

AMAZON SNS (SIMPLE NOTIFICATION SERVICE)

Wednesday, 18 August 2021 18.48

Amazon Simple Notification Service

Fully managed pub/sub messaging, SMS, email, and mobile push notifications

From <<https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>>

Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication.

TURKISH

Amazon Simple Notification Service (Amazon SNS), hem uygulamadan uygulamaya (A2A) hem de uygulamadan kişiye (A2P) iletişim için tam olarak yönetilen bir mesajlaşma hizmetidir.

From <<https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>>

The A2A pub/sub functionality provides topics for high-throughput, push-based, many-to-many messaging between distributed systems, microservices, and event-driven serverless applications. A2A yayın/alt işlevi, dağıtılmış sistemler, mikro hizmetler ve olaya dayalı sunucusuz uygulamalar arasında yüksek verimli, anında iletme tabanlı, çoktan çoğa mesajlaşma için konular sağlar

From <<https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>>

Using Amazon SNS topics, your publisher systems can fanout messages to a large number of subscriber systems including Amazon SQS queues, AWS Lambda functions and HTTPS endpoints, for parallel processing, and Amazon Kinesis Data Firehose. The A2P functionality enables you to send messages to users at scale via SMS, mobile push, and email.

From <<https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>>

1 million requests free
with the [AWS Free Tier](#)
[Get started for free »](#)

From <<https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>>

Benefits

Modernize and decouple your applications

Amazon SNS enables you to modernize your applications and decouple them into smaller, independent components that are easier to develop, deploy, and maintain. Leveraging a pub/sub [event-driven architecture](#) for your application improves performance, reliability, and allows each component to scale independently.

Send messages directly to millions of users

Amazon SNS enables you to send messages or notifications directly to users with SMS text messages to over 200 countries, mobile push on Apple, Android, and other platforms or email (SMTP). Amazon SNS provides redundancy across multiple SMS providers and allows you to push mobile notifications using a single endpoint for all platforms.

Reliably deliver messages

Amazon SNS uses a number of strategies that work together to provide message durability. To start, published messages are stored across multiple, geographically-separated servers and data centers. If a subscribed endpoint isn't available, Amazon SNS executes a [message delivery retry policy](#). To preserve any messages that aren't delivered before the delivery retry policy ends, you can create a [dead-letter queue](#). You can also subscribe [Amazon Kinesis Data Firehose delivery streams to SNS topics](#), which allows messages to be sent to durable endpoints such as Amazon S3 buckets or Amazon Redshift tables.

Automatically scale your workload

Amazon SNS leverages the proven AWS cloud to dynamically scale with your application. Amazon SNS is a fully managed service, taking care of the heavy lifting related to capacity planning, provisioning, monitoring, and patching. The service is designed to handle high-throughput, bursty traffic patterns and enables you to send millions of messages per second.

Ensure accuracy with message ordering and deduplication

[Amazon SNS FIFO topics](#) work with Amazon SQS FIFO queues to ensure messages are delivered in a strictly ordered manner and are only processed once (deduplicated). This enables you to maintain consistency when processing transactions across a single or multiple independent services where it's critical that messages are in the correct order. It also allows you to offload the effort of writing custom code for ordering and message deduplication.

Simplify your architecture with Message Filtering

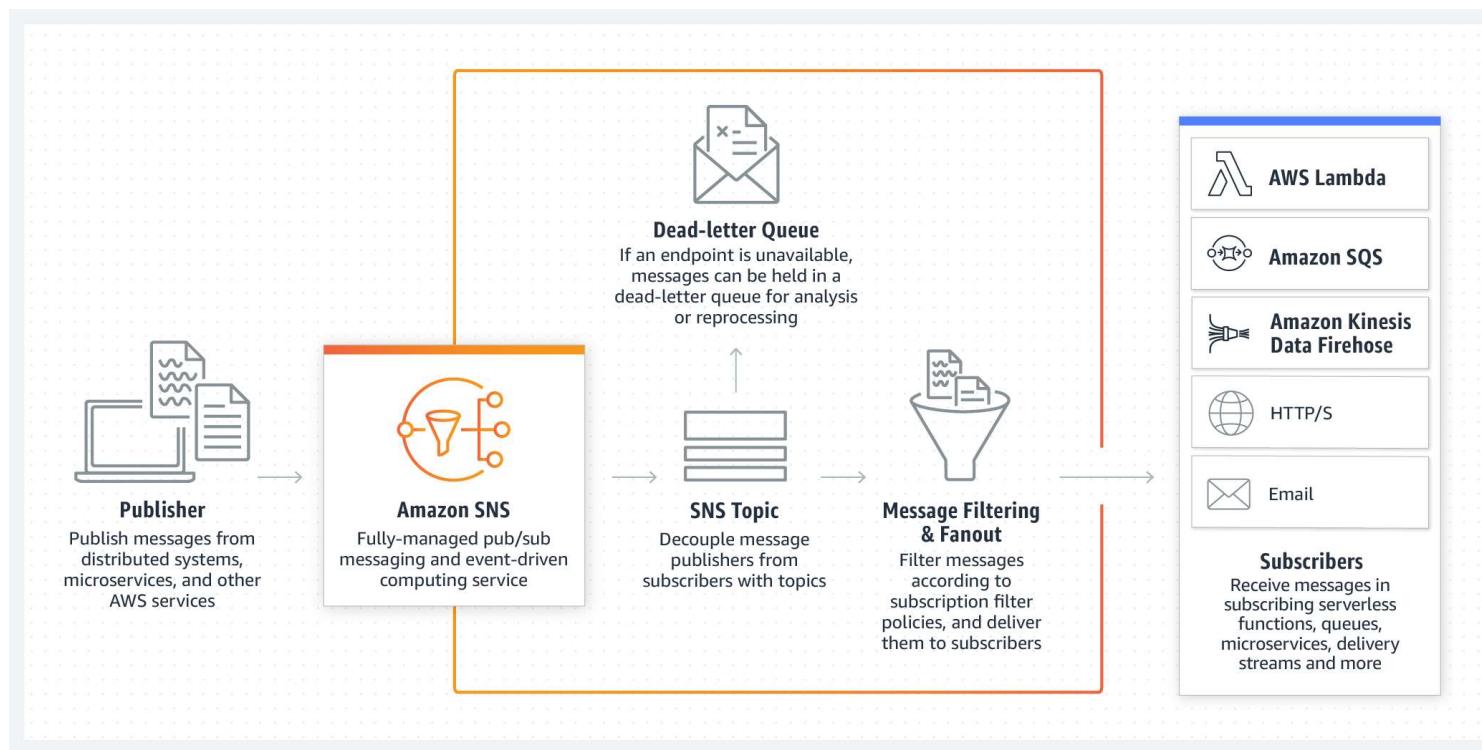
Amazon SNS helps you simplify your pub/sub messaging architecture by offloading the message filtering logic from your subscriber systems, and message routing logic from your publisher systems. With Amazon SNS message filtering, subscribing endpoints receive only the messages of interest, instead of all messages published to the topic. Amazon CloudWatch gives visibility into your filtering activity, and AWS CloudFormation allows you to deploy subscription filter policies in an automated and secure manner.

