

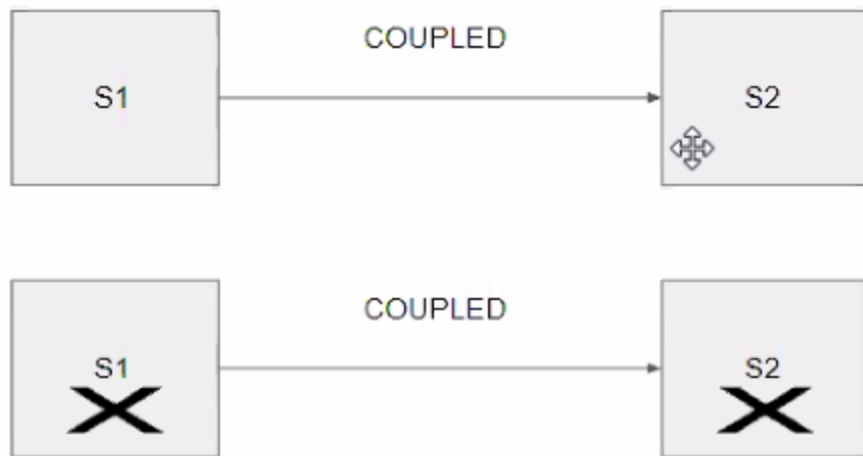
# SQS & SNS

## Simple Queue Service (SQS)

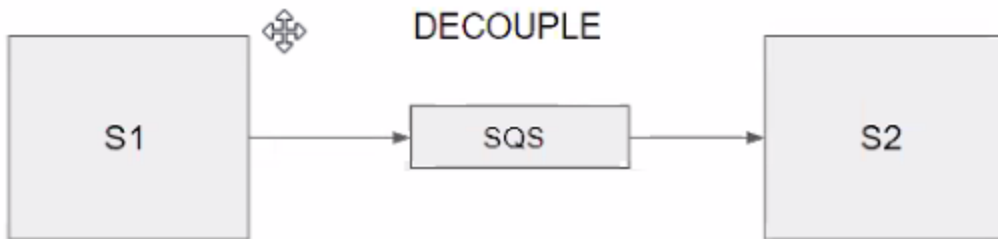
Basit sıra hizmeti diyebiliriz. AWS'nin en eski servislerinden.

### Decoupled

Birbiri ile ilişkili (bağlanan) servislere coupled ya da akuple deriz.



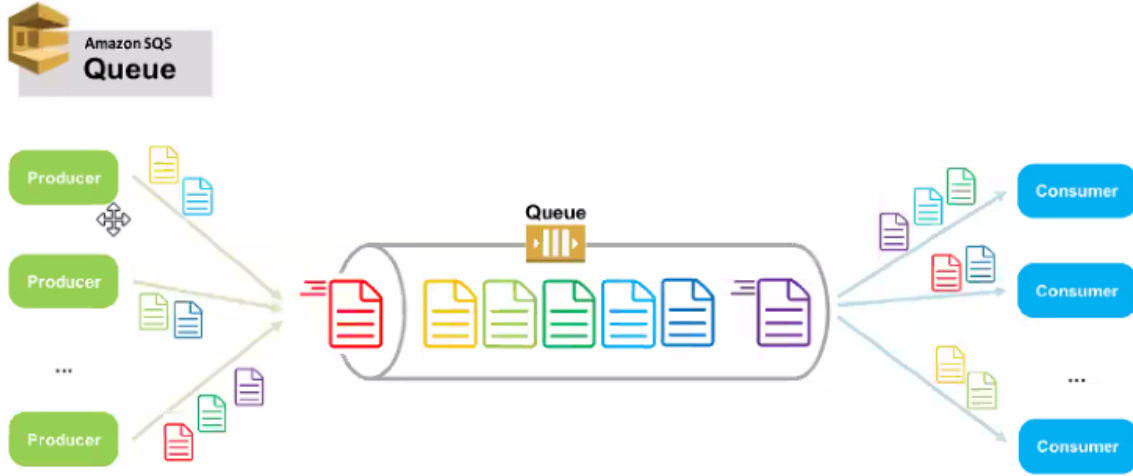
Bunun dezavantajı S1 çalışmazsa S2'de çalışmaz. Yani birinde sıkıntı varsa ikisi de gider. Yukarıda bir işlem gözüküyor ama normal şartlarda 10'larca işlem olduğunda biri patlarsa hepsi gidiyor.



