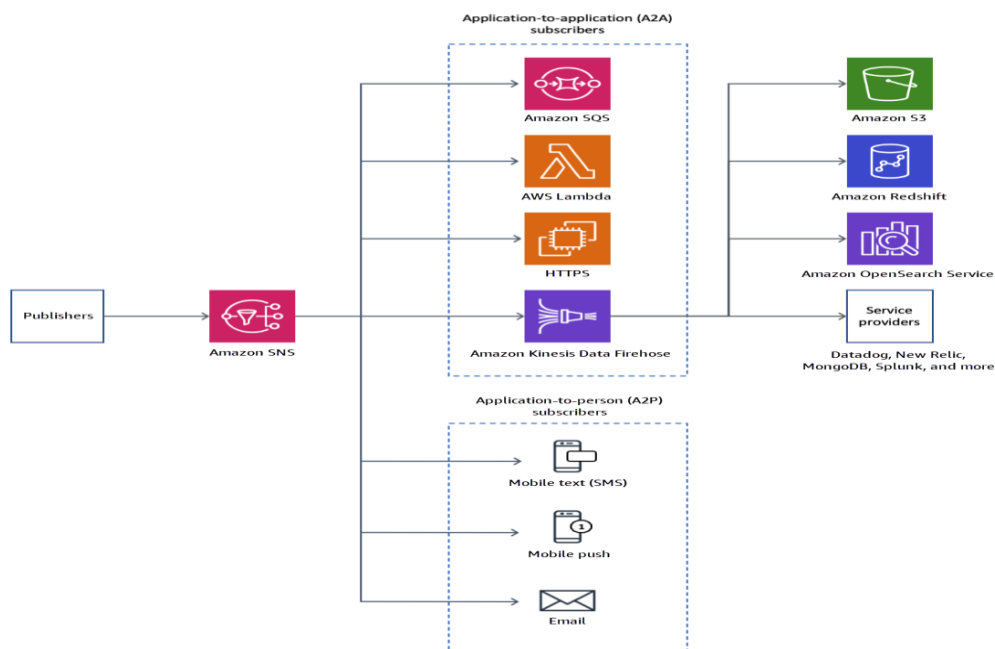


APPLICATION SERVICES

1. What is SNS

Amazon Simple Notification Service (Amazon SNS) is a managed service that provides message delivery from publishers to subscribers (also known as producers and consumers). Publishers communicate asynchronously with subscribers by sending messages to a topic, which is a logical access point and communication channel. Clients can subscribe to the SNS topic and receive published messages using a supported endpoint type, such as Amazon Data Firehose, Amazon SQS, AWS Lambda, HTTP, email, mobile push notifications, and mobile text messages (SMS).



2. Create a topic

Prerequisites

Step 1: Create a topic

Step 2: Create a subscription to the topic

Step 3: Publish a message to the topic

Step 4: Delete the subscription and topic

Next steps

3.Create subscription

To subscribe an email address to an Amazon SNS topic using the AWS Management Console. Sign in to the Amazon SNS console . In the left navigation pane, choose Subscriptions. On the Subscriptions page, choose Create subscription.

4.Subscribe to the subscription

To subscribe an endpoint to an Amazon SNS topic

1. Sign in to the [Amazon SNS console](#).
2. In the left navigation pane, choose **Subscriptions**.
3. On the **Subscriptions** page, choose **Create subscription**.
4. On the **Create subscription** page, in the **Details** section, do the following:
 - a. For **Topic ARN**, choose the Amazon Resource Name (ARN) of a topic. This value is the AWS ARN that was generated when you created the Amazon SNS topic, for example `arn:aws:sns:us-east-2:123456789012:your_topic`.
 - b. For **Protocol**, choose an endpoint type. The available endpoint types are:
 - [HTTP/HTTPS](#)
 - [Email/Email-JSON](#)
 - [Amazon Data Firehose](#)
 - [Amazon SQS](#)
 - [Platform application endpoint](#)
 - [SMS](#)
 - c. For **Endpoint**, enter the endpoint value, such as an email address or the ARN of an Amazon SQS queue.
 - d. Firehose endpoints only: For **Subscription role ARN**, specify the ARN of the IAM role that you created for writing to Firehose delivery streams. For more information, see [Prerequisites for subscribing Firehose delivery streams to Amazon SNS topics](#).
 - e. (Optional) For Firehose, Amazon SQS, HTTP/S endpoints, you can also enable raw message delivery. For more information, see [Amazon SNS raw message delivery](#).
 - f. (Optional) To configure a filter policy, expand the **Subscription filter policy** section. For more information, see [Amazon SNS subscription filter policies](#).
 - g. (Optional) To enable payload-based filtering, configure Filter Policy Scope to MessageBody. For more information, see [Amazon SNS subscription filter policy scope](#).

- h. (Optional) To configure a dead-letter queue for the subscription, expand the **Redrive policy (dead-letter queue)** section. For more information, see [Amazon SNS dead-letter queues \(DLQs\)](#).
- i. Choose **Create subscription**.

5.SQS

SQS stands for Simple Queue Service, which is a fully managed message queuing service provided by Amazon Web Services (AWS). It allows you to decouple and scale microservices, distributed systems, and serverless applications. With SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available. It's particularly useful for building systems that need to communicate asynchronously and reliably, ensuring that messages are processed exactly once and in the right order

6.SES

SES stands for Amazon Simple Email Service. It's a cloud-based email sending service provided by Amazon Web Services (AWS) that allows developers to send transactional, marketing, and notification emails to their customers. SES simplifies the process of sending emails without having to maintain email servers. It provides features such as email sending, receiving, and email list management, as well as monitoring and analytics to track email sending metrics. SES is commonly used by businesses to send automated emails such as purchase confirmations, password resets, newsletters, and promotional campaigns.

7. Lambda and Elastic Beanstalk

Lambda and Elastic Beanstalk are both services provided by Amazon Web Services (AWS) for deploying and managing applications, but they serve different purposes and have different characteristics.

AWS Lambda:

AWS Lambda is a serverless compute service that allows you to run code without provisioning or managing servers.

With Lambda, you upload your code and AWS automatically scales and manages the infrastructure required to run it.

Lambda functions are event-driven and can be triggered by various AWS services, such as S3, API Gateway, SNS, DynamoDB, etc.

You only pay for the compute time consumed by your function, measured in milliseconds.

Lambda is well-suited for microservices, event-driven architectures, and small, focused functions.

AWS Elastic Beanstalk:

AWS Elastic Beanstalk is a platform as a service (PaaS) offering that simplifies the deployment and management of applications in the AWS cloud.

Elastic Beanstalk automatically handles the provisioning and configuration of the underlying infrastructure (e.g., EC2 instances, load balancers, auto-scaling groups) based on the application's requirements.

It supports various programming languages and frameworks, such as Java, Python, Ruby, Node.js, PHP, .NET, and Docker containers.

Elastic Beanstalk provides flexibility and control over the underlying infrastructure, allowing you to customize configurations and manage resources manually if needed.

You pay for the AWS resources (e.g., EC2 instances, load balancers) used by your Elastic Beanstalk environment.

In summary, Lambda is a serverless compute service ideal for event-driven, small-scale functions, while Elastic Beanstalk is a PaaS offering suitable for deploying and managing more complex applications with control over the underlying infrastructure. Depending on your application requirements and architecture, you may choose one or the other, or even use them together in a hybrid deployment.