From <<https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>>

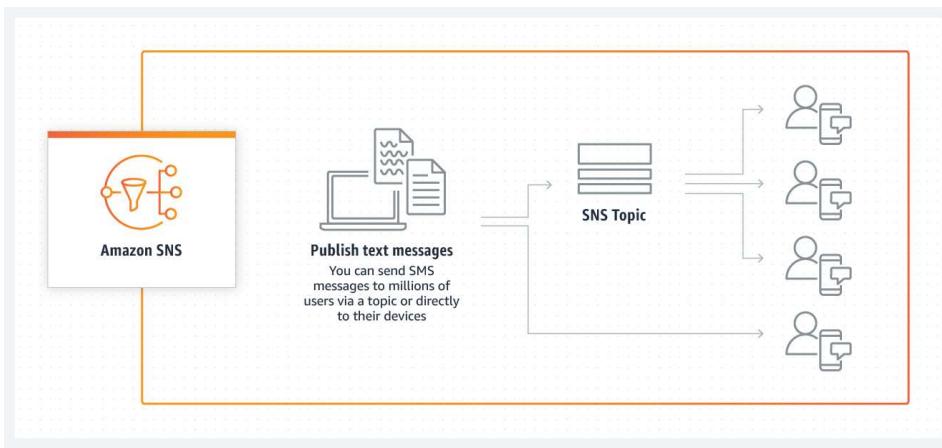
How it works

From <<https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>>

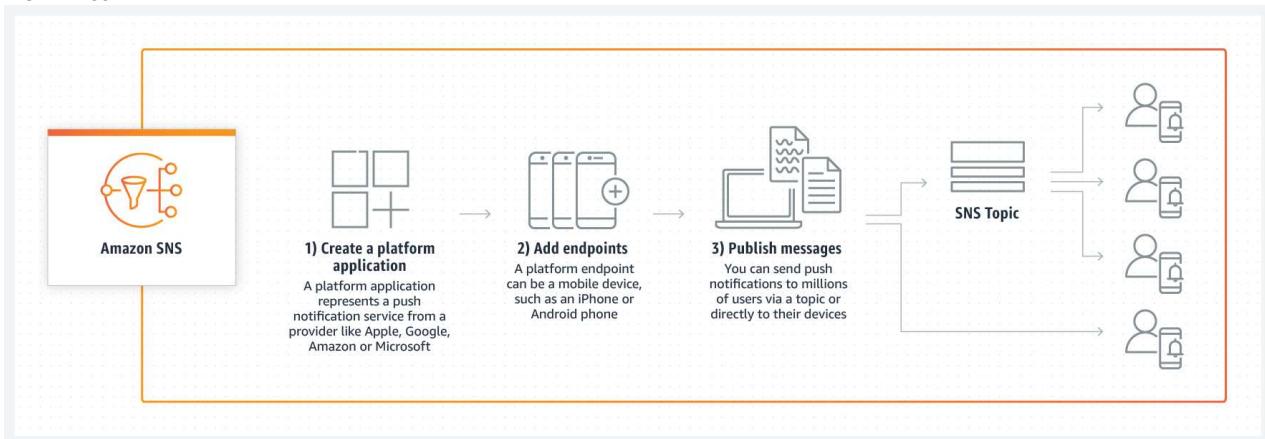
PUB-SUB



SMS



MOBILE PUSH

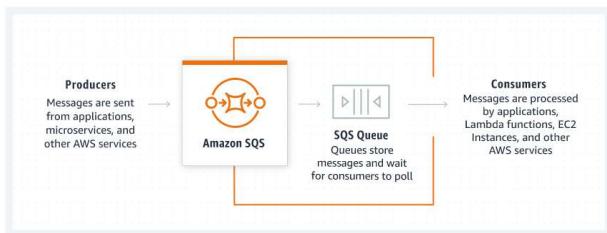


Amazon SQS

A message queuing service

Amazon SQS provides queues for high-throughput, system-to-system messaging. You can use queues to decouple heavyweight processes and to buffer and batch work. Amazon SQS stores messages until microservices and serverless applications process them.

From <<https://console.aws.amazon.com/sqs/v2/home?region=us-east-1#/>>



Benefits and features

Highly scalable Standard and FIFO queues

Queues scale elastically with your application. Nearly unlimited throughput and no limit to the number of messages per queue in Standard queues. First-In-First-Out delivery and exactly once processing in FIFO queues.

Durability and availability

Your queues are distributed on multiple servers. Redundant infrastructure provides highly concurrent access to messages.

Security

Protection in transit and at rest. Transmit sensitive data in encrypted queues. Send messages in a Virtual Private Cloud.

Batching

Send, receive, or delete messages in batches of up to 10 messages or 256KB to save costs.

From <<https://console.aws.amazon.com/sqs/v2/home?region=us-east-1#/>>

Pricing (US)

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.

From <<https://console.aws.amazon.com/sqs/v2/home?region=us-east-1#/>>

Works with

Amazon SNS

Pub-sub messaging for microservices and serverless applications.

From <<https://console.aws.amazon.com/sqs/v2/home?region=us-east-1#/>>

AMAZON SNS ve SQS en eski servislerinden

SNS= notification servis

SQS de Queue servis

<https://aws.amazon.com/blogs/compute/building-loosely-coupled-scalable-c-applications-with-amazon-sqs-and-amazon-sns/>

From <<https://app.slack.com/client/T0227UVRJU8/C021BG84YJJ>>



What is SQS?



- Amazon **Simple Queue Service (SQS)** is a fully managed **message queuing service** that enables you to decouple and scale microservices, distributed systems, and serverless applications.

CLARUSWAY

Decouple onemli konu sinavlarda cikabilir.



What is SQS?

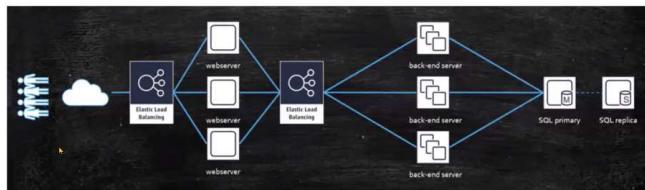


CLARUSWAY

Consumer gelen siraya gore mesahlari cekiyor
Islem bittikten sonra da silme islemlerini uyapıyor

▶ SQS

What is SQS?



CLARUSWAY
WAY TO REINVENT YOURSELF

6

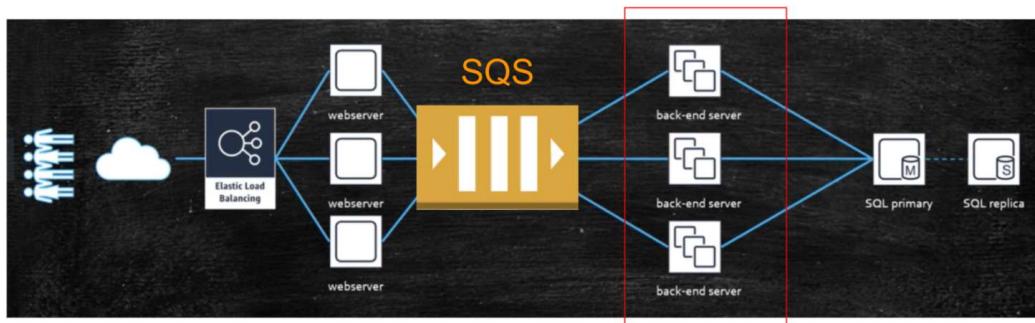
Decouple mantiginda sqs kullaniyoreuz
On ve arkada taraftaki backend serverler birbirleriyle iliskili

▶ SQS

What is SQS?

CloudWatch Metric – Queue Length
ApproximateNumberOfMessages

ASG



SQS mesajlari gelen mesajları belli bir sure tutuyor backend server işlem yapamadığı zaman mesaj sqs içinde sırada bekliyor. İşlem bitince sqs içindeki mesaj siliniyor.
Decouple microservis uygulamalarında birbirinden ayılıyor.
Fault tolerance i azaltıyoruz bu mimariyle.

Yine ASG ile SQS in içinde visible olan metric e göre backend serverların sayısını artırıp azaltabiliyoruz

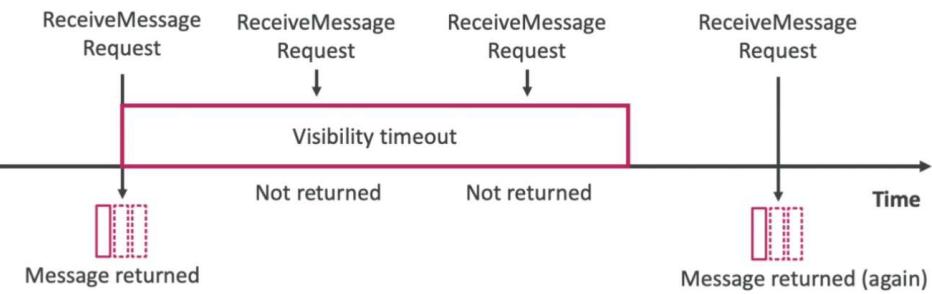
What is decouple SQS?

SQS lets you decouple application components so that they run and fail independently, increasing the overall fault tolerance of the system. Multiple copies of every message are stored redundantly across multiple availability zones so that they are available whenever needed.

SQS, uygulama bileşenlerini birbirinden bağımsız olarak çalışacak ve arızalanacak şekilde ayırmaya olanak tanıyarak sistemin genel hata toleransını artırır. Her mesajın birden çok kopyası, gerektiğinde kullanılabilir olmaları için birden çok kullanılabilirlik bölgesinde yedekli olarak depolanır.