Bunu engellemek için araya bir aracı eklenir. S1 işini bitirdikten sonra SQS'e atar. S2 daha sonra gelip kendi işini SQS'ten alır.

AWS Fully-managed sistem. Serverless sistemdir. 1 milyon request free-tier

## What is SQS?



Producer mesajı üreten taraf. Frontend kısmı. Consumer ise gereği hanesi. EC2 vs gibi işlemi yapacak ekip.

Producerlar mesajlarını (task) sırayla gönderir. queue'da sıraya sokulur. Consumerlar sırayla gelir bana görev var mı. Bu sorgulama işleme poll denir. Her poll işlemi ücretli. Varsa alır gibi. Bu tasklara örnek bucket'a resim koyduğumuzda herhangi bir işlem yapacak bir lambda kodu yazmıştık. Bu tüm işlemi buna sokabiliriz.

Analoji:

Bankaya sıraya girdik. Müşteriler producerlar. Her müşterinin farklı işlemi var. Para yatırma kredi çekme vs. Veznedarlar ise Consumer. Numaratör ise Queue işlemi.

## Type

**1) Standart Queue:** Nerdeyse saniye başına sınırsız sayıda işlem yapabilme imkanı verir. Aşağıdaki resime de bakarsak herhangi bir sıra yok. Her mesaj kesinlikle 1 kez işlem göörecektir ancak daha fazla da olabilir. 3. işlemdeki, gibi.



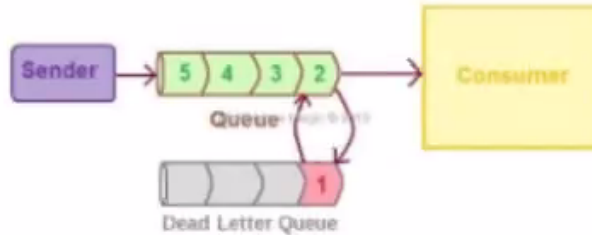
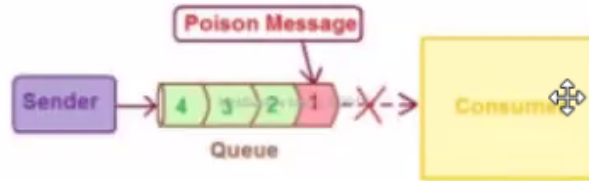
**2) FIFO Queue:** First in First out, Önce bayat ekmek yenir. Çünkü önce o alınmıştır.

Sıralama önemli. Bir işlemi bir kez yapar. Aynı işlemi bir tane daha varsa kopyasını siler.1 Ancak daha pahalı. Ayrıca saniye başına max. 300 işlem limiti var.



### Dead Letter Queue (DLQ)

Consumer benlik bişey var mı diye poll attı. İşi aldı ancak gereğini yapamıyor. Bu durumda boşta olan başka bir consumer geliyor. Bir şekilde onda da hata oluyor. Yani bir kısır döngü oluyor burada. Default olarak 10 (kösoldan sayı değışebilir.) poll işlemi yapıldı ve gereği yapılamadı ise alttaki dead letter sırasına alır. Daha sonra 2. işlemi sıra devam eder. Bitirdiği işlemi siler. Sırada kalmaz. Daha sonra bu dead letter kısmını açıp bakabiliyoruz neden arıza vermiş diye.



Konsol üzerinden ayarlama yapabiliriz.

Mesajın bekleme süresi 1 dk.dan 14 güne kadar ayarlayabiliriz. Bunu troubleshooting yapmak için istediğimiz gibi ayarlarız. İstersek sıraya gelen işlemi 15 dk.ya kadar

bekletebiliyoruz.

Message retention period [Info](#)

Days ▼

Should be between 1 minute and 14 days.

Maximum message size [Info](#)

KB

Should be between 1 KB and 256 KB.

