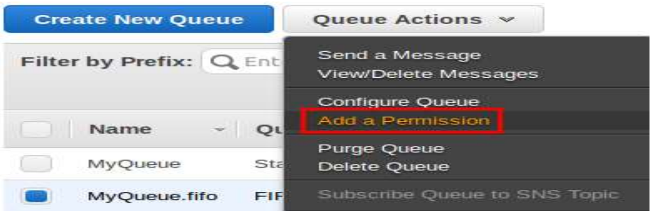
You can specify to whom you allow (or explicitly deny) the ability to interact with your queue in specific ways by adding permissions to a queue.

1. Sign in to the Amazon SQS console.
2. From the queue list, select a queue.



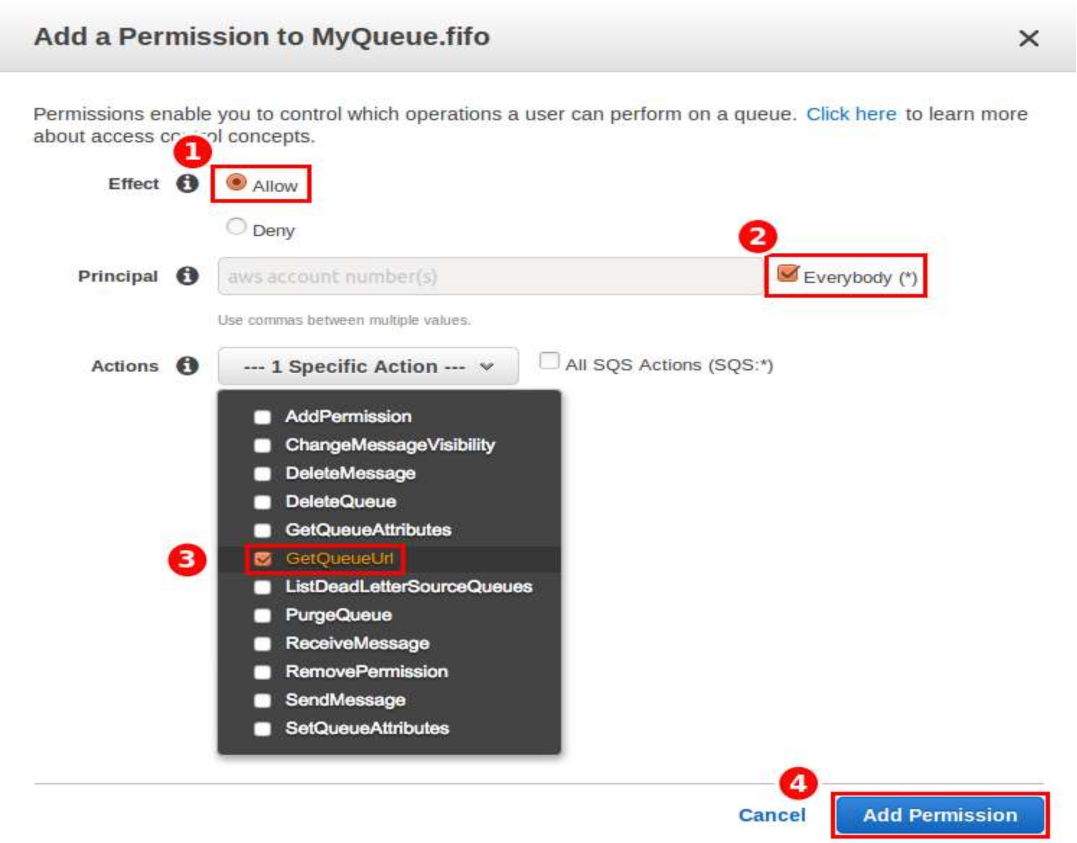
	Name	Queue Type
□	MyQueue	Standard
■	MyQueue.fifo	FIFO

3. From **Queue Actions**, select **Add a Permission**.



The **Add a Permission** dialog box is displayed.

4. In this example, you allow anyone to get the queue's URL:



Add a Permission to MyQueue.fifo

Permissions enable you to control which operations a user can perform on a queue. Click here to learn more about access co~~nt~~rol concepts.

Effect: ● Allow ○ Deny

Principal: aws account number(s) ☑ Everybody (*)

Use commas between multiple values.

Actions: --- 1 Specific Action --- □ All SQS Actions (SQS:*)

1 Ensure that next to **Effect**, **Allow** is selected.

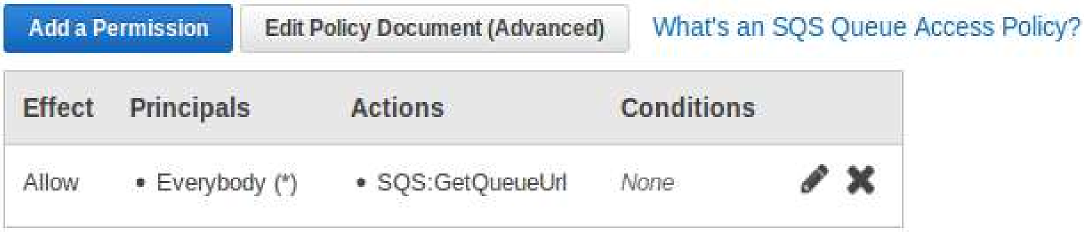
2 Next to **Principal**, check the **Everybody** box.