SQS

Message Visibility Timeout

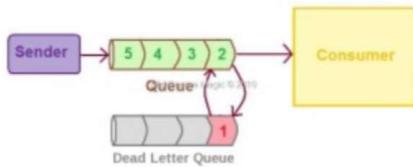
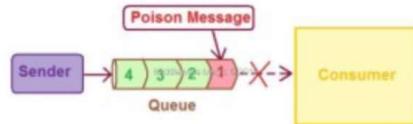


CLARUSWAY
WAY TO REINVENT YOURSELF

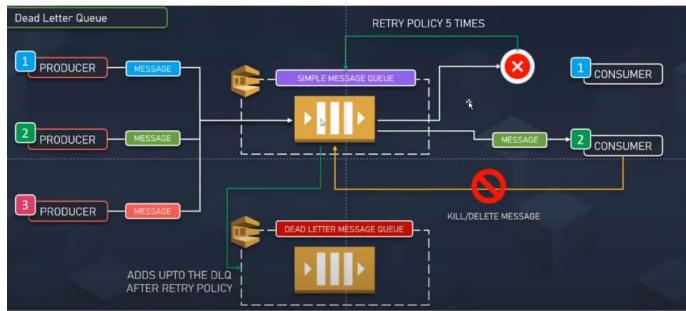
Gelen mesaj consumer tarafindan isleniyor.
Aynı mesajın diğer consumer tarafından da işlenmesini engellenmek için message visibility time out ayarlanıyor. Bu süre içinde diğer consumerlar mesajı göremiyor.

SQS

Dead Letter Queue (DLQ)

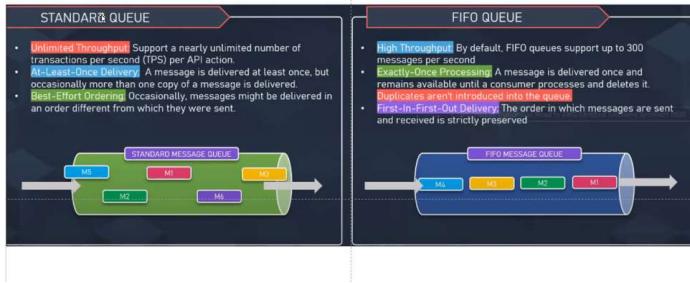
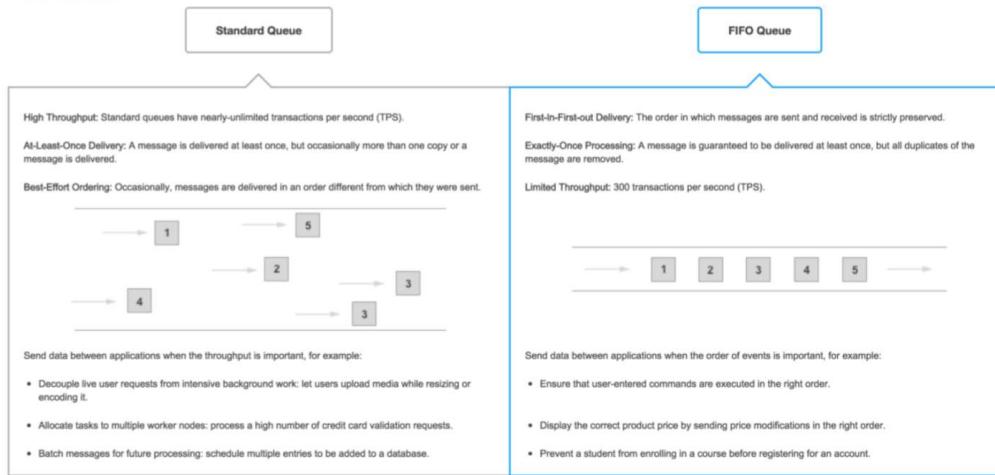


CLARUSWAY
WAY TO REINVENT YOURSELF



DLQ standart Q'nun ayını. Consumer gelen mesajı işlemee calisti ancak isleyemedi tekrar Q'ya gonderiyor consumer deniyo belli bir sayıda bizim ayarladığımız sayıda. Olmazsa DLQ'ye gonderiliyor. Dah asında troubleshooting yaparken içindeki mesajlara bakılabilir. HATALI MESAJLARIN GONDERILDIGI YER.

STANDART Q'nun içinde mesajlar 14 güne kadar duruyor. Death LQ'nın standartdan daha fazla olması lazımkı buraya gönderilen hata mesajını daha sonra inceleyebilelim.



Saniyeeki işlem sayısında Standart Q da limit yok. Ancak FIFO Q da hightthroughput dedigimiz 300 sınırlaması var, bosphorus ozelligi kullanarak $300 \times 10 = 3\,000$ 3000 e cikarılabiliriyor,

Standart Q da sıraya giren mesajın sırası girileceğinden dolayı comsumerlar için mesajın sırası önemli değil.

Mesaj birden fazla process işlem görebilir Standart Q da.

Fifo da duplicate diye bir kavram yok. Standart Q da olabiliyor.

FIFO DA ILK GIREN MESAJIK ILK CIKIYOR.



- Pay only for what you use
- AWS **Free Tier** includes **1 million requests** with Amazon Simple Queue Service (SQS).

Amazon SQS		Overview	Features	Pricing	Getting Started	Resources	FAQs
API Actions	Every Amazon SQS action counts as a request.						
FIFO Requests	API actions for sending, receiving, deleting, and changing visibility of messages from FIFO queues are charged at FIFO rates. All other API requests are charged at standard rates.						
Contents of Requests	A single request can have from 1 to 10 messages, up to a maximum total payload of 256 KB.						
Size of Payloads	Each 64 KB chunk of a payload is billed as 1 request (for example, an API action with a 256 KB payload is billed as 4 requests).						
Interaction with Amazon S3	When using the Amazon SQS Extended Client Library to send payloads using Amazon S3, you incur Amazon S3 charges for any Amazon S3 storage you use to send message payloads.						
Interaction with AWS KMS	When using the AWS Key Management Service to manage keys for SQS server-side encryption, you incur charges for calls from Amazon SQS to AWS KMS. For more information see KMS pricing and How Do I Estimate My AWS KMS Usage Costs? in the Amazon SQS Developer Guide .						

Amazon SQS requests are listed on the [product features page](#).

SQS HANDSON

SQS Hands-on

Part 1 - Creating Queue, Sending and Receiving Messages

**Manuel yapacagiz mesaj yazcاز gondercez cosumer gibi pull edecez vs
SQS i lambda ile entegre edecegiz, mesaj gidince lambda fonksiyonun trigger edecek daha sonra cloudwatchda loglara yazacak**

Step 1 : Create Queue

- Go to 'SQS' service on AWS console.
- Click 'Create queue'.

Screenshot of the AWS SQS home page (<https://console.aws.amazon.com/sqs/v2/home?region=us-east-1#/>)

The page title is "Amazon SQS" and subtitle is "A message queuing service". A brief description states: "Amazon SQS provides queues for high-throughput, system-to-system messaging. You can use queues to decouple heavyweight processes and to buffer and batch work. Amazon SQS stores messages until microservices and serverless applications process them."

Get started: Learn how to use Amazon SQS by creating a queue, sending a message to the queue, and receiving and processing the message. A blue box highlights the "Create queue" button.

Pricing (US): 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. Includes a "Cost calculator" link.

Documentation: Includes links to the "Developer guide" and other documentation resources.

How it works: A diagram showing the flow from "Producers" (HTTP request from applications, microservices, and other AWS services) to "Amazon SQS" (represented by a central box with a circular icon), then to an "SQS Queue" (represented by a box with three vertical bars), and finally to "Consumers" (Messages are processed by applications, Lambda functions, EC2 Instances, and other AWS services).

AWS Cloud Practitioner sınavlarında servislerin ilk sayfadaki açıklamalarını anlamak çok önemlidir.

Mesela SQS için :

A message queuing service

Amazon SQS provides queues for high-throughput, system-to-system messaging. You can use queues to decouple heavyweight processes and to buffer and batch work. Amazon SQS stores messages until microservices and serverless applications process them." önemlidir

From <<https://console.aws.amazon.com/sqs/v2/home?region=us-east-1#/>>

- 'Details' :
 - Type: Standart (Keep default)
 - Name: My-First-Queue
 - Display Name: My-First-Topic
- Talk about Configuration, Access policy and Dead-letter queue (Keep default)
- Keep rest default.
- Click 'Create queue'.

Screenshot of the "Create queue" wizard in the AWS SQS console.

Create queue

Details

Type: Choose the queue type for your application or cloud infrastructure.

Standard (Info): At-least-once delivery, message ordering isn't preserved

- At-least once delivery
- Best-effort ordering

FIFO (Info): First-in-first-out delivery, message ordering is preserved

- First-in-first-out delivery
- Exactly-once processing

Name: My-First-Topic

A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (_).

