PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

Amazon SQS FAQs

Overview

Q: What is Amazon SQS?

Amazon Simple Queue Service (Amazon SQS) is a web service that gives you access to message queues that store messages waiting to be processed. With Amazon SQS, you can quickly build message queuing applications that can run on any computer.

Amazon SQS offers a reliable, highly-scalable, hosted queue for storing messages in transit between computers. With Amazon SQS, you can move data between diverse, distributed application components without losing messages and without requiring each component to be always available.

Amazon SQS can help you build a distributed application with decoupled components, working closely with the Amazon Elastic Compute Cloud (Amazon EC2) and other AWS infrastructure web services.

Q: What can I do with Amazon SQS?

Because Amazon SQS is highly-scalable, you pay only for what you use. You can start small and grow your application alongside your business needs, with no performance or reliability compromises. Amazon SQS lets you stop worrying about how your messages are stored and managed and helps you focus on building robust, sophisticated message-based applications.

Here are just a few ideas:

- Integrate Amazon SQS with other AWS services to make applications more flexible and reliable.
- Use Amazon SQS to create work queues with each message as a task to be completed by a process. Let one (or many) computers read tasks from the message queue and then process them.
- Build a microservice architecture and use message queues to connect your microservices.

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

- Keep notifications of significant business events in an Amazon SQS message queue. Each event can have a corresponding message in a message queue, and applications that need to be aware of the event can read and process the messages.

Q: How can I get started using Amazon SQS?

You can create an Amazon SQS queue and send a message in a few steps.

1. On the AWS Management Console, select **SQS**.
2. On the **Create New Queue** page, ensure you're in the correct region and then type the **Queue Name**, for example, *MyQueue*.
3. Review the descriptions of **Standard Queue** and **FIFO Queue**, and then select a queue type.
4. Select **Quick-Create Queue**.
   Your queue is created.
5. Select **Go to Dashboard**.
   Your new queue is selected in the queue list.
6. From the **Queue Actions** drop-down list, select **Send a Message**.
   The **Send a Message to *MyQueue*** dialog box is displayed.
7. Send your message.
   • To send a message to a standard queue, type text into the **Message Body** and then choose **Send Message**.
   • To send a message to a FIFO queue, type text into the **Message Body**, type the **Message Group ID** (required) and the **Message Deduplication ID** (required) and then choose **Send Message**.
   The message is sent and a confirmation with the message attributes is displayed.
8. To finish, choose **Close**.

For more information, see the *Amazon SQS Developer Guide*, and sample code in the Resource Center.

Q: What are the benefits of Amazon SQS over homegrown or packaged message queuing systems?

Amazon SQS provides several advantages over building your own software for managing message queues or using commercial or open-source message queuing systems that require significant up-front time for development and configuration.

These alternatives require ongoing hardware maintenance and system administration resources. The complexity of configuring and managing these systems is compounded by the need for redundant storage of messages that ensures messages are not lost if hardware fails.

In contrast, Amazon SQS requires no administrative overhead and little configuration. Amazon SQS works on a massive scale, processing billions of messages per day. You can scale the amount of traffic you send to Amazon SQS up or down without any configuration. Amazon SQS also provides extremely high message durability, giving you and your stakeholders added confidence.

Q: Does Amazon SQS provide message ordering?

Yes. FIFO (first-in-first-out) queues preserve the exact order in which messages are sent and received. If you use a FIFO queue, you don't have to place sequencing information in your messages. For more information, see FIFO Queue Logic in the *Amazon SQS Developer*

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

Guide.

Standard queues provide a loose-FIFO capability that attempts to preserve the order of messages. However, because standard queues are designed to be massively scalable using a highly distributed architecture, receiving messages in the exact order they are sent is not guaranteed.

Q: Does Amazon SQS guarantee delivery of messages?

Standard queues provide at-least-once delivery, which means that each message is delivered at least once.

FIFO queues provide exactly-once processing, which means that each message is delivered once and remains available until a consumer processes it and deletes it. Duplicates are not introduced into the queue.

Q: How is Amazon SQS different from Amazon SWF?

Both Amazon SQS and Amazon SWF are services that facilitate the integration of applications or microservices:

- Amazon Simple Queue Service (Amazon SQS) offers reliable, highly-scalable hosted queues for storing messages while they travel between applications or microservices. Amazon SQS lets you move data between distributed application components and helps you decouple these components.
- Amazon Simple Workflow Service (Amazon SWF) is a web service that makes it easy to coordinate work across distributed application components.