3 From the **Actions** drop-down list, select **GetQueueUrl** box.

4 Choose **Add Permission**.

The permission is added to the queue.

Your queues' policy **Effect**, **Principals**, **Actions**, and **Conditions** are displayed on your queue's **Permissions** tab.



Add a Permission
Edit Policy Document (Advanced)
What's an SQS Queue Access Policy?

Effect	Principals	Actions	Conditions
Allow	• Everybody (*)	• SQS:GetQueueUrl	*None*

6. Configuring Long Polling for an Amazon SQS Queue

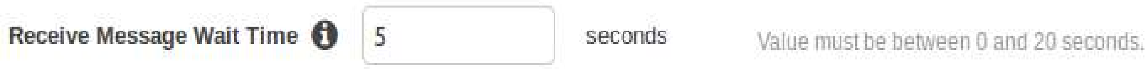
- When the **wait time** for the **`ReceiveMessage`** API action is **greater than 0**, *long polling* is in effect.

- Long polling helps **reduce the cost** of using Amazon SQS by **eliminating the number of empty responses** (when there are no messages available for a `ReceiveMessage` request) and false empty responses (when messages are available but aren't included in a response).

1. Sign in to the Amazon SQS console.
2. Choose **Create New Queue.**
3. On the **Create New Queue** page, ensure that you're in the correct region and then type the **Queue Name**.

Note: The name of a FIFO queue must end with the .fifo suffix.

4. **Standard** is selected by default. Choose **FIFO**.
5. Choose **Configure Queue**.
6. For **Receive Message Wait Time**, type a number between 1 and 20.



Receive Message Wait Time ℹ 5 seconds Value must be between 0 and 20 seconds.

Note: Setting the value to 0 configures ***short polling***.

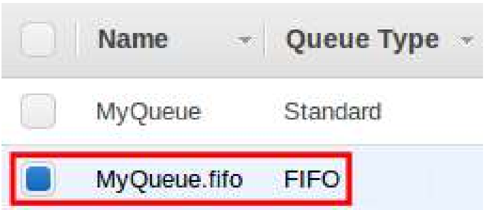
7. Choose **Create Queue**.

   Your new queue is configured to use long polling, created, and selected in the queue list.

7. Sending a Message with Attributes to an Amazon SQS Queue

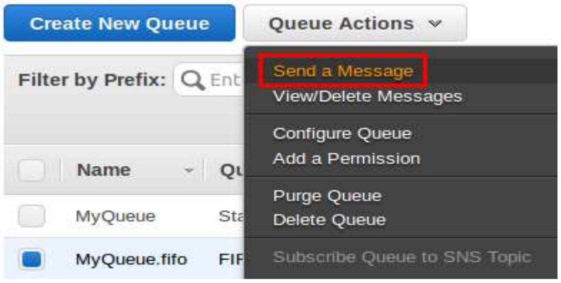
You can include structured metadata (such as timestamps, geospatial data, signatures, and identifiers) with messages using ***message attributes***.

1. Sign in to the **Amazon SQS console**.
2. From the queue list, select a queue.



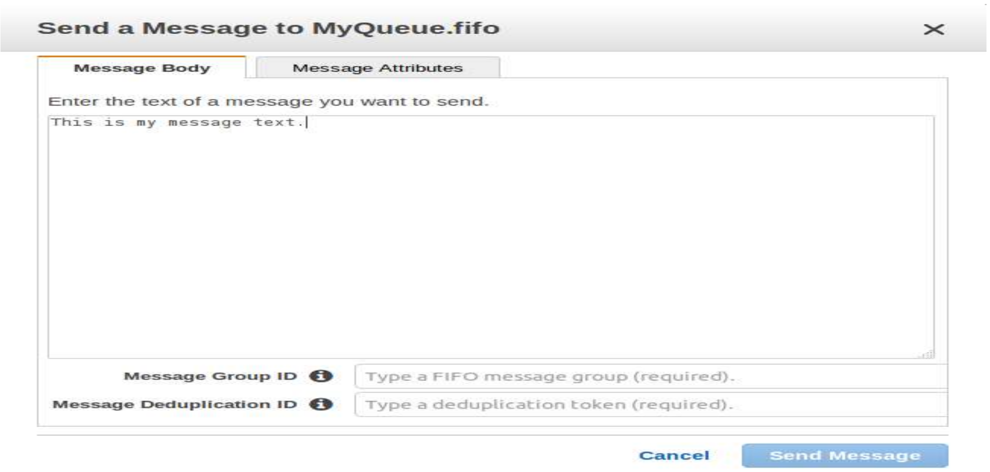
□	Name	Queue Type
□	MyQueue	Standard
■	MyQueue.fifo	FIFO

3. From **Queue Actions**, select **Send a Message**.



The **Send a Message to *QueueName*** dialog box is displayed.