Configuration

Set the maximum message size, visibility to other consumers, and message retention. [Info](#)

Visibility timeout [Info](#) Message retention period [Info](#)

Tip daha sonra değiştiriliyor. Bastan seçim yaarken unutmamak lazım

Configuration

Set the maximum message size, visibility to other consumers, and message retention. [Info](#)

Visibility timeout Info	Message retention period Info
<input type="text" value="30"/> Seconds ▾	<input type="text" value="4"/> Days ▾
Should be between 0 seconds and 12 hours.	
Delivery delay Info	Maximum message size Info
<input type="text" value="0"/> Seconds ▾	<input type="text" value="256"/> KB
Should be between 0 seconds and 15 minutes.	
Receive message wait time Info	
<input type="text" value="0"/> Seconds	
Should be between 0 and 20 seconds.	

Receive message time: 0 ise short poning, 20 sn ise long poning

Q nu icine mesaj geldi mesaj bos, consumer devamlı request gonderiyor. Bu bos mesajlarin alinip islenmesini engellemek icin long poning diye bir sey cikartilmis, bos mesajlarin islenmesini engelleyerek maliyet tasarrufu sagliyor.

Queue configuration

The configuration section is pre-filled with default values that are suitable for basic queue operation. Configure the following values to customize your queue:

- **Visibility timeout:** the length of time that a message received from a queue (by one consumer) will not be visible to the other message consumers.

Visibility timeout sets the length of time that a message received from a queue (by one consumer) will not be visible to the other message consumers.

The visibility timeout begins when Amazon SQS returns a message. If the consumer fails to process and delete the message before the visibility timeout expires, the message becomes visible to other consumers. If a message must be received only once, your consumer must delete it within the duration of the visibility timeout.

The default visibility timeout setting is 30 seconds. This setting applies to all messages in the queue. Typically, you should set the visibility timeout to the maximum time that it takes your application to process and delete a message from the queue.

For optimal performance, set the visibility timeout to be larger than the AWS SDK read timeout. This applies to using the ReceiveMessage API action with either short polling or long polling.

From <https://console.aws.amazon.com/sqs/v2/home?region=us-east-1#/create-queue#>

- **Message retention period:** the amount of time that Amazon SQS retains a message that does not get deleted. (Mesaj saklama süresi: Amazon SQS'nin silinmeyen bir mesajı saklama süresi.)
- **Delivery delay:** the amount of time to delay the first delivery of each message added to this queue.

Delivery delay

If your consumers need additional time to process messages, you can delay each new message coming to the queue. The delivery delay is the amount of time to delay the first delivery of each message added to the queue. Any messages that you send to the queue remain invisible to consumers for the duration of the delay period. The default (minimum) delay for a queue is 0 seconds. The maximum is 15 minutes.

For standard queues, the per-queue delay setting is not retroactive; changing the setting doesn't affect the delay of messages already in the queue.

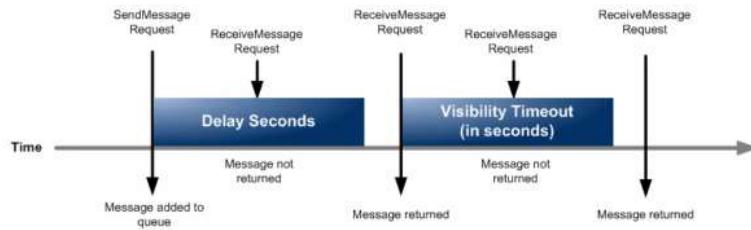
For FIFO queues, the per-queue delay setting is retroactive; changing the setting affects the delay of messages already in the queue.

From <https://console.aws.amazon.com/sqs/v2/home?region=us-east-1#/create-queue#>

Q nun icinde mesaj consumer tarafından gorunmuyor.

- **Maximum message size:** the maximum message size for this queue.
- **Receive message wait time:** the maximum amount of time that polling will wait for messages to become available.
- **Content-based deduplication:** Amazon SQS can automatically create deduplication IDs, based on the body of the message.

From <https://console.aws.amazon.com/sqs/v2/home?region=us-east-1#/create-queue#>



Choose method

Basic Use simple criteria to define a basic access policy.

Advanced Use a JSON object to define an advanced access policy.

Define who can send messages to the queue

Only the queue owner Only the owner of the queue can send messages to the queue.

Only the specified AWS accounts, IAM users and roles Only the specified AWS account IDs, IAM users and roles can send messages to the queue.

Define who can receive messages from the queue

Only the queue owner Only the owner of the queue can receive messages from the queue.

Only the specified AWS accounts, IAM users and roles Only the specified AWS account IDs, IAM users and roles can receive messages from the queue.

JSON (read-only)

```
{
  "Sid": "__owner_statement",
  "Effect": "Allow",
  "Principal": {
    "AWS": "547187538673"
  },
  "Action": [
    "SQS:*"
  ],
  "Resource": "arn:aws:sqs:us-east-1:547187538673:My-First-Topic"
}
```

Encryption - Optional
Amazon SQS provides in-transit encryption by default. To add at-rest encryption to your queue, enable server-side encryption. [Info](#)

Dead-letter queue - Optional
Send undeliverable messages to a dead-letter queue. [Info](#)

Bunlar da default

Create q diyoruz

Amazon SOS > Queues > My-First-Topic

My-First-Topic

Edit Delete Purge Send and receive messages

Details [Info](#)

Name	Type	ARN
My-First-Topic	Standard	arn:aws:sqs:us-east-1:547187538673:My-First-Topic
Encryption	URL	Dead-letter queue
-	https://sns.us-east-1.amazonaws.com/547187538673/My-First-Topic	-

[More](#)

SNS subscriptions Lambda triggers Dead-letter queue Monitoring Tagging Access policy Encryption

Subscription region us-east-1

SNS subscriptions (0) [Info](#)

Search subscriptions

Subscription ARN Topic ARN

View in SNS [Delete](#) [Subscribe to Amazon SNS topic](#)

Sns ve lambda ile ilişkilendirilebiliyoruz

Step 2 : Send Message

Queue My-First-Topic created successfully
You can now send and receive messages.

My-First-Topic

Details [Info](#)

Name My-First-Topic	Type Standard	ARN arn:aws:sqs:us-east-1:547187538673:My-First-Topic
Encryption -	URL https://sqs.us-east-1.amazonaws.com/547187538673/My-First-Topic	Dead-letter queue -

[More](#)

SNS subscriptions | Lambda triggers | Dead-letter queue | Monitoring | Tagging | Access policy | Encryption

Subscription region
us-east-1

SNS subscriptions (0) [Info](#)

[View in SNS](#) [Delete](#) [Subscribe to Amazon SNS topic](#)

[Search subscriptions](#)

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#) [Cookie preferences](#)

Send message yapacagiz

Amazon SQS > Queues > My-First-Topic > Send and receive messages

Send and receive messages

Send messages to and receive messages from a queue.

Send message [Info](#)

Message body
Enter the message to send to the queue.

Delivery delay [Info](#)
0 Seconds

Should be between 0 seconds and 15 minutes.

Message attributes - *Optional* [Info](#)

Receive messages [Info](#)

Messages available 0	Polling duration 30	Maximum message count 10	Polling progress 0 receives/second 0%
-------------------------	------------------------	-----------------------------	--

[Edit poll settings](#) [Stop polling](#) [Poll for messages](#)

Messages (0)

[View details](#) [Delete](#)

[Search messages](#)

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#) [Cookie preferences](#)

- On My-First-Queue page Click 'Send and receive messages'.
- 'Send message'.
 - Message body: "This is the first message for sqs."
- Keep rest default.
- Click 'Send message'.
- Show 'Receive messages' >> 'Messages available' = 1.

Amazon SQS > Queues > My-First-Topic > Send and receive messages

Send and receive messages

Send messages to and receive messages from a queue.

Send message Info

Your message has been sent and is ready to be received.

Message body
Enter the message to send to the queue.
This is my first Queue

Delivery delay Info
0 Seconds
Should be between 0 seconds and 15 minutes.

Message attributes - Optional Info

Receive messages Info

Messages available 1 Polling duration 30 Maximum message count 10 Polling progress 0 receives/second

Edit poll settings **Stop polling** **Poll for messages**

Step 3 : Poll for Messages

- Click 'Poll for messages' under 'Receive messages'.
- Click on the polled message under 'Messages'.
- Show the message.
- Click 'Done'.
- Select the polled message and click 'Delete' and delete the message.
- Show 'Status' >> 'Confirmed'.

Your message has been sent and is ready to be received.

Message body
Enter the message to send to the queue.
This is my first Queue

Delivery delay Info
0 Seconds
Should be between 0 seconds and 15 minutes.

