



AWS Decision guide

Amazon SQS, Amazon SNS, or EventBridge?



Amazon SQS, Amazon SNS, or EventBridge?: AWS Decision guide

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

- Decision guide 1**
 - Introduction 1
 - Details on the differences 3
 - Use 7
- Document history 10**

Amazon SQS, Amazon SNS, or Amazon EventBridge?

Understand the differences and pick the one that's right for you

Purpose	Understand the differences between Amazon SQS, Amazon SNS, and EventBridge and determine which service is the best fit for your needs.
Last updated	July 31, 2024
Covered services	<ul style="list-style-type: none">• Amazon Simple Queue Service• Amazon Simple Notification Service• Amazon EventBridge

Introduction

When building applications on AWS, you might want help in choosing the right service for handling messaging, event-driven architectures, and decoupling components. AWS offers three key services for these purposes—Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS), and Amazon EventBridge (formerly known as CloudWatch Events).

- Amazon SQS is a fully managed message queuing service that enables decoupling and scaling of microservices, distributed systems, and serverless applications.
- Amazon SNS is a highly available, durable, and secure pub/sub messaging service that allows decoupled applications to communicate with each other using a publish-subscribe model.
- Amazon EventBridge is a serverless event bus designed to make it easier to build event-driven architectures by allowing you to connect applications using data from various sources and route it to targets like AWS Lambda.

While all three services facilitate communication between decoupled components, they differ in their underlying architecture, use cases, and capabilities.

Here's a high-level view of the key differences between these services to get you started.

Category	Amazon SQS	Amazon SNS	EventBridge
Communication model	Pull-based (consumers poll messages from the queue)	Push-based (subscribers receive messages when published)	Pushed-based. Event-driven (rules match events and route to targets)
Persistence	Messages persist until consumed or expired	Messages do not persist; delivered in real-time to subscribers	Events do not persist; processed in real-time
Delivery guarantees	At-least-once delivery	At-least-once delivery for HTTP/S, exactly-once for Lambda and Amazon SQS	Exactly-once processing
Message ordering	FIFO (First-In-First-Out) queues ensure strict ordering	Amazon SNS FIFO topics guarantee order	No ordering guarantees
Message filtering	Amazon SQS can't decide consumer based on message. Use Amazon SNS message filtering with Amazon SQS to achieve.	Message filtering using subscription filter policies based on message metadata and for FIFO topics, message content	Complex event pattern matching and content-based filtering
Supported subscribers	Pull-based consumers (such as Amazon EC2 or Lambda)	HTTP/S endpoints, email, SMS, mobile push, Lambda, Amazon SQS	AWS services, Lambda, API destinations, event buses in other AWS accounts
Typical use cases	Decoupling microservices, buffering requests, processing tasks asynchronously	Fanout notifications, pub/sub messaging, mobile push notifications	Event-driven architectures, real-time stream processing

Category	Amazon SQS	Amazon SNS	EventBridge
			g, cross-account event sharing
Integration with other AWS services	Lambda, CloudWatch, AWS KMS, IAM	Lambda, Amazon SQS, Mobile Push, AWS KMS, IAM	Lambda, Step Functions, Amazon SQS, Amazon SNS, Kinesis, SageMaker, CloudWatch, IAM

Details on the differences

Explore differences between Amazon SQS, Amazon SNS, and EventBridge in eight key areas. These cover communication model, persistence, message ordering, filtering, integrations, use cases, scalability, and pricing.

Communication model

Amazon SQS

- Pull-based model where consumers actively poll messages from the queue, allowing for fine-grained control over message processing rates and independent scaling of consumers.

Amazon SNS

- Push-based model where subscribers receive messages in real-time as they are published, enabling immediate message delivery to multiple subscribers.

Amazon EventBridge

- Event-driven model where events are matched against predefined rules and routed to target services for processing, facilitating the building of reactive, event-driven architectures.

Persistence and delivery guarantees

Amazon SQS

- Messages are persisted in the queue until consumed or expired, ensuring no message loss. Provides at-least-once delivery, guaranteeing that every message is delivered at least once, with the possibility of duplicates.

Amazon SNS

- Messages are not persisted and are delivered to subscribers in real-time. Offers at-least-once delivery for HTTP/S subscribers and exactly-once delivery for Lambda and Amazon SQS subscribers.

Amazon EventBridge

- Events are not persisted and are processed in real-time. Ensures exactly-once processing, guaranteeing that each event is processed only once by the target services.

Message ordering

Amazon SQS and Amazon SNS

- Amazon SQS FIFO queues and Amazon SNS FIFO topics support First-In-First-Out guaranteed message ordering, making them suitable for scenarios requiring sequential processing. See [message ordering details for FIFO topics](#) for more information on message ordering using Amazon SNS and Amazon SQS.

Amazon EventBridge

- EventBridge does not provide message ordering guarantees, instead delivering events to targets in an arbitrary order.

Message filtering and routing

Amazon SQS

- Offers basic filtering capabilities through visibility timeouts (to prevent duplicate processing) and dead-letter queues (to handle failed messages).

Amazon SNS

- Provides message filtering using subscription filter policies, allowing subscribers to selectively receive messages based on message attributes.

Amazon EventBridge

- Supports advanced event pattern matching and content-based filtering, enabling fine-grained event processing and routing based on event content.