The following example shows the **Message Group ID** and **Message Deduplication ID** parameters specific to FIFO queues (content-based deduplication is disabled).


Send a Message to MyQueue.fifo

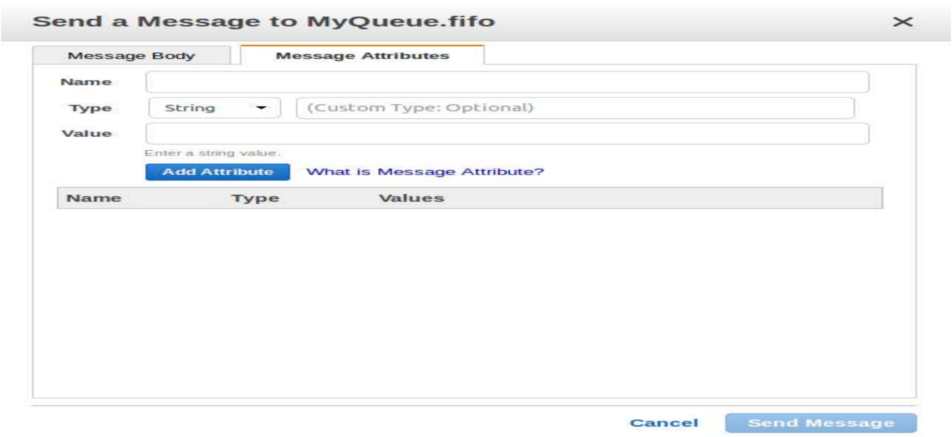
Enter the text of a message you want to send.

```
This is my message text.
```

Message Group ID: Type a FIFO message group (required).

Message Deduplication ID: Type a deduplication token (required).

4. To send a message to a FIFO queue, type the **Message Body**, the **Message Group ID** MyMessageGroupId1234567890, and the **Message Deduplication ID** MyMessageDeduplicationId1234567890, and then choose **Message Attributes**.


Send a Message to MyQueue.fifo

Name:

Type: String (Custom Type: Optional)

Value:

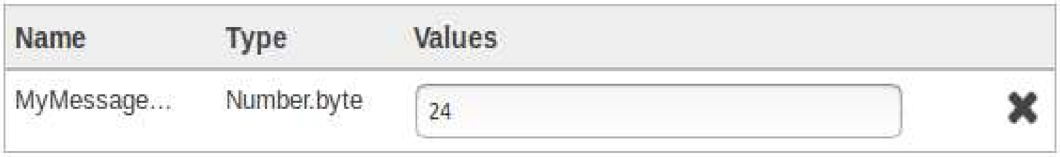
Enter a string value.

Name	Type	Values

5. Define the message attribute parameters. For more information, see Message Attribute Components and Message Attribute Data Types.
    a. For the message attribute **Name** type MyMessageAttribute.
    b. For the message attribute data **Type**, select **Number** and type byte for the optional custom type.
    c. For the message attribute **Value**, type 24.

    Choose **Add Attribute**.

    The attribute is added to the message as **Number.byte**.



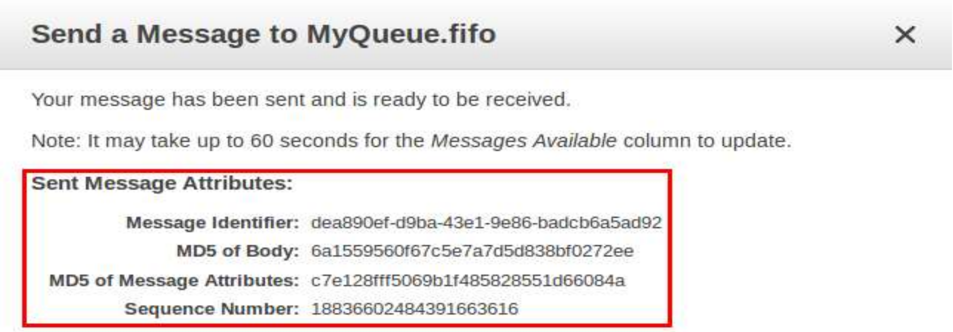
Name	Type	Values	
MyMessage...	Number.byte	24	✖

    You can modify the value before sending the message. To delete the attribute, choose ✖ .

6. When you finish adding attributes to the message, choose **Send Message**.

    Your message is sent and the **Send a Message to *QueueName*** dialog box is displayed, showing the attributes of the sent message.

    The following example shows the **MD5 of Message Attributes** specific to your custom message attribute and the **Sequence Number** attribute specific to FIFO queues.



Send a Message to MyQueue.fifo ×

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

Note: It may take up to 60 seconds for the *Messages Available* column to update.