Message attributes - Optional Info

Receive messages Info

Messages available 0 Polling duration 30 Maximum message count 10 Polling progress 0 receives/second

Poll for messages

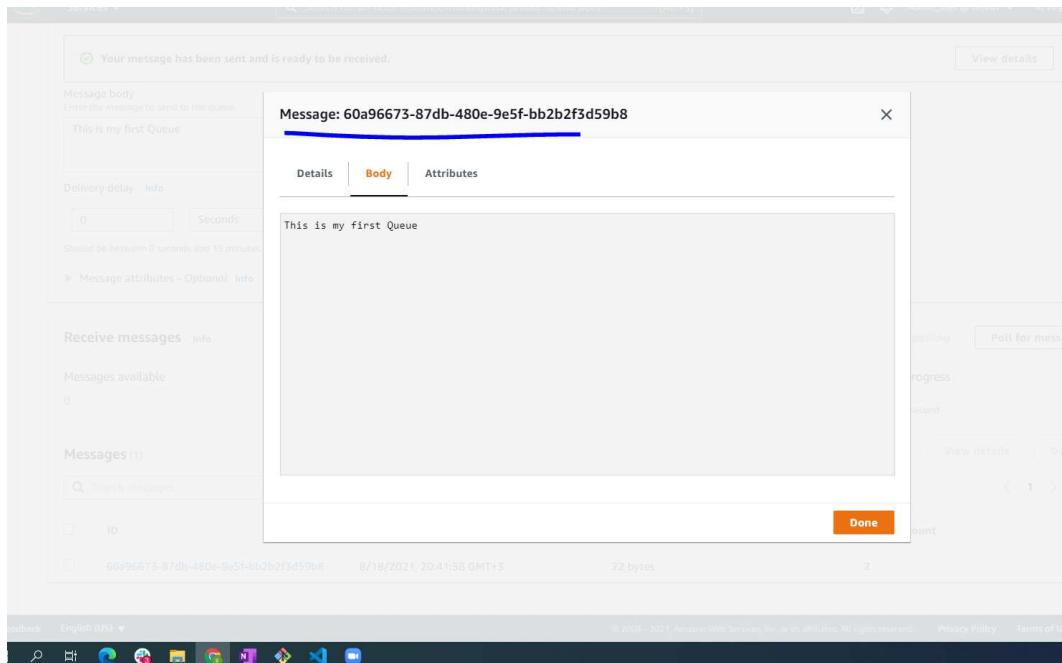
Messages (1)

Search messages

ID	Sent	Size	Receive count
60a96673-87db-480e-9e5f-bb2b2f3d59b8	8/18/2021, 20:41:58 GMT+3	22 bytes	2

Poll mesajı cekme işlemi

Mesaj id ye tıklayınca



Mesajı görebiliyoruz

Bekledikte bir muddet sonra mesajlar gitmisti
Tekrar poll yapınca geldiler

Silme işlemi

The screenshot shows the 'Receive messages' section of the AWS Lambda Queue interface. It displays a table of messages with columns: ID, Sent, Size, and Receive count. Five messages are listed, each with a checkmark next to it. A blue arrow points to the 'Delete' button at the top right of the message table. The table shows the following data:

ID	Sent	Size	Receive count
ed6df597-8407-4055-b56a-5da5245207db	8/18/2021, 20:44:55 GMT+3	22 bytes	2
60a96673-87db-480e-9e5f-bb2b2f3d59b8	8/18/2021, 20:41:58 GMT+3	22 bytes	4
03f85feb-34a1-4499-b181-15e1891f9b43	8/18/2021, 20:44:34 GMT+3	22 bytes	3
c6f54e68-2a83-453f-89f0-ff5f98d51c26	8/18/2021, 20:44:45 GMT+3	23 bytes	2
134d48af-53fa-4f64-b018-9c259f0cae85	8/18/2021, 20:44:40 GMT+3	23 bytes	2

Silindiler

The screenshot shows the 'Receive messages' section of the AWS Lambda Queue interface. A green box at the top indicates '1 message deleted successfully.' Below it, the message table shows the same five messages as before. The table includes columns for 'Messages available' (0), 'Polling duration' (30), 'Maximum message count' (10), and 'Polling progress' (1 receives/second, 40%). The bottom part of the interface shows a message table with 0 messages available, a 'Poll for messages' button, and a note: 'No messages. To view messages in the queue, poll for messages.'

Step 4 : Populate Queue

- Do the Step 2 again with 3 messages.
- Send 3 new messages consecutively to the queue ("This is the 2nd/3rd/4th message for sqs.")
- Show the messages in the queue.
- Do the Step 3 again and poll the messages.
- Delete the messages in the queue.

Part 2 - Creating Lambda Function to Be Triggered by SQS

User blueprint i sececegiz

Lambda > Functions > Create function

Create function [Info](#)

Choose one of the following options to create your function.

- Author from scratch
- Use a blueprint **SQS**
- Container image
- Browse serverless app repository

Blueprints [Info](#)

Filter by top level category Search by keyword 1 match

Blueprint attributes [Clear filters](#)

Blueprint name
Runtime
Keyword

An Amazon SQS trigger that logs messages in a queue.

nodejs - sqs

Cancel [Configure](#)

Sqs yazip ariyoruz

Br sonuc cikto
onu secip configure e basiyoruz

Step 1 : Create Lambda Function

- Go to 'Lambda' service on AWS console.
- Click 'Create function' .
- Select 'Use a blueprint' .
- Search 'sqc' in the the blueprints search bar.
- Select 'sqc-poller' and click 'Configure' .

Lambda > Functions > Create function > Configure blueprint sqs-poller

Basic information Info

Function name

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

- Create a new role with basic Lambda permissions
- Use an existing role
- Create a new role from AWS policy templates

Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Role name
Enter a name for your new role.

Use only letters, numbers, hyphens, or underscores with no spaces.

Policy templates - optional Info
Choose one or more policy templates.

Amazon SQS poller permissions SQS

SQS trigger

Feedback English (US) ▾ Remove © 2008 - 2021, Amazon Web Services, Inc. or its affiliates

- Basic information
 - Function name : sqs-poller
 - Execution role : Create a new role from AWS policy templates (Keep default)
 - Role name : sqs-poller-role
 - Policy templates : Keep the default "Amazon SQS poller permissions"

Function name

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

- Create a new role with basic Lambda permissions
- Use an existing role
- Create a new role from AWS policy templates

Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Role name
Enter a name for your new role.

Use only letters, numbers, hyphens, or underscores with no spaces.

Policy templates - optional Info
Choose one or more policy templates.

Amazon SQS poller permissions SQS

SQS trigger Remove

SQS queue
Choose or enter the ARN of an SQS queue.

Batch size
The maximum number of messages to retrieve in a single batch.

Batch window
The maximum amount of time to gather records before invoking the function, in seconds.

In order to read from the SQS trigger, your execution role must have proper permissions.

Enable trigger
Enable the trigger now, or create it in a disabled state for testing (recommended).

Lambda function code

- Keep rest default.

- Click 'Create function'.

Lambda function code

Code is preconfigured by the chosen blueprint. You can configure it after you create the function. [Learn more](#) about deploying Lambda functions.

Runtime
Node.js 12.x

```

1 console.log('Loading function');
2
3 exports.handler = async (event) => {
4   //console.log('Received event:', JSON.stringify(event, null, 2));
5   for (const { messageId, body } of event.Records) {
6     console.log(`SQS message ${messageId}: ${body}`);
7   }
8   return `Successfully processed ${event.Records.length} messages.`;
9 };
10

```

[Cancel](#) [Create function](#)

Lambda > Functions > sqs-poller-

sqs-poller-

Congratulations! Your Lambda function "sqs-poller-" has been successfully created and configured with My-First-Topic as a trigger in a disabled state. We recommend testing the function behavior before enabling the trigger. [X](#)

[Function overview](#) [Info](#)

Code source [Info](#)

[Upload from](#)

[File](#) [Edit](#) [Find](#) [View](#) [Go](#) [Tools](#) [Window](#) [Test](#) [Deploy](#) [Changes deployed](#)

index.js

```

1 console.log('Loading function');

```

Description
An Amazon SQS trigger that logs messages in a queue.

Last modified
1 minute ago

Function ARN
[arn:aws:lambda:us-east-1:547187538673:function:sqs-poller-](#)

[Code](#) [Test](#) [Monitor](#) [Configuration](#) [Aliases](#) [Versions](#)

TESTI TEST ETTIK

[Code](#) [Test](#) [Monitor](#) [Configuration](#) [Aliases](#) [Versions](#)

Code source [Info](#)

[Upload from](#)

[File](#) [Edit](#) [Find](#) [View](#) [Go](#) [Tools](#) [Window](#) [Test](#) [Deploy](#) [Changes deployed](#)

Execution results

Test Event Name
myfirstevent

Response
Successfully processed 1 messages.