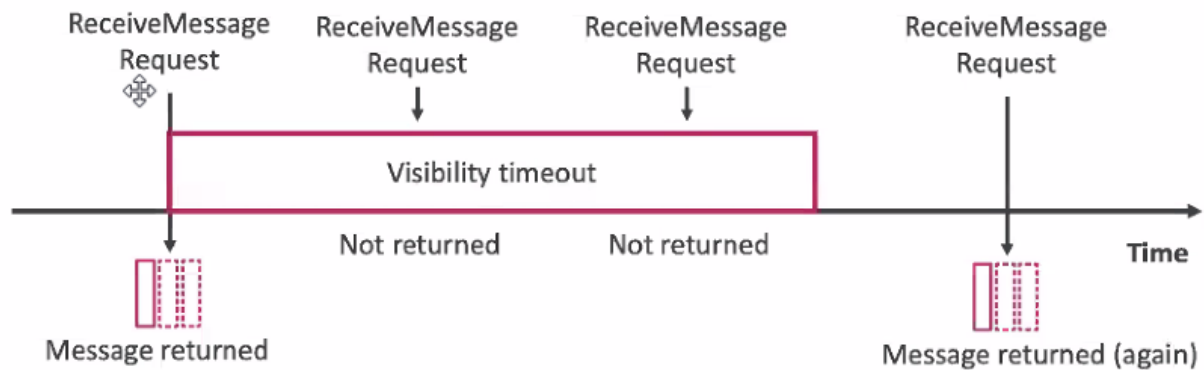
Delivery delay [Info](#)

Seconds ▼

Should be between 0 seconds and 15 minutes.

## Message Visibility Timeout

### Message Visibility Timeout



Bir işlemi bir consumer aldıysa belirli bir süre bu işlemi diğer consumerlar göremesin diye ayarlama yapıyoruz. Çünkü bir işlem hemen bitmeyebilir. O sırada boşta kalan consumer gelip aynı işlemi alabilir. Her poll işlemi paralı ayrıca boş yere diğer consumerları meşgul ediyoruz. Bu nedenle işlem süresine göre burayı ayarlayabiliyoruz.

## Configuration

Set the maximum message size, visibility to other

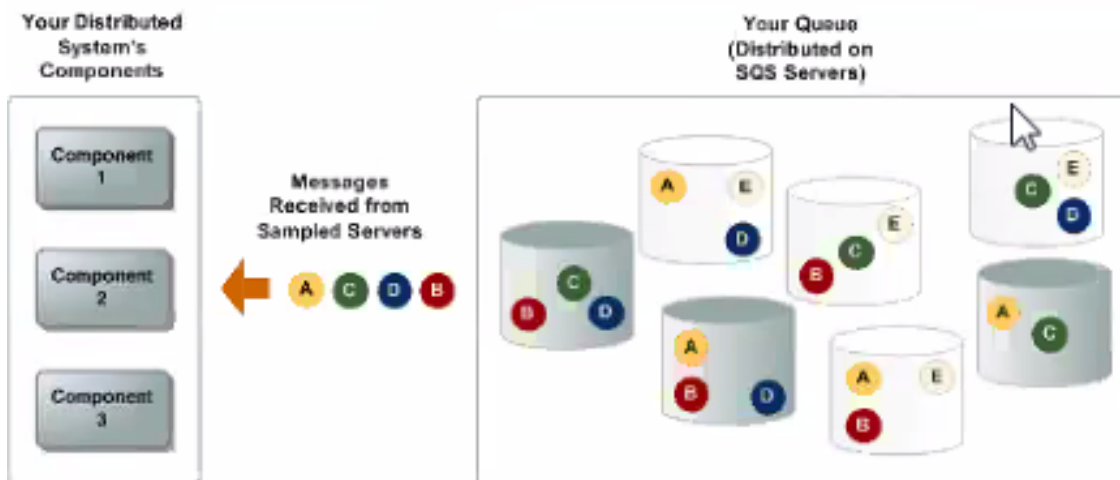
Visibility timeout [Info](#)

Seconds ▼

Should be between 0 seconds and 12 hours.

## Short poll - Long poll

Short poll çok kısa olduğu için tüm servislere bakamıyor. Sadece grilere bakabili. Böylece E işlemini kaçırdı. 2. poll işleminde onu da görür yapar. Bu nedenle genellikle long poll kullanılır. Uzun uzun bakar tüm işlemlere hiçbirini kaçırmaz.