Sent Message Attributes:

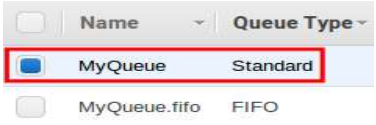
Message Identifier:	dea890ef-d9ba-43e1-9e86-badcb6a5ad92
MD5 of Body:	6a1559560f67c5e7a7d5d838bf0272ee
MD5 of Message Attributes:	c7e128fff5069b1f485828551d66084a
Sequence Number:	18836602484391663616

7. Choose **Close**.

8. Sending a Message with a Timer to an Amazon SQS Queue

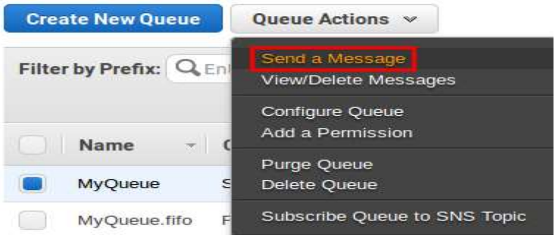
- Message timers let you specify an initial invisibility period for a message added to a queue.
- The default (minimum) delay for a message is 0 seconds. The maximum is 15 minutes.
- For example, if you send a message with a 45-second timer, the message isn't visible to consumers for its first 45 seconds in the queue.

1. Sign in to the Amazon SQS console.
2. From the queue list, select a queue.

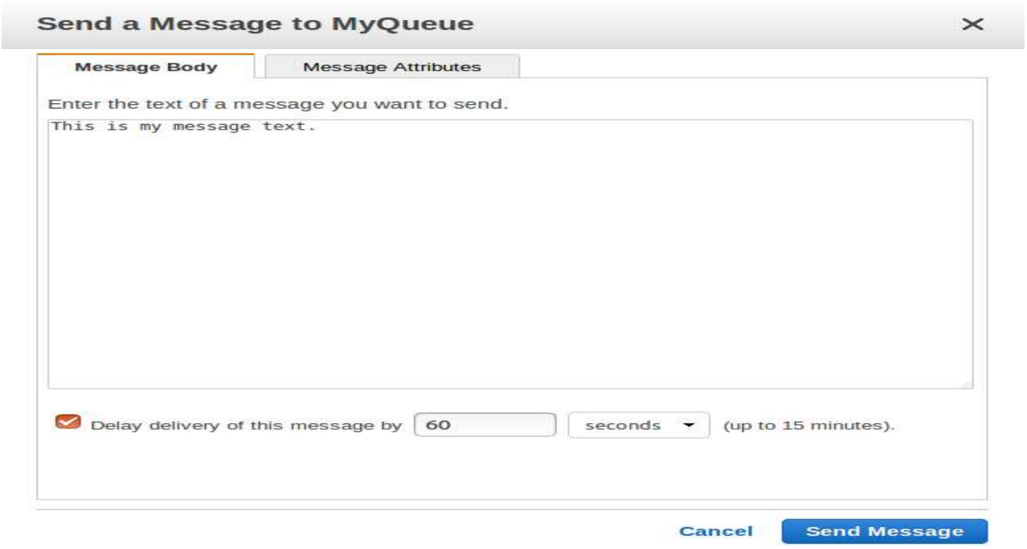


	Name	Queue Type
■	MyQueue	Standard
□	MyQueue.fifo	FIFO

3. From **Queue Actions**, select **Send a Message**.



The **Send a Message to *QueueName*** dialog box is displayed.



Send a Message to MyQueue

Message Body | Message Attributes

Enter the text of a message you want to send.

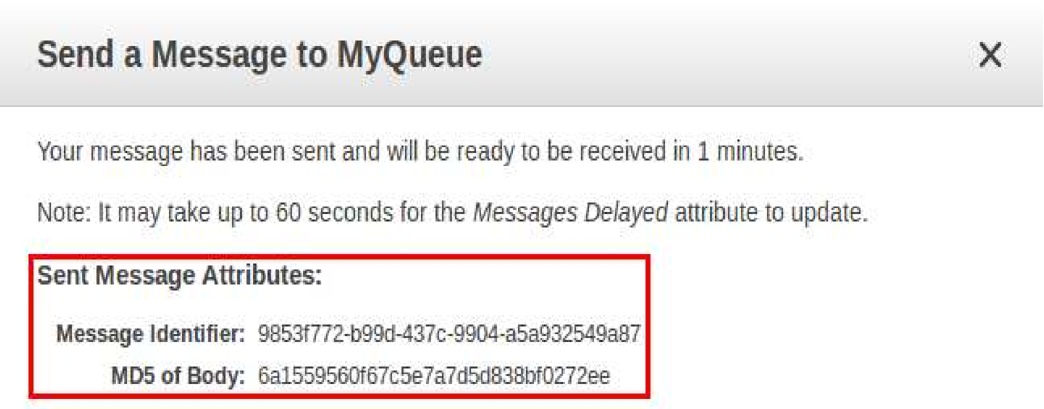
```
This is my message text.
```

☑ Delay delivery of this message by 60 seconds (up to 15 minutes).