The following are the main differences between Amazon SQS and Amazon SWF:

- Amazon SWF API actions are task-oriented. Amazon SQS API actions are message-oriented.
- Amazon SWF keeps track of all tasks and events in an application. Amazon SQS requires you to implement your own application-level tracking, especially if your application uses multiple queues.
- The Amazon SWF Console and visibility APIs provide an application-centric view that lets you search for executions, drill down into an execution's details, and administer executions. Amazon SQS requires implementing such additional functionality.
- Amazon SWF offers several features that facilitate application development, such as passing data between tasks, signaling, and flexibility in distributing tasks. Amazon SQS requires you to implement some application-level functionality.
- In addition to a core SDK that calls service APIs, Amazon SWF provides the AWS Flow Framework with which you can write distributed applications using programming constructs that structure asynchronous interactions.

While you can use Amazon SQS to build basic workflows to coordinate your distributed application, you can get this facility out-of-the-box with Amazon SWF, alongside other application-level capabilities.

We recommend trying both Amazon SQS and Amazon SWF to determine which solution best fits your needs.

Q: How is Amazon SQS different from Amazon Kinesis Streams?

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

Amazon SQS offers a reliable, highly-scalable hosted queue for storing messages as they travel between applications or microservices. It moves data between distributed application components and helps you decouple these components. Amazon SQS provides common middleware constructs such as dead-letter queues and poison-pill management. It also provides a generic web services API and can be accessed by any programming language that the AWS SDK supports. Amazon SQS supports both standard and FIFO queues.

Use Amazon SQS when you need each unique message to be consumed only once and for cases such as the following:

- **Decoupling the components of an application:** You have a queue of work items and want to track the successful completion of each item independently. Amazon SQS tracks the ACK/FAIL results, so the application does not have to maintain a persistent checkpoint or cursor. After a configured visibility timeout, Amazon SQS deletes acknowledged messages and redelivers failed messages.
- **Configuring individual message delay:** You have a job queue and you need to schedule individual jobs with a delay. With Amazon SQS, you can configure individual messages to have a delay of up to 15 minutes.
- **Dynamically increasing concurrency or throughput at read time:** You have a work queue and want to add more readers until the backlog is cleared. With Amazon Kinesis Streams, you can scale up to a sufficient number of shards (however, you must provision enough shards ahead of time). Amazon SQS requires no pre-provisioning.
- **Scaling transparently:** You buffer requests and the load changes as a result of occasional load spikes or the natural growth of your business. Because Amazon SQS can process each buffered request independently, Amazon SQS can scale transparently to handle the load without any provisioning instructions from you.

Amazon Kinesis Streams allows real-time processing of streaming big data and the ability to read and replay records to multiple Amazon Kinesis Applications. The Amazon Kinesis Client Library (KCL) delivers all records for a given partition key to the same record processor, making it easier to build multiple applications that read from the same Amazon Kinesis stream (for example, to perform counting, aggregation, and filtering).

Use Amazon Kinesis Streams when you need multiple consumers to be able to process each record and for use cases such as the following:

- **Routing related records to the same record processor:** You stream MapReduce. Actions such as counting and aggregation are simpler when all records for a given key are routed to the same record processor.
- **Allowing multiple applications to consume the same stream concurrently:** You have one application that updates a real-time dashboard and another that archives data to Amazon Redshift. You want both applications to consume data from the same stream concurrently and independently.

Q: Does Amazon use Amazon SQS for its own applications?

Yes. Developers at Amazon use Amazon SQS for a variety of applications that process large numbers of messages every day. Key business processes in both Amazon.com and Amazon Web Services use Amazon SQS.

Billing

Q: How much does Amazon SQS cost?

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

You pay only for what you use, and there is no minimum fee.

The cost of Amazon SQS is calculated per request, plus data transfer charges for data transferred out of Amazon SQS (unless data is transferred to Amazon EC2 instances or to AWS Lambda functions within the same region). For detailed pricing breakdowns per queue type and region, see Amazon SQS Pricing.

Q: What can I do with the Amazon SQS Free Tier?

The Amazon SQS Free Tier provides you with 1 million requests per month at no charge.

Many small-scale applications are able to operate entirely within the limits of the Free Tier. However, data transfer charges might still apply. For more information, see Amazon SQS Pricing.

The Free Tier is a monthly offer. Free usage does not accumulate across months.

Q: Will I be charged for all Amazon SQS requests?

Yes, for any requests beyond the free tier. All Amazon SQS requests are chargeable, and they are billed at the same rate.

Q: Do Amazon SQS batch operations cost more than other requests?

No. Batch operations (SendMessageBatch, DeleteMessageBatch, and ChangeMessageVisibilityBatch) all cost the same as other Amazon SQS requests. By grouping messages into batches, you can reduce your Amazon SQS costs.

Q: How will I be charged and billed for my use of Amazon SQS?

There are no initial fees to begin using Amazon SQS. At the end of the month, your credit card will be automatically charged for the month's usage.

You can view your charges for the current billing period at any time on the AWS website:

1. Log into your AWS account.
2. Under **Your Web Services Account**, select **Account Activity**.

Q: How can I track and manage the costs associated with my Amazon SQS queues?