Function Log
START RequestId: 9abb07f4-25ee-4b87-b5ba-ae7540032e24 Version: \$LATEST
2020-08-18T10:20:09Z 9abb07f4-25ee-4b87-b5ba-ae7540032e24 INFO SQS message 19dd0b57-b21e-4ac1-bd88-01bb068cb78: "Hello from SQS!"
END RequestId: 9abb07f4-25ee-4b87-b5ba-ae7540032e24
REPORT RequestId: 9abb07f4-25ee-4b87-b5ba-ae7540032e24 Duration: 37.44 ms Billed Duration: 38 ms Memory Size: 128 MB Max Memory Used: 65 MB

Request ID
9abb07f4-25ee-4b87-b5ba-ae7540032e24

Add trigger

Execution result: succeeded (logs)

Details

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

```
"Successfully processed 1 messages."
```

Summary

Code SHA-256	Request ID
RznHSDt7Kuo0sJUrka6v0b8Jze2GICQ5rFTQ69ORkg=	9abb07f4-25e4-4b87-b5ba-ae7540032e24
Duration	Billed duration
37.44 ms	38 ms
Resources configured	Max memory used
128 MB	65 MB

Log output

The section below shows the logging calls in your code. [Click here](#) to view the corresponding CloudWatch log group.

START RequestId: 9ahh07f4-25e4-4b87-b5ba-ae7540032e24 Version: \$1 ATFST

Cloudwatcha da gelince loglarımızın buraya geldiğini goruyoruz

CloudWatch > Log groups

Log groups (7)

By default, we only load up to 10000 log groups.

Log group	Retention	Metric filters	Contributor Insights	Subscription filters
/aws/lambda/NumberGenerator	Never expire	-	-	-
/aws/lambda/sqs-poller-	Never expire	-	-	-
/aws/lambda/start-instance	Never expire	-	-	-
/aws/lambda/Stop_Instance	Never expire	-	-	-
/var/log/messages	Never expire	-	-	-
AccessLog	Never expire	-	-	-
ErrorLog	Never expire	-	-	-

- Click 'Configuration' tab under 'sq-s-poller' function.

- Select 'SQS: My-First-Queue' and click 'Enable' to enable trigger.(Wait for it to be "Enabled")

SQS

+ Add destination

8 minutes ago

Function ARN
arn:aws:lambda:us-east-1:547187538673:function:sqs-poller

Code Test Monitor Configuration Aliases Versions

General configuration Triggers (1)

Find triggers C Enable Disable Fix errors Delete Add trigger

Trigger

SQS: My-First-Topic (Disabled)
arn:aws:sqs:us-east-1:547187538673:My-First-Topic

Details

Enable ediyoruz

Layers (0)

+ Add destination

Last modified 8 minutes ago

Function ARN arn:aws:lambda:us-east-1:547187538673:function:sqs-poller

Code Test Monitor Configuration Aliases Versions

General configuration Triggers (1)

Find triggers C Enable Disable Fix errors Delete Add trigger

Trigger

SQS: My-First-Topic (Enabling)
arn:aws:sqs:us-east-1:547187538673:My-First-Topic

Details

SQS in icinde lambda trigger'e bakınca da enabled olduğunu goruyoruz

Amazon SQS > Queues > My-First-Topic

My-First-Topic

Edit Delete Purge Send and receive messages

Details Info

Name	Type	ARN
My-First-Topic	Standard	arn:aws:sqs:us-east-1:547187538673:My-First-Topic
Encryption	URL	Dead-letter queue
-	https://sqs.us-east-1.amazonaws.com/547187538673/My-First-Topic	-
> More		

SNS subscriptions Lambda triggers Dead-letter queue Monitoring Tagging Access policy Encryption

Lambda triggers (1) Info

Search triggers C View in Lambda Delete Configure Lambda function trigger

UUID	ARN	Status	Last modified
970fd2da-e0cd-44db-9cd8-2052e2f99869	arn:aws:lambda:us-east-1:547187538673:function:sqs-poller	Enabled	8/18/2021, 9:16:10 PM

SQS servisi backendlerin mail atarken kullandıkları servislerden

Q da sırada bekletiyor.

Timeout durumu olan servislerde bu q mantığı çalışıyor .

MW Group, sürücülerine dinamik olarak güncellenen harita bilgileri sunmak için BMW 7 Serisi araçlardan sensör verilerini toplayan bağlı araç uygulaması için AWS'yi kullanıyor. BMW, sensör olarak araç (CARASSO) adını verdiği yeni hizmetini Amazon SQS, Amazon S3, Amazon DynamoDB, Amazon RDS ve

AWS Elastic Beanstalk'tan yararlanarak yalnızca altı ayda oluşturdu.

From <<https://app.slack.com/client/T0227UVRJU8/C021BG84YJJ>>
<https://aws.amazon.com/tr/sqs/>

From <<https://app.slack.com/client/T0227UVRJU8/C021BG84YJJ>>

Step 2 : Send Message to Invoke Lambda

- Go back to 'SQS' service on AWS console.
- Send a new message like "This message is sent from sqs to trigger lambda" to the 'My-First-Queue'.
- Show there is no message since it has been polled by lambda.

The screenshot shows the AWS SQS console. In the 'Send message' section, a success message is displayed: 'Your message has been sent and is ready to be received.' Below this, there is a 'Message body' input field containing 'send message', a 'Delivery delay' field set to 0 seconds, and a 'Message attributes' section. In the 'Receive messages' section, it shows 0 messages available, a polling duration of 30 seconds, and a maximum message count of 10. The polling progress bar is at 0%.

Biz available message da gormiyoruz bile

Lambda otomatik yapiyor çok hızlı oluyor biz gormemiyoruz

The screenshot shows the AWS CloudWatch Logs interface. It displays the 'Log group details' for the '/aws/lambda/sqs-poller-' log group. The retention is set to 'Never expire', creation time is '19 minutes ago', and ARN is 'arn:aws:logs:us-east-1:547187538673:log-group:/aws/lambda/sqs-poller-*'. There are two log streams listed: '2021/08/18/[\$LATEST]ee9b850677dd428f9798764ef0338415' (last event time: 2021-08-18 21:25:20 UTC+03:00) and '2021/08/18/[\$LATEST]a657e059e10148e8a87fcbe623b595bd' (last event time: 2021-08-18 21:10:21 UTC+03:00).

Cloudwatche gidince goruyoruz loga dusuyor

No older events at this moment. <i>Retry</i>
▶ 2021-08-18T21:25:20.732+03:00 START RequestId: 29ca6c47-56a6-5ce0-b7b2-35d004671d18 Version: \$LATEST
▶ 2021-08-18T21:25:20.732+03:00 2021-08-18T18:25:20.732Z undefined INFO Loading function
▶ 2021-08-18T21:25:20.737+03:00 2021-08-18T18:25:20.737Z 29ca6c47-56a6-5ce0-b7b2-35d004671d18 INFO SQS message a36585e7-e571-45ab-8a64-966bf1a3346f:
▶ 2021-08-18T21:25:20.741+03:00 END RequestId: 29ca6c47-56a6-5ce0-b7b2-35d004671d18
▶ 2021-08-18T21:25:20.741+03:00 REPORT RequestId: 29ca6c47-56a6-5ce0-b7b2-35d004671d18 Duration: 4.08 ms Billed Duration: 5 ms Memory Size: 128 MB Max Mem
No newer events at this moment. <i>Auto retry paused. Resume</i>

Step 3 : Check Logs

- Go back to `CloudWatch` service on AWS console.
- From left-hand menu `Logs` >> `Log groups`
- Click on `/aws/lambda/sqs-poller`

Daha sonra tekrar mesaj gönderdik tekrar gorduk

Timestamp	Message
2021-08-18T21:25:20.732+03:00	START RequestId: 29ca6c47-56a6-5ce0-b7b2-35d004671d18 Version: \$LATEST
2021-08-18T21:25:20.732+03:00	2021-08-18T18:25:20.732Z undefined INFO Loading function
2021-08-18T21:25:20.737+03:00	2021-08-18T18:25:20.737Z 29ca6c47-56a6-5ce0-b7b2-35d004671d18 INFO SQS message a36585e7-e571-45ab-8a64-966bf1a3346f: "send me"
2021-08-18T21:25:20.741+03:00	END RequestId: 29ca6c47-56a6-5ce0-b7b2-35d004671d18
2021-08-18T21:25:20.741+03:00	REPORT RequestId: 29ca6c47-56a6-5ce0-b7b2-35d004671d18 Duration: 4.08 ms Billed Duration: 5 ms Memory Size: 128 MB Max Mem
2021-08-18T21:28:15.051+03:00	START RequestId: cdc4df44-3088-533c-aec3-86d363fd913e Version: \$LATEST
2021-08-18T21:28:15.102+03:00	2021-08-18T18:28:15.102Z cdc4df44-3088-533c-aec3-86d363fd913e INFO SQS message 3a82ee47-3c30-4863-ad48-276d07595c3a: "send me"
2021-08-18T21:28:15.103+03:00	END RequestId: cdc4df44-3088-533c-aec3-86d363fd913e
2021-08-18T21:28:15.103+03:00	REPORT RequestId: cdc4df44-3088-533c-aec3-86d363fd913e Duration: 41.50 ms Billed Duration: 42 ms Memory Size: 128 MB Max Mem
No newer events at this moment. Auto retry paused. <i>Resume</i>	

- Click on the log stream and show the message sent from sqs and processed by lambda.