4. To send a message to a standard queue, type the **Message Body**, choose **Delay delivery of this message by** and type a value, for example 60 seconds.

5. Choose **Send Message**.

   Your message is sent and the **Send a Message to *QueueName*** dialog box is displayed, showing the attributes of the sent message.



Send a Message to MyQueue

Your message has been sent and will be ready to be received in 1 minutes.

Note: It may take up to 60 seconds for the *Messages Delayed* attribute to update.

Sent Message Attributes:

Message Identifier: 9853f772-b99d-437c-9904-a5a932549a87

MD5 of Body: 6a1559560f67c5e7a7d5d838bf0272ee

6. Choose **Close**.

9. Configuring an Amazon SQS Dead-Letter Queue

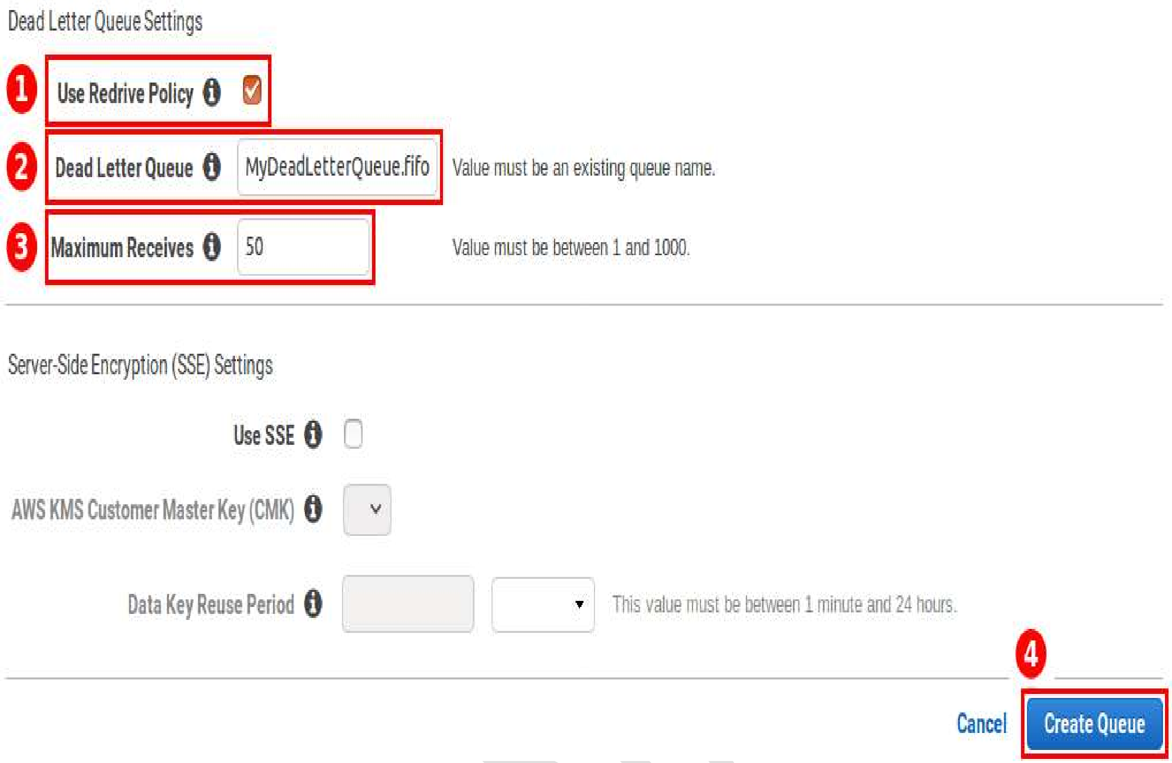
- A dead-letter queue is a queue that *other* (source) queues can target for messages that can't be processed (consumed) successfully.

- The dead-letter queue of a FIFO queue must also be a FIFO queue. Similarly, the dead-letter queue of a standard queue must also be a standard queue.

1. Sign in to the Amazon SQS console.
2. Choose **Create New Queue.**
3. On the **Create New Queue** page, ensure that you're in the correct region and then type the **Queue Name**.

Note: The name of a FIFO queue must end with the `.fifo` suffix.

4. **Standard** is selected by default. Choose **FIFO**.
5. Choose **Configure Queue**.
6. In this example, you enable the redrive policy for your new queue, set the `MyDeadLetterQueue.fifo` queue as the dead-letter queue, and set the number of maximum receives to 50.



Dead Letter Queue Settings

1 Use Redrive Policy ☑

2 Dead Letter Queue MyDeadLetterQueue.fifo Value must be an existing queue name.

3 Maximum Receives 50 Value must be between 1 and 1000.

Server-Side Encryption (SSE) Settings

Use SSE □

AWS KMS Customer Master Key (CMK)

Data Key Reuse Period This value must be between 1 minute and 24 hours.

