Application Integration

# **Amazon SNS**

Amazon Simple Notification Service (Amazon SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud.

Amazon SNS is used for building and integrating loosely-coupled, distributed applications.

Provides instantaneous, push-based delivery (no polling).

Uses simple APIs and easy integration with applications.

Flexible message delivery is provided over multiple transport protocols.

Offered under an inexpensive, pay-as-you-go model with no up-front costs.

The web-based AWS Management Console offers the simplicity of a point-and-click interface.

Data type is JSON.

SNS supports a wide variety of needs including event notification, monitoring applications, workflow systems, time-sensitive information updates, mobile applications, and any other application that generates or consumes notifications.

SNS Subscribers:

* HTTP.
* HTTPS.
* Email.
* Email-JSON.
* SQS.
* Application.
* Lambda.

SNS supports notifications over multiple transport protocols:

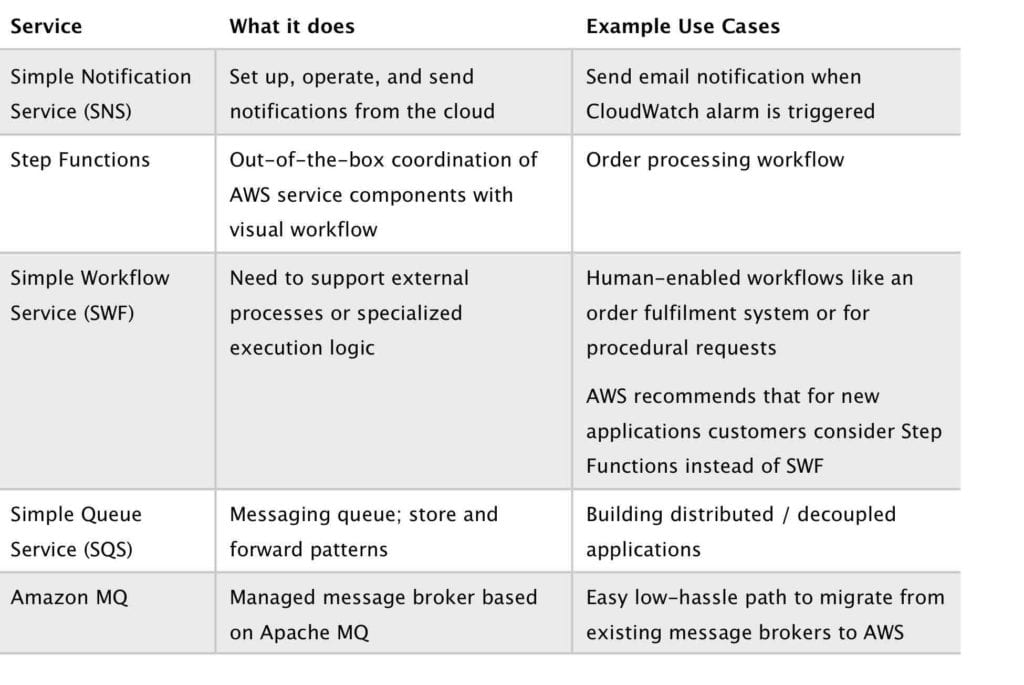
* HTTP/HTTPS – subscribers specify a URL as part of the subscription registration.
* Email/Email-JSON – messages are sent to registered addresses as email (text-based or JSON-object).
* SQS – users can specify an SQS standard queue as the endpoint.
* SMS – messages are sent to registered phone numbers as SMS text messages.

Topic names are limited to 256 characters.

SNS supports CloudTrail auditing for authenticated calls.

SNS provides durable storage of all messages that it receives (across multiple AZs).

Users pay $0.50 per 1 million Amazon SNS Requests, $0.06 per 100,000 notification deliveries over HTTP, and $2.00 per 100,000 notification deliveries over email.

