

UNIVERSITY of WASHINGTON

AWS SNS

Team 12

Chhavi Gupta
RamaSoumya Naraparaju
Sathwika Suddala



Introduction

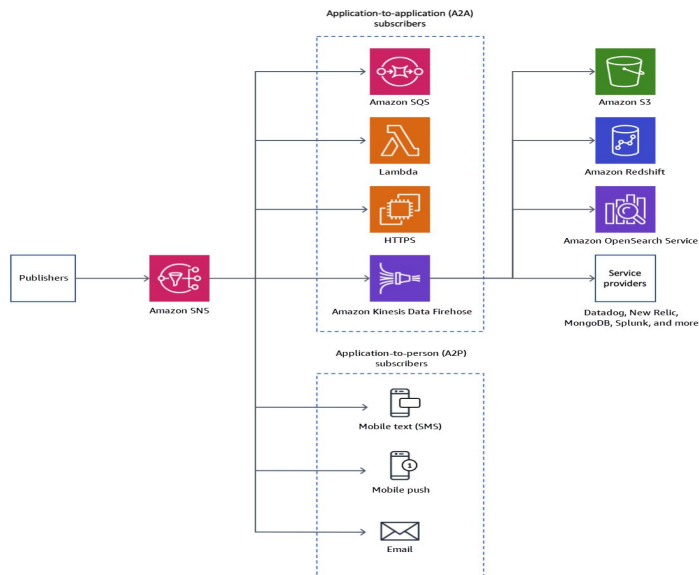
- Simple Notification Service is used for publishing messages to Consumers or applications
- SQS is a PubSub model where the publisher communicates with consumer endpoints like Email, HTTP/HTTPS, SMS, Push Notification etc. asynchronously by sending messages to SNS topic
- One to Many relation model where many consumers can subscribe to one topic. Topic owner creates the policies which defines who can subscribe to the topic.
- Limit: 100K topics per account and 12.5 million Consumers per topic
- Fully managed and scalable service by Amazon
- Pricing is based on the number of calls made to publish a message, size of the message and number of deliveries to various endpoints
- Benefits – Instantaneous, Flexible, Cheap, Easy to Use.



Introduction

Two types of subscriber models:

- *Application to Application*
- *Application to Consumer*



W

History

- **Who invented the Technology?**
 - Amazon announced the service on April 7th 2010. There are no details of the Individuals that worked on this technology it is listed a proprietary service of Amazon
- **Why was the technology invented?**
 - The service was primarily developed for enabling users to build highly reliable, event-driven workflows and messaging applications without needing complex 3rd party middleware and application management tools.
- **Competing Alternatives?**
 - Apache Kafka, IBM MQ, Google Cloud Pub/Sub, TIBCO, Rabbit MQ, One Signal

W

Features

- **Different Topic Types**
 - Standard Topic – Applications can process duplicate and out of order messages
 - FIFO Topic - when the order of messages is critical, and duplicates can't be tolerated
- **Message Publishing and Batching**
 - Service allows publisher to batch messages while sending it to a topic for reducing costs
- **Message Filtering**
 - Subscriber can create filter policies to receive only the notifications that they are interested in
- **Message Fanout**
 - Replicates and delivers the published message to multiple consumers

W

Features

- **Message Encryption**
 - When published to encrypted topic encryption on the server, using a 256-bit AES-GCM algorithm and a Customer Master Key (CMK) issued with AWS Key Management Service (KMS)
- **Message Privacy**
 - Use VPC endpoints to publish messages privately to SNS topics
- **Message Data Protection**
 - Topic owners can define data protection policies that can discover and protect sensitive data
- **Message Durability**
 - All messages sent to SNS stored redundantly across multiple Availability Zones

W

Features

- **Message Archiving and Analytics**
 - direct connection to Amazon Kinesis Data Firehose enables message storage in services like S3, Redshift, MongoDB, Datadog, Splunk etc.
- **SMS Text Messaging**
 - 200+ countries, create and manage sender ID
- **Mobile Push Notification**
 - To iOS, Android, Fire, Windows, and Baidu devices
- **Email Notification**
- **Subscription Policy**
 - Topic owner can define the subscription policies which will decide who can subscribe to the topic.

