



# AWS SQS & SNS

## 1 SQS

Amazon Simple Queue Service (SQS), dağıtılmış sistemleri ve sunucusuz uygulamaları birbirinden ayırmaya ve ölçeklendirmenize imkan tanıyor, tam olarak yönetilen bir iletileri kuyruğa alma hizmetidir.

SQS, mesajlaşmaya yönelik ara yazılımları yönetmenin ve işletmenin getirdiği karmaşıklık ile ek iş yükünü ortadan kaldırarak geliştiricilerin farklı işlere odaklanmasına imkan tanır. SQS ile ileti kaybı yaşamadan veya diğer hizmetlerin erişilebilir olmasına gereksinim duymadan yazılım bilesenleri arasında dilediğiniz hacimde ileti gönderebilir, depolayabilir ve alabilirsiniz. AWS konsolunu, tercih ettiğiniz Komut Satırı Arabirimini veya SDK'yi ve üç basit komutu kullanarak SQS'yi dakikalar içinde kullanmaya başlayabilirsiniz. SQS iki tür ileti kuyruğu sunar. Standart kuyruklar tarafından en yüksek aktarım hızı, en iyi çaba ilkesine göre sıralama ve en az bir kez teslim olanakları sunulur. SQS FIFO kuyrukları, iletilerin tam olarak bir kez ve tam olarak gönderildikleri sırada işlenmesi konusunda güvence sağlayacak şekilde tasarlanmıştır.==>  
<https://aws.amazon.com/tr/sqs/> SQS servisi Elastic Load Balancer' a bir alternatif servis diyebiliriz

Amazon Simple Notification Service Fully managed pub/sub messaging, SMS, email, and mobile push notifications From 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

## ▶ SQS 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.

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

A2A pub/sub işlevi; dağıtılmış sistemler, mikro hizmetler ve olay tabanlı sunucusuz uygulamalar arasında yüksek aktarım hızlı, gönderme tabanlı, çok çok mesajlaşmaya yönelik konular sunar. Yayımcı sistemleriniz, Amazon SNS konularını kullanarak mesajları paralel işleme için Amazon SQS kuyrukları, AWS Lambda işlevleri, HTTPS uç noktaları ve Amazon Kinesis Data Firehose gibi çok sayıda abone sisteme dağıtabilir. A2P işlevi kullanıcılarına SMS, mobil anlık bildirimler ve e-posta yoluya uygun ölçüde mesajlar göndermenizi sağlar.

From <<https://aws.amazon.com/tr/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-&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 çoğu mesajlaşma için konular sağlar

**From** 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 at scale via SMS, mobile push, and email.

**From 1 million requests free**

**the AWS Free Tier Get started for free » From**

FCBarcelona web platformı 6000'den fazla sayfaya ve 12.000'den fazla dijital fotoğrafa ev sahipliği yapmaktadır. Altı dilde sunulan bu sayfa beş elit profesyonel takımla ilgili güncel bilgileri sunmaktadır. Çözüm birden fazla platform bildirimi için Amazon SNS hizmetini kullanmaktadır.

From <<https://aws.amazon.com/tr/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort=&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 (SIMPLE NOTIFICATION SERVICE)** Wednesday, 18 August 2021 18.48 AWS Page 1  
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

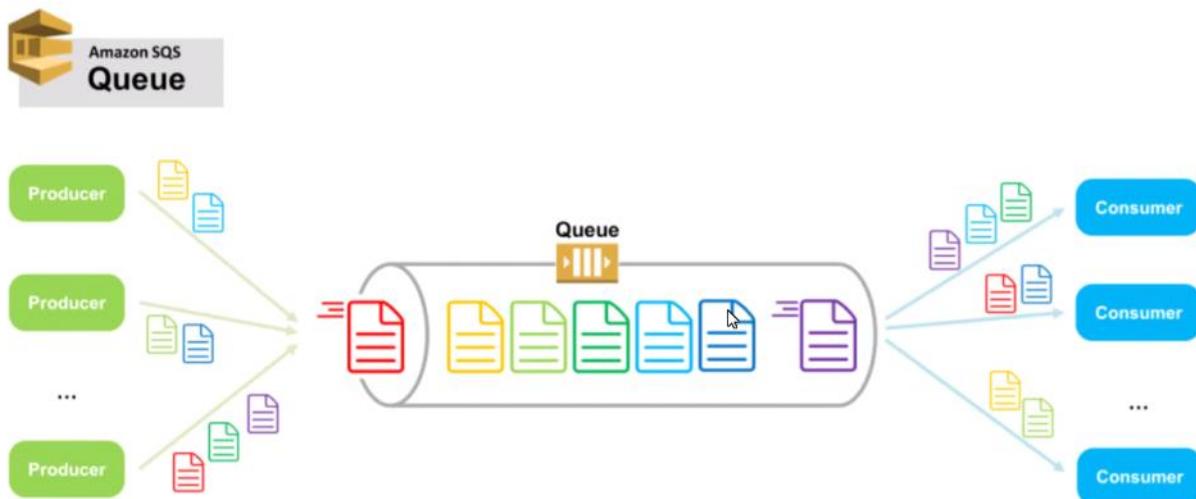
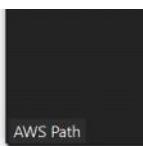
Amazon SNS FIFO topics work with Amazon SQS FIFO queues to ensure messages are delivered in a strictly ordered manner and are only processed once (deduplicated). This enables you to maintain consistency when processing transactions across a single or multiple independent services where it's critical 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

How it works From

Producer	Client, application, mesaj gönderen
Consumer	Mesajı işleyen taraf

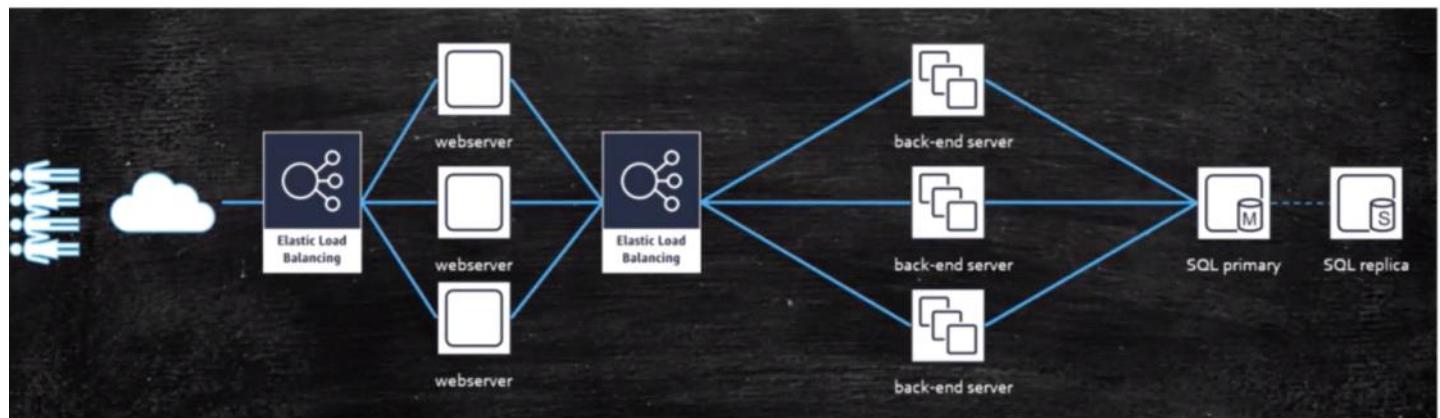
# SQS

## What is SQS?



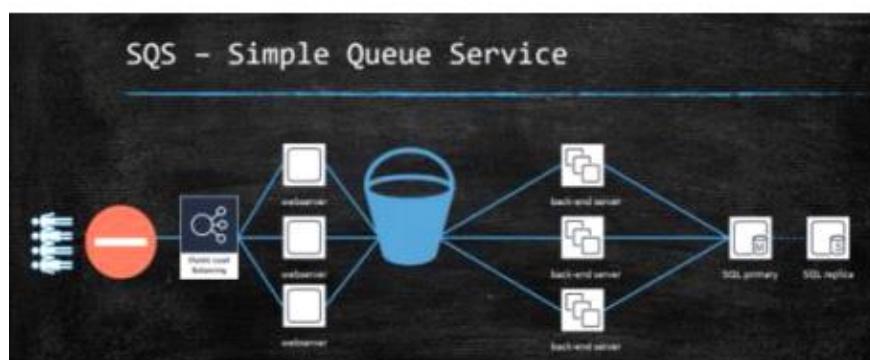
ARUSWAY  
TO REINVENT YOURSELF

## What is SQS?



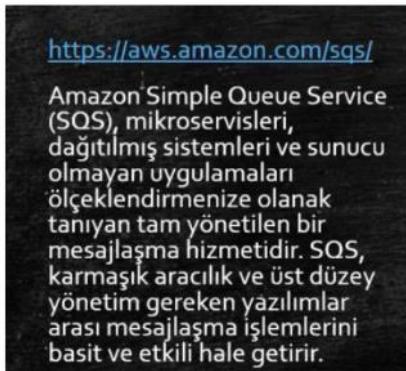
Online otel hizmeti Kullanıcılar front end e baglanıp rezervasyon işlemi ve kredi kartı işlemleri yapacak. Front end back end e yönlendirecek. Back end de cevap verecek mesajları front ende diğer kısmını da database ye gönderecek Normalde arada load balancerlar var ve yükü dengelemektedir. Ya yük artar ve load balancer gibi dengeleyicilerde yükü karşılayamazlarsa. Yük artarsa front end talepleri back endde ulaşamayacak ve işlem gerçekleşemedi hatası dener

Decouple mantığında sqs kullanıyoruz On ve arka taraftaki backend serverler birbirleriyle ilişkili

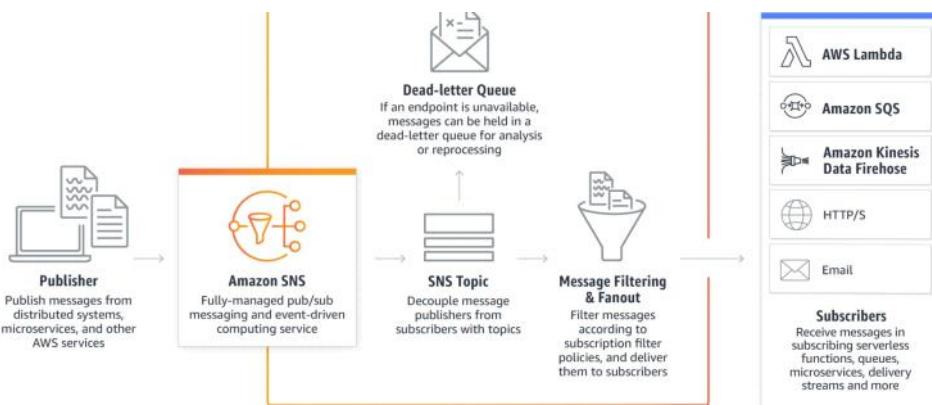
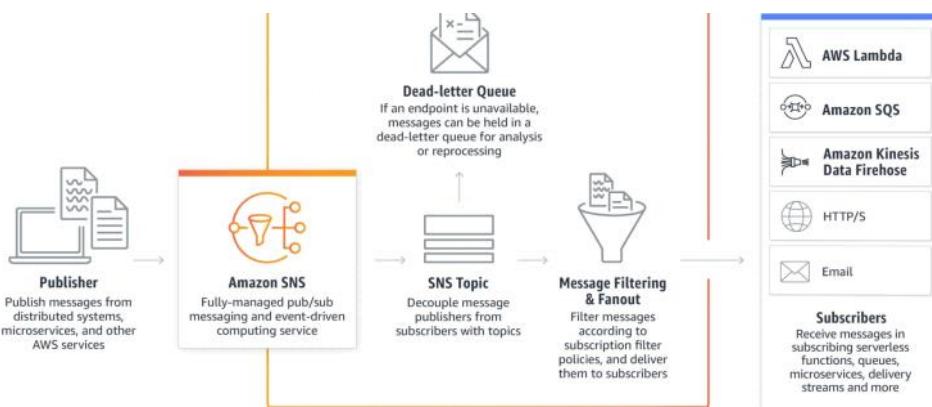


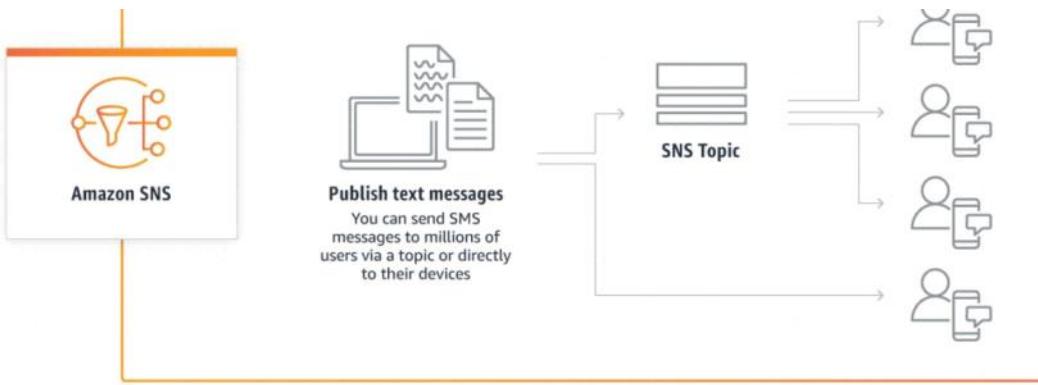
SQS yukarıdaki gorseldeki gibi gelen mesajları beklettiği bir kova sistemi uygulamaktadır. Ve musteriye işleminiz yapılıyor bir sure bekleyiniz şeklinde bilgilendirme yapılmaktadır (artık musteriye hata mesajı gitmemektedir).