You can tag and track your queues for resource and cost management using cost allocation tags. A *tag* is a metadata label comprised of a key-value pair. For example, you can tag your queues by cost center and then categorize and track your costs based on these cost centers.

For more information, see Tagging Your Amazon SQS Queues in the *Amazon SQS Developer Guide*. For more information on cost allocation tagging of AWS resources, see Using Cost Allocation Tags in the *AWS Billing and Cost Management User Guide*.

Q: Do your prices include taxes?

Except as noted otherwise, our prices don't include any applicable taxes and duties such as VAT or applicable sales tax.

For customers with a Japanese billing address, the use of AWS in any region is subject to Japanese Consumption Tax. For more information, see the Amazon Web Services Consumption Tax FAQ.

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

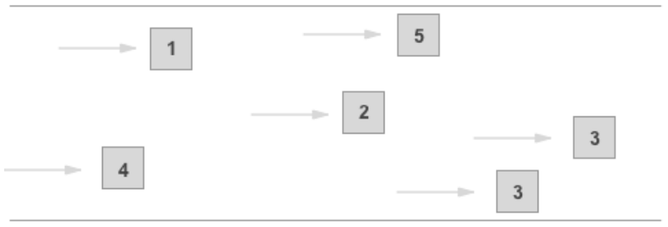
Manage Your Resources

Sign In to the Console

Q: What are the differences between a *standard queue* and a *FIFO queue?*

Standard Queues

Amazon SQS offers *standard* as the default queue type. A standard queue lets you have a nearly-unlimited number of transactions per second. Standard queues guarantee that a message is delivered at least once. However, occasionally (because of the highly-distributed architecture that allows high throughput), more than one copy of a message might be delivered out of order. Standard queues provide best-effort ordering which ensures that messages are generally delivered in the same order as they are sent.



FIFO Queues

The *FIFO queue* complements the standard queue. The most important features of this queue type are *FIFO (first-in-first-out) delivery* and *exactly-once processing*: 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 are not introduced into the queue. FIFO queues also support message groups that allow multiple ordered message groups within a single queue. FIFO queues are limited to 300 transactions per second (TPS), but have all the capabilities of standard queues.

You can use standard message queues in many scenarios, as long as your application can process messages that arrive more than once and out of order, for example:

- Decouple live user requests from intensive background work: Let users upload media while resizing or encoding it.
- Allocate tasks to multiple worker nodes: Process a high number of credit card validation requests.
- Batch messages for future processing: Schedule multiple entries to be added to a database.

FIFO queues are designed to enhance messaging between applications when the order of operations and events is critical, or where duplicates can't be tolerated, for example:

- Ensure that user-entered commands are executed in the right order.
- Display the correct product price by sending price modifications in the right order.
- Prevent a student from enrolling in a course before registering for an account.

Q: What regions are FIFO queues available in?

FIFO queues are currently available in US East (Ohio), US East (N. Virginia), US West (Oregon), and EU (Ireland) regions. This feature will be available in more regions over the coming months.

Q: How many copies of a message will I receive?

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

FIFO queues are designed to never introduce duplicate messages. However, your message producer might introduce duplicates in certain scenarios: for example, if the producer sends a message, does not receive a response, and then resends the same message. Amazon SQS APIs provide deduplication functionality that prevents your message producer from sending duplicates. Any duplicates introduced by the message producer are removed within a 5-minute deduplication interval.

For standard queues, you might occasionally receive a duplicate copy of a message (at-least-once delivery). If you use a standard queue, you must design your applications to be idempotent (that is, they must not be affected adversely when processing the same message more than once).

Q: Are the Amazon SQS queues I used previously changing to FIFO queues?

No. Amazon SQS *standard* queues (the new name for existing queues) remain unchanged, and you can still create standard queues. These queues continue to provide the highest scalability and throughput; however, you will not get ordering guarantees and duplicates might occur.

Standard queues are appropriate for many scenarios, such as work distribution with multiple idempotent consumers.

Q: Can I convert my existing standard queue to a FIFO queue?

No. You must choose the queue type when you create it. However, it is possible to move to a FIFO queue. For more information, see Moving From a Standard Queue to a FIFO Queue in the *Amazon SQS Developer Guide*.

Q: Are Amazon SQS FIFO queues backwards-compatible?

To take advantage of FIFO queue functionality, you must use the latest AWS SDK.

FIFO queues use the same API actions as standard queues, and the mechanics for receiving and deleting messages and changing the visibility timeout are the same. However, when sending messages, you must specify a message group ID. For more information, see FIFO Queue Logic in the *Amazon SQS Developer Guide.*

Important: You can't convert an existing standard queue into a FIFO queue. To make the move, you must either create a new FIFO queue for your application or delete your existing standard queue and recreate it as a FIFO queue. For more information, see Moving from a Standard Queue to a FIFO Queue in the *Amazon SQS Developer Guide*.

