**AWS SQS**

**Table of Content:**

1. SQS understanding
2. Hands on SQS
3. Case study
4. Pros and Cons
5. Cost Estimation

## What is Amazon SQS (Simple Queue Service)?

AWS SQS (**Amazon Simple Queue Service**) is a service which helps to align the message. Moreover, it also helps to enable the user to separate and scale microservices, distributed system, and serverless applications.

Amazon SQS makes it easy to manage the operating message-oriented middleware and enhances the developers to focus on their work. Amazon Simple Queue Service works at any volume without losing the message or requiring other services to be available.

It helps to send, store, and receive messages between software components. AWS SQS can start with the help of the tools such as Amazon Console, command line interface, and SDK.

**In Amazon SQS there are two types of queues which are:**

**Standard Queue :** Standard queue offers at least one delivery and maximum throughput.

**AWS SQS FIFO :** The FIFO queues guarantee that the processed message takes place only once in the first in first out basis

#### **Standard Queues**

Unlimited Throughput: Standard queues support a nearly unlimited number of transactions per second (TPS) per API action.

At-Least-Once Delivery: A message is delivered at least once, but occasionally more than one copy of a message is delivered.

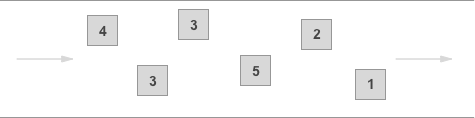
Best-Effort Ordering: Occasionally, messages might be delivered in an order different from which they were sent.

#### **FIFO Queues:**

High Throughput: By default, FIFO queues support up to 300 messages per second (300 send, receive, or delete operations per second). When you batch 10 messages per operation (maximum), FIFO queues can support up to 3,000 messages per second. If you require higher throughput, you can enable high throughput mode for FIFO on the Amazon SQS console, which will support up to 30,000 messages per second with batching, or up to 3,000 messages per second without batching.

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.

First-In-First-Out Delivery: The order in which messages are sent and received is strictly preserved (i.e. First-In-First-Out).



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.

sqs-what-is-sqs-fifo-queue-diagram

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.

**Case studies of AWS SQS:**

# **Environmental Monitoring Solutions (EMS) Case Study:**

AWS Services Used: IoT Device Management, Amazon ECS, & Amazon SQS

Petroleum retailers in Australia are improving the performance and safety of their service stations with an AWS IoT–enabled solution called Fuelsuite from EMS. EMS specializes in solutions that provide petrol retailers with performance data gathered from sensors located around petrol stations. Fuelsuite uses AWS IoT Device Management to control the edge devices collecting petrol station data, processes the data with Amazon Elastic Compute Cloud (EC2), and schedules messages to and from the edge devices by using Amazon SQS.

# **AWS Partner Story: NASA**

AWS Services Used: Amazon SNS & Amazon SQS

The NASA Image and Video Library provides easy access to more than 140,000 still images, audio recordings, and videos—documenting NASA’s more than half a century of achievements in exploring the vast unknown. The architecture includes Amazon SQS to decouple incoming jobs from pipeline processes and [Amazon Simple Notification Service (SNS)](https://aws.amazon.com/sns/) to trigger the processing pipeline when new content is updated.

# **BMW Case Study**

AWS Services Used: Amazon SQS, Amazon S3, Amazon DynamoDB, Amazon RDS, & AWS Elastic Beanstalk

The BMW Group is using AWS for its connected car application that collects sensor data from BMW 7 Series cars to give drivers dynamically updated map information. BMW built its new car-as-a-sensor (CARASSO) service in only six months leveraging Amazon SQS, Amazon Simple Storage Service (S3), Amazon DynamoDB, Amazon Relational Database Service (RDS), and AWS Elastic Beanstalk.

**CapitalOne:**

AWS Services Used: Amazon SQS

Capital One is modernizing their retail message queuing by migrating from self-managed message-oriented middleware systems to Amazon SQS. Capital One is using SQS to migrate several core banking applications to the cloud to ensure high availability and cost efficiency while simplifying administrative complexity and overhead.

# **redBus Case Study:**

AWS Services Used: Amazon SQS & Amazon SNS

redBus is expanding their AWS solution to include Amazon SQS and SNS for monitoring, alerts, and intercommunication. “Amazon SQS is an especially good solution for enabling messaging between external applications and our applications.” –

# **Pros of Amazon SQS:**

# **Pay for what you use**

You only get charged for the messages you read and write when using SQS. There are no recurring or base charges.

## ****Ease of setup****

Since SQS is a managed service, no infrastructure needs to be set up to start using SQS. To read and write messages, you may simply use the API, or use the SQS & Lambda integration.

## ****Options for Standard and FIFO queues****

You can choose between a regular queue and a FIFO queue out of the box when building an SQS queue. For various purposes, both of these queue forms may be useful.