SQS mesajları 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 ayıiyor. Fault tolerance i azaltıyoruz bu mimariyle.

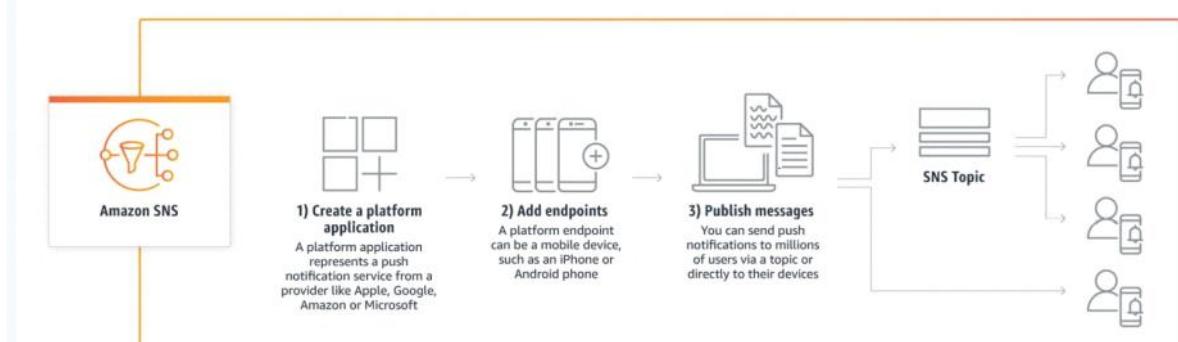


Yine ASG ile SQS in içinde visible olan metric e gore backend serverların sayisini artirip azaltabiliyoruz What is decouple SQS? SQS lets you decouple application components so that they run and fail independently, increasing the overall fault tolerance of the system. Multiple copies of every message 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 tanıyor sistem genel hata toleransını artırır. Her mesaj



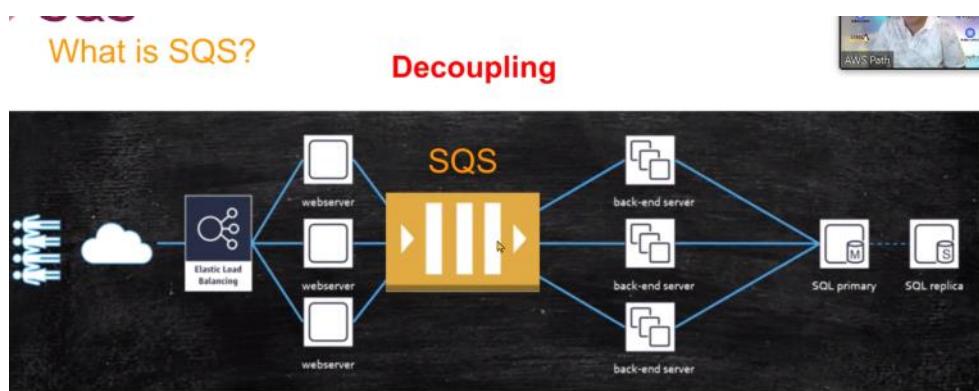


#### MOBILE PUSH



#### Amazon Simple Notification Service (Amazon SNS)

From <<https://app.slack.com/client/T0227UVRJU8/C021BG84YIJ/thread/C025D0P6R7B-1629234556.001700>>



Load balancer yerine SQS eklenmis oldu ve belli bir surelige mesajları muhafaza edebiliriz (14 gün)

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.



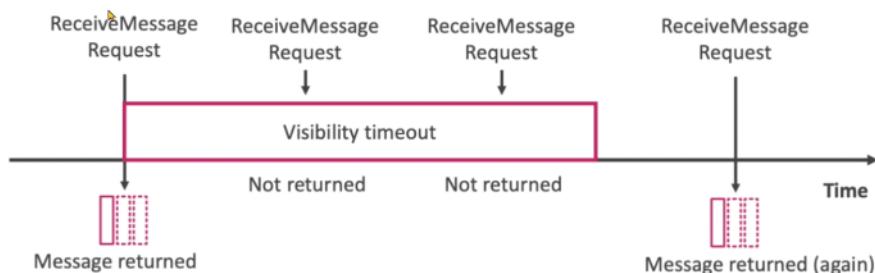
**Amazon Simple Queue Service (SQS)**, dağıtılmış sistemleri ve sunucusuz uygulamaları birbirinden ayırmaya ve ölçeklendirmeye imkan tanıyan, tam olarak yönetilen bir iletileri kuyruğa alma hizmetidir. SQS, mesajlaşmaya yönelik ara yazılımları yönetmenin ve işletmenin getirdiği karmaşıklık ile ek iş yükünü ortadan kaldırarak geliştiricilerin farklı işlere odaklanmasına imkan tanır. SQS ile ileti kaybı yaşamadan veya diğer hizmetlerin erişilebilir olmasına gereksinim duymadan yazılım bileşenleri arasında dilediğiniz hacimde ileti gönderebilir, depolayabilir ve alabilirsiniz. AWS konsolunu, tercih ettiğiniz Komut Satırı Arayüzü veya SDK'yi ve üç basit komutu kullanarak SQS'yi dakikalar içinde kullanmaya başlayabilirsiniz.

SQS iki tür ileti kuyruğu sunar. Standart kuyruklar tarafından en yüksek aktarım hızı, en iyi çaba ilkesine göre sıralama ve en az bir kez teslim olanakları sunulur. SQS FIFO kuyrukları, iletlerin tam olarak bir kez ve tam olarak gönderildikleri sırada işlenmesi konusunda güvence sağlayacak şekilde tasarlanmıştır.

From <<https://aws.amazon.com/tr/sqs/>>

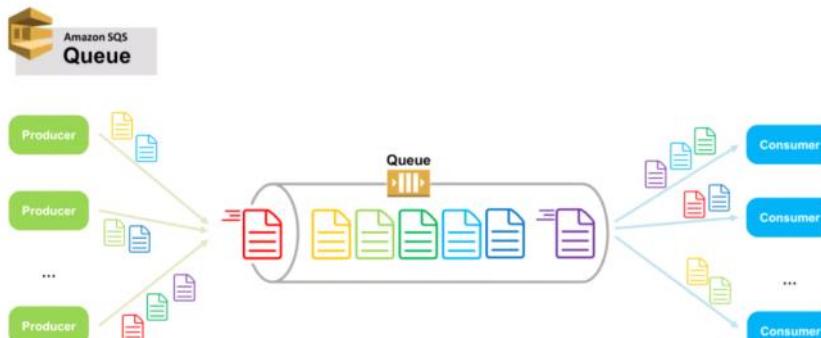
## ▶ SQS

### Message Visibility Timeout



Gelen mesaj consumer tarafından işleniyor. Aynı mesajın diğer consumer tarafından da işlenmesini engellenmek için message visibility time out ayarlanıyor. Bu sure içinde diğer consumerlar mesajı görmüyor

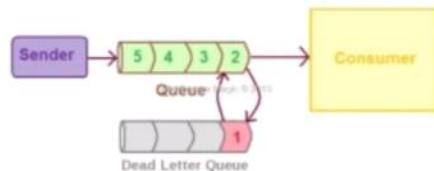
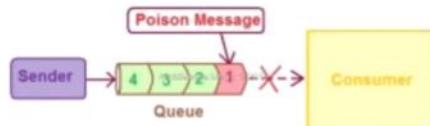
## What is SQS?



Decouple onemli konu sınavlarda çıkabilir

# SQS

## Dead Letter Queue (DLQ)

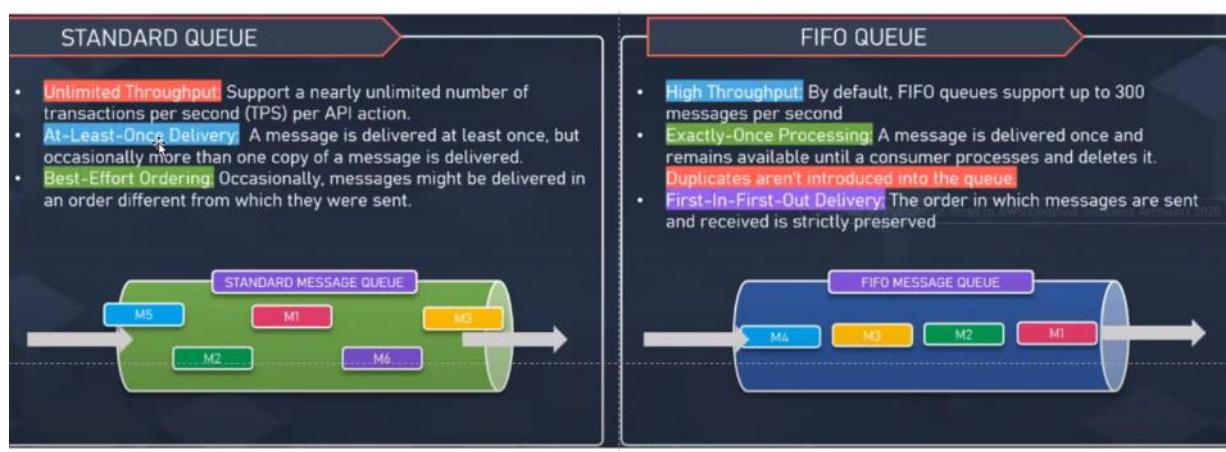


CLARUSWAY

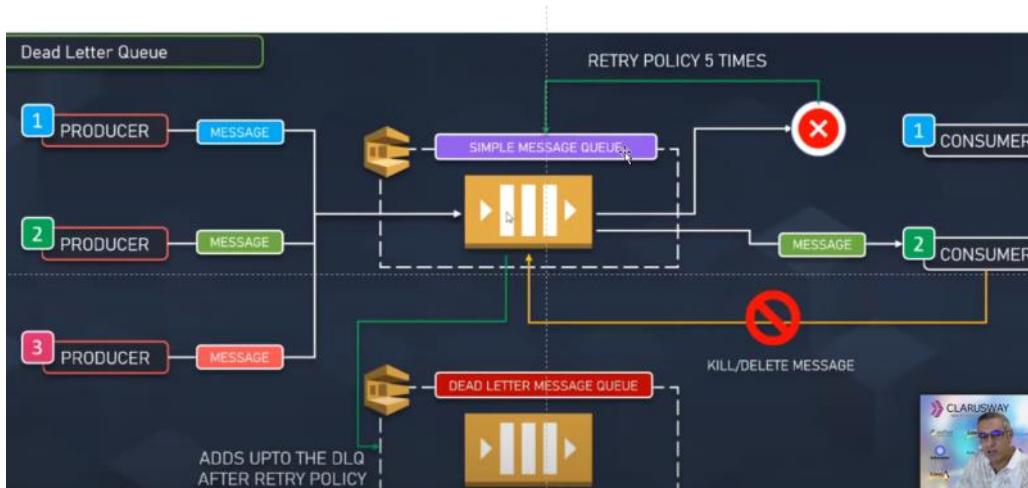
What is decouple SQS?

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.

From <<https://app.slack.com/client/T0227UVRJU8/C021BG84YJJ/thread/C025D0P6R7B-1629234556.001700>>



Saniyeeki işlem sayısında Standart q da limit yok. Ancak FIFO Q da hightthroughput dedigimiz 300 sınırlaması var, bu şebeke kullanılarak  $300 \times 10 = 3000$  mesaj e cıkırılıyabilir, Standart q da sıraya giren mesajın sırası göre olağanüstü olmamalıdır, consumerlar için mesajın sırası önemli değil. Mesaj birden fazla process işlem görebiliyor Standart q da. FIFO da duplicate diye bir kavram yok. Standart q da olabiliyor. FIFO DA İLK GİREN MESAJ KİMDİR CİKTIYOR



DLQ standart Q nun aynisi. Consumer gelen mesaji islemee calisti ancak isleyemedi tekrar Q ya gonderiyor consumer deniro belli bir sayıda bizim ayarladigimiz sayıda. Olmazsa DLQ ye gonderiliyor. Dah asonra troubleshooting yaparken içindeki mesajlara bakılabilir. HATALI MESAJLARIN GONDERILDIGI YER. STANDART q nun icinde mesajlar 14 gune kadar duruyor. Death LQ nin standarddan daha fazla olması lazımkı buraya gonderilen hata mesajını daha sonra inceleyebilelim

Sqs in un limited bir kapasitesi var 14 gune kadar bir kapasitesi var

Amazon SQS, ileti kuyruklarını yönetmek için kendi yazılımınızı derlemeye veya geliştirmeye ve yapılandırma için önemli hazırlık süresi gerektiren ücretli ya da açık kaynaklı ileti kuyruğa alma sistemlerini kullanmaya göre çok sayıda avantaj sağlar.