Q: With which AWS or external services are Amazon SQS FIFO queues compatible?

Some AWS or external services that send notifications to Amazon SQS might not be compatible with FIFO queues, despite allowing you to set a FIFO queue as a target.

The following features of AWS services aren't currently compatible with FIFO queues:

- Amazon CloudWatch Events
- Amazon S3 Event Notifications
- Amazon SNS Topic Subscriptions

  Note: You can use Amazon SNS to forward messages to Amazon SQS standard queues.
- Auto Scaling Lifecycle Hooks

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

- AWS IoT Rule Actions
- AWS Lambda Dead Letter Queues

For information about compatibility of other services with FIFO queues, see your service documentation.

Q: Are Amazon SQS FIFO queues compatible with the Amazon SQS Buffered Asynchronous Client, the Amazon SQS Extended Client Library for Java, or the Amazon SQS Java Message Service (JMS) Client?

FIFO queues aren't currently compatible with these clients. However, we expect to update these client libraries at a later date. You can continue to use these clients with standard queues.

Q: Which AWS CloudWatch metrics do Amazon SQS FIFO queues support?

FIFO queues support all metrics that standard queues support. For FIFO queues, all *approximate* metrics return accurate counts. For example, the following AWS CloudWatch metrics are supported:

- ApproximateNumberOfMessagesDelayed - The number of messages in the queue that are delayed and not available for reading immediately.
- ApproximateNumberOfMessagesVisible - The number of messages available for retrieval from the queue.
- ApproximateNumberOfMessagesNotVisible - The number of messages that are in flight (sent to a client but have not yet been deleted or have not yet reached the end of their visibility window).

Q: What are message groups?

Messages are grouped into distinct, ordered "bundles" within a FIFO queue. For each message group ID, all messages are sent and received in strict order. However, messages with different message group ID values might be sent and received out of order. You must associate a message group ID with a message. If you don't provide a message group ID, the action fails.

If multiple hosts (or different threads on the same host) send messages with the same message group ID are sent to a FIFO queue, Amazon SQS delivers the messages in the order in which they arrive for processing. To ensure that Amazon SQS preserves the order in which messages are sent and received, ensure that multiple senders send each message with a unique message group ID.

Q: Do Amazon SQS FIFO queues support multiple producers?

Yes. One or more producers can send messages to a FIFO queue. Messages are stored in the order that they were successfully received by Amazon SQS.

If multiple producers send messages in parallel, without waiting for the success response from SendMessage or SendMessageBatch actions, the order between producers might not be preserved. The response of SendMessage or SendMessageBatch actions contains the final ordering sequence that FIFO queues use to place messages in the queue, so your multiple-parallel-producer code can determine the final order of messages in the queue.

Q: Do Amazon SQS FIFO queues support multiple consumers?

Yes. We recommend using a single consumer per message group to support processing messages in the correct order. FIFO queues provide additional ReceiveMessage options that support multiple-consumer scenarios:

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

- In highly-available single-consumer systems, where two or more consumers compete for the next message in the queue, you can configure the queue to make the message visible to other consumers only after the inflight message is deleted or the visibility timeout expires. Thus, only one of the consumers will process the next batch of messages. Other messages must wait until the consumer finishes or fails.
- In parallel message processing, where multiple consumers receive messages, although a FIFO queue returns messages in order, your consumers might see them out of order due to networking issues.

Q: Can I use a dead letter queue with FIFO queues?

Yes. However, you must use a FIFO dead letter queue with a FIFO queue. (Similarly, you can use only a standard dead letter queue with a standard queue.)

Q: What is the throughput limit for an Amazon SQS FIFO queue?

A single FIFO queue currently supports throughput of up to 300 transactions per second (TPS). (A standard queue has unlimited throughput.)

Q: Are there any limits specific to FIFO queue attributes?

The name of a FIFO queue must end with the .fifo suffix. The suffix counts towards the 80-character queue name limit. To determine whether a queue is FIFO, you can check whether the queue name ends with the suffix.

Features, Functionality, and Interfaces

Q: Can I use Amazon SQS with other AWS services?

Yes. You can make your applications more flexible and scalable by using Amazon SQS with compute services such as Amazon EC2, Amazon EC2 Container Service (Amazon ECS), and AWS Lambda, as well as with storage and database services such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB.

One common use case is a distributed, decoupled application whose multiple components and modules need to communicate with each other, but can't do the same amount of work simultaneously. In this case, Amazon SQS message queues carry messages to be processed by the application running on Amazon EC2 instances.