Short poll, 0 sn; Long poll 20 sn olarak belirlenir.

Receive message wait time [Info](#)

Seconds

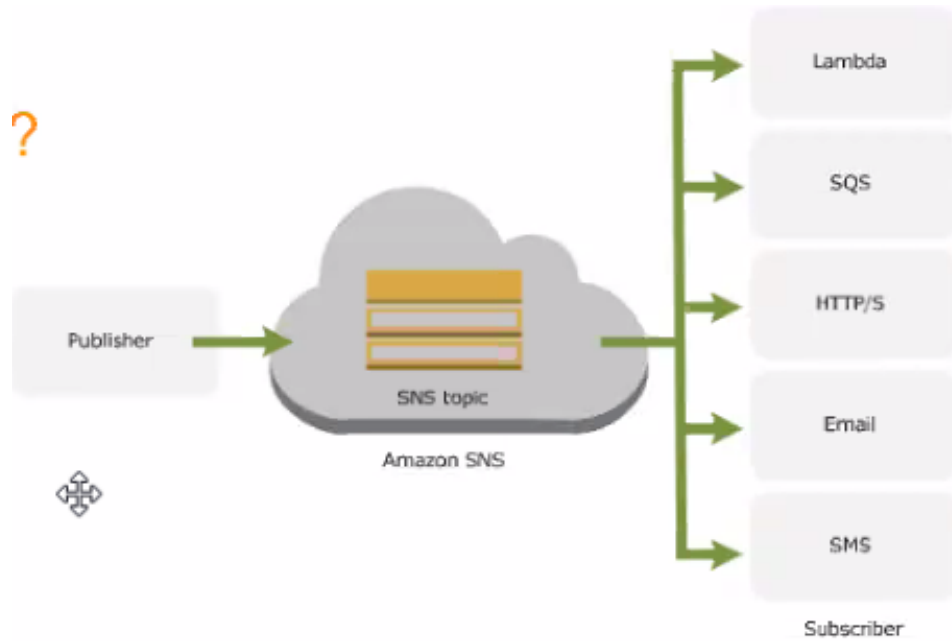
Should be between 0 and 20 seconds.

## Simple Notification Service (SNS)

Bildirim servisi diyebiliriz. İki ucu Publisher ve subscriber olarak adlandırılır (producer ve consumer yerine). Yapılan işlem de topic. PubSub modeli de denir.

Pay as you go servistir. Regional sistemdir. Bölgeye göre fiyatlandırma değişebilir. 1 milyon bildirim free-tier dan ücretsiz

Bir olay olduğu zaman Publisher bir SNS topic'i tetikler. O da subscriber'lara mesaj yollar. SQS'teki gibi poll işlemi yok yani her consumer'a bir işlem yok. Tüm subscriberlara **push** edilir.



bunlara günlük hayattan örnek verirse telefonlara gelen uygulama bilgilendirme mesajları

Type olarak bunda da FIFO ve standart var. Mantık SQS ile aynı

A2A: Application to Application. Uygulamadan uygulamaya

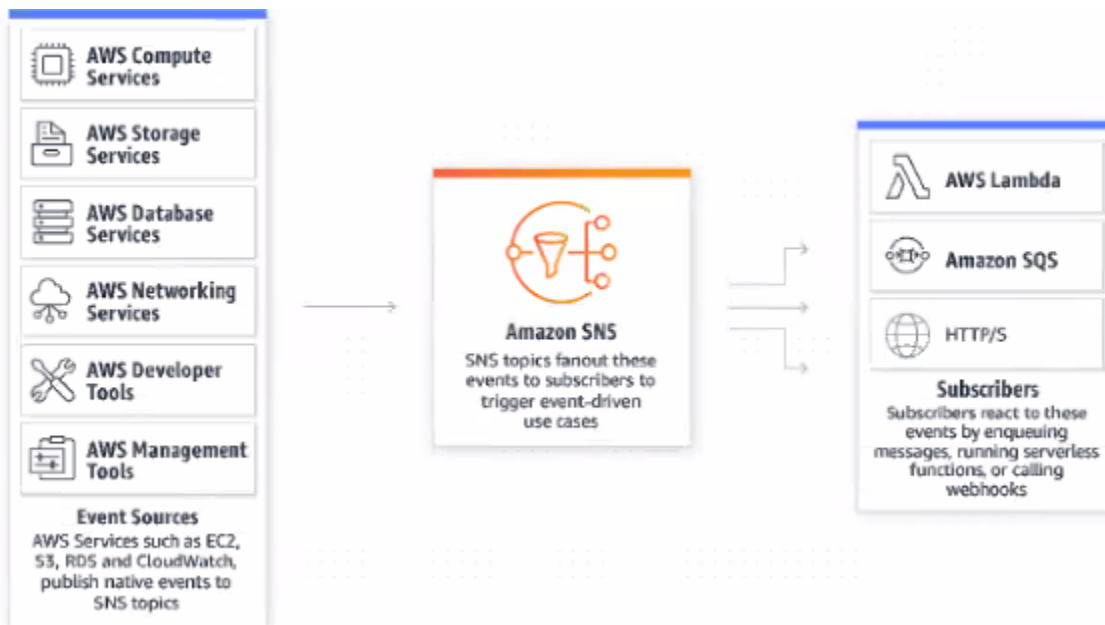
A2P: Application to Person. Uygulamadan insana