1 To configure the dead-letter queue, choose **Use Redrive Policy**.

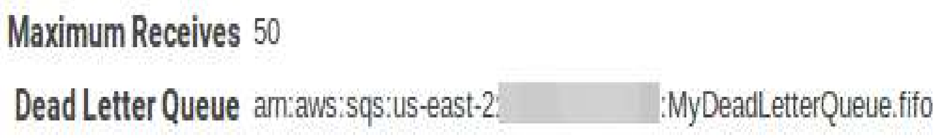
2 Enter the name of the existing **Dead Letter Queue** to which you want sources queues to send messages.

3 To configure the number of times that a message can be received before being sent to a dead-letter queue, set **Maximum Receives** to a value between 1 and 1,000.

Note: The **Maximum Receives** setting applies only to individual messages.

4 Choose **Create Queue**.

- Your new queue is configured to use a dead-letter queue, created, and selected in the queue list.
- Your queue's **Maximum Receives** and **Dead Letter Queue** ARN are displayed on the **Redrive Policy** tab.



Maximum Receives 50

Dead Letter Queue arn:aws:sqs:us-east-2:[redacted]:MyDeadLetterQueue.fifo

10. Configuring Messages Arriving in an Amazon SQS Queue to Trigger an AWS Lambda Function

- How to configure an existing Amazon SQS queue to trigger an AWS Lambda function when new messages arrive in a queue.

- Lambda functions let you run code without provisioning or managing a server.

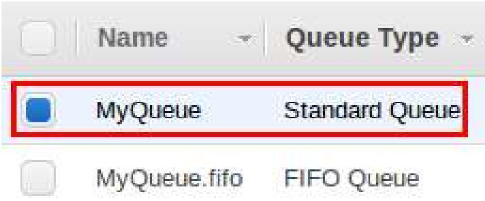
- For example, you can configure a Lambda function to process messages from one queue while another queue acts as a *dead-letter queue* for messages that your Lambda function can't process successfully.

  - Your queue and Lambda function must be in the same AWS Region.
  - FIFO queues don't support Lambda function triggers.
  - You can associate only one queue with one or more Lambda functions.
  - You can't associate an encrypted queue that uses an AWS managed customer master key for Amazon SQS with a Lambda function in a different AWS account.
  - If you associate an encrypted queue with a Lambda function but Lambda doesn't poll for messages, add the kms:Decrypt permission to your Lambda role.

- To configure Lambda function triggers using the console, you must ensure the following:

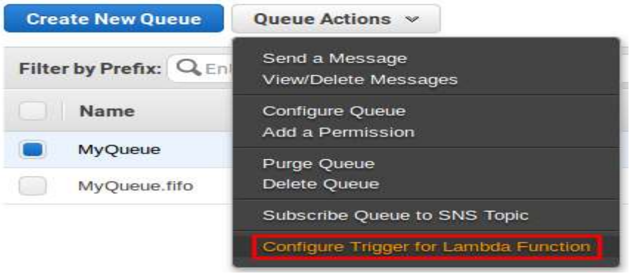
  - If you use an IAM user, your Amazon SQS role must include the following permissions:
    - `lambda:CreateEventSourceMapping`
    - `lambda:ListEventSourceMappings`
    - `lambda:ListFunctions`
  - Your Lambda role must include the following permissions:
    - `sqs:ChangeMessageVisibility`
    - `sqs:DeleteMessage`
    - `sqs:GetQueueAttributes`
    - `sqs:ReceiveMessage`

1. Sign in to the **Amazon SQS console**.
2. From the list of queues, choose the queue which you want to trigger a Lambda function.

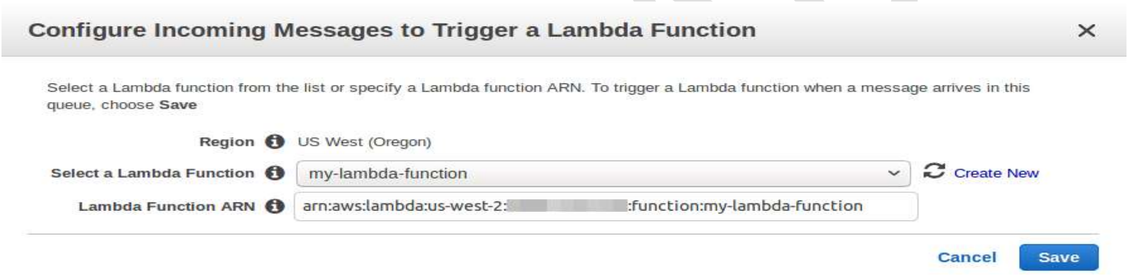


☐	Name	Queue Type
■	MyQueue	Standard Queue
☐	MyQueue.fifo	FIFO Queue

3. From **Queue Actions**, select **Configure Trigger for Lambda Function**.



4. In the **Configure Incoming Messages to Trigger a Lambda Function** dialog box, do one of the following:
   - To use an existing Lambda function, **Select a Lambda Function** from the list.
   - To create a new Lambda function in the AWS Lambda console, choose **Create New**.