The following table describes related services and typical use cases for them: 

# **Amazon SQS**

## General SQS Concepts

Amazon Simple Queue Service (Amazon SQS) is a web service that gives you access to message queues that store messages waiting to be processed.

SQS offers a reliable, highly-scalable, hosted queue for storing messages in transit between computers.

SQS is used for distributed/decoupled applications.

SQS can be used with RedShift, DynamoDB, EC2, ECS, RDS, S3 and Lambda.

SQS uses a message-oriented API.

SQS uses pull based (polling) not push based.

Messages are 256KB in size.

Messages can be kept in the queue from 1 minute to 14 days (default is 4 days).

The visibility timeout is the amount of time a message is invisible in the queue after a reader picks up the message.

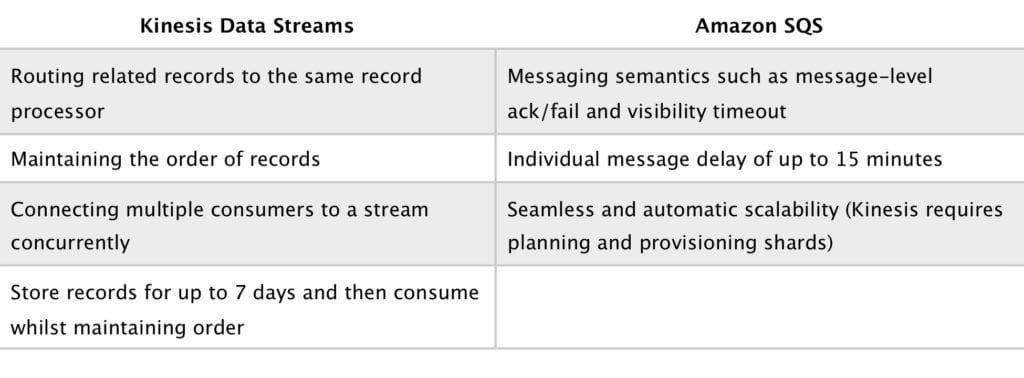
If a job is processed within the visibility timeout the message will be deleted.

If a job is not processed within the visibility timeout the message will become visible again (could be delivered twice).

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

An Amazon SQS message can contain up to 10 metadata attributes.

The table below compares solution requirements that are more suitable for Amazon Kinesis Data Streams and Amazon SQS:



Polling

SQS uses short polling and long polling.

Short polling:

* Does not wait for messages to appear in the queue.
* It queries only a subset of the available servers for messages (based on weighted random execution).
* Short polling is the default.
* ReceiveMessageWaitTime is set to 0.
* More requests are used, which implies higher cost.

Long polling:

* Uses fewer requests and reduces cost.
* Eliminates false empty responses by querying all servers.
* SQS waits until a message is available in the queue before sending a response.
* Requests contain at least one of the available messages up to the maximum number of messages specified in the ReceiveMessage action.
* Shouldn’t be used if your application expects an immediate response to receive message calls.
* ReceiveMessageWaitTime is set to a non-zero value (up to 20 seconds).
* Same charge per million requests as short polling.

## Queues

Queue names must be unique within a region.

Queues can be either standard or first-in-first-out (FIFO).

Standard queues provide a loose-FIFO capability that attempts to preserve the order of messages.

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.