## SNS is integrated with many AWS Services

I

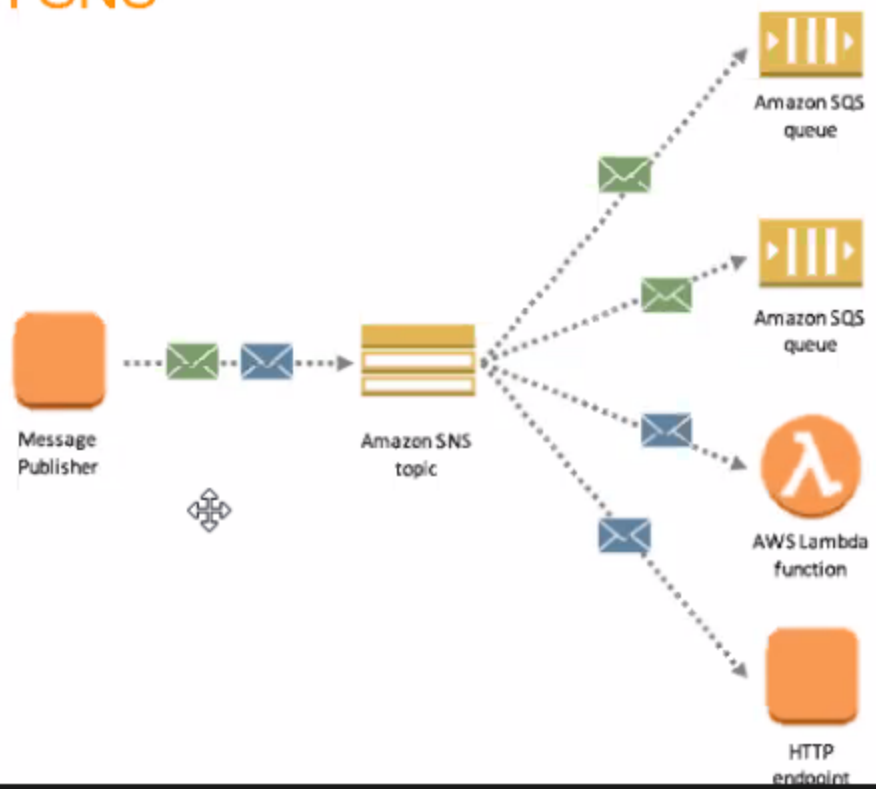
- CloudWatch (Alarms/Events)
- S3 (Bucket Events)
- CloudFormation (State changes etc.)
- Auto Scaling Groups
- And many others **can invoke SNS.**

Entegre olduğu belirli uygulamalar yukarıda. SQS ve SNS AWS'nin en temel servislerinden. Tek başlarına pek de önemli değil gibi görülse de başka servislerle birleşince oldukça önemli.



SNS olay bazlı çalışır. Bir olay olduğu zaman bir çok şeyi tetikler. (Fan out: Yaymak) Ayrıca bu tetikleme işlemini filtreleyebilir. Aşağıdaki mavi ve yeşil mesajları farklı yerlere gönderebilir.

## Fan Out with SNS



RUSWAY