- Delete/terminate the resources created.

From <<https://app.slack.com/client/T0227UVRJU8/C021BG84YJJ>>

SNS

2

SNS

CLARUSWAY

► SNS

What is SNS?



Amazon
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**).

CLARUSWAY
WAY TO REINVENT YOURSELF

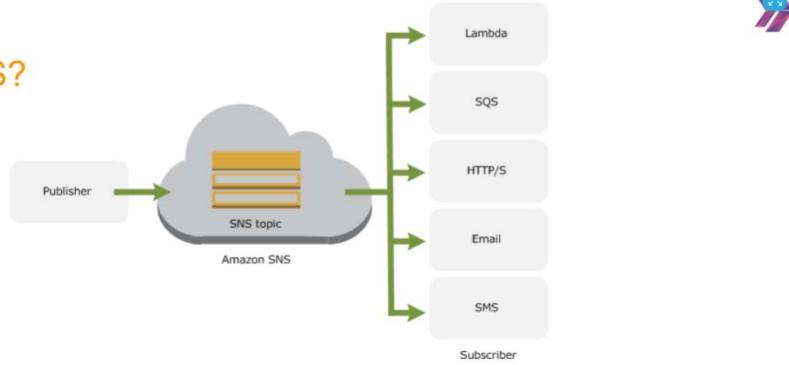
14

Puböisher mesajı yayınlayanlar
Mesajı oluşturup iletiyoruz subscriberlara

SNS de belirledigimiz topici farklı protokollerle farklı kullanıcılara gönderebiliriz

► SNS

What is SNS?



- Clients can subscribe to the **SNS topic** and receive published messages using a supported protocol, such as Amazon SQS, AWS Lambda, HTTP, email, mobile push notifications, and mobile text messages (SMS).

CLARUSWAY
WAY TO REINVENT YOURSELF

15

► SNS

What is SNS?

Application-to-Application (A2A)



CLARUSWAY
WAY TO REINVENT YOURSELF

16

Burda application

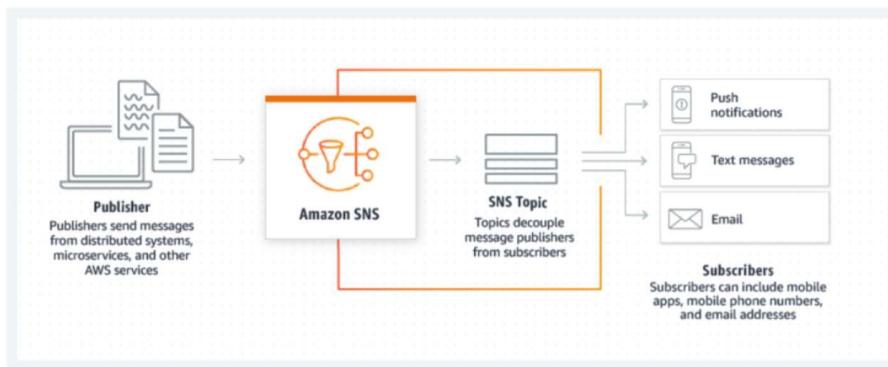
Bir topic de mesaj yazıyoruz lambda'ya sqs e http s e giriyor.

Bir topic oluşturuyoruz bu topic i birden fazla kullanıcıya gönderebiliyoruz veya birden farklı protokollerdeki kisiye gönderebiliyoruz

SNS

What is SNS?

Application-to-Person (A2P)



CLARUSWAY

WAY TO REINVENT YOURSELF

17

SNS

What is SNS?

SNS is integrated with many AWS Services

- CloudWatch (Alarms/Events)
- S3 (Bucket Events)
- CloudFormation (State changes etc.)
- Auto Scaling Groups
- And many others **can invoke SNS.**

CLARUSWAY

18

Autoscaling Cloud Watch S3 CloudFormation vb aws servisleriyle ilintili

SNS

What is SNS?



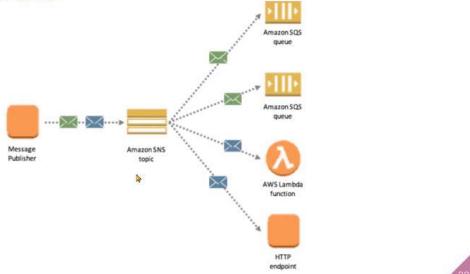
CLARUSWAY
WAY TO REINVENT YOURSELF

19

Farklı protokollerde gele mesajları iletmesine fanout diyoruz.
SNS e bilgiler farklı sistemlerden geliyor.

SNS

Fan Out with SNS



CLARUSWAY
WAY TO REINVENT YOURSELF

20



SNS

Pricing



- Amazon SNS has **no upfront costs** and you can **pay as you go**. You pay based on the number of notifications you publish, the number of notifications you deliver, and any additional API calls for managing topics and subscriptions. Delivery **pricing varies by endpoint type**. You can get started for free with the SNS free tier.

CLARUSWAY
WAY TO REINVENT YOURSELF

21

Ucretlendirme degisiklik gosteryor.
Pay as you go

▶ SQS & SNS

Hands on



- Hands-on!!!
- Let's go to the AWS Management Console

CLARUSWAY
WAY TO REINVENT YOURSELF

22

HANDSON

```
# SNS Hands-on
## Part 1 - Creating Topic, Subscription and Publishing Message
```

The screenshot shows the AWS SNS Topics page. On the left, there's a sidebar with links for Dashboard, Topics (which is selected and highlighted in orange), Subscriptions, Mobile (Push notifications, Text messaging (SMS), Origination numbers), and Analytics. The main content area has a header 'Topics (3)'. Below it is a search bar and a table with three rows. The table columns are Name, Type, and ARN. The rows are:

Name	Type	ARN
Clarus_Alarms_SNS	Standard	arn:aws:sns:us-east-1:547187538673:Clarus_Alarms_SNS
My_Aws_Billing_Topics	Standard	arn:aws:sns:us-east-1:547187538673:My_Aws_Billing_Topics
dynamodb	Standard	arn:aws:sns:us-east-1:547187538673:dynamodb

Step 1 : Create Topic

- Go to `SNS` service on AWS console.
- Click `Topics` >> `Create topic`.
- `Details`.
 - Type: Standard
 - Name: Demo-topic
 - Display Name: My-First-Topic
- Keep rest default.
- Click `Create`.

AWS Services ▾ Search for services, features, marketplace products, and docs [Alt+S] Admin_user @ ser007 N. Virginia

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, Lambda, HTTP, SMS, email, mobile application endpoints

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
Demo-topic
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](#)
My-First-Topic
Minimum 100 characters, including hyphens (-) and underscores (_).

Encryption - optional
Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

Access policy - optional
This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic. [Info](#)

Delivery retry policy (HTTP/S) - optional
The policy defines how Amazon SNS retries failed deliveries to HTTP/S endpoints. To modify the default settings, expand this section. [Info](#)

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookies

Amazon SNS X Topic Demo-topic created successfully. You can create subscriptions and send messages to them from this topic. Publish message X

Dashboard Topics Subscriptions Mobile Push notifications Text messaging (SMS) Origination numbers

Amazon SNS > Topics > Demo-topic

Demo-topic Edit Delete Publish message

Details	
Name	Display name
Demo-topic	My-First-Topic
ARN	Topic owner
arn:aws:sns:us-east-1:547187538673:Demo-topic	547187538673
Type	Encryption
Standard	

Subscriptions Access policy Delivery retry policy (HTTP/S) Delivery status logging Encryption Tags

Subscriptions (0) Create subscription

ID	Endpoint	Status	Protocol

Step 2 : Create Subscription

- On Demo-topic page Click `Create subscription`.
- `Details`.
 - Topic ARN: arn:aws:sns:us-east-1:046402772087:Demo-topic (comes default)
 - Protocol: Email
 - Endpoint: test@example.com (your mail here)
- Keep rest default.
- Click `Create subscription`.
- Show `Status` >> `Pending Confirmation`.