Bu alternatif sürekli donanım bakımı ve sistem yönetim kaynakları gerektirir. Bu sistemleri yapılandırma ve yönetmenin karmaşıklığı, donanımın arıza yapması durumunda iletelerin kaybedilmemesini sağlayan yedek ileti deposu gereksinimi ile birleştir.

Buna karşılık, Amazon SQS idari bir yük gerektirmez ve çok az yapılandırma içerir. Amazon SQS, çok büyük ölçekte çalışır ve bir günde milyarlarca mesajı işler. Herhangi bir yapılandırma olmadan Amazon SQS'ye gönderdiğiniz trafik miktarının ölçüğini artırıp azaltabilirsiniz. Amazon SQS, ayrıca son derece yüksek ileti dayanıklılığı sağlayarak size ve paydaşlarınıza daha fazla güven verir.

From <<https://aws.amazon.com/tr/sqs/faqs/>>



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

CLAPICIAZ

Application Integration

## 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.

#### Visibility timeout

The default visibility timeout setting is 30 seconds.

Message retention period Info Days Should be between 1 minute and 14 days.

TURKISH

Mesaj saklama süresi Bilgi Günler 1 dakika ile 14 gün arasında olmalıdır.

#### Delivery delay

TURKISH

Teslim gecikmesi

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.

TURKISH

Kuyruğa gönderdiğiniz tüm iletiler, gecikme süresi boyunca tüketicilere görünmez kalır. Bir kuyruk için varsayılan (minimum) gecikme 0 sanjedir. Maksimum 15 dakikadır.

## ----Hands -on !!!

### • Let's go to the AWS Management Console

### # SNS Hands-on

#### ## Part 1 - Creating Topic, Subscription and Publishing Message

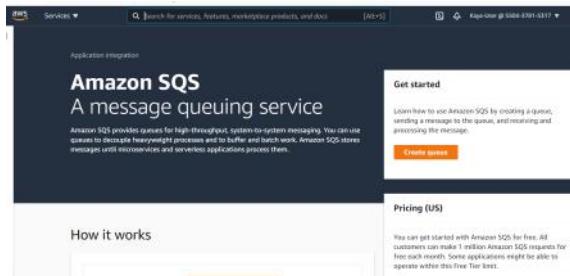
##### ### Step 1 : Create Topic

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

-----

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' .



Delay queues let you postpone the delivery of new messages to a queue for a number of seconds, for example, when your consumer application needs additional time to process messages. If you create a delay 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 information about configuring delay queues using the console see [Configuring queue parameters \(console\)](#).

FrOM < <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-delay-queues.html>>

**Sayısal değerler önemli sınavda sorulabilir**

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

---

**Access policy**  
 Define who can access your queue. [Info](#)

Choose method

<input checked="" type="radio"/> Basic Use simple criteria to define a basic access policy.	<input type="radio"/> Advanced Use a JSON object to define an advanced access policy.
--	--

---

Access policy  
 Define who can access your queue. [Info](#)

Choose method

<input checked="" type="radio"/> Basic Use simple criteria to define a basic access policy.	<input type="radio"/> 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)

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__owner_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "550437815317"
      },
      "Action": [
        "SQS:*"
      ],
      "Resource": "arn:aws:sqs:us-east-1:550437815317:My-First-Queue"
    }
  ]
}
```

Manuel olarak mesaj gönderecegiz

Amazon SQS > Queues > My-First-Queue

**My-First-Queue**

[Edit](#) [Delete](#) [Purge](#) [Send and receive messages](#)

<b>Details</b> <a href="#">Info</a>		
Name <input checked="" type="checkbox"/> My-First-Queue	Type <a href="#">Standard</a>	ARN <a href="arn:aws:sqs:us-east-1:550437815317:My-First-Queue">arn:aws:sqs:us-east-1:550437815317:My-First-Queue</a>
Encryption	URL <a href="https://sqs.us-east-1.amazonaws.com/550437815317/My-First-Queue">https://sqs.us-east-1.amazonaws.com/550437815317/My-First-Queue</a>	Dead-letter queue
<a href="#">More</a>		
<a href="#">SNS subscriptions</a> <a href="#">Lambda triggers</a> <a href="#">Dead-letter queue</a> <a href="#">Monitoring</a> <a href="#">Tagging</a> <a href="#">Access policy</a> <a href="#">Encryption</a>		

- ### Step 2 : Send Message**
- 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.

## Send and receive messages

Send messages to and receive messages from a queue.

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

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

> Message attributes - Optional Info

Send message Info Clear content View details X

Poll for messages

**Menusune basinca bir tane mesaj geldigini gorebiliriz**

**Menusune basinca bir tane mesaj geldigini gorebiliriz**

**Poll ==> mesaji request sayesinde consumer (aliciya) a gonderiliyor yani cekiliyor**

### ### 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.

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

> Message attributes - Optional Info

Receive messages Info Edit poll settings Stop polling Poll for messages

Messages available	Polling duration	Maximum message count	Polling progress
0	30	10	57% 3 receives/second

**Messages (3)**

ID	Sent	Size	Receive count
231df626-0e3a-4ee8-b4b8-8f53dc45395e	8/19/2021, 02:49:21 GMT+3	14 bytes	2
76391e6e-2889-40b1-9d19-a59716a330ef	8/19/2021, 02:40:22 GMT+3	24 bytes	3
16381a0f-23af-46ad-8a8c-c1cfa5f313d	8/19/2021, 02:48:56 GMT+3	45 bytes	2

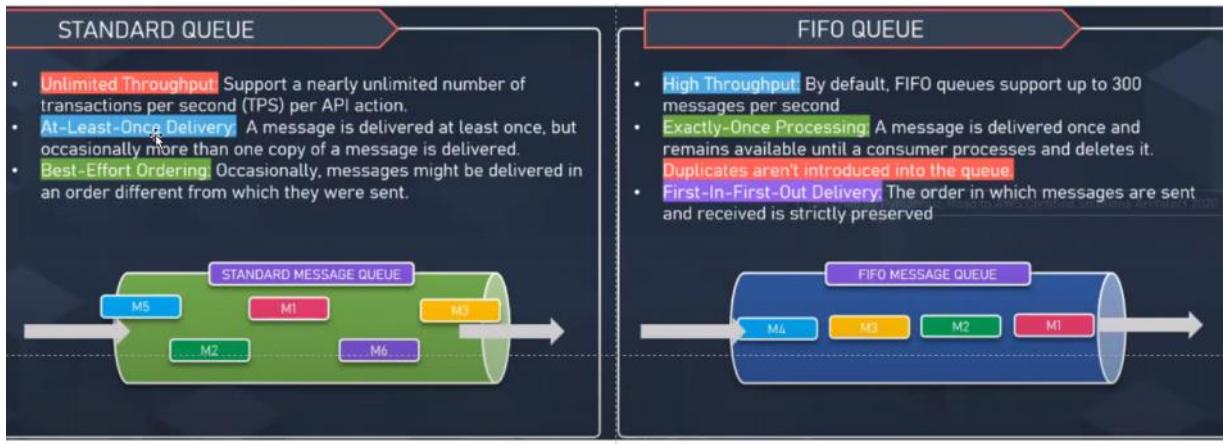
View details Delete

**Gönderdigim diger mesajlar**

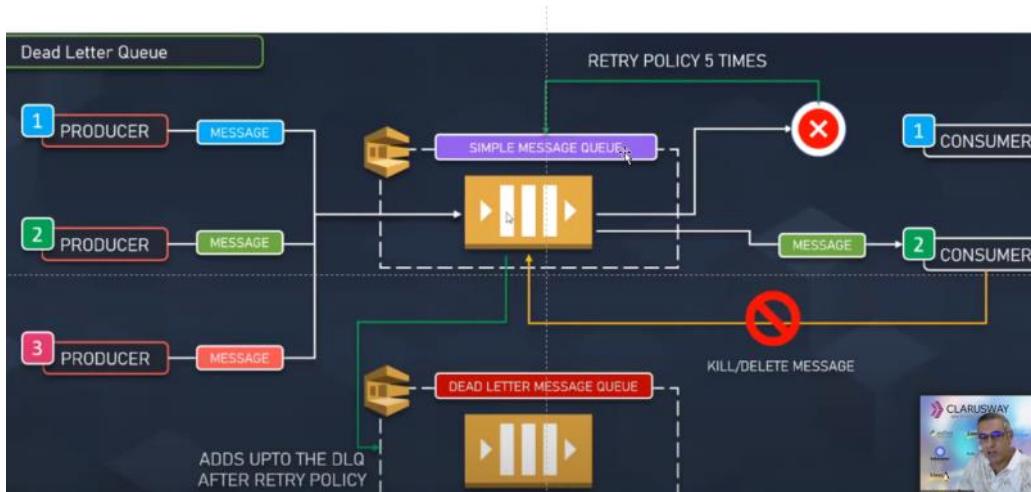
## Building LOosely Coupled, Scalable, C# Applications with Amazon SQS and Amazon SNS

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

From <<https://app.slack.com/client/T0227UVRJU8/C021BG84YIJ/thread/C025D0P6R7B-1629234556.001700>>



Saniyeki işlem sayısında Standart Q da limit yok. Ancak FIFO Q da hightthroughput dedigimiz 300 sınırlaması var, bu özelliği kullanırsak  $300 \times 10 = 3\,000$  mesajı 10 saniyede çırçırlıyor, Standart Q da sıraya giren mesajın sırası girise göre olmuyabiliyor, consumerlar için mesajın sırası önemli değil. Mesaj birden fazla process işlem görebiliyor Standart Q da. FIFO da duplicate diye bir kavram yok. Standart Q da olabiliyor. FIFO DA ILK GİREN MESAJK İLK ÇIKIYOR



DLQ standart Q'nun aynısı. Consumer gelen mesajı işlemee çalıştı ancak isleyemedi tekrar Q'ya gönderiyor consumer deneyip belli bir sayıda bizim ayarladığımız sayıda. Olmazsa DLQ'ye gönderiliyor. Dah asıra troubleshooting yaparken içindeki mesajlara bakılabilir. HATALI MESAJLARIN GÖNDERİLDİĞİ 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

SQS'in un limited bir kapasitesi var 14 güne kadar bir kapasitesi var

Amazon SQS, ileti kuyruklarını yönetmek için kendi yazılımınızı derlemeye veya geliştirmeye ve yapılandırma için önemli hazırlık süresi gerektiren ücretli ya da açık kaynaklı ileti kuyruğa alma sistemlerini kullanmaya göre çok sayıda avantaj sağlar.

Bu alternatif sürekli donanım bakımını ve sistem yönetim kaynakları gerektirir. Bu sistemleri yapılandırma ve yönetmenin karmaşaklılığı, donanımın arıza yapması durumunda iletelerin kaybedilmemesini sağlayan yedek ileti deposu gereksinimi ile birleştir.

Buna karşılık, Amazon SQS idari bir yük gerektirmez ve çok az yapılandırma içerir. Amazon SQS, çok büyük ölçekte çalışır ve bir günde milyarlarca mesaj işler. Herhangi bir yapılandırma olmadan Amazon SQS'ye gönderdiğiniz trafik miktarının ölçüğünü artırıp azaltabilirsiniz. Amazon SQS, ayrıca son derece yüksek ileti dayanıklılığı sağlayarak size ve paydaşlarınıza daha fazla güven verir.

From <<https://aws.amazon.com/tr/sqs/faqs/>>



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

## ► Application Integration

Application integration

# 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.

### Visibility timeout

The default visibility timeout setting is 30 seconds.

Message retention period Info Days Should be between 1 minute and 14 days.

TURKISH

Mesaj saklama süresi Bilgi Günler 1 dakika ile 14 gün arasında olmalıdır.

### Delivery delay

TURKISH

Teslim gecikmesi

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.

TURKISH

Kuyruğa gönderdiğiniz tüm iletiler, gecikme süresi boyunca tüketicilere görünmez kalır. Bir kuyruk için varsayılan (minimum) gecikme 0 saniyedir. Maksimum 15 dakikadır.

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

## ----Hands -on !!!

### • Let's go to the AWS Management Console

•

### # SNS Hands-on

#### ## Part 1 - Creating Topic, Subscription and Publishing Message

##### ### Step 1 : Create Topic

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

-----

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' .

The screenshot shows the AWS SQS homepage. At the top, there's a search bar and a user profile. Below the header, the main content area has a dark header "Amazon SQS" and a sub-header "A message queuing service". A paragraph explains what SQS does, mentioning high-throughput, system-to-system messaging. It includes a "Create queue" button. Below this, there's a "How it works" link.

The screenshot shows the "Create queue" wizard. Step 1: Details. It asks for the queue type. Two options are shown: "Standard" (selected) and "FIFO". Both have an "Info" link. The "Standard" option says "At-least-once delivery, message ordering isn't preserved" with points "At-least once delivery" and "Best-effort ordering". The "FIFO" option says "First-in-first-out delivery, message ordering is preserved" with points "First-in-first-out delivery" and "Exactly-once processing".

Delay queues let you postpone the delivery of new messages to a queue for a number of seconds, for example, when your consumer application needs additional time to process messages. If you create a delay 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 information about configuring delay queues using the console see [Configuring queue parameters \(console\)](#).

From < <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-delay-queues.html>>

### Sayısal değerler önemli sınavda sorulabilir

The screenshot continues the "Create queue" wizard. Step 1: Details. It shows configuration settings: "Visibility timeout" (30 seconds), "Message retention period" (4 days), "Delivery delay" (0 seconds), and "Maximum message size" (256 KB). Below each setting is a note about the allowed range. It also shows "Receive message wait time" (0 seconds) and "Access policy" (Basic).

**Access policy**  
Define who can access your queue. [Info](#)

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)

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__owner_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "550437815317"
      },
      "Action": [
        "SQS:*"
      ],
      "Resource": "arn:aws:sqs:us-east-1:550437815317:My-First-Queue"
    }
  ]
}
```

Manuel olarak mesaj göndereceğiz

Amazon SQS > Queues > My-First-Queue

**My-First-Queue**

[Edit](#) [Delete](#) [Purge](#) [Send and receive messages](#)

**Details** [Info](#)

Name	Type	ARN
My-First-Queue	Standard	arn:aws:sqs:us-east-1:550437815317:My-First-Queue
Encryption	URL	Dead-letter queue
-	<a href="https://sqs.us-east-1.amazonaws.com/550437815317/My-First-Queue">https://sqs.us-east-1.amazonaws.com/550437815317/My-First-Queue</a>	-
<a href="#">More</a>		
<a href="#">SNS subscriptions</a>   <a href="#">Lambda triggers</a>   <a href="#">Dead-letter queue</a>   <a href="#">Monitoring</a>   <a href="#">Tagging</a>   <a href="#">Access policy</a>   <a href="#">Encryption</a>		

### ### Step 2 : Send Message

- 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-Queue > Send and receive messages

**Send and receive messages**

Send messages to and receive messages from a queue.