W

Example Use Cases

- **Concert Ticket Sale Notification to Registered users**
 - As soon as tickets for an event are on sale, a publisher which can be a ticket management system will send a message to the SNS standard topic. All the registered user's phone numbers will be added as consumers to the topic. As soon as the topic receives the message from the publisher it will instantaneously create a copy of the message and send it to all of the subscribers.
- **Order Ship Notification**
 - Order Shipment API will publish a message to Shipment Topic as soon as any customer order shipped. Topic will send the update instantaneously to other applications like order management for updating order status, billing for crediting amount etc.

W

Technology Advantages

> Reliable and Simple

- > Simple APIs that can be easily integrated with other AWS services
- > SNS works across AWS regions and stores messages in cross availability zones

> Secure

- > AWS SNS has different security policies such as MFA, that are enforced to maintain and protect the data and infrastructure. Additional features such as using a VPC (Virtual Private Cloud) are also offered by AWS.

> Easy-to-Use

- > AWS CLI, and the SNS console helps the user to create topics, subscriptions, send and receive messages, and monitor events and logs.

W

Technology Advantages

> Scalable

- > Amazon SNS works for high throughput and traffic patterns differences. It also helps users to plan workload with capacity planning and provisioning.

> Flexible

- > Message delivery works across multiple transport protocols, also multiple notification formats are supported, such as mobile push notifications, HTTPS endpoints, email addresses, and SMS messages.

> Fully Managed

- > The service takes care of scalability and also the management of the function, taking the burden of management off the development team.

> Pricing

- > The price is cost effective compared to other vendors but only for use cases suitable for SNS.

W

Technology Disadvantages

> Troubleshooting

- > The process of working on bugs and debugging the errors has been reported to be complicated by the users

> Lack of control with the user (Fully Managed)

- > Being a fully managed service, it does not allow the users/developers much freedom on how to control the service.

> Dashboard unavailability

- > The Dashboard of SNS only allows the users to change a minimum amount of settings and displays minimal information of the service.

W

Technology Disadvantages

> High cost depending on situation

- > Service cost can become expensive if the number of topics, subscribers, or messages increases.

> Difficult to format email and SMS notifications

- > SNS is designed to distribute notifications, but it does not support much on formatting the email and SMS formats.

W

Cost of SNS - Notification Deliveries

Endpoint Type	Free Tier	Price
Mobile Push Notifications	1 million notifications	\$0.50 per million notifications
Email/Email-JSON	1,000 notifications	\$2.00 per 100,000 notifications
HTTP/s	100,000 notifications	\$0.60 per million notifications

W

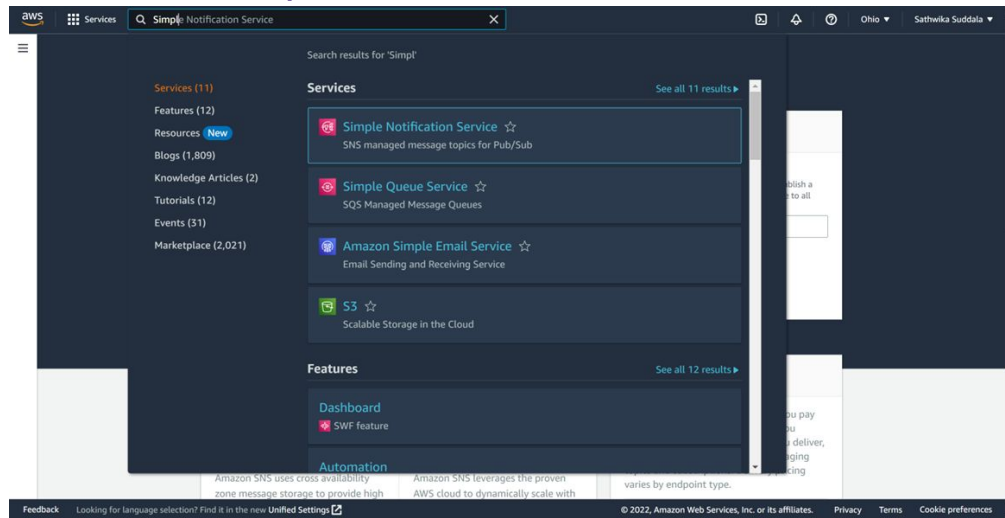
Cost of SNS - Data Transfer

Data Transfer IN / OUT	Pricing
All data transfer in	\$0.00 per GB
Data Transfer OUT	
First 9.999 TB / Month	\$0.09 per GB
Next 40 TB / Month	\$0.085 per GB
Next 100 TB / Month	\$0.07 per GB
Greater than 150 TB / Month	\$0.05 per GB

