AWS SIMPLE NOTIFICATION SERVICE

The Tech Stuff

OVERVIEW

1

INTRODUCTION

INTRODUCTION

A fully managed messaging service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SNS makes it easy to set up, operate, and send notifications from the cloud, supporting a variety of messaging patterns and delivery protocols.

KEY FEATURES

KEY FEATURES

Topic-Based Publish/Subscribe

Publishers send messages to topics, and subscribers receive messages from those topics.
Supports multiple subscriber types, including AWS Lambda, SQS, HTTP/S endpoints, and email addresses.

Multiple Protocols

Supports multiple delivery protocols such as HTTP/S, email, SMS, Lambda functions, and SQS.
Flexibility in how notifications are delivered to end-users or applications.

KEY FEATURES

Message Filtering

Allows subscribers to receive only the messages of interest based on message attributes.
Reduces the volume of unwanted messages and helps in efficient processing.

Fanout

Messages sent to an SNS topic can be delivered to multiple endpoints simultaneously.
Facilitates parallel processing of messages by multiple subscribers.

KEY FEATURES

Mobile Push Notifications

Supports sending push notifications to mobile devices via services like Amazon Device Messaging (ADM), Apple Push Notification Service (APNs), Google Cloud Messaging (GCM), and Baidu Cloud Push.

Security

Integrates with AWS Identity and Access Management (IAM) for fine-grained access control.
Supports encryption of messages at rest and in transit.

HOW IT WORKS

HOW IT WORKS

- **Publisher**

  Applications or services that send messages to an SNS topic.

- **Topic**

  A logical access point and communication channel for publishers and subscribers.

- **Subscriber**

  Applications, services, or endpoints that receive messages from an SNS topic.

  Subscribers can be other AWS services like Lambda functions, SQS queues, or external services like HTTP/S endpoints.

USE CASES

USE CASES

- **Event-Driven Architectures:**

  Decouples components in an event-driven system, allowing them to react to events asynchronously.

- **Application Alerts and Notifications**

  Sends alerts and notifications to users or administrators via email, SMS, or mobile push notifications.

USE CASES

- **Microservices Communication**

  Facilitates communication between microservices by sending messages to topics that microservices subscribe to.

- **Message Fanout**

  Distributes messages to multiple systems for parallel processing, such as sending the same message to a Lambda function and an SQS queue.

5

BENEFITS

BENEFITS

- **Scalability**

  Automatically scales to handle a large number of messages and subscribers.
  Handles high throughput with low latency.

- **Reliability**

  Delivers messages reliably across multiple regions.
  Ensures high availability and fault tolerance.

BENEFITS

- **Flexibility**

  Supports various messaging patterns and protocols, making it suitable for a wide range of applications.
  Allows message filtering for more targeted delivery.

- **Ease of Use**

  Simple APIs and management console for setting up and managing topics and subscriptions.
  Easily integrates with other AWS services and third-party applications.

BENEFITS

- **Cost-Effective**

  Pay-as-you-go pricing model with no upfront costs or minimum fees. Charges based on the number of messages published and delivered.

THANK YOU

Any Questions?

The Tech Stuff