[Send message](#) [Info](#) [Clear content](#) [Send message](#)

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 message

Delivery delay [Info](#)  Seconds [▼](#)

Should be between 0 seconds and 15 minutes.

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

[Poll for messages](#)

Menüsünde basınca bir tane mesaj geldigini görebiliriz

Menüsünde basınca bir tane mesaj geldigini görebiliriz

Poll ==> mesajı request sayesinde consumer (alıcıya) gönderiliyor yani çekiliyor

### ### 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.

ID	Sent	Size	Receive count
231df626-0e3a-4ee8-b4b8-8f53dc45395e	8/19/2021, 02:49:21 GMT+3	14 bytes	2
76391e6e-2889-40b1-9d19-a59716a330ef	8/19/2021, 02:40:22 GMT+3	24 bytes	3
16381a0f-23af-46ad-8a8c-c1c6a55f313d	8/19/2021, 02:48:56 GMT+3	45 bytes	2

Gönderdigim diger mesajlar

## Building LOOsely Coupled, Scalable, C# Applications with Amazon SQS and Amazon SNS

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

Delete menusu ile silebiliriz Bir tane Lambda function olusturacagiz

### ## Part 2 - Creating Lambda Function to Be Triggered by SQS

#### ### Step 1 : Create Lambda Function

- Go to `Lambda` service on AWS console.
- Click `Create function`.
- Select `Use a blueprint`.
- Search `sqrs` in the the blueprints search bar.
- Select `sqrs-poller` and click `Configure`.
- `Basic information`
  - Function name : sqrs-poller
  - Execution role : Create a new role from AWS policy templates (Keep default)
  - Role name : sqrs-poller-role
  - Policy templates : Keep the default "Amazon SQS poller permissions"

Aws nin bize hazir sundugu template/function kullanacagiz

## Sqs yazıp enter diyelim

Lambda > Functions > Create function

### Create function Info

Choose one of the following options to create your function.

- Author from scratch Start with a simple Hello World example.
- Use a blueprint Build a Lambda application from sample code and configuration presets for common use cases.
- Container image Select a container image to deploy for your function.
- Browse serverless repository Deploy a sample Lambda function from the AWS Serverless Application Repository.

#### Blueprints Info

Filter by tags and attributes or search by keyword  1 match

<b>sqs-poller</b> <input type="radio"/> An Amazon SQS trigger that logs messages in a queue.  nodejs · sqs
---

---

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

i 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 <input type="button" value="X"/>
--

---

SQS trigger

```
`SQS trigger`  
- SQS queue : My-First-Queue  
- Keep rest default.  
- Click 'Create function'.  
- Click `Configuration` tab under `sq-poller` function.  
- Select `SQS: My-First-Queue` and click `Enable` to enable trigger.(Wait for it to be "Enabled")
```