W

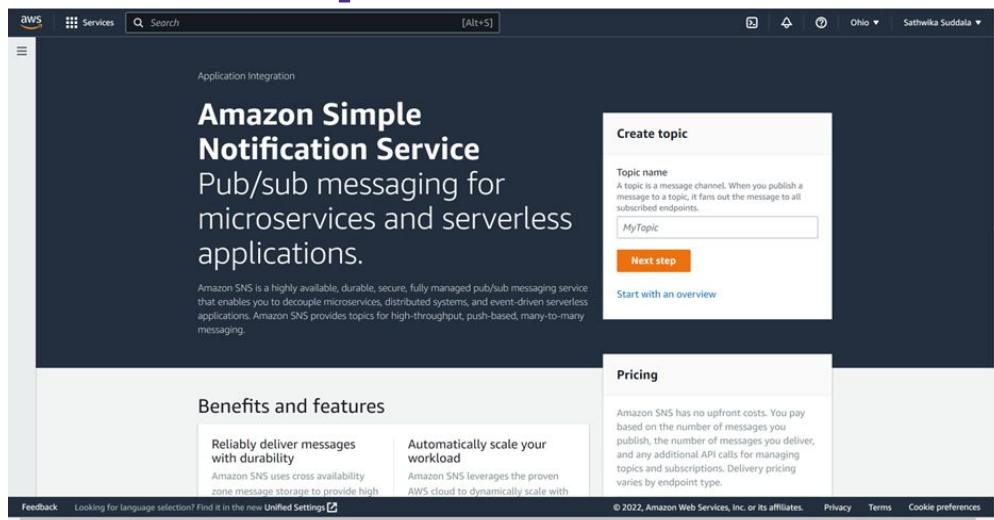
DEMO

Search for Simple Notification Service in AWS console



W

Create a topic



W

Create a topic

The screenshot shows the 'Create topic' page in the AWS console. The breadcrumb navigation is 'Amazon SNS > Topics > Create topic'. The page title is 'Create topic'. Under the 'Details' section, there is a 'Type' dropdown set to 'Info'. Below it, a note states 'Topic type cannot be modified after topic is created'. There are two radio button options: 'FIFO (first-in, first-out)' and 'Standard'. The 'Standard' option is selected. Below the radio buttons, there are two text input fields. The first is labeled 'Name' and contains the text 'sample'. Below it, a note says 'Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).' The second field is labeled 'Display name - optional' and contains the text 'sample'. Below it, a note says 'Maximum 100 characters.' At the bottom of the form, there is a section for 'Encryption - optional' which is currently collapsed. The footer of the console shows 'Feedback', a link to 'Looking for language selection? Find it in the new Unified Settings', the copyright '© 2022, Amazon Web Services, Inc. or its affiliates.', and links for 'Privacy', 'Terms', and 'Cookie preferences'.

Amazon SNS > Topics > Create topic

Create topic

Details

Type [Info](#)
Topic type cannot be modified after topic is created

☐ FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

☒ Standard

- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

sample

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - optional

To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message. [Info](#)

sample

Maximum 100 characters.

► Encryption - optional

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

W

Create Subscription

The screenshot shows the 'Create subscription' page in the AWS console. The breadcrumb navigation is 'Amazon SNS > Subscriptions > Create subscription'. The page title is 'Create subscription'. Under the 'Details' section, there is a 'Topic ARN' text input field containing 'arn:aws:sns:us-east-2:248248909336:sample'. Below it is a 'Protocol' dropdown menu set to 'Email'. The 'Endpoint' text input field contains 'sathwikasuddala4@gmail.com'. Below the endpoint field, there is a blue box with a note: 'After your subscription is created, you must confirm it.' Below this, there are two sections: 'Subscription filter policy - optional' and 'Redrive policy (dead-letter queue) - optional', both of which are currently collapsed. At the bottom of the form, there are 'Cancel' and 'Create subscription' buttons. The footer of the console shows 'Feedback', a link to 'Looking for language selection? Find it in the new Unified Settings', the copyright '© 2022, Amazon Web Services, Inc.', and links for 'Privacy', 'Terms', and 'Cookie preferences'.