## ****Scalability****

Size the SQS queues to the number of messages you write and read. You don’t need to scale the queues; AWS takes care of all of the scaling and performance-at-scale aspects.

## ****Automatic deduplication for FIFO queues****

When using queues, deduplication is necessary, and SQS will do the job of removing any duplicate messages for you for FIFO queues. This makes SQS FIFO queues ideal for tasks where it is necessary to have each task performed exactly once.

## ****A separate queue for unprocessed messages****

For debugging, this feature of SQS is useful. All messages that can not be processed are sent to a “dead-letter” queue where they can be checked. This queue has allowed all the normal integrations, so you can subscribe to it using, for example, an AWS Lambda event to send a notification when it is not possible to process an object

# **AWS SQS Limits:**

Amazon SQS is highly scalable, but there are some documented limits. We check your queues to see if they are approaching these limits:

1. Inflight Messages
2. Message Size
3. FIFO Queue Throughput

## Effects

Exceeding SQS limits can result in:

* Errors returned from the API
* Messages being lost without errors returned from the API

1. Inflight Messages

Inflight messages are messages in SQS that have been received by a consumer but not yet deleted. Each SQS queue is limited to 120,000 inflight messages, or 20,000 if it is a FIFO queue. When sending a message to a queue with too many inflight messages, SQS returns the "OverLimit" error message. To avoid hitting this limit, consider the following actions:

1. Delete messages as soon as they are processed so that they are no longer inflight.
2. Split up your messages between multiple queues. You can send messages to separate queues round-robin, and consume from multiple queues in code.
3. Message Size:

The *maximum message size* attribute on a queue controls the size of messages entering the queue. Our system detects when the average "SentMessageSize" metric is approaching this value. Since messages over the limit are rejected, it is actually impossible to hit the limit, so our recommendation is to leave at least 10% overhead when configuring your maximum message size.

To avoid SendMessage errors, ensure the maximum message size is larger than the messages you send. If your messages are larger than the SQS limit of 256 KB, you can use the [Amazon SQS Extended Library for Java](https://github.com/awslabs/amazon-sqs-java-extended-client-lib) to store messages in S3 and send references to them.

1. FIFO Queue Throughput

*FIFO queues* are a type of SQS queue that offer exactly-once processing, message deduplication, and in-order processing. The drawback is that FIFO queues can only support 300 send, receive, *or* delete operations per second. When using message batching at 10 messages, this effectively is increased to 3,000 operations per second. Our system will detect when your FIFO queues are near the 3,000 operations per second limit.

To get around this limit you can either use multiple queues to split up the workload, or request a limit increase from AWS support.

# **Amazon SQS pricing**

* Pay only for what you use
* No minimum fee

**Amazon SQS Free Tier\***

You can get started with Amazon SQS for free. All customers can make 1 million Amazon SQS requests for free each month. Some applications might be able to operate within this Free Tier limit.

**AWS SES**

1. SES understanding
2. Hands on SQS SES
3. Case study
4. Pros and Cons
5. Cost Estimation

## What is Amazon SES?

Amazon SES (Simple Email Service) is a service which sends an email regarding marketing, transaction, and notifications. It is suitable for small as well as large industries as the cost is less and it is reliable.

Amazon SES can be directly integrated to the existing application with the help of SMTP Interface and Amazon SDK. Email sending capabilities can also be introduced in Amazon SES such as ticketing system and email clients.

Building a large-scale email answer is a complex and expensive challenge for a business: you’ve got to make your infrastructure, assemble your network, warm up your IP addresses and shield your sender name. Several third-party email solutions need contract negotiations and important up-front prices.

Amazon Simple Email Service eliminates these challenges, permitting you to start out causation email in minutes. You have the benefit of the years of expertise and therefore the refined email infrastructure Amazon.com engineered to serve its own large-scale client base.

With Amazon SES, there are not any direct prices and no minimum commitments. You pay as you go, and you pay just for what you utilize.

If you send email using an application hosted in an **Amazon EC2 instance**, the primary 62,000 emails you send area unit free, with an awfully low rate for every email sent thereafter.

# **Amazon SES customers**

## Seetickets

Seetickets retails and distributes tickets for music, festival, theatre, sport, comedy, exhibition and lifestyle events, working with more than 5,000 clients globally.

Our legacy system for marketing campaigns and customer relationship management was complicated to use and expensive to manage. When evaluating new options, Amazon Simple Email Service stood out to us for its ease of use, reliability, and cost. We were able to integrate it into our new owned-and-operated CRM solution by ourselves, and using it has been very intuitive. In 2021, we have been able to send over 250 million emails, with a 99.8% delivery rate. Amazon SES has helped us build a cost-efficient CRM solution for our own B2C business and for our B2B clients in Europe, and we are looking forward to deploying it globally in 2022.

**Reddit**

Reddit is a network of communities where individuals can find experiences built around their interests, hobbies, and passions. With over 430 million global users monthly, Reddit is home to the most open and authentic conversations on the internet.