Permission otomatik geldi

## 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

## Yukariya tıkladığımızda my first queue otomatik geliyor

Amazon SQS poller permissions X

▼ | G

## SQS trigger

Remove

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

arn:aws:sqs:us-east-1:550437815317:My-First-Queue X G

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

10

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

0

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

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.`;
}
```

## Function olustur ve triger da sqs

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 X

Throttle Copy ARN Actions ▾

▼ Function overview Info

sq-s-poller- Layers (0) + Add destination

SQS + Add trigger

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

Last modified  
1 minute ago

Function ARN  
arn:aws:lambda:us-east-1:547187538675:function:sqs-poller

Code Test Monitor Configuration Aliases Versions

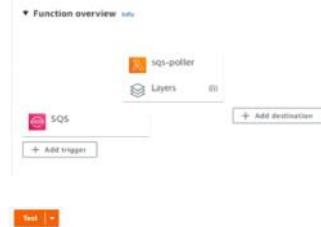
Code source Info

File Edit Find View Go Tools Window Test Deploy Changes deployed

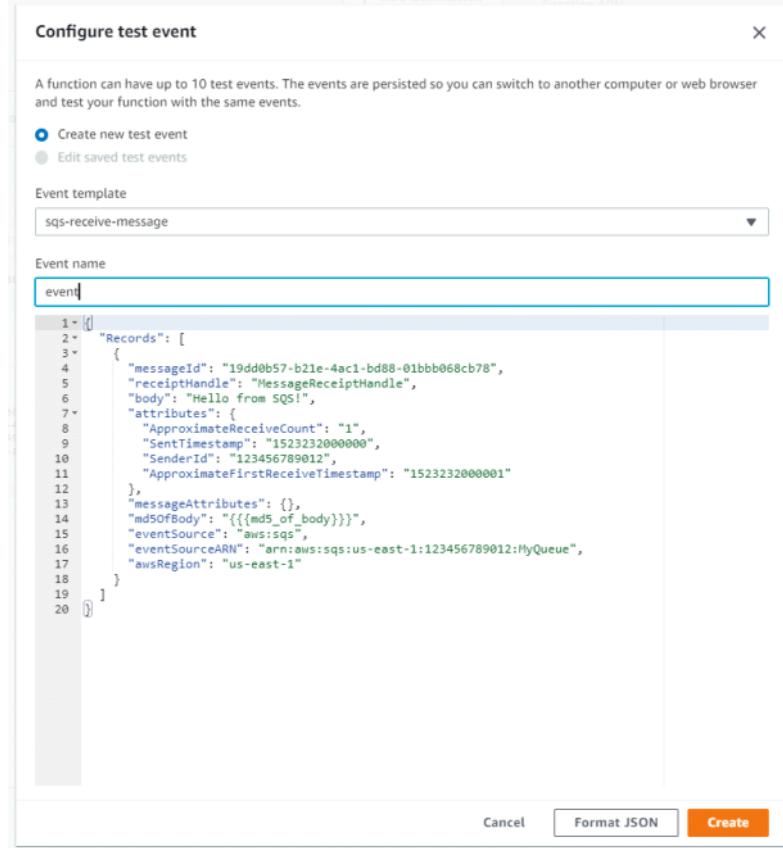
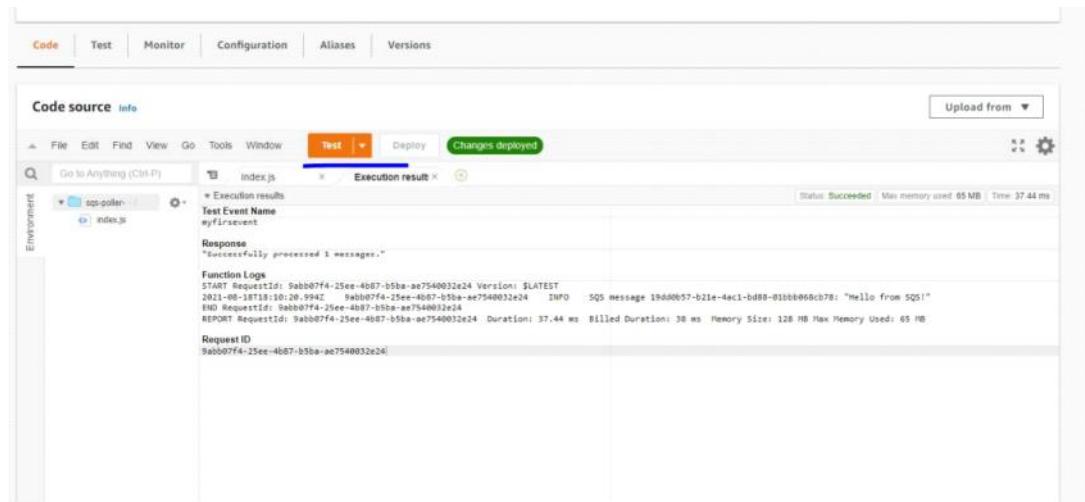
index.js

Upload from

Function olustur ve trigger da sqs



## TEST TEST ETTIK



The test event event was successfully saved.

The screenshot shows the AWS Lambda function configuration page. In the top navigation bar, the 'Test' tab is selected. Below it, the 'Execution result' tab is active. The results show a successful execution with the status 'Succeeded'. The response message is "Successfully processed 1 messages." The function logs detail the execution process, including the start and end request IDs, duration, and memory usage. The request ID is 0fe382f2-aa9c-4057-8404-a6496141be77.

### Create and test

The screenshot shows the AWS Lambda function configuration page. In the top navigation bar, the 'Test' tab is selected. Below it, the 'Execution result' tab is active. The results show a successful execution with the status 'succeeded'. The response message is "Successfully processed 1 messages." The function logs detail the execution process, including the start and end request IDs, duration, and memory usage. The request ID is 978ed3ad-c2c3-4794-84ce-8bb6517389ef.

### ## 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.

### Cloudwatch ekranina gidelim

The screenshot shows the AWS CloudWatch Logs interface. The left sidebar shows navigation options like CloudWatch, Favorites, Dashboards, Alarms, Logs, Log groups, Metrics, and Explorer. The 'Logs' section is expanded, and 'Log groups' is selected. The main area displays a table of log groups. There are three log groups listed: '/aws/lambda/sqs-poller', '/aws/lambda/start\_instance', and '/aws/lambda/stop\_instance'. Each log group has its retention period set to 'Never expire'.

Logs ---- Logs groups sekmesinin altında sqs-poller i görebiliyoruz

Amacımız : biz consumer yerine lambda yi configure ettik. Mesaj koyduğumuzda consumer yerine lambda mesajı cekerek ve cloudwatch e yazacak.

BMW 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.

Amazon SQS > Queues > My-First-Queue

My-First-Queue

Details info

Name: My-First-Queue Type: Standard ARN: arn:aws:sqs:us-east-1:556437815317:My-First-Queue

Encryption URL: https://sqs.us-east-1.amazonaws.com/556437815317/My-First-Queue Dead-letter queue:

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

Lambda triggers (1) Info

UUID	ARN	Status	Last modified
839a3c5a-9cab-450c-aafa-82be2f5d8ebd	arn:aws:lambda:us-east-1:556437815317:function:sqspoller	Enabled	8/19/2021, 3:10:22 AM

Amazon SQS > Queues > My-First-Queue > 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.

Hey mitä kulu hyvää kulu ihana

Delivery delay Info

0 Seconds

Should be between 0 seconds and 15 minutes.

Message attributes - Optional Info

Lambda da mesaj log kayıtlarını gorebiliriz

Send diyelim

Salesforce

Search for services, features, marketplace products, and docs [Alt+S]

Kaya-User @ 5504-3781-5317 N. Virginia Support

## CloudWatch

New menu experience

- Favorites
- Dashboards
- Alarms ▾ 0 2 0
- In alarm
- All alarms
- Billing
- Logs ▾
- Log groups
- Logs Insights
- Metrics ▾
- All metrics
- Explorer
- Streams
- Events ▾
- Rules
- Event Buses

CloudWatch > Log groups > /aws/lambda/sqs-poller > 2021/08/19/[...LATEST]e8c95c23de4141bcb412f101a986ac4d

### Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

View as text Actions Create Metric Filter

Filter events Clear 1m 30m 1h 12h Custom

Timestamp	Message
2021-08-19T03:31:36.764+03:00	No older events at this moment. <a href="#">Retry</a>
2021-08-19T03:31:36.764+03:00	START RequestId: 2d0f6c35-16c1-51c2-9114-35a2be66fb40 Version: \$LATEST
2021-08-19T03:31:36.764+03:00	2021-08-19T00:31:36.7632 undefined INFO Loading function
2021-08-19T03:31:36.773+03:00	2021-08-19T00:31:36.7712 2d0f6c35-16c1-51c2-9114-35a2be66fb40 INFO SQS message 25e28fe5-24ff-4f8a-95d0-d0178009c2de: "Hey mit..
2021-08-19T03:31:36.793+03:00	EID RequestId: 2d0f6c35-16c1-51c2-9114-35a2be66fb40
2021-08-19T03:31:36.793+03:00	REPORT RequestId: 2d0f6c35-16c1-51c2-9114-35a2be66fb40 Duration: 23.76 ms Billed Duration: 24 ms Memory Size: 128 MB Max Memo..
2021-08-19T03:33:59.527+03:00	START RequestId: 9a72bec9-51c1-5a77-85e6-e75efbf22bcb Version: \$LATEST
2021-08-19T03:33:59.573+03:00	2021-08-19T00:33:59.5732 9a72bec9-51c1-5a77-85e6-e75efbf22bcb INFO SQS message 7ca9a83b-de55-48b8-898b-7921e9dfa7f: "bejfb l..
2021-08-19T03:33:59.593+03:00	EID RequestId: 9a72bec9-51c1-5a77-85e6-e75efbf22bcb
2021-08-19T03:33:59.593+03:00	REPORT RequestId: 9a72bec9-51c1-5a77-85e6-e75efbf22bcb Duration: 63.67 ms Billed Duration: 64 ms Memory Size: 128 MB Max Memo..
2021-08-19T03:34:57.525+03:00	START RequestId: a6550603-f110-5b7c-bb10-5f94587ffefa Version: \$LATEST
2021-08-19T03:34:57.553+03:00	2021-08-19T00:34:57.5532 a6550603-f110-5b7c-bb10-5f94587ffefa INFO SQS message 1a5c87e8-8307-4a01-a516-7fab1a77adcc: "cbkjdb..
2021-08-19T03:34:57.554+03:00	EID RequestId: a6550603-f110-5b7c-bb10-5f94587ffefa
2021-08-19T03:34:57.554+03:00	REPORT RequestId: a6550603-f110-5b7c-bb10-5f94587ffefa Duration: 26.29 ms Billed Duration: 27 ms Memory Size: 128 MB Max Memo..
	No newer events at this moment. <a href="#">Auto retry paused.</a>

### ### 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`
- Click on the log stream and show the message sent from sqs and processed by lambda.
- Delete/terminate the resources created.



## SNS

What is SNS?



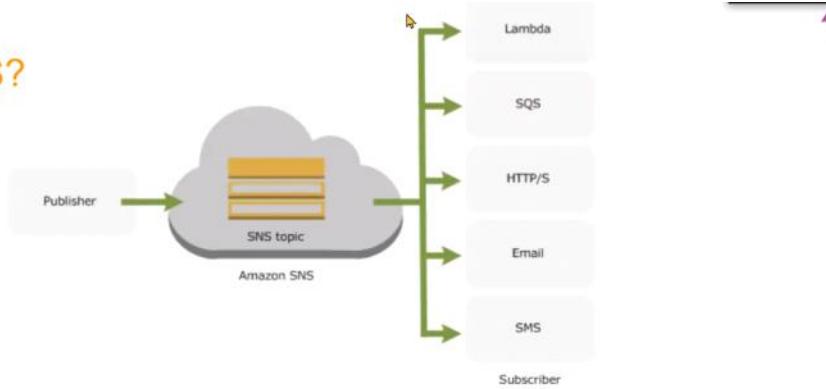
Amazon  
SNS

AWS Path

- Amazon **Simple Notification Service** (Amazon SNS) is a managed service that provides message delivery from **publishers** to **subscribers** (also known as **producers** and **consumers**).

## ► 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).

Cİ ARIUSWAY

>Aynı anda bir çok farklı yere aynı mesajı gönderebiliyoruz

## ► SNS

### What is SNS?

#### Application-to-Application (A2A)



## ► SNS

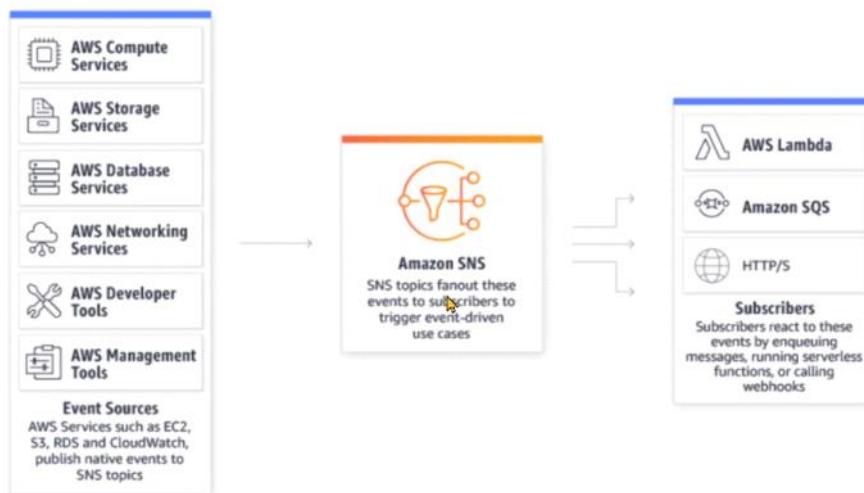
### What is SNS?

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.**

# SNS

## What is SNS?



Farklı protokolden gelen mesajları iletmesine denir

## ▶ SNS

### What is SNS?

### Application-to-Person (A2P)

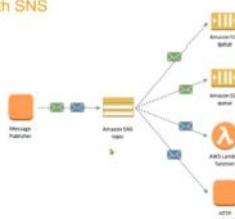


Burda application Bir topic de mesaj yazıyoruz lambda sqs e http s e gitiyor.

Bir topic oluşturuyoruz bu topic i birden fazla kullanıcıya gönderebiliyoruz veya birden farklı protokoldeki kişiye gönderebiliyoruz

## ▶ SNS

Fan Out with SNS



CLARUSWAY

## 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.

## SQS & SNS

Hands on

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

AWS CONSOLE

### SNS Hands-on

**## Part 1 - Creating Topic, Subscription and Publishing Message**  
**### Step 1 : Create Topic**

**- Go to `SNS` service on AWS console.**

**- Click `Topics` >> `Create topic`.**

The screenshot shows the AWS SNS Topics page. On the left, there's a sidebar with 'Amazon SNS' and links for 'Dashboard', 'Topics' (which is selected), and 'Subscriptions'. Below that is a 'Mobile' section with 'Push notifications', 'Text messaging (SMS)', and 'Origination numbers'. The main content area has a header 'Topics (4)'. It lists four topics: 'Billing-Alarm' (Standard, ARN: arn:aws:sns:us-east-1:550437815317:Billing-Alarm), 'My-Billing-Topic' (Standard, ARN: arn:aws:sns:us-east-1:550437815317:My-Billing-Topic), 'clarous-alarm' (Standard, ARN: arn:aws:sns:us-east-1:550437815317:clarous-alarm), and 'dynamodb' (Standard, ARN: arn:aws:sns:us-east-1:550437815317:dynamodb). There are buttons for 'Edit', 'Delete', 'Publish message', and 'Create Topic'.

**- `Details`.**

- **Type: Standard**
- **Name: Demo-topic**
- Display Name: My-First-Topic**
- **Keep rest default.**
- **Click `Create`.**

## 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 500 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

Maximum 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](#)

### ### 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-1:550437815317:Demo-topic

#### Protocol

The type of endpoint to subscribe

Email

#### Endpoint

An email address that can receive notifications from Amazon SNS.

kayakadir58067@gmail.com

i 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**

Amazon SNS > Topics > Demo-topic

## Demo-topic

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

Details	
Name Demo-topic	Display name My-First-Topic
ARN arn:aws:sns:us-east-1:550437815317:Demo-topic	Topic owner 550437815317
Type Standard	

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

Subscriptions (1)				
<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Request confirmation</a> <a href="#">Confirm subscription</a> <a href="#">Create subscription</a>				
ID	Endpoint	Status	Protocol	
Pending confirmation	kayakadir38067@gmail.com	Pending confirmation	EMAIL	<a href="#">Edit</a>

### ### Step 3 : Confirm subscription

- Go to your mail and check inbox.

### ### 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.

Cloudwatch ekranina gidelim

CloudWatch X

New menu experience

- Favorites
- Dashboards
- Alarms
- Logs
- Metrics

CloudWatch > Log groups

**Log groups (3)**  
By default, we only load up to 10000 log groups.

<input type="checkbox"/> Log group	Retenti...	Metric filters	Contributor Insights	Subscription ...
/aws/lambda/sqs-poller	Never expire	-	-	-
/aws/lambda/start_instance	Never expire	-	-	-
/aws/lambda/stop_instance	Never expire	-	-	-

Logs ---- Logs groups sekmesinin altında sqs-poller i görebiliyoruz

Amacımız : biz consumer yerine lambda yi configure ettik. Mesaj koyduğumuzda consumer yerine lambda mesajı cekerek ve cloudwatch e yazacak.