Configure Incoming Messages to Trigger a Lambda Function

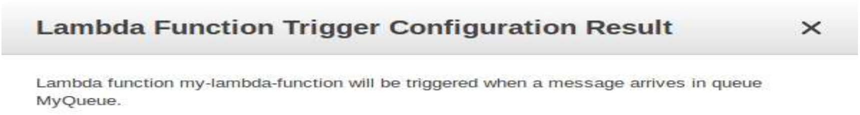
Select a Lambda function from the list or specify a Lambda function ARN. To trigger a Lambda function when a message arrives in this queue, choose **Save**

Region: US West (Oregon)

Select a Lambda Function: my-lambda-function

Lambda Function ARN: arn:aws:lambda:us-west-2:[redacted]:function:my-lambda-function

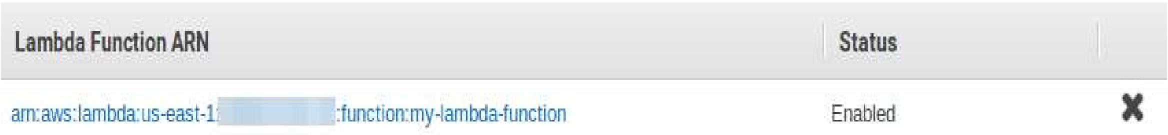
5. Choose **Save**.
6. In the **Lambda Function Trigger Configuration Result** dialog box, review the Lambda function that will be triggered by your Amazon SQS queue and choose **OK**.



Lambda Function Trigger Configuration Result

Lambda function my-lambda-function will be triggered when a message arrives in queue MyQueue.

➢ The Lambda function and its status are displayed on the **Lambda Triggers** tab.



Lambda Function ARN	Status	
arn:aws:lambda:us-east-1:[redacted]:function:my-lambda-function	Enabled	✖

- To verify the results of the configuration, you can send a message to your queue and then view the triggered Lambda function in the Lambda console.
- To delete the association between a Lambda function and your queue, choose ✖ next to a Lambda function ARN.

11. Sending a Message to an Amazon SQS Queue from Amazon Virtual Private Cloud (VPC)

- How to send messages to an Amazon SQS queue over a secure, private network.
- This network consists of a VPC that contains an Amazon EC2 instance. The instance connects to Amazon SQS through an *interface VPC endpoint*, allowing you to connect to the Amazon EC2 instance and send messages to the Amazon SQS queue even though the network is disconnected from the public internet.
- You can use Amazon Virtual Private Cloud only with HTTPS Amazon SQS endpoints.
- When you configure Amazon SQS to send messages from Amazon VPC, you must enable private DNS and specify endpoints in the format `sqs.us-east-2.amazonaws.com`.
- Private DNS doesn't support legacy endpoints such as `queue.amazonaws.com` or `us-east-2.queue.amazonaws.com`.

Step 1: Create an Amazon EC2 Key Pair

A *key pair* lets you connect to an Amazon EC2 instance. It consists of a public key that encrypts your login information and a private key that decrypts it.

1. Sign in to the **Amazon EC2 console.**
2. On the navigation menu, under **Network & Security**, choose **Key Pairs**.
3. Choose **Create Key Pair**.
4. In the **Create Key Pair** dialog box, for **Key pair name**, enter `SQS-VPCE-Tutorial-Key-Pair`, and then choose **Create**.
5. Your browser downloads the private key file `SQS-VPCE-Tutorial-Key-Pair.pem` automatically.

   Important

   Save this file in a safe place. EC2 does not generate a .pem file for the same key pair a second time.

6. To allow an SSH client to connect to your EC2 instance, set the permissions for your private key file so that only your user can have read permissions for it, for example:

```
chmod 400 SQS-VPCE-Tutorial-KeyPair.pem
```

Step 2: Create AWS Resources

To set up the necessary infrastructure, you must use an AWS **CloudFormation *template***, which is a blueprint for creating a *stack* comprised of AWS resources, such as Amazon EC2 instances and Amazon SQS queues.

The stack for this tutorial includes the following resources:

- A VPC and the associated networking resources, including a subnet, a security group, an internet gateway, and a route table
- An Amazon EC2 instance launched into the VPC subnet
- An Amazon SQS queue

1. Download the AWS CloudFormation template named `SQS-VPCE-Tutorial-CloudFormation.yaml` from GitHub.
2. Sign in to the **AWS CloudFormation console**.
3. Choose **Create Stack**.
4. On the **Select Template** page, choose **Upload a template to Amazon S3**, select the `SQS-VPCE-SQS-Tutorial-CloudFormation.yaml` file, and then choose **Next**.
5. On the **Specify Details** page, do the following:
   a. For **Stack name**, enter `SQS-VPCE-Tutorial-Stack`.
   b. For **KeyName**, choose **SQS-VPCE-Tutorial-Key-Pair**.
   c. Choose **Next**.
6. On the **Options** page, choose **Next**.
7. On the **Review** page, in the **Capabilities** section, choose **I acknowledge that AWS CloudFormation might create IAM resources with custom names.**, and then choose **Create**.