At Reddit, we want to help our users stay up-to-date with content and discussions that interest them the most. As we continue to evolve our engagement strategy, we needed an email provider that was flexible and could scale to support our millions of global users. We started using Amazon SES a few years ago to meet those needs and haven’t looked back.

**Netflix**

Netflix is the world's leading internet television network, with more than 150 million members in 190 countries. The Netflix Messaging team builds the systems and applications to communicate with members across various channels and relies on Amazon’s SES for email delivery.

Before we migrated to Amazon Simple Email Service (SES), Netflix had to maintain an in-house solution for sending emails. This in-house solution carried its own operational overheads, including running dedicated servers with email delivery software and optimizing email sending practices for each Internet Service Provider. We evaluated several email delivery solutions and decided on Amazon SES because it is flexible, affordable, highly scalable, has global reach, and promises excellent deliverability. Through a combination of the SES Dedicated IP offering and some strategies implemented on our end, after migrating to Amazon SES, we were able to streamline operations, improve Inbox Placement and increase our email sender reputation score.

**Academia**

Academia.edu is a platform for academics to share research papers. The company's mission is to accelerate the world's research. Academics use Academia.edu to share their research, monitor deep analytics around the impact of their research, and track the research of academics they follow. Over 115 million academics have signed up to Academia.edu, adding 24 million papers. Academia.edu attracts over 70 million unique visitors a month.

Academia is a network for academics to connect, collaborate on work or find relevant insights in a single place, like citations or grant opportunities. This platform is only as powerful as the individuals who engage in it, so email outreach and reminders are a critical piece of the peer review and research process. We needed an email solution that could dependably enable us to communicate to tens of millions of researchers every day. Amazon SES was a natural fit.”

**Iterable**

Iterable is empowering the world’s leading brands to achieve world-class customer engagement at scale. Powered by the same technologies that propelled leaders like Twitter and Facebook, we are building tools for growth marketers wishing to captivate customers with messaging that matters when it matters.

"Here at Iterable, we are a customer engagement platform, which means we help businesses connect with their customers over various channels, like email. Email continues to be one of the most effective and preferred channels for communicating with consumers and businesses. Amazon SES has become an integral infrastructure partner to Iterable. The speed, reliability, and deliverability of Amazon SES has enabled our clients to spin up email marketing with lightening speed. In addition to supporting increased sending volumes, Amazon SES helps ensure stellar inbox deliverability across the board. We couldn't be happier with the performance of Amazon SES

## Benefits of AWS Simple Email Service

### **a. High Deliverability**

Content filtering technologies, dedicated IP addresses, and a name dashboard facilitate defend and enhance your sender name. Maintaining a decent reputation ensures that your messages reach your customers’ inboxes.

### **b. Cost-Effective**

Pay as you go, and pay just for what you utilize. There aren’t any upfront fees, any long pricing negotiations, any mounting expenses, and no minimum charges. And, if you send from an associate application hosted in Amazon EC2, the primary 62,000 emails you send each month are free.

### **c. Configurable**

Use configuration sets to make rules that are applied to the emails you send using Amazon SES. Send email metrics to Amazon CloudWatch for more analysis, or receive notifications through **Amazon SNS**.

### **d. Reliable**

Amazon SES runs on the extremely reliable Amazon Internet Services Infrastructure. Multiple data centres and redundant systems make sure the highest levels of availableness.

### **Limitations of Amazon SES**

### **Extremely Technical**

If there’s one thing you’ve noticed from this review (and the numerous terms I’ve had to explain), it’s that Amazon SES is super-technical.

To put it simply, if you don’t have a developer’s knowledge and skill level, configuring and setting up Amazon SES can be really difficult.

Whether you are using its SMTP Interface or API, or even going deeper into configuring specific features like DKIM (DomainKeys Identified Mail), the process is highly complicated and technical.

### **Limited SMTP Functionality**

While Amazon SES offers a powerful SMTP server, it might not be as powerful as most other dedicated SMTP providers.

Why?

Unlike other SMTP providers that allow you to send an **unlimited** number of emails, Amazon SES still restricts the number of daily emails you can send.

Sure, it’s more than what Gmail can offer you, but for large email marketing campaigns that involve more than 10,000 emails a day, Amazon SES might not be able to help you.

Additionally, because of separate AWS regions, you’ll need different SMTP credentials for customer bases that are geographically dispersed. This limits your reach when you’re trying to send a consolidated, global email campaign.

# **Amazon SES pricing:**

Amazon Simple Email Service (SES) is a pay-as-you-go service based on the volume of emails sent and received. There are no subscriptions, no contract negotiations, and no minimum charges.

### **AWS Free Usage Tier**

When you call Amazon SES from an application hosted in Amazon EC2 or via AWS Lambda, you can send 62,000 messages per month at no charge. This Free Usage Tier benefit does not expire.