The Amazon EC2 instances can read the message queue, process the job, and then post the results as messages to another Amazon SQS message queue (for example, for further processing by another application). Because Amazon EC2 allows applications to scale up and down dynamically, application developers can vary the number of compute instances based on the amount of messages in the Amazon SQS queues using Auto Scaling, to ensure that jobs are executed in a timely manner.

Q: Can you give me an example use case for Amazon SQS?

Here is how a video transcoding website uses Amazon EC2, Amazon SQS, Amazon S3, and Amazon DynamoDB together:

1. End users submit videos to be transcoded to the website.

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

2. The videos are stored in Amazon S3, and a request message is placed in an incoming Amazon SQS queue with a pointer to the video and to the target video format within the message.
3. The transcoding engine that runs on a set of Amazon EC2 instances reads the request message from the incoming queue, retrieves the video from Amazon S3 using the pointer, and transcodes the video into the target format.
4. The converted video is put back into Amazon S3 and another response message is placed in another outgoing Amazon SQS queue with a pointer to the converted video.
5. At the same time, metadata about the video (format, date created, length, and so on) is indexed into Amazon DynamoDB for querying.

During this workflow, a dedicated Auto Scaling instance can constantly monitor the incoming queue. Based on the number of messages in the incoming queue, the Auto Scaling instance dynamically adjusts the number of transcoding Amazon EC2 instances to meet the response time requirements of the website's customers.

Q: How can I interact with Amazon SQS?

You can access Amazon SQS using the AWS Management Console, which helps you create Amazon SQS queues and send messages easily.

Amazon SQS also provides a web services API. It is also integrated with the AWS SDKs, allowing you to work in the programming language of your choice.

Q: What are the available operations for message queues?

For information on message queue operations, see Amazon SQS Product Details.

Q: Who can perform operations on a message queue?

Only an AWS account owner (or an AWS account that the account owner has delegated rights to) can perform operations on an Amazon SQS message queue.

Q: Can I use Java Message Service (JMS) with Amazon SQS?

Yes. You can take advantage of the scale, low cost, and high availability of Amazon SQS without the worry and high overhead of running your own JMS cluster.

Amazon provides the Amazon SQS Java Messaging Library that implements the JMS 1.1 specification and uses Amazon SQS as the JMS provider. For more information, see Using JMS with Amazon SQS in the *Amazon SQS Developer Guide.*

FIFO queues are not currently supported by the JMS client.

Q: How does Amazon SQS identify messages?

All messages have a global unique ID that Amazon SQS returns when the message is delivered to the message queue. The ID isn't required to perform any further actions on the message, but it is useful for tracking the receipt of a particular message in the message queue.

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

When you receive a message from the message queue, the response includes a receipt handle that you must provide when deleting the message.

Q: How does Amazon SQS handle unsuccessfully-processed messages?

In Amazon SQS, you can use the API or the console to configure *dead letter queues*, which are queues that receive messages from other source queues.

If you make a queue into a dead letter queue, it receives messages after a maximum number of processing attempts cannot be completed. You can use dead letter queues to isolate messages that can't be processed for later analysis.

For more information, see "Can I use a dead letter queue with FIFO queues?" on this page and Using Amazon SQS Dead Letter Queues in the *Amazon SQS Developer Guide*.

Q: What is a visibility timeout?

The visibility timeout is a period of time during which Amazon SQS prevents other consuming components from receiving and processing a message. For more information, see Visibility Timeout in the *Amazon SQS Developer Guide*.

Q: How does Amazon SQS allow multiple readers to access the same message queue without losing messages or processing them multiple times?

Every Amazon SQS queue has a configurable visibility timeout. A message is not visible to any other reader for a designated amount of time when it is read from a message queue. As long as the amount of time it takes to process the message is less than the visibility timeout, every message is processed and deleted.

If the component processing of the message fails or becomes unavailable, the message again becomes visible to any component reading the message queue once the visibility timeout ends. This allows multiple components to read messages from the same message queue, each one working to process different messages.

Q: What is the maximum limit for message visibility?

The maximum visibility timeout for an Amazon SQS message is 12 hours.

Q: Does Amazon SQS support message metadata?

Yes. An Amazon SQS message can contain up to 10 metadata attributes. You can use message attributes to separate the body of a message from the metadata that describes it. This helps process and store information with greater speed and efficiency because your applications don't have to inspect an entire message before understanding how to process it.

Amazon SQS message attributes take the form of name-type-value triples. The supported types include string, binary, and number (including integer, floating-point, and double). For more information, see Using Amazon SQS Message Attributes in the *Amazon SQS Developer Guide.*

Q: How can I determine the time-in-queue value?

To determine the time-in-queue value, you can request the SentTimestamp attribute when receiving a message. Subtracting that value from the current time results in the time-in-queue value.

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

Q: What is the typical latency for Amazon SQS?