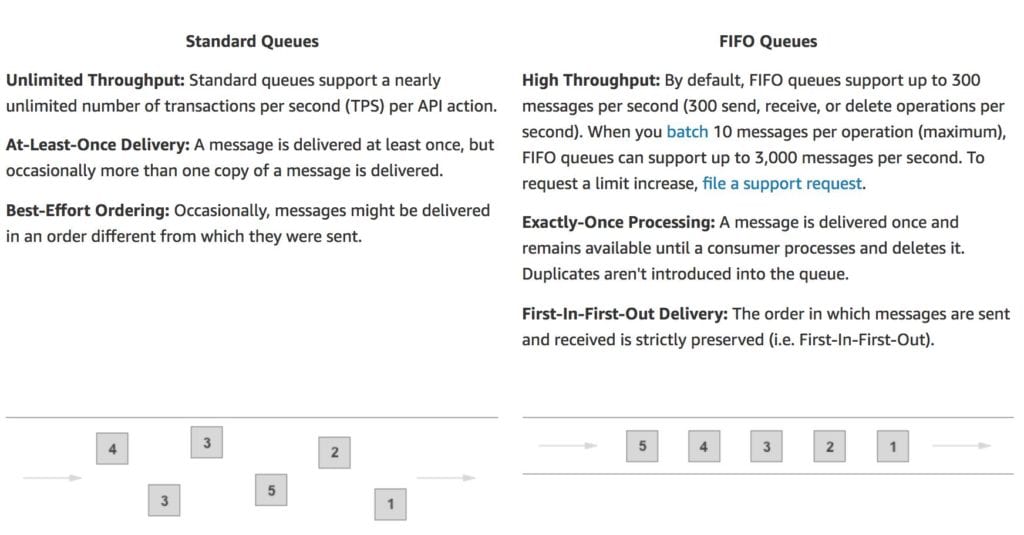
Standard queues provide at-least-once delivery, which means that each message is delivered at least once.

FIFO (first-in-first-out) queues preserve the exact order in which messages are sent and received.

FIFO queues are available in limited regions currently.

If you use a FIFO queue, you don’t have to place sequencing information in your message.

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.



## Limits

In-flight messages are messages that have been picked up by a consumer but not yet deleted from the queue.

Standard queues have a limit of 120,000 in-flight messages per queue.

FIFO queues have a limit of 20,000 in-flight messages per queue.

Queue names can be up to 80 characters.

Messages are retained for 4 days by default up to 14 days.

FIFO queues support up to 3000 messages per second when batching or 300 per second otherwise.

The maximum messages size is 256KB.

## Scalability and Durability

You can have multiple queues with different priorities.

Scaling is performed by creating more queues.

SQS stores all message queues and messages within a single, highly-available AWS region with multiple redundant AZs.

## Security

You can use IAM policies to control who can read/write messages.

Authentication can be used to secure messages within queues (who can send and receive).

SQS supports HTTPS and supports TLS versions 1.0, 1.1, 1.2.

SQS is PCI DSS level 1 compliant and HIPAA eligible.

Server-side encryption (SSE) lets you transmit sensitive data in encrypted queues (AWS KMS):

* SSE encrypts messages as soon as SQS receives them.
* The messages are stored in encrypted form and SQS decrypts messages only when they are sent to an authorized consumer.
* Uses AES 256 bit encryption.
* Not available in all regions.
* Standard and FIFO queues.
* Body of message is encrypted.
* The following is not encrypted:
  + Queue metadata.
  + Message metadata.
  + Per-queue metrics.

Monitoring

CloudWatch is integrated with SQS and you can view and monitor queue metrics.

