

Introducing the Amazon Simple Queue Service

What is Amazon Simple Queue Service?

Amazon Simple Queue Service (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 familiar 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**.

What Can I Use Amazon SQS For?

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 standard queues, 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 consumers until the backlog is cleared. 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.

What Type of Queue Do I Need?

Standard Queue

1. Available in all regions.
2. **High Throughput** – Standard queues have nearly-unlimited transactions per second (TPS).
3. **At-Least-Once Delivery** – A message is delivered at least once, but occasionally more than one copy of a message is delivered.
4. **Best-Effort Ordering** – Occasionally, messages might be delivered in an order different from which they were sent.
5. Send data between applications when the throughput is important,

FIFO Queue

1. Available in the US West (Oregon) and US East (Ohio) regions.
2. **First-In-First-Out Delivery** – The order in which messages are sent and received is strictly preserved.
3. **Exactly-Once Processing** – A message is delivered once and remains available until a consumer processes and deletes it. Duplicates are not introduced into the queue.
4. **Limited Throughput** – 300 transactions per second (TPS).
5. Send data between applications when the order of events is important

What Are the Main Features of Amazon SQS?

Amazon SQS provides the following major features:

- **Redundant infrastructure** – Standard queues support at-least-once message delivery, while FIFO queues support exactly-once message processing. Amazon SQS provides highly-concurrent access to messages and high availability for producing and consuming messages.
- **Multiple producers and consumers** – Multiple parts of your system can send or receive messages at the same time. Amazon SQS locks the message during processing, keeping other parts of your system from processing the message simultaneously.
- **Configurable settings per queue** – All of your queues don't have to be exactly alike. For example, you can optimize one queue for messages that require a longer processing time than others.
- **Variable message size** – Your messages can be up to 262,144 bytes (256 KB) in size. You can store the contents of larger messages using the Amazon Simple

Storage Service (Amazon S3) or Amazon DynamoDB, with Amazon SQS holding a pointer to the Amazon S3 object.

- **Access control** – You control who can send messages to a queue, and who can receive messages from a queue.
- **Delay queues** – You can set a default delay on a queue, so that delivery of all enqueued messages is postponed for the specified duration. You can set the delay value when you create a queue with `CreateQueue`, and you can update the value with `SetQueueAttributes`. If you update the value, the new value affects only messages enqueued after the update.
- **PCI compliance** – Amazon SQS supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as compliant with Payment Card Industry (PCI) Data Security Standard (DSS).

What is the Basic Architecture of Amazon SQS?

There are three main actors in the overall system:

- The components of your distributed system
- Queues
- Messages in the queues