Typical latencies for SendMessage, ReceiveMessage, and DeleteMessage API requests are in the tens or low hundreds of milliseconds.

Q: For anonymous access, what is the value of the SenderId attribute for a message?

When the AWS account ID is not available (for example, when an anonymous user sends a message), Amazon SQS provides the IP address.

Q: What is Amazon SQS long polling?

Amazon SQS long polling is a way to retrieve messages from your Amazon SQS queues. 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.

Long polling makes it inexpensive to retrieve messages from your Amazon SQS queue as soon as the messages are available. Using long polling might reduce the cost of using SQS, because you can reduce the number of empty receives. For more information, see Amazon SQS Long Polling in the *Amazon SQS Developer Guide.*

Q: Is there an additional charge for using Amazon SQS long polling?

No. Long-polling ReceiveMessage calls are billed exactly the same as short-polling ReceiveMessage calls.

Q: When should I use Amazon SQS long polling, and when should I use Amazon SQS short polling?

In almost all cases, Amazon SQS long polling is preferable to short polling. Long-polling requests let your queue consumers receive messages as soon as they arrive in your queue while reducing the number of empty ReceiveMessageResponse instances returned.

Amazon SQS long polling results in higher performance at reduced cost in the majority of use cases. However, if your application expects an immediate response from a ReceiveMessage call, you might not be able to take advantage of long polling without some modifications to your application.

For example, if your application uses a single thread to poll multiple queues, switching from short polling to long polling will probably not work, because the single thread will wait for the long-poll timeout on any empty queues, delaying the processing of any queues that might contain messages.

In such an application, it is a good practice to use a single thread to process only one queue, allowing the application to take advantage of the benefits that Amazon SQS long polling provides.

Q: What value should I use for my long-poll timeout?

In general, you should use maximum 20 seconds for a long-poll timeout. Because higher long-poll timeout values reduce the number of empty ReceiveMessageResponse instances returned, try to set your long-poll timeout as high as possible.

If the 20-second maximum doesn't work for your application (see the example in the previous question), set a shorter long-poll timeout, as low as 1 second.

All AWS SDKs work with 20-second long polls by default. If you don't use an AWS SDK to access Amazon SQS, or if you configured your AWS SDK to specifically have a shorter timeout, you might need to modify your Amazon SQS client to allow longer requests or to use a

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

shorter long-poll timeout.

Q: What is the AmazonSQSBufferedAsyncClient for Java?

The AmazonSQSBufferedAsyncClient for Java provides an implementation of the AmazonSQSAsyncClient interface and adds several important features:

- Automatic batching of multiple SendMessage, DeleteMessage, or ChangeMessageVisibility requests without any required changes to the application
- Prefetching of messages into a local buffer that allows your application to immediately process messages from Amazon SQS without waiting for the messages to be retrieved

Working together, automatic batching and prefetching increase the throughput and reduce the latency of your application while reducing your costs by making fewer Amazon SQS requests. For more information, see Client-Side Buffering and Request Batching in the *Amazon SQS Developer Guide.*

The Amazon SQS Buffered Asynchronous Client does not currently support FIFO queues.

Q: Where can I download the AmazonSQSBufferedAsyncClient for Java?

You can download the AmazonSQSBufferedAsyncClient as part of the AWS SDK for Java.

Q: Do I have to rewrite my application to use the AmazonSQSBufferedAsyncClient for Java?

No. The AmazonSQSBufferedAsyncClient for Java is implemented as a drop-in replacement for the existing AmazonSQSAsyncClient.

If you update your application to use the latest AWS SDK and change your client to use the AmazonSQSBufferedAsyncClient for Java instead of the AmazonSQSAsyncClient, your application will receive the added benefits of automatic batching and prefetching.

Q: How can I subscribe Amazon SQS message queues to receive notifications from Amazon SNS topics?

1. In the Amazon SQS console, select an Amazon SQS standard queue.
2. Under **Queue Actions**, select **Subscribe Queue to SNS Topic** from the drop-down list.
3. In the dialog box, select the topic from the **Choose a Topic** drop-down list, and click **Subscribe**.

For more information, see Subscribing a Queue to an Amazon SNS Topic in the *Amazon SQS Developer Guide.*

Q: How can I fan out identical messages to multiple Amazon SQS queues?

1. Use Amazon SNS to create a topic.
2. Create and subscribe multiple Amazon SQS standard queues to the Amazon SNS topic.
3. Whenever a message is sent to the Amazon SNS topic, it is fanned out to the Amazon SQS message queues.

Amazon SNS delivers the message to all Amazon SQS message queues subscribed to the topic.

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

Q: Does Amazon SNS provide at-least-once delivery of messages to Amazon SQS queues?

Amazon SNS is designed such that each message is delivered at least once to Amazon SQS standard queues.