Supported subscribers and integrations

Amazon SQS

- Supports pull-based consumers, such as Amazon EC2 instances or Lambda functions, which actively retrieve messages from the queue.

Amazon SNS

- Supports a wide range of subscribers, including HTTP/S endpoints, email, SMS, mobile push notifications, Lambda functions, and Amazon SQS queues.

Amazon EventBridge

- Integrates with numerous AWS services, such as Lambda, Step Functions, Amazon SQS, Amazon SNS, Kinesis, and SageMaker, allowing event routing based on predefined rules. EventBridge also has numerous built-in integrations with third party products such as PagerDuty, DataDog, NewRelic. To see the complete list visit [Amazon EventBridge targets](#).

Typical use cases

Amazon SQS

- Commonly used for decoupling microservices, buffering requests, and processing tasks asynchronously, enabling independent scaling and graceful failure handling.

Amazon SNS

- Often used for fanout notifications, pub/sub messaging, and mobile push notifications, facilitating the broadcasting of messages to multiple subscribers simultaneously.

Amazon EventBridge

- Ideal for building event-driven architectures, real-time stream processing, and cross-account event sharing, allowing reactive systems to be built across multiple services.

Scalability and performance

Amazon SQS

- Highly scalable, automatically scaling based on the number of messages and consumers, providing high throughput for message processing.

Amazon SNS

- Highly scalable, capable of delivering messages to a large number of subscribers, with elastic scaling to handle increasing publish and subscribe demands.

Amazon EventBridge

- Automatically scales based on incoming event traffic, offering low-latency event processing and near real-time delivery to targets.

Pricing

Amazon SQS

- Pricing based on the number of API requests and data transferred, with a free tier including a monthly allowance of free API requests and data transfer.

Amazon SNS

- Pricing is based on the number of API requests, notifications delivered, and data transferred. Amazon SNS SMS messages are billed through AWS End User Messaging.

Amazon EventBridge

- Pricing based on the number of events published and target invocations, with a free tier that includes a monthly allowance of free events and invocations.

Use

Amazon SQS

- **Get started with Amazon SQS**

Get step-by-step instructions on setting up and using Amazon SQS. It covers topics such as creating a queue, sending and receiving messages, and configuring queue properties.

[Explore the guide](#)

- **Amazon SQS tutorial**

Walk through a practical example of using Amazon SQS to decouple the components of a simple application. It demonstrates how to create a queue, send messages to the queue, and process messages from the queue using AWS SDKs.

[Explore the tutorial](#)

- **Orchestrate Queue-based Microservices**

Learn how to design and run a serverless workflow that orchestrates a message queue-based microservice.

[Explore the tutorial](#)

- **Send Fanout Event Notifications**

Learn how to implement a fanout messaging scenario using Amazon SQS and Amazon SNS.

[Explore the tutorial](#)

Amazon SNS

- **Get started with Amazon SNS**

A step-by-step walkthrough of setting up and using Amazon SNS. It covers topics such as creating a topic, subscribing endpoints to a topic, publishing messages, and configuring access permissions.

[Explore the guide](#)

- **Filter Messages Published to Topics with Amazon SNS and Amazon SQS**

Learn how to use the message filtering feature of Amazon SNS

[Explore the tutorial](#)

- **Amazon SNS - Troubleshooting**

Learn how to view configuration information, monitor processes, and gather diagnostic data about Amazon SNS.

[Explore the course](#)

EventBridge

- **Amazon EventBridge User Guide**

This comprehensive documentation covers topics such as creating event buses, defining event rules, setting up targets, and integrating with various AWS services.

[Explore the guide](#)

- **Amazon EventBridge Tutorials**

The AWS documentation offers a series of tutorials that walk users through different use cases and scenarios using Amazon EventBridge. These tutorials cover topics like scheduling automated tasks, reacting to changes in AWS resources, and integrating with AWS services.

[Explore the tutorials](#)

- **AWS Serverless Workshops - Event-Driven Architecture**

Build event-driven architectures using Amazon EventBridge and other AWS serverless services. This workshop guides participants through the process of creating event buses, defining event rules, and triggering actions based on events.

[Explore the workshop](#)

- **AWS Online Tech Talk - Introduction to Amazon EventBridge**

Get an introduction to Amazon EventBridge, explaining key concepts, features, and use cases. The tech talk includes demonstrations and practical examples to help users understand how to leverage EventBridge in their applications.

[Watch the video](#)

- **Building Event-Driven Applications with Amazon EventBridge**

This blog post explores the process of building event-driven applications using Amazon EventBridge. It provides a step-by-step guide on creating event buses, defining event patterns, and setting up targets to process events.

[Read the blog post](#)

- **Create Point-to-Point Integrations Between Event Producers and Consumers with Amazon EventBridge Pipes**

Explore Amazon EventBridge Pipes, a feature of EventBridge that makes it easier for you to build event-driven applications by providing a consistent and cost-effective way to create point-to-point integrations between event producers and consumers, removing the need to write undifferentiated glue code.

[Read the blog post](#)

Document history

The following table describes the important changes to this decision guide. For notifications about updates to this guide, you can subscribe to an RSS feed.

Change	Description	Date
Initial publication	Guide first published.	July 31, 2024