➢ AWS CloudFormation begins to create the stack and displays the **CREATE_IN_PROGRESS** status. When the process is complete, AWS CloudFormation displays the **CREATE_COMPLETE** status.

Step 3: Confirm That Your EC2 Instance Isn't Publicly Accessible

➢ Your AWS CloudFormation template launches an EC2 instance named `SQS-VPCE-Tutorial-EC2-Instance` into your VPC.

➢ This EC2 instance doesn't allow outbound traffic and isn't able to send messages to Amazon SQS.

➢ To verify this, you must connect to the instance, try to connect to a public endpoint, and then try to message Amazon SQS.

1. Sign in to the **Amazon EC2 console.**
2. On the navigation menu, under **Instances**, choose **Instances**.
3. Select **SQS-VPCE-Tutorial-EC2Instance**.
4. Copy the hostname under **Public DNS (IPv4)**, for example, **ec2-203-0-113-0.us-west-2.compute.amazonaws.com**.
5. From the directory that contains the key pair that you created earlier, connect to the instance using the following command, for example:

   ```
   ssh -i SQS-VPCE-Tutorial-KeyPair.pem ec2-user@ec2-203-0-113-
   0.us-east-2.compute.amazonaws.com
   ```

6. Try to connect to any public endpoint, for example:

   ```
   ping amazon.com
   ```

   The connection attempt fails, as expected.

7. Sign in to the **Amazon SQS console.**
8. From the list of queues, select the queue created by your AWS CloudFormation template, for example, **VPCE-SQS-Tutorial-Stack-CFQueue-1ABCDEFGH2IJK**.
9. On the **Details** table, copy the URL, for example, **https://sqs.us-east-2.amazonaws.com/123456789012/**.
10. From your EC2 instance, try to publish a message to the queue using the following command, for example:

    ```
    aws sqs send-message --region us-east-2 --endpoint-url
    https://sqs.us-east-2.amazonaws.com/ --queue-url https://sqs.us-east-
    2.amazonaws.com/123456789012/ --message-body "Hello from Amazon SQS."
    ```

    The sending attempt fails, as expected.

Important: Later, when you create a VPC endpoint for Amazon SQS, your sending attempt will succeed.

Step 4: Create an Amazon VPC Endpoint for Amazon SQS

To connect your VPC to Amazon SQS, you must define an interface VPC endpoint. After you add the endpoint, you can use the Amazon SQS API from the EC2 instance in your VPC. This allows you to send messages to a queue within the AWS network without crossing the public internet.

Note: The EC2 instance still doesn't have access to other AWS services and endpoints on the internet.

1. Sign in to the **Amazon VPC console.**
2. On the navigation menu, choose **Endpoints**.
3. Choose **Create Endpoint**.
4. On the **Create Endpoint** page, for **Service Name**, choose the service name for Amazon SQS.

Note: The service names vary based on the current AWS Region. For example, if you are in US East (Ohio), the service name is **com.amazonaws.*us-east-2*.sqs**.

5. For **VPC**, choose **SQS-VPCE-Tutorial-VPC**.
6. For **Subnets**, choose the subnet whose **Subnet ID** contains **SQS-VPCE-Tutorial-Subnet**.
7. For **Security group**, choose **Select security groups**, and then choose the security group whose **Group Name** contains **SQS VPCE Tutorial Security Group**.
8. Choose **Create endpoint**.

   The interface VPC endpoint is created and its ID is displayed, for example, **vpce-0ab1cdef2ghi3j456k**.

9. Choose **Close**.

   The Amazon VPC console opens the **Endpoints** page.

Amazon VPC begins to create the endpoint and displays the **pending** status. When the process is complete, Amazon VPC displays the **available** status.

Step 5: Send a Message to Your Amazon SQS Queue

Now that your VPC includes an endpoint for Amazon SQS, you can connect to your EC2 instance and send messages to your queue.

1. Reconnect to your EC2 instance, for example:

```
ssh -i SQS-VPCE-Tutorial-KeyPair.pem ec2-user@ec2-203-0-113-0.us-
east-2.compute.amazonaws.com
```

2. Try to publish a message to the queue again using the following command, for example:

```
aws sqs send-message --region us-east-2 --endpoint-url
https://sqs.us-east-2.amazonaws.com/ --queue-url https://sqs.us-east-
2.amazonaws.com/123456789012/ --message-body "Hello from Amazon SQS."
```

The sending attempt succeeds and the MD5 digest of the message body and the message ID are displayed, for example:

```
{

      "MD5OfMessageBody": "a1bcd2ef3g45hi678j90klmn12p34qr5",

      "MessageId": "12345a67-8901-2345-bc67-d890123e45fg"

}
```

Step 6: Delete Your VPC, SQS, Instance, Stack, Keypair and VPC Endpoint

1. Deleting a VPC Endpoint
2. Deleting an Amazon SQS Queue
3. Terminate Your Instance
4. Deleting Your VPC
5. Deleting a Stack on the AWS CloudFormation Console
6. Deleting Your Key Pair