Q: Can I delete all messages in a message queue without deleting the message queue itself?

Yes. You can delete all messages in an Amazon SQS message queue using the PurgeQueue action.

When you purge a message queue, all the messages previously sent to the message queue are deleted. Because your message queue and its attributes remain, there is no need to reconfigure the message queue; you can continue using it.

To delete only specific messages, use the DeleteMessage or DeleteMessageBatch actions.

Security and Reliability

Q: How reliable is the storage of my data in Amazon SQS?

Amazon SQS stores all message queues and messages within a single, highly-available AWS region with multiple redundant Availability Zones (AZs), so that no single computer, network, or AZ failure can make messages inaccessible. For more information, see Regions and Availability Zones in the *Amazon Relational Database Service User Guide*.

Q: How can I secure the messages in my message queues?

Authentication mechanisms ensure that messages stored in Amazon SQS message queues are secured against unauthorized access. You can control who can send messages to a message queue and who can receive messages from a message queue. For additional security, you can build your application to encrypt messages before they are placed in a message queue.

Amazon SQS has its own resource-based permissions system that uses policies written in the same language as AWS Identity and Access Management (IAM) policies: for example, you can use variables, just like in IAM policies. For more information, see Amazon SQS Policy Examples in the *Amazon SQS Developer Guide*.

Amazon SQS supports the HTTP over SSL (HTTPS) and Transport Layer Security (TLS) protocols. Most clients can automatically negotiate to use newer versions of TLS without any code or configuration change. Amazon SQS supports versions 1.0, 1.1, and 1.2 of the Transport Layer Security (TLS) protocol in all regions.

Q: Why are there separate ReceiveMessage and DeleteMessage operations?

When Amazon SQS returns a message to you, the message stays in the message queue whether or not you actually receive the message. You're responsible for deleting the message and the deletion request acknowledges that you're done processing the message.

If you don't delete the message, Amazon SQS will deliver it again on when it receives another receive request. For more information, see Visibility Timeout in the *Amazon SQS Developer Guide.*

Q: Can a deleted message be received again?

No. FIFO queues never introduce duplicate messages.

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

For standard queues, under rare circumstances, you might receive a previously-deleted message a second time. This can happen in the rare situation when a DeleteMessage operation doesn't delete all copies of a message because one of the servers in the distributed Amazon SQS system isn't available at the time of deletion. This message copy can be delivered again. If you use standard queues, design your application to be *idempotent* (that is, no errors or inconsistencies occur if you receive a deleted message a second time).

Q: What happens if I issue a DeleteMessage request on a previously-deleted message?

When you issue a DeleteMessage request on a previously-deleted message, Amazon SQS returns a *success* response.

Compliance

Q: Is Amazon SQS PCI DSS certified?

Yes. Amazon SQS is PCI DSS Level 1 certified. For more information, see PCI Compliance.

Q: Is Amazon SQS HIPAA compliant?

No. Amazon SQS is not yet eligible for HIPAA compliance.

However, you can use the Extended Client Library to send Amazon SQS message payloads through Amazon S3 (Amazon S3 is an HIPAA-eligible service).You can achieve HIPAA compliance in this manner, because no personally identifiable information (PII) is transferred via Amazon SQS.

For more information, and Using the Amazon SQS Extended Client Library for Java in the *Amazon SQS Developer Guide*.

Limits and Restrictions

Q: How long can I keep my messages in Amazon SQS message queues?

Longer message retention provides greater flexibility to allow for longer intervals between message production and consumption.

You can configure the Amazon SQS message retention period to a value from 1 minute to 14 days. The default is 4 days. Once the message retention limit is reached, your messages are automatically deleted.

Q: How do I configure Amazon SQS to support longer message retention?

To configure the message retention period, set the MessageRetentionPeriod attribute using the console or using the Distributiveness method. Use this attribute to specify the number of seconds a message will be retained in Amazon SQS.

You can use the MessageRetentionPeriod attribute to set the message retention period from 60 seconds (1 minute) to 1,209,600 seconds (14 days). For more information on working with this message attribute, see the *Amazon SQS API Reference*.

Q: How do I configure the maximum message size for Amazon SQS?

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

To configure the maximum message size, use the console or the SetQueueAttributes method to set the MaximumMessageSize attribute. This attribute specifies the limit on bytes that an Amazon SQS message can contain. Set this limit to a value between 1,024 bytes (1 KB), and 262,144 bytes (256 KB). For more information, see Using Amazon SQS Message Attributes in the *Amazon SQS Developer Guide*.

To send messages larger than 256 KB, use the Amazon SQS Extended Client Library for Java. This library lets you send an Amazon SQS message that contains a reference to a message payload in Amazon S3 that can be as large as 2 GB. For FIFO queues, see "Are Amazon SQS FIFO queues compatible with the Amazon SQS Buffered Asynchronous Client, the Amazon SQS Extended Client Library for Java, or the Amazon SQS Java Message Service (JMS) Client?"