BMW 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.

Amazon SQS > Queues > My-First-Queue

### My-First-Queue

**Details** [Info](#)

Name	Type	ARN
My-First-Queue	Standard	arn:aws:sqs:us-east-1:550437815317:My-First-Queue
Encryption	URL	Dead-letter queue
<a href="#">More</a>		

SNS subscriptions | **Lambda triggers** [Info](#) | Dead-letter queue | Monitoring | Tagging | Access policy | Encryption

**Lambda triggers (1) [Info](#)**

UUID	ARN	Status	Last modified
839a3c5a-9cab-450c-aafa-82ne2f5d8bd	arn:aws:lambda:us-east-1:550437815317:function:sqs-poller	Enabled	8/19/2021, 3:10:22 AM

[View in Lambda](#) [Delete](#) [Configure Lambda function trigger](#)

Amazon SQS > Queues > My-First-Queue > 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.  
Hey mitä kulu hyvää kulu ihana

Delivery delay [Info](#)  
0 Seconds

Should be between 0 seconds and 15 minutes.

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

### Lambda da mesaj log kayitlarini gorebiliriz

#### Send diyelim

AWS Services ▾ [Search for services, features, marketplace products, and docs](#) [Alt+S] Kaya-User @ 5504-3781-5317 ▾ N: Virginia ▾ Support

**CloudWatch** [New menu experience](#)

- Favorites
- Dashboards
- Alarms [0](#) [2](#) [0](#)
- In alarm
- All alarms
- Billing
- Logs [Log groups](#)
- Metrics
- Events
- Application management

CloudWatch > Log groups > /aws/lambda/sqs-poller > 2021/08/19/[LATEST]e8c95c23de4141lcb412f101a986ac4d

**Log events**  
You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

View as text [Actions](#) [Create Metric Filter](#)

Filter events Clear 1m 30m 1h 12h Custom

Timestamp	Message
2021-08-19T03:31:36.764+03:00	START RequestId: 2d0ff6c35-16c1-51c2-9114-35a2be66fb40 Version: \$LATEST
2021-08-19T03:31:36.764+03:00	2021-08-19T03:31:36.763Z undefined INFO Loading function
2021-08-19T03:31:36.772+03:00	2021-08-19T03:31:36.771Z 2d0ff6c35-16c1-51c2-9114-35a2be66fb40 INFO SQS message 23e28fe5-24ff-4f8a-95d0-db178009c2de: "Hey mit..
2021-08-19T03:31:36.793+03:00	EID RequestId: 2d0ff6c35-16c1-51c2-9114-35a2be66fb40
2021-08-19T03:31:36.793+03:00	REPORT RequestId: 2d0ff6c35-16c1-51c2-9114-35a2be66fb40 Duration: 23.76 ms Billed Duration: 24 ms Memory Size: 128 MB Max Memo..
2021-08-19T03:33:59.527+03:00	START RequestId: 9a72bec9-51c1-5a77-85e6-e75ef8f22ccb Version: \$LATEST
2021-08-19T03:33:59.527+03:00	2021-08-19T03:33:59.523Z 9a72bec9-51c1-5a77-85e6-e75ef8f22ccb INFO SQS message 7ca9a83b-de55-48b8-898b-7921e9d3fa7f: "bejfb l..
2021-08-19T03:33:59.593+03:00	EID RequestId: 9a72bec9-51c1-5a77-85e6-e75ef8f22ccb
2021-08-19T03:33:59.593+03:00	REPORT RequestId: 9a72bec9-51c1-5a77-85e6-e75ef8f22ccb Duration: 63.67 ms Billed Duration: 64 ms Memory Size: 128 MB Max Memo..
2021-08-19T03:34:57.525+03:00	START RequestId: a6550603-f110-5b7c-bb10-5f94587fffea Version: \$LATEST
2021-08-19T03:34:57.525+03:00	2021-08-19T03:34:57.533Z a6550603-f110-5b7c-bb10-5f94587fffea INFO SQS message 1a5c87e8-8307-4a01-aa16-7fab1a77adcc: "cbkjdbc..
2021-08-19T03:34:57.554+03:00	EID RequestId: a6550603-f110-5b7c-bb10-5f94587fffea
2021-08-19T03:34:57.554+03:00	REPORT RequestId: a6550603-f110-5b7c-bb10-5f94587fffea Duration: 26.29 ms Billed Duration: 27 ms Memory Size: 128 MB Max Memo..
No newer events at this moment. Auto retry paused. <a href="#">Resume</a>	

### ### 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`

- Click on the log stream and show the message sent from sqs and processed by lambda.
- Delete/terminate the resources created.

▶

## 2 SNS

### SNS

#### What is SNS?

AWS Path

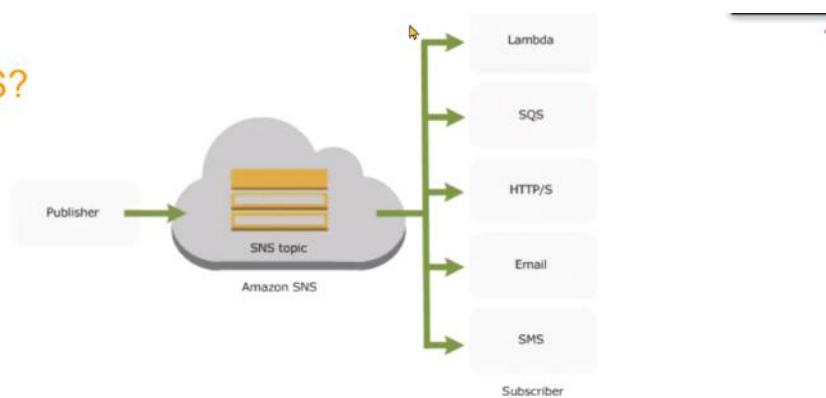


Amazon  
SNS

- Amazon **Simple Notification Service** (Amazon SNS) is a managed service that provides message delivery from **publishers** to **subscribers** (also known as **producers** and **consumers**).

### 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).

CI ARUSWAY

>Aynı anda bir çok farklı yere aynı mesajı gönderebiliyoruz

## SNS

### What is SNS?

Application-to-Application (A2A)



## 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.**

## SNS

### What is SNS?



Farklı protokolden gelen mesajları iletmesine denir

# SNS

## What is SNS?

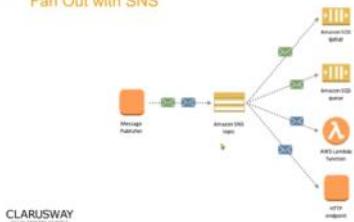
### Application-to-Person (A2P)



Burda application Bir topic de mesaj yaziyoruz sqs e http s e gidiyor.  
 Bir topic olusturuoruz bu topic i birden fazla kullanici ya gondereiliyoruz veya birden farkli protokoldeki kisye gonderebiliyoruz

## SNS

### Fan Out with SNS



## 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.

# SQS & SNS

## Hands on

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

AWS HOWTO

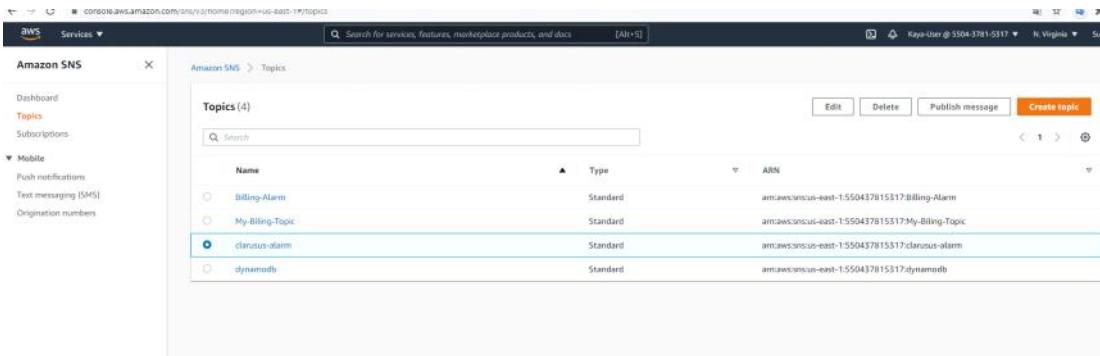
### SNS Hands-on

#### **## Part 1 - Creating Topic, Subscription and Publishing Message**

##### **### Step 1 : Create Topic**

**- Go to `SNS` service on AWS console.**

**- Click `Topics` >> `Create topic`.**



The screenshot shows the AWS SNS Topics page. The left sidebar has links for Dashboard, Topics (which is selected), Subscriptions, Mobile, Push notifications, Text messaging (SMS), and Origination numbers. The main content area is titled "Topics (4)". It lists four topics:

Name	Type	ARN
Billing-Alarm	Standard	arn:aws:sns:us-east-1:550437815317:Billing-Alarm
My-Billing-Topic	Standard	arn:aws:sns:us-east-1:550437815317:My-Billing-Topic
clausus-alarm	Standard	arn:aws:sns:us-east-1:550437815317:clausus-alarm
dynamodb	Standard	arn:aws:sns:us-east-1:550437815317:dynamodb

**- Details :**

- Type: Standard
- Name: Demo-topic
- Display Name: My-First-Topic
- Keep rest default.
- Click `Create`.

## 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 500 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

Maximum 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](#)

### ### 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-1:550437815317:Demo-topic

#### Protocol

The type of endpoint to subscribe

Email

#### Endpoint

An email address that can receive notifications from Amazon SNS.

kayakadir58067@gmail.com

i 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**

Amazon SNS > Topics > Demo-topic

## Demo-topic

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

Details	
Name Demo-topic	Display name My-First-Topic
ARN arn:aws:sns:us-east-1:550437815317:Demo-topic	Topic owner 550437815317
Type Standard	

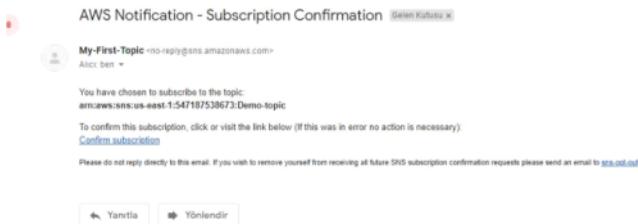
[Subscriptions](#) [Access policy](#) [Delivery retry policy \(HTTP/S\)](#) [Delivery status logging](#) [Encryption](#) [Tags](#)

Subscriptions (1)				
<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Request confirmation</a> <a href="#">Confirm subscription</a> <a href="#">Create subscription</a>				
ID	Endpoint	Status	Protocol	
Pending confirmation	kayakadir3806@gmail.com	Pending confirmation	EMAIL	

### ### 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`.

Maile gidip confirmation aciyoruz



### ### Step 4 : Publish message

- Select `Topics` from the left-hand menu and click on `Demo-topic`.
- Click `Publish message`.

Amazon SNS > Topics > Demo-topic

## Demo-topic

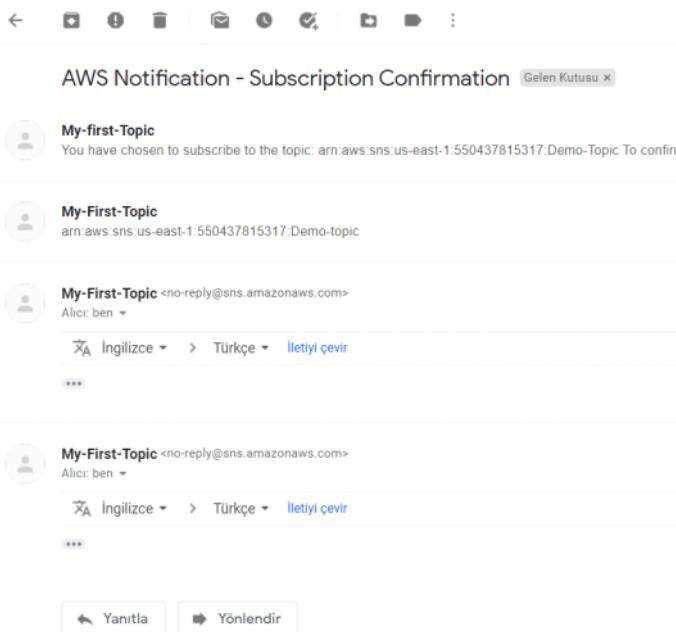
[Edit](#) [Delete](#) [Publish message](#)

Details	
Name Demo-topic	Display name My-First-Topic
ARN arn:aws:sns:us-east-1:550437815317:Demo-topic	Topic owner 550437815317
Type Standard	

- `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.

En alttta publish yapıyoruzx Mailimizi kontrol ediyoruz Mesaj endpoint suscription olarak mail adresini verdigiiz icin oraya gitmi



## ## 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.

Status	Name	ARN
Success	EventBridge -> SNS topic	arn:aws:lambda:2018-04-04:123456789012:Function:my-function

- Click `Create Rule`.
- `Event Source`:
  - Event Pattern
  - Service Name : EC2
  - Event Type : EC2 Instance State-change Notification
- `Targets` >> Add target
  - SNS topic
  - Topic : Demo-topic

- Click `Configure details`.
- Name: Instance-State-Change
- Keep rest default.
- Click `Create Rule`.

The screenshot shows the 'Step 1: Create rule' page. In the 'Event Pattern' section, 'Service Name' is set to 'EC2' and 'Event Type' is 'EC2 Instance State-change Notification'. Under 'Any instance', the 'Any state' radio button is selected. In the 'Targets' section, a 'Topic' named 'sns:Demo-Alarm' is chosen. Below the main form, an 'Event Pattern Preview' shows the JSON structure of the event being sent to the topic.

The screenshot shows the 'Rules' section of the CloudWatch Events console. A success message indicates 'Rule Instance-State-Change was created.' Below it, a note about Amazon EventBridge is present. The 'Actions' tab is selected, showing a table of rules. The 'Instance-State-Change' rule is listed with a status of 'All' and a green status indicator.

Status	Name	Description
All	Instance-State-Change	EVENT

### ### 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).

The screenshot shows the 'Instances' page in the EC2 console. It lists two instances: one with Instance ID 'i-0a93bc24468c9f0ef' which is 'Stopped', and another with Instance ID 'i-08758a1946c110571' which is 'Running'.

- 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.

## AWS Notification - Subscription Confirmation Gelen Kut

### My-first-Topic

You have chosen to subscribe to the topic: arn:aws:sns:us-east-1:550437815317:Dem

### My-First-Topic

You have chosen to subscribe to the topic: arn:aws:sns:us-east-1:550437815317:Dem

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

Alici: ben ▾

 Ingilizce ▾ > Türkçe ▾ [İletiyi çevir](#)

...



AWS SQS & SNS



## Table of Contents

- ▶ SQS
- ▶ SNS

1

### SQS

Amazon Simple Queue Service (SQS), dağıtılmış sistemleri ve sunucusuz uygulamaları birbirinden ayırmamız ve eşzamanlı olarak yönetmememize imkan tanınan, tam olarak yönetilen bir iletişim kuyruğu alma hizmetidir. SQS, mesajlaşmaya yönelik ore yazılımları yönetmenin ve işletmenin getirdiği karmaşıklik ile ek iş yükünü ortadan kaldırarak geliştiricilerin farklı işlevlerin dokunulmasına imkan tanır. SQS ile ileti kaybı yaşamadıkça veya diğer hizmetlerin erişilebilir olmasına gereksinim duymadan yazılım bileşenleri arasında dilediğiniz hacimde ileti gonderebilir, depolayabilir ve abdillerstir. AWS konsolunu, terminal ettiğiniz Komut Satırı Arasährimini veya SDK yi ve iş beşti komutu kullanarak SQS yi dosyalar içinde kullanmaya başlayabilirsiniz.

SQS iki tür ileti kuyruğu sunar. Standart kuyruklar tarafından en yüksek aktarım hızı, en iyi çağrı ilkesine göre sıralanır ve en az bir kez teslim olacakları sunulur. SQS FIFO kuyrukları, iletiların tam olarak bir kez ve tam olarak gönderildikleri sırada silinmesi konusunda güvenle sağlanacaktır şekilde tasarılmıştır=><https://aws.amazon.com/tr/sqs/>