Amazon SNS > Subscriptions > Create subscription

Create subscription

Details

Topic ARN

arn:aws:sns:us-east-2:248248909336:sample

Protocol

The type of endpoint to subscribe

Email

Endpoint

An email address that can receive notifications from Amazon SNS.

sathwikasuddala4@gmail.com

After your subscription is created, you must confirm it.

► Subscription filter policy - optional

This policy filters the messages that a subscriber receives.

► Redrive policy (dead-letter queue) - optional

Send undeliverable messages to a dead-letter queue.

Cancel Create subscription

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc.

aws

Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's ID is:
arn:aws:sns:us-east-2:248248909336:sample:8f89a7ed-84b8-4ce5-b4b0-8ec79c9612a3

If it was not your intention to subscribe, [click here to unsubscribe](#).

W

Publish a message

The screenshot shows the Amazon SNS console interface for publishing a message to a topic. The breadcrumb navigation is 'Amazon SNS > Topics > sample > Publish message'. The main heading is 'Publish message to topic'. The form is divided into two main sections: 'Message details' and 'Message body'.

Message details

- Topic ARN: `arn:aws:sns:us-east-2:248248909336:sample`
- Subject - optional:
- Time to Live (TTL) - optional:

Message body

Message structure

- ☐ Identical payload for all delivery protocols. The same payload is sent to endpoints subscribed to the topic, regardless of their delivery protocol.
- ☒ Custom payload for each delivery protocol. Different payloads are sent to endpoints subscribed to the topic, based on their delivery protocol.

Message body to send to the endpoint

The message body must be a JSON object with an attribute for each delivery protocol.

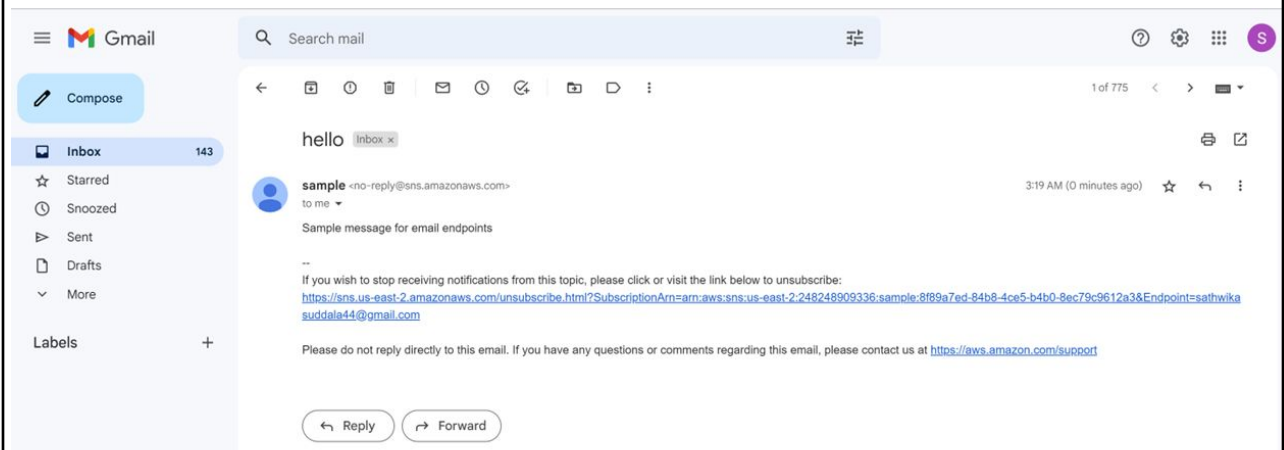
`{ "sns": "Sample message for SNS endpoints", }`

Feedback: Looking for language selection? Find it in the new Unified Settings.

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

W

Publish a message



W

Conclusion

Amazon SNS is a pub/sub service fully managed that is used for message delivery from application to application or application to person.

It is reliable, and can be integrated with AWS services.

W