Create subscription

Details

Topic ARN
arn:aws:sns:us-east-1:547187538673:Demo-topic

Protocol
The type of endpoint to subscribe
Email

Endpoint
An email address that can receive notifications from Amazon SNS.
sersah95@gmail.com

After your subscription is created, you must confirm it. [Info](#)

Subscription filter policy - optional
This policy filters the messages that a subscriber receives. [Info](#)

Redrive policy (dead-letter queue) - optional
Send undeliverable messages to a dead-letter queue. [Info](#)

[Cancel](#) [Create subscription](#)

Demo-topic

[Edit](#) [Delete](#) [Publish message](#)

Details

Name Demo-topic	Display name My-First-Topic
ARN arn:aws:sns:us-east-1:547187538673:Demo-topic	Topic owner 547187538673
Type Standard	

[Subscriptions](#) [Access policy](#) [Delivery retry policy \(HTTP/S\)](#) [Delivery status logging](#) [Encryption](#) [Tags](#)

Subscriptions (1)

ID	Endpoint	Status	Protocol
Pending confirmation	sersah95@gmail.com	Pending confirmation	EMAIL

[Edit](#) [Delete](#) [Request confirmation](#) [Confirm subscription](#) [Create subscription](#)

Maile gidip confirmation aciyoruz

AWS Notification - Subscription Confirmation [Gelen Kutusu](#) [X](#)

My-First-Topic <no-reply@sns.amazonaws.com>
Alici: ben

You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:547187538673:Demo-topic

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)

[Yanıtlala](#) [Yönlendir](#)

Confirm subscription yapı

Step 3 : Confirm subscription

- Go to your mail and check inbox.
- Open mail from `My-First-Topic`.
- Click `Confirm subscription`.
- Go back to Demo-topic subscription and refresh the page.
- Show `Status` >> `Confirmed`.

Step 4 : Publish message

- Select `Topics` from the left-hand menu and click on `Demo-topic`.
- Click `Publish message`.
- `Message details`:
 - Subject: sns-test
 - Time to Live (TTL) : -
- `Message body`:
 - Message structure: Identical payload for all delivery protocols.
 - "This is a test message for sns inclass session".
- Keep rest default.
- Click `Publish message`.
- Go to your mail and check inbox.
- Open mail from `My-First-Topic`.
- Show the topic and the test message sent from SNS.

Amazon SNS > Topics > Demo-topic > Publish message

Publish message to topic

Message details

Topic ARN
arn:aws:sns:us-east-1:547187538673:Demo-topic

Subject - optional
sns-test
Maximum 100 printable ASCII characters

Time to Live (TTL) - optional
This setting applies only to mobile application endpoints. The number of seconds that the push notification service has to deliver the message to the endpoint. [Info](#)

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
1 this message is for SNS testing

En alttta publish yapıyoruz

Mailimizi kontrol ediyoruz

Mesaj endpoint suscription olarak mail adresini verdigiiz icin oraya gitmis

SNS-test Gelen Kutusu x

My-First-Topic <no-reply@sns.amazonaws.com>
Alici: ben ✓
21:45 (0 dakika önce)

İngilizce ▾ > Türkçe ▾ İlettiştirme çevir İngilizce

this is test message SNS in class session

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
<https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:547187538673:Demo-topic-d473ac84-eb57-4655-bfe3-a32239fe651b&Endpoint=sersah95@gmail.com>

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>

Yanıtla Yönleendir

Part 2 - Creating a CloudWatch Event to Invoke SNS

Step 1 : Create Rule

- Go to `CloudWatch` service on AWS console.
- Click `Events` >> `Rules` from the left-hand menu.

console.aws.amazon.com/cloudwatch/home?region=us-east-1#rules

CloudWatch Services ▾ Q Search for services, features, marketplace products, and docs [Alt+S] Admin_user @ ser007 N. Virginia Support

CloudWatch Events is now Amazon EventBridge

Amazon EventBridge (formerly CloudWatch Events) provides all functionality from CloudWatch Events and also launched new features such as Custom event buses, 3rd party event sources and Schema registry to better support our customers in the space of event-driven architecture and applications.

Amazon EventBridge documentation

Rules

Rules route events from your AWS resources for processing by selected targets. You can create, edit, and delete rules.

Create rule Actions ▾

Status	All	Name	Description
<input type="radio"/>		Event_Start	This event provides start action.
<input type="radio"/>		Event_stop	This event provides stop action.

Viewing 1 to 2 of 2 Rules

Feedback English (US) ▾ © 2020-2021 Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookies Preferences

- Click `Create Rule`.
- `Event Source`
 - Event Pattern

- Service Name : EC2
- Event Type : EC2 Instance State-change Notification

Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

Event Pattern Schedule

Build event pattern to match events by service

Service Name: EC2

Event Type: EC2 Instance State-change Notification

Any state Specific state(s)

Any instance Specific instance Id(s)

Event Pattern Preview

```
{
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "EC2 Instance State-change Notification"
  ]
}
```

Targets

Select Target to invoke when an event matches your Event triggered.

Add target*

- `Targets` >> Add target
 - SNS topic
 - Topic : Demo-topic
- Click `Configure details`.
 - Name: Instance-State-Change
 - Keep rest default.
- Click `Create Rule`.

CloudWatch

Success
Rule **instance-state** was created.

CloudWatch Events is now Amazon EventBridge
Amazon EventBridge (formerly CloudWatch Events) provides all functionality from CloudWatch Events and also launched new features such as Custom event buses, 3rd party event sources and Schema registry to better support our customers in the space of event-driven architecture and applications.
[Amazon EventBridge documentation](#)

Rules

Rules route events from your AWS resources for processing by selected targets. You can create, edit, and delete rules.

Create rule		Actions	
Status	All	Name	
<input type="radio"/>	<input checked="" type="radio"/>	Event_Start	This event provides start action.
<input type="radio"/>	<input checked="" type="radio"/>	Event_stop	This event provides stop action.
<input type="radio"/>	<input checked="" type="radio"/>	instance-state	

Step 2 : Invoke SNS

- Go to `EC2` service on AWS console.
- Change state of any available instance like starting a stopped one (Launch a new one if you don't have any).
- Go to your mail and check inbox.
- Open mail from `My-First-Topic`.
- Show the topics and the messages sent from SNS.
- Delete/terminate the resources created.

Mailimize gittigimizde cloudwatchden bize mesajlarin geldigini goruyoruz

1.767 İleti dizisinden 1.

AWS Notification Message [Gelen Kutusu](#)

My-First-Topic <no-reply@sns.amazonaws.com>

Alici: ben

22:12 (4 dakika önce)

İngilizce [Türkçe](#) [İletyi çevir](#) [İngilizce için kapat](#)

{"version":"0","id":"6ced1920-92cd-3624-427b-77e80d412648","detail-type":"EC2 Instance State-change Notification","source":"aws.ec2","account":"547187538673","time":"2021-08-18T19:12:12Z","region":"us-east-1","resources":["arn:aws:ec2:us-east-1:547187538673:instance|i-087d2d3152c6e455f"],"detail":{"instance-id": "i-087d2d3152c6e455f","state": "stopping"}}

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe.
<https://sns.us-east-1.amazonaws.com/jnsubscrib.htm?SubscriptionArn=arn:aws:sns:us-east-1:547187538673:Demo-topic:d473ac84-ab57-4655-bfe3-a32239fe651b&Endpoint=sersah95@gmail.com>

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>

My-First-Topic <no-reply@sns.amazonaws.com>

Alici: ben

22:12 (4 dakika önce)

İngilizce [Türkçe](#) [İletyi çevir](#) [İngilizce için kapat](#)

My-First-Topic <no-reply@sns.amazonaws.com>

Alici: ben

22:12 (3 dakika önce)

İngilizce [Türkçe](#) [İletyi çevir](#) [İngilizce için kapat](#)

{"version":"0","id":"7644edb7-0570-4834-c8e0-606b928820b6","detail-type":"EC2 Instance State-change Notification","source":"aws.ec2","account":"547187538673","time":"2021-08-18T19:12:35Z","region":"us-east-1","resources":["arn:aws:ec2:us-east-1:547187538673:instance|i-087d2d3152c6e455f"],"detail":{"instance-id": "i-087d2d3152c6e455f","state": "stopped"}}

My-First-Topic <no-reply@sns.amazonaws.com>

Alici: ben

22:12 (3 dakika önce)

İngilizce [Türkçe](#) [İletyi çevir](#) [İngilizce için kapat](#)



THANKS!

Any questions?



CLARUSWAY
WAY TO REINVENT YOURSELF

23