

Application Services in AWS

Analytics Tensor

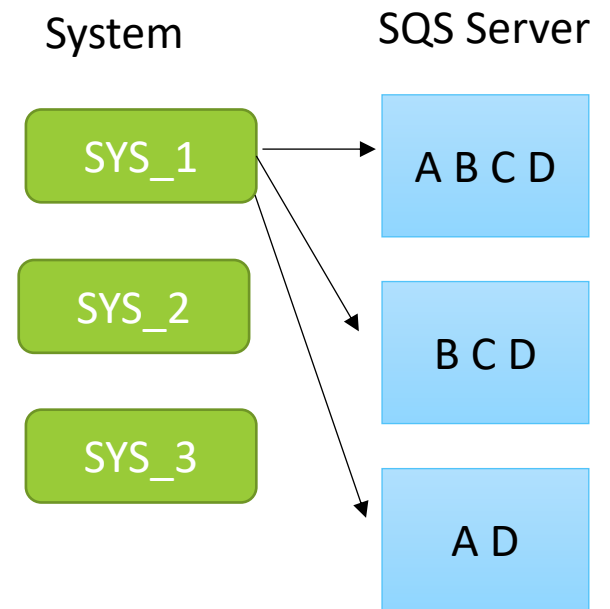
Mahesh KC

mahesh.kc@analyticstensor.com

<https://analyticstensor.com>

Amazon Simple Queue Services (SQS)

- It is highly-available, fault-tolerant and fully managed message queuing services.
- It enables to decouple and scale microservices, distributed systems, and serverless applications.
- Queue is generally used to loose coupling into the application. Loose coupling is the ability to separate different application so that they can scale independently, they don't have any inheritance knowledge of each other. It can scale out and handles large amount traffic and data.
- It can handle million to billions of messages without understanding the underlying infrastructure of the service.
- The message will be stored redundantly across multiple servers and AZs.
- In SQS, the messages will be pushed in and pulled out. We need ask the message, it is not published like other services.
- The message order is not guarantee but we get guarantee of at-least-once delivery.
- We can apply IAM/resource policy to specify who is allowed to use the queue.
- We can also allow read/write anonymously into the queue.

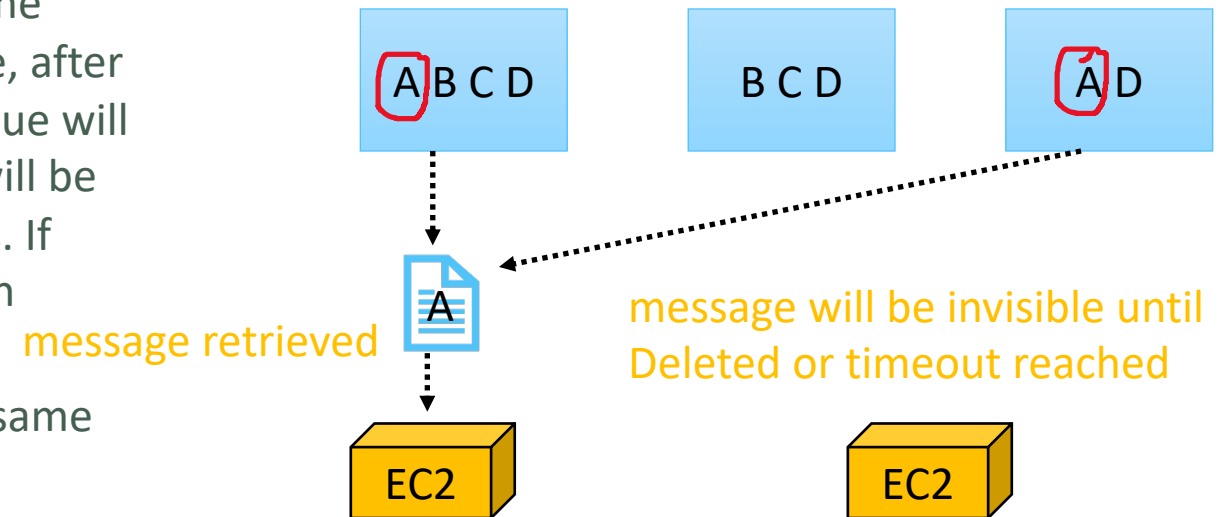


Message: A B C D

Amazon Simple Queue Services (SQS) (cont.)

Visibility Timeout

- The other feature of SQS is visibility timeout. Once the message is received from application into the queue, after acknowledging the message being received, the queue will mark the message across all of the partition and it will be invisible. The default visibility timeout is 30 seconds. If processing fails