SQ5 servisi Elastic Load Balancer' a bir alternatif servis direbiliriz

## ► SQS

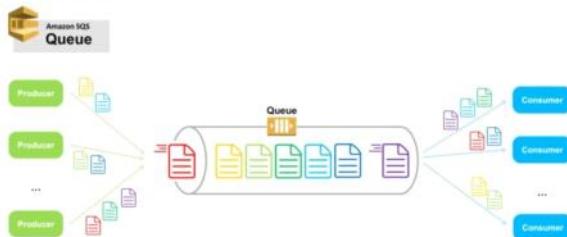
What is SQS?



- 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.

## ► SQS

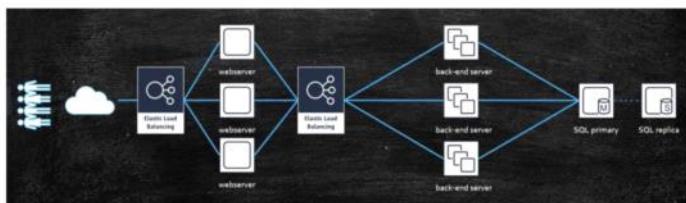
What is SQS?



Producer	Client, application, mesaj gönderen taraf
Consumer	Mesajı işleyen taraf

## ► SQS

### What is SQS?

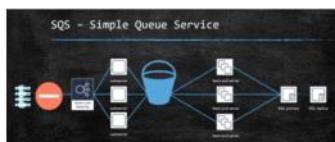


#### Online otel hizmeti

Kullanıcılar front end e bağlanıp rezervasyon işlemi ve kredi kartı işlemleri yapacak.

Front end back end e yönlendirilecek. Back end de cevap verecek mesajları front ende diğer kısmını da database ye gönderecek

Normalde arada load balancerler var ve yük dengelemektedir. Ya yük artar ve load balancer gibi dengeleyicilerde yükü karşılayamazlarsa. Yük artarsa front end talepleri back ende ulaşamayacak ve işlem gerçekleştirmedi hata donec



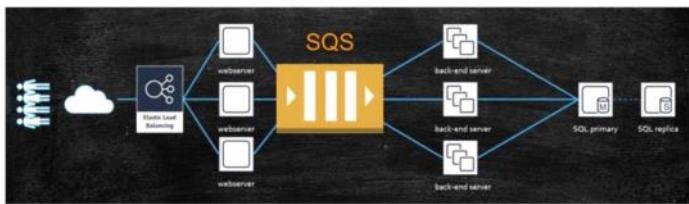
SQS yukarıdaki gorseldeki gibi gelen mesajları beklettiği bir kova sistemi uygulmaktadır. Ve müşteriye işleminiz yapılmıyor bir süre bekleyiniz şeklinde bilgilendirme yapılmaktadır (artık müşteriye hata mesajı gitmemektedir)

<https://aws.amazon.com/sqs/>  
Amazon Simple Queue Service (SQS), mikroservisleri, dağıtılmış sistemleri ve sunucu olmayan uygulamaları ölçeklendirmenize olanak tanıyan tam yönetilen bir mesajlaşma hizmetidir. SQS, karmaşık aracılık ve üst düzey yönetim gereken yazılımlar arası mesajlaşma işlemlerini basit ve etkili hale getirir.

## ▶ SQS

What is SQS?

Decoupling



Load balancer yerine SQS eklenmiş oldu ve belli bir süreliğine mesajları muhafaza edebiliriz (14 gün)

What is decouple SQS?

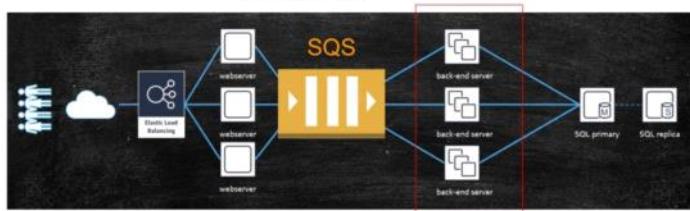
SQS lets you decouple application components so that they run and fail independently, increasing the overall fault tolerance of the system. Multiple copies of every message are stored redundantly across multiple availability zones so that they are available whenever needed.

## ▶ SQS

What is SQS?

ASG

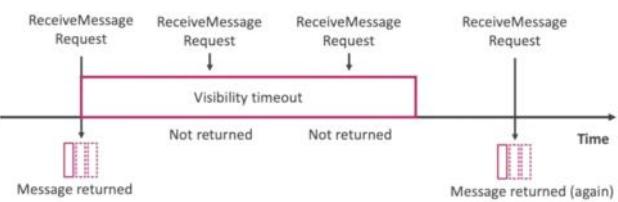
CloudWatch Metric - Queue Length  
ApproximateNumberOfMessages



100%

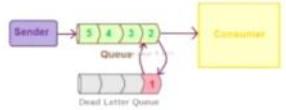
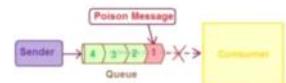
## ► SQS

### Message Visibility Timeout

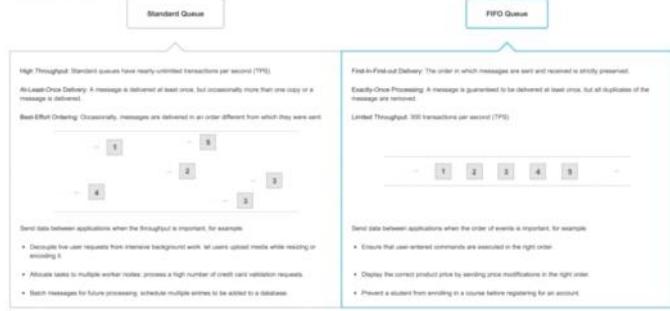


## ► SQS

### Dead Letter Queue (DLQ)



## ▶ SQS



## Standard vs. FiFo

### Standard

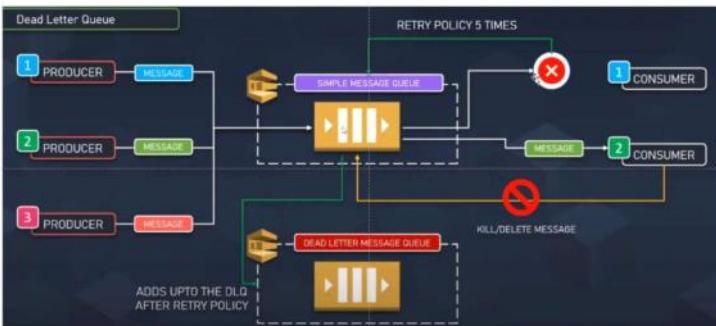
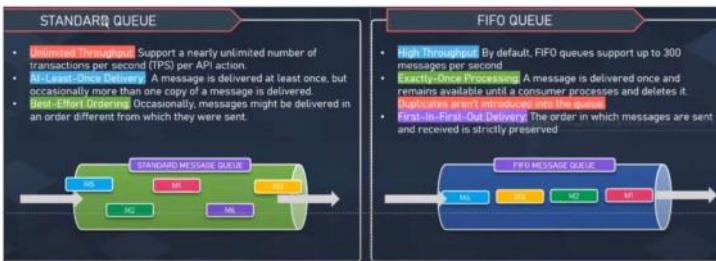
- Saniyede sınırsız yakın işlem kapasitesi
- Mesajın tek bir defa iletilmesi ve giriş çıkış sırası garanti değil

### FiFo

- Saniyede 300 mesaj işleme kapasitesi, batch request ile bu 3000 mesaja kadar çıkabilir
- İlk giren mesaj ilk işlenir-FIFO
- Mesajın tek sefer işleneceği garanti edilir



100% ▶



## ► SQS Pricing



- Pay only for what you use

- AWS **Free Tier** includes **1 million requests** with Amazon Simple Queue Service (SQS).

<https://aws.amazon.com/sqs/pricing/>

100% ✓

## 2 SNS

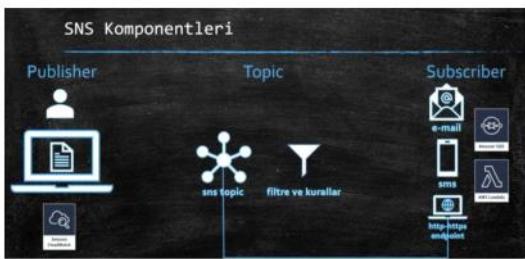
### SNS

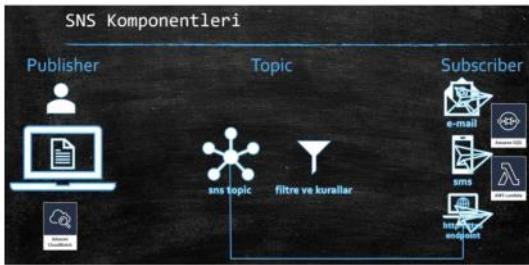
What is SNS?



Amazon  
SNS

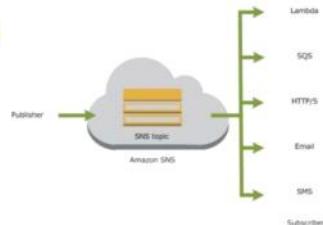
- Amazon **Simple Notification Service** (Amazon SNS) is a managed service that provides message delivery from **publishers** to **subscribers** (also known as **producers** and **consumers**).





## 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).

## SNS

What is SNS?

Application-to-Application (A2A)



## ► SNS

### What is SNS?

#### Application-to-Person (A2P)



Bir topic oluşturuluyor ve aynı anda birden çok kullanıcıya gonderebiliyoruz.

## ► 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.**

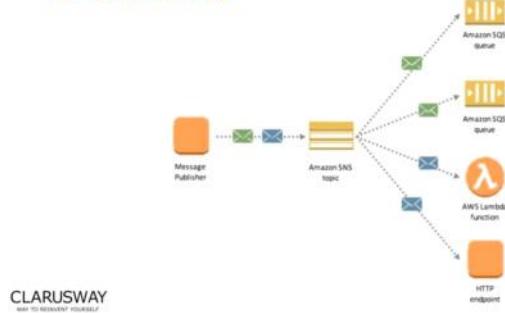
## SNS

### What is SNS?



## SNS

### Fan Out with SNS



CLARUSWAY  
WE TRY TO REINVENT YOURSELF

## 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.

<https://aws.amazon.com/sns/pricing/>

## ▶ SQS & SNS

### Hands on

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

The screenshot shows the AWS Management Console interface for creating a new queue. At the top, there is a navigation bar with 'AWS Lambda' selected. Below it, a 'Services' dropdown menu is open, showing 'Simple Queue Service' as the selected option. A large orange 'Create queue' button is prominently displayed. The main area is titled 'Create queue' and contains a 'Details' section. Under 'Type', it says 'Standard - with at least one message delivery confirmation' and has a note: 'You can't change the queue type after you create a queue.' There are two tabs: 'Standard' and 'FIFO'. Under 'Standard', there are three options: 'At least one delivery', 'Delivery on success', and 'Delivery on failure'. Under 'FIFO', there are four options: 'At least one delivery', 'Delivery on success', 'Delivery on failure', and 'Delivery on error'. A 'Name' field is present with the value 'MyFirstQueue'. At the bottom, there is a note about the queue URL and a link to the developer guide: <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-deleting-queues.html>.

**Configuration**

For this configuration message size availability for other consumers and message retention - info

Availability timeout - info	Message retention period - info
100	4 hours
Seconds	Days
Message retention if consumer is idle 10 days	Should be between 1 minute and 14 days
Delivery delay - info	Maximum message size - info
0	256
Seconds	KB
Message retention if consumer is idle 10 minutes	Should be between 1 KB and 256 KB
Receive message wait time - info	Message retention if consumer is idle 10 seconds
0	Seconds
Seconds	Should be between 0 and 10 seconds

**Access policy**

Before allowing access your account - info

**Classic endpoint**

Basic (Use this endpoint to define a basic authentication)

Advanced (Use a JSON object to define an advanced access policy)

(Define a basic endpoint)

Define who can send messages to this endpoint

Only this consumer (This consumer can send messages to this endpoint)

Any consumer (Any consumer can send messages to this endpoint)

Only this user (Only this user can send messages to this endpoint)

Any user (Any user can send messages to this endpoint)

Define who can receive messages from this endpoint

Only this consumer (This consumer can receive messages from this endpoint)

Any consumer (Any consumer can receive messages from this endpoint)

Only this user (Only this user can receive messages from this endpoint)

Any user (Any user can receive messages from this endpoint)

**Encryption - Optional**

None (Not recommended as it leaves messages unencrypted by default). To enable an end-to-end encryption for your queue, enable [TLS 1.3 or later](#) - info

Transport layer encryption

Enabled

Disabled

**Dead-letter queue - Optional**

None (Send undeliverable messages to a dead-letter queue) - info

Set this endpoint to receive undeliverable messages

Undelivered

Delivered

**Tags - Optional**

A tag is a label assigned to an AMQP message. Use tags to search and filter your messages or track your AMQP links - [Update queue](#) - info

Tag	Value (optional)
AMQP tag	AMQP tag
Add new tag	Remove tag

Cancel Create queue

**Send and receive messages**

**Manual olarak mesaj göndereceğiz**

**Send message** - info

Clear content	Send message
Message body	Text area for message content
Content type	Text area for message content type
Use as the first <a href="#">topic</a>	Text area for message content

Send message diyeлим

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

**View details**

**Post New message**

**Menüsunu basınca bir tane mesaj görebiliriz**

Poll => mesaji request sayesinde consumer a gonderiliyor yani cekiliyor

The screenshot shows the 'Receive messages' interface for an AWS Lambda function. It has tabs for 'Edit poll settings', 'Stop polling', and 'Poll for messages'. Configuration options include 'Messages available' (3), 'Polling duration' (30), 'Maximum message count' (10), and 'Polling progress' (10 messages polled). The 'Messages' list contains three items:

ID	Approximate arrival time	Size	Receive count
1	8/18/2021, 20:44:51 GMT+1	23 bytes	1
2	8/18/2021, 20:44:51 GMT+1	23 bytes	1
3	8/18/2021, 20:44:51 GMT+1	23 bytes	1

Gonderdigim diger mesajlar

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

The screenshot shows the 'Receive messages' interface for an AWS Lambda function. It has tabs for 'Edit poll settings', 'Stop polling', and 'Poll for messages'. Configuration options are identical to the previous interface. The 'Messages' list contains four items:

ID	Approximate arrival time	Size	Receive count
1	8/18/2021, 20:44:51	23 bytes	3
2	8/18/2021, 20:44:51	23 bytes	4
3	8/18/2021, 20:44:51	23 bytes	2

Delete menusü ile silebiliriz

Bir tane Lambda function oluşturucu

The screenshot shows the 'Functions' list interface for AWS Lambda. It includes a search bar, a 'Create function' button, and a table with columns: Function name, Description, Package type, Runtime, Code size, and Last modified. A note at the bottom states: 'There is no data to display.'

Aşas nı bize hazır sunduğu template/function kullanacağız

The screenshot shows the first step of the 'Create function' wizard. It asks for the runtime ('AWS Lambda') and the code source ('Author from scratch'). Other options shown are 'Use a blueprint' (selected), 'Container image', and 'Borrow serverless app repository'.

Sqs yazıp enter diyelim

The screenshot shows the second step of the 'Create function' wizard. It lists a trigger named 'sqs-poller' (an Amazon SQS trigger that logs messages in a queue) and a 'Configure' button.

**Basic information** [Info](#)

Function name: `sqs-poller`

Execution role: [Create a new role with basic Lambda permissions.](#) [Create a new role with a custom role.](#) [Go to IAM console.](#)

Create a new role with basic Lambda permissions.

Role name: `sqs-poller-role`

Role description: [AWS Lambda service role](#) [AWS Lambda service role documentation](#)

Policy templates - optional: [Choose one or more policy templates.](#)

Amazon SQS Poller permissions: [Edit](#)

**Permission otomatik geldi**

SQS trigger: [View details](#)

SQS queue: [Amazon SQS queue named MyFirstQueue](#)

Batch size: 10

Batch window: 0

In order to read from the SQS trigger, your execution role should have proper permissions. [Read the Amazon SQS documentation for a detailed way of setting permissions.](#)

**Lambda function code**

Code is configured for the chosen blueprint. You can configure it after you create the function. [Learn more about AWS Lambda function code.](#)