Q: What kind of data can I include in a message?

Amazon SQS messages can contain up to 256 KB of text data, including XML, JSON and unformatted text. The following Unicode characters are accepted:

#x9 | #xA | #xD | [#x20 to #xD7FF] | [#xE000 to #xFFFD] | [#x10000 to #x10FFFF]

For more information, see the XML 1.0 Specification.

Q: How large can Amazon SQS message queues be?

A single Amazon SQS message queue can contain an unlimited number of messages. However, there is a 120,000 limit for the number of inflight messages for a standard queue and 20,000 for a FIFO queue. Messages are inflight after they have been received from the queue by a consuming component, but have not yet been deleted from the queue.

Q: How many message queues can I create?

You can create any number of message queues.

Q: Is there a size limit on the name of Amazon SQS message queues?

Queue names are limited to 80 characters.

Q: Are there restrictions on the names of Amazon SQS message queues?

You can use alphanumeric characters, hyphens (-), and underscores (_).

Q: Can I reuse a message queue name?

A message queue's name must be unique within an AWS account and region. You can reuse a message queue's name after you delete the message queue.

Queue Sharing

Q: How do I share a message queue?

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

You can associate an access policy statement (and specify the permissions granted) with the message queue to be shared. Amazon SQS provides APIs for creating and managing access policy statements:

- AddPermission
- RemovePermission
- SetQueueAttributes
- GetQueueAttributes

For more information, see the *Amazon SQS API Reference*.

Q: Who pays for shared queue access?

The message queue owner pays for shared message queue access.

Q: How do I identify another AWS user I want to share a message queue with?

The Amazon SQS API uses the AWS account number to identify AWS users.

Q: What do I need to provide to an AWS user I want to share a message queue with?

To share a message queue with an AWS user, provide the full URL from the message queue you want to share. The CreateQueue and ListQueues operations return this URL in their responses.

Q: Does Amazon SQS support anonymous access?

Yes. You can configure an access policy that allows anonymous users to access a message queue.

Q: When should I use the permissions API?

The permissions API provides an interface for sharing access to a message queue to developers. However, this API cannot allow conditional access or more advanced use cases.

Q: When should I use the SetQueueAttributes operation with JSON objects?

The SetQueueAttributes operation supports the full access policy language. For example, you can use the policy language to restrict access to a message queue by IP address and time of day. For more information, see Amazon SQS Policy Examples in the *Amazon SQS Developer Guide*.

Service Access and Regions

Q: What regions is Amazon SQS available in?

For service region availability, see the AWS Global Infrastructure Region Table. FIFO queues are currently available only in the US East (Ohio) and US West (Oregon) regions.

Q: Can I share messages between queues in different regions?

No. Each Amazon SQS message queue is independent within each region.

Q: Is there a pricing difference between regions?

Amazon SQS pricing is the same for all regions, except China (Beijing). For more information, see Amazon SQS Pricing.

Q: What is the pricing structure between various regions?

You can transfer data between Amazon SQS and Amazon EC2 or AWS Lambda free of charge within a single region.

When you transfer data between Amazon SQS and Amazon EC2 or AWS Lambda in different regions, you will be charged the normal data transfer rate. For more information, see Amazon SQS Pricing.

PRODUCTS & SERVICES

Amazon SQS
Product Details
Pricing
Developer Resources
SQS FAQs
Getting Started

RELATED LINKS

Documentation
Management Console
Release Notes
Discussion Forum

Architecture Center
Security Center
What's New
Whitepapers
AWS Blog
Events
Sustainable Energy
Press Releases
AWS in the News
Analyst Reports
Legal

Solutions

Websites & Website Hosting
Business Applications
Backup & Recovery
Disaster Recovery
Data Archive
DevOps
Serverless Computing
Big Data
High Performance Computing
Mobile Services
Digital Marketing
Game Development
Digital Media
Government & Education
Health
Financial Services
Windows on AWS

Resources & Training

Developers
Java on AWS
JavaScript on AWS
Mobile on AWS
PHP on AWS
Python on AWS
Ruby on AWS
.NET on AWS
SDKs & Tools
AWS Marketplace
User Groups
Support Plans
Service Health Dashboard

Discussion Forums
FAQs
Documentation
Articles & Tutorials
Test Drives
AWS Business Builder

Manage Your Account
Management Console
Billing & Cost Management
Subscribe to Updates
Personal Information
Payment Method
AWS Identity & Access Management
Security Credentials
Request Service Limit Increases
Contact Us

Language Deutsch | English | Español | Français | Italiano | Português | Русский | 日本語 | 한국어 | 中文 (简体) | 中文 (繁體)

Site Terms | Privacy