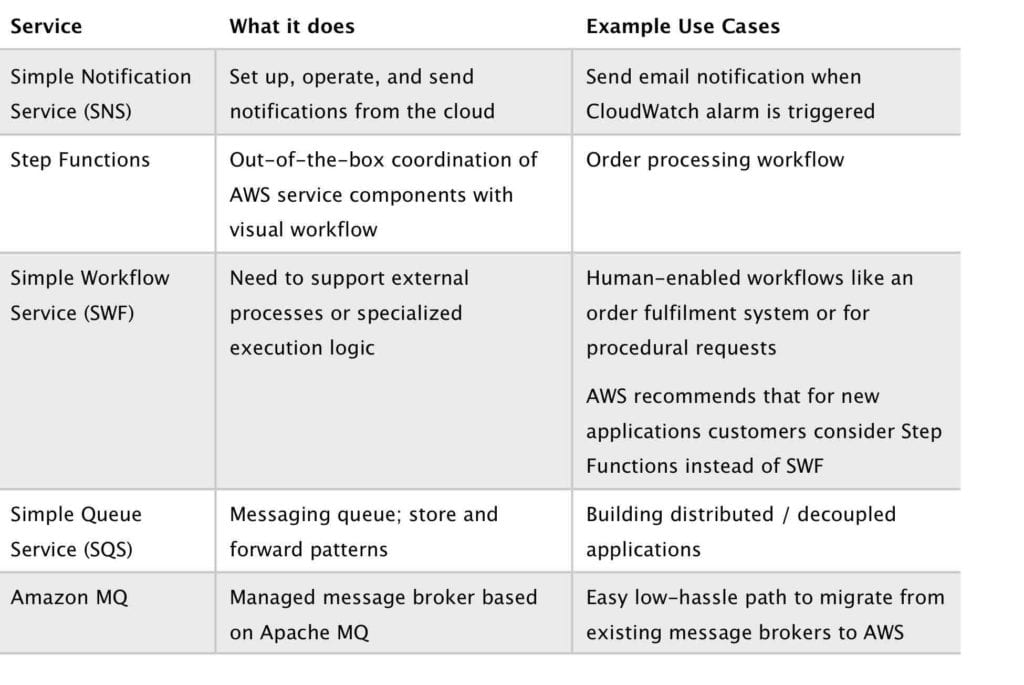
CloudWatch metrics are automatically collected every 5 minutes.

CloudWatch considers a queue to be active for up to 6 hours if it contains any messages or if any API action accesses it.

No charge for CloudWatch (no detailed monitoring).

CloudTrail captures API calls from SQS and logs to a specified S3 bucket.

The following table describes related services and typical use cases for them:



# **Amazon SWF**

Amazon Simple Workflow Service (SWF) is a web service that makes it easy to coordinate work across distributed application components.

Create distributed asynchronous systems as workflows.

Supports both sequential and parallel processing.

Tracks the state of your workflow which you interact and update via API.

Best suited for human-enabled workflows like an order fulfilment system or for procedural requests.

AWS recommends that for new applications customers consider Step Functions instead of SWF.

SWF enables applications for a range of use cases, including media processing, web application back-ends, business process workflows, and analytics pipelines, to be designed as a coordination of tasks.

Registration is a one-time step that you perform for each different type of workflow and activity.

SWF has a completion time of up to 1 year for workflow executions.

SWF uses a task-oriented API.

SWF ensures a task is assigned once and never duplicated.

SWF keeps track of all the tasks and events in an application.

A domain is a logical container for application resources such as workflows, activities, and executions.

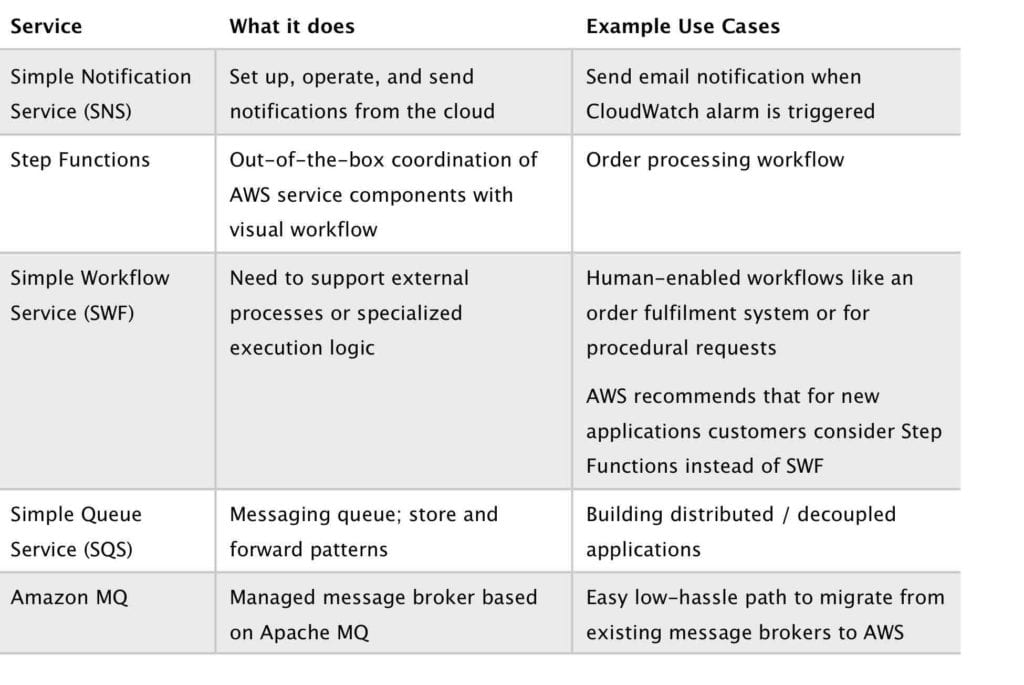
Workers are programs that interact with Amazon SWF to get tasks, process received tasks, and return the results.

The decider is a program that controls the coordination of tasks, i.e. their ordering, concurrency, and scheduling according to the application logic.

SWF applications include the following logical components:

* Domains.
* Workflows.
* Activities.
* Task Lists.
* Workers.
* Workflow Execution.

The following table describes related services and typical use cases for them:



# **Amazon MQ**

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

Amazon MQ supports industry-standard APIs and protocols so you can migrate messaging and applications without rewriting code.

Amazon MQ provides cost-efficient and flexible messaging capacity – you pay for broker instance and storage usage as you go.

Amazon MQ manages the administration and maintenance of ActiveMQ brokers and automatically provisions infrastructure for high availability.

With Amazon MQ, you can use the AWS Management Console, AWS CloudFormation, the Command Line Interface (CLI), or simple API calls to launch a production-ready message broker in minutes.

It’s a managed implementation of Apache ActiveMQ.

Fully managed and highly available within a region.

Amazon MQ stores your messages redundantly across multiple Availability Zones (AZs).

Active/standby brokers are designed for high availability. In the event of a failure of the broker, or even a full AZ outage, Amazon MQ automatically fails over to the standby broker so you can continue sending and receiving messages.

ActiveMQ API and support for JMS, NMS, MQTT, and WebSockets.

Designed as a drop-in replacement for on-premise message brokers.

Use SQS if you’re creating a new application from scratch.

Use MQ if you want an easy low-hassle path to migrate from existing message brokers to AWS.

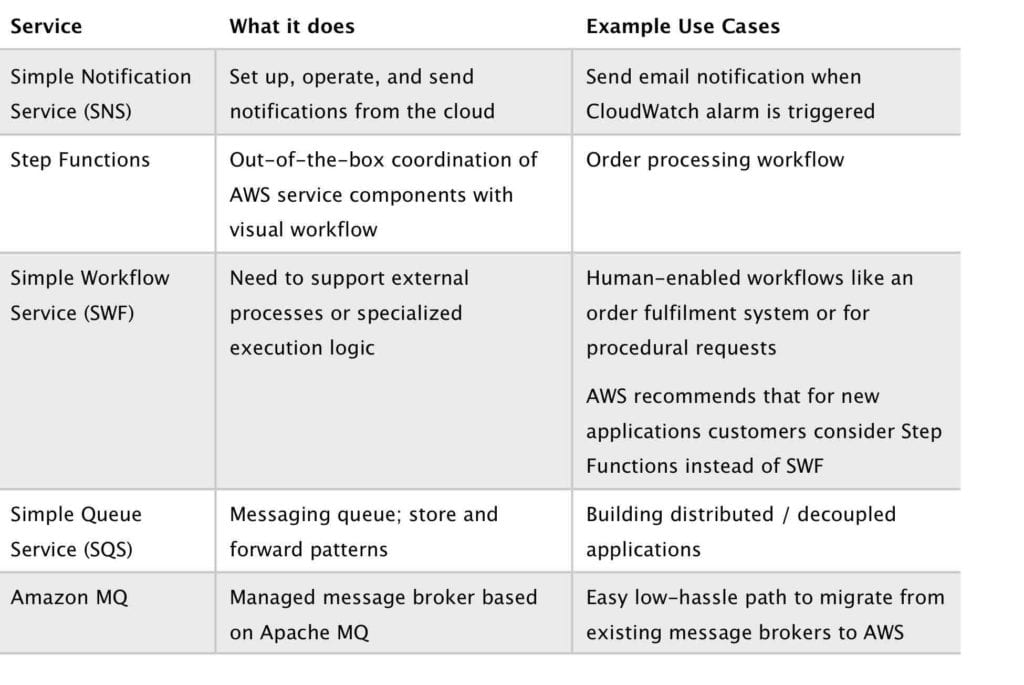
Amazon MQ provides encryption of your messages at rest and in transit.