```
functionHandler: 'index.handler'
```

Index.js (24)

```
1: //console.log('Loading function');
2: exports.handler = async (event) => {
3:   // Grab the event records
4:   const messages = event.Records;
5:   let result = '';
6:   for (let i = 0; i < messages.length; i++) {
7:     const message = messages[i];
8:     result += `Message ${i}: ${message.body}\n`;
9:   }
10:  return `Successfully processed ${messages.length} messages!`;
11}
```

[Edit](#)

[Create function](#)

**Function oluşturul ve trigger da sqs**

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

Triggers: [SQS](#)

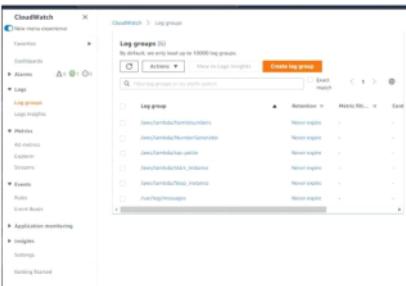
Layers: [Add layer](#)

Add destination

[Edit trigger](#)

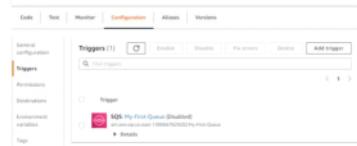
[View](#) [Edit](#)





Sqs-poller adı altında log grubunun oluşturduğunu görebiliriz

Lambda ekranına gidelim



Enable edelim



Amacımız : biz consumer yerine lambda yi configure ettik. Mesaj keydugumuzda consumer yerine lambda mesajı çekerek ve cloudwatch e yazacık.

BMW Group, sürücülerine dinamik olarak güncellenen harita bilgileri sunmak için BMW 7 Serisi araçlarından sensor verilerini toplayan bağlı araç uygulaması için AWS'yi kullanıyor. BMW, sensorlerin aracın (CARASSO) odaklı olduğu yeni hizmetini Amazon SQS, Amazon S3, Amazon DynamoDB, Amazon RDS ve AWS Elastic Beanstalk'tan yararlanarak yalnızca altı ayda oluşturdu.

Amazon SQS > Queues > My-First-Queue

**My-First-Queue**

**Details** [Edit](#) [Delete](#) [Purge](#) [Send and receive messages](#)

**Name:** My-First-Queue **Type:** Standard **ARN:** arn:aws:sqs:us-east-1:123456789012:My-First-Queue

**Encryption:** [TLS](#) [AWS Lambda trigger](#) [Dead letter queue](#)

[More](#)

[SMS subscriptions](#) [Lambda triggers](#) [Dead letter queue](#) [Monitoring](#) [Tagging](#) [Access policy](#) [Encryption](#)

**Lambda triggers (1) [Info](#)** [View in Lambda](#) [Edit](#) [Configure Lambda function trigger](#)

**Search triggers**

**UUID:** ARN: Status Last modified

arn:aws:lambda:us-east-1:123456789012:function:My-First-Queue 2023-07-18T16:48:48Z Enabled 2023-07-18T16:48:48Z

**Send and receive messages**

Send messages to and receive messages from a queue.

**Send message** [Info](#) [Clear content](#) [Send message](#)

**Message body** Enter the message to send to the queue.

This message is sent from [AWS Lambda](#).

**Delivery delay** [Info](#)

0 ms

Set... Should be between 0 seconds and 10 minutes.

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

**Receive messages** [Info](#) [Edit poll settings](#) [Time polling](#) [Put for messages](#)

**Messages available:** 0 **Polling duration:** 10 **Maximum message count:** 10 **Polling progress:** 0%

**Messages (0)** [Search messages](#) [View details](#) [Delete](#)

No messages. To view messages in the queue, poll for messages.

[Put for messages](#)

**Lambda da mesaj log kayitlarini gorebiliriz**

**Send diyelim**

**Send and receive messages**

Send messages to and receive messages from a queue.

**Send message** [Info](#) [Clear content](#) [Send message](#)

**Message body** Enter the message to send to the queue.

This message is sent from [AWS Lambda](#).

**Delivery delay** [Info](#)

0 ms



**Details**

Type: **HTTP** This subscription type is used when the topic is created.

**HTTP via AWS Direct Connect**

- Service generated message ordering
- Service generated message ID
- Subscription ARN
- Subscription creation date
- Subscription status

**Standard**

- Best effort message ordering
- All event notifications
- Subscription acknowledgement
- Subscription metadata (ARN, Lambda ARN)
- Subscription status

Name: **Amazon SNS - 1620804762022-Demo-Topic**

Display name: **Amazon SNS - 1620804762022-Demo-Topic** Can include alphanumeric characters, hyphens (-), and underscores (\_).

**We trust this** This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic. [Info](#)

**Encryption - optional** Amazon SNS provides no standard encryption by default. Enabling server-side encryption adds an extra layer of security for your topic. [Info](#)

**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** This policy defines how many times to attempt delivery of an HTTP(S) message. To modify the default settings, expand this section. [Info](#)

**Delivery status logging - optional** This setting configures the logging of message delivery status to CloudWatch Logs. [Info](#)

**Tags - optional** An tag is a metadata label that you can assign to an Amazon SNS topic. Each tag consists of a key and an optional value. You can use tags to search and filter your topics and track your costs. [Learn more](#) [Edit](#)

[Cancel](#) [Create topic](#)

[Create subscription](#)

Amazon SNS > Subscriptions > Create subscription

**Create subscription**

**Details**

Type: **HTTP** arn:aws:sns:eu-central-1:1620804762022:Demo-Topic

Protocol: **HTTP** The protocol for notifications selected.

Email: **hannadiyakupcu@gmail.com** an email address that can receive notifications from Amazon SNS.

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

**Subscription filter policy - optional** The 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](#)

[Create subscription](#)

Pending statusundaki aboneliğinizi görebilirsiniz

Pending confirmation  **Hannadiyakupcu@gmail.com**  Pending confirmation  Email  Demo Topic

Mail olarak confirmation message gelecektir

AWS Notification - Subscription Confirmation (100%)

My First Topic [mehmet@preprocess.com](#) 541 ms (2 minutes ago)

By me

You have chosen to subscribe to the topic.  
You can now use it directly in the AWS Lambda function you created.

To confirm this subscription, click or copy the URL of this page (or enter its address in the browser).

Please do not reply directly to this email. If you want to receive yourself from sending an SNS message confirmation message, please send it to:

aws-sns-tester@preprocess.com

**Once you've done this, you're good to go!**

SMS Email Lambda Function SNS Lambda

Amazon SNS > Topics > Demo-Topic > Publish message

**Publish message diylelim**

Amazon SNS > Topics > Demo-Topic > Publish message

**Publish message to topic**

**Message details**

Topic ARN: arn:aws:sns:eu-central-1:000000000000:Demo-Topic

Subject: replicated

One text message

Message body (optional)  
This is just a test message for one recipient. Please note that the push notifications are now being delivered to the recipient's mobile device.

Please note to turn on the requested:

Identify pushed for all delivery protocols  
The recipient will receive an SMS message, which is the best way to verify if the topic has been successfully subscribed to the topic. Based on their provider's preference.

Custom pushed for each delivery protocol  
The recipient will receive an SMS message, which is the best way to verify if the topic has been successfully subscribed to the topic. Based on their provider's preference.

Message body:  
This is just a test message for one recipient.

**Publish message**

**Message e-mail (end point) gitti**

Recipient: [mehmet@preprocess.com](mailto:mehmet@preprocess.com)

Primary Secondary Promotions

My First Topic one test message This is a test message for one recipient account. 541 ms

44 More > Creating a CloudWatch Event to invoke me

Amazon SNS

**CloudWatch**

New name: CloudWatch Events to new Amazon EventBridge

CloudWatch Events triggers an AWS Lambda function or an AWS Step Functions state machine when CloudWatch Events receives new resources such as an AWS event source. The party would subscribe and define a registry to handle request and update the topic or trigger per your architecture and application.

**CloudWatch Events documentation**

**Rules**

Rule rule available from your AWS account for processing by selected targets. You can create, edit, and delete rules.

**Create rule**

**Bir tane ec2 ocalim (ozellikler onemli degil)**

**Step 1: Create rule**

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

**Event Source**

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

**Event Pattern**

Match events pattern to match events by services.

**Interval**: 600

**Event Type**: EC2 instance state-change notifications

**Any source**  Specific source

**Any destination**  Specific destination

**Event Pattern Preview**

```
{ "source": [ "aws.ec2" ], "detail-type": [ "AWS Lambda - New instance state-change notification" ] }
```

**Targets**

Select Target to invoke when an event matching your Event Pattern or when schedule is triggered.

**SNSTopic**

**Type**: Event Type

**Add target**

**Configure Details**

**Step 2: Configure rule details**

**Rule definition**

**Name**: instance state-change

**Description**:  Enabled

**CloudWatch Events will add necessary permissions for targets so they can be invoked when the rule is triggered.**

**Required**

**Create rule**

Done 40 / 100

**Rules > instance-state-change**

**Summary**

Event pattern (1) `aws.ec2.event.detail(instance.state.name == 'running' OR instance.state.name == 'pending')`

Description

Monitoring: Once instance has been started.

**Targets**

Filter: Viewing 1 to 1 Targets ▾

Type	Name	Arn	Date	Additional permissions
AWS Lambda	CloudWatch Metrics	arn:aws:lambda:eu-central-1:318594768165:CloudWatchMetrics	10/19/19	

**Ec2 servisine gecelim. Herhangi instance de durum degisikliginde bize mail atacak**

- Instance yi stop veya start edelim

**AWS Notification Message** (888888)

MyFirstTopic -> myfirsttopic@amazonaws.com

To the "MyFirstTopic" topic, the "aws:ec2:instance-state-change" notification type was triggered at 12:19 PM (10 minutes ago).

Subscription: "arn:aws:sns:eu-central-1:318594768165:MyFirstTopic" "arn:aws:lambda:eu-central-1:318594768165:instanceStateChangeHandler" "arn:aws:kinesis:eu-central-1:318594768165:instanceStateChangeKinesis" "arn:aws:lambda:eu-central-1:318594768165:instanceStateChangeLambda"

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe.

This message was sent via Amazon Simple Notification Service (Amazon SNS). To learn more about SNS, visit [the Amazon SNS developer documentation](#).

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at [AWS customer support](#).

**Pending statusundede oldugunu gorabiliriz**

**Temizlik asaması:**

**Cloudwatch rule silelim**

**Devamli mesaj gelmesini istemiyoruz**

**Rules**

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

**Create Rule** Actions ▾

Topics ▾ Create Topic

Filter: Viewing 1 to 1 Rules ▾

Name	Type	ARN
instance-state-change	Standard	arn:aws:lambda:eu-central-1:318594768165:instance-state-change

**Amazon SNS** (1)

Topics ▾ Create Topic

Filter: Viewing 1 to 1 Topics ▾

Name	Type	ARN
awslogs-health-data	Standard	arn:aws:kinesis:eu-central-1:318594768165:awslogs-health-data
awslogs-cloudwatch-logs	Standard	arn:aws:kinesis:eu-central-1:318594768165:awslogs-cloudwatch-logs
awslogs-cloudwatch-metrics	Standard	arn:aws:kinesis:eu-central-1:318594768165:awslogs-cloudwatch-metrics
awslogs-cloudwatch-metric-streams	Standard	arn:aws:kinesis:eu-central-1:318594768165:awslogs-cloudwatch-metric-streams
awslogs-cloudwatch-log-groups	Standard	arn:aws:kinesis:eu-central-1:318594768165:awslogs-cloudwatch-log-groups
awslogs-cloudwatch-metric-meters	Standard	arn:aws:kinesis:eu-central-1:318594768165:awslogs-cloudwatch-metric-meters