It’s easy to ensure that your messages are securely stored in an encrypted format. Connections to the broker use SSL, and access can be restricted to a private endpoint within your Amazon VPC, which allows you to isolate your broker in your own virtual network.

You can configure security groups to control network access to your broker.

Amazon MQ is integrated with Amazon CloudWatch and AWS CloudTrail. With CloudWatch you can monitor metrics on your brokers, queues, and topics.

The following table describes related services and typical use cases for them:



# **AWS Step Functions**

AWS Step Functions makes it easy to coordinate the components of distributed applications as a series of steps in a visual workflow.

You can quickly build and run state machines to execute the steps of your application in a reliable and scalable fashion.

How it works:

1. Define the steps of your workflow in the JSON-based Amazon States Language. The visual console automatically graphs each step in the order of execution.
2. Start an execution to visualize and verify the steps of your application are operating as intended. The console highlights the real-time status of each step and provides a detailed history of every execution.
3. AWS Step Functions operates and scales the steps of your application and underlying compute for you to help ensure your application executes reliably under increasing demand.

Managed workflow and orchestration platform.

Scalable and highly available.

Define your app as a state machine.

Create tasks, sequential steps, parallel steps, branching paths or timers.

Amazon State Language declarative JSON.

Apps can interact and update the stream via Step Function API.

Visual interface describes flow and real-time status.

Detailed logs of each step execution.

Check out this AWS Hands-On Labs video in which we teach you what AWS Step Functions is, discuss design patterns, and create a simple Step Function to show you how it works:

Benefits and Features:

* **Built-in error handling –** AWS Step Functions tracks the state of each step, so you can automatically retry failed or timed-out tasks, catch specific errors, and recover gracefully, whether the task takes seconds or months to complete.
* **Automatic Scaling –**AWS Step Functions automatically scales the operations and underlying compute to run the steps of your application for you in response to changing workloads. Step Functions scales automatically to help ensure the performance of your application workflow remains consistently high as the frequency of requests increases.
* **Pay per use –** With AWS Step Functions, you pay only for the transition from one step of your application workflow to the next, called a state transition. Billing is metered by state transition, regardless of how long each state persists (up to one year).
* **Execution event history –** AWS Step Functions creates a detailed event log for every execution, so when things do go wrong, you can quickly identify not only where, but why. All of the execution history is available visually and programmatically to quickly troubleshoot and remediate failures.
* **High availability –** AWS Step Functions has built-in fault tolerance. Step Functions maintains service capacity across multiple Availability Zones in each region to help protect application workflows against individual machine or data center facility failures. There are no maintenance windows or scheduled downtimes.
* **Administrative security –** AWS Step Functions is integrated with AWS Identity and Access Management (IAM). IAM policies can be used to control access to the Step Functions APIs.

The following table describes related services and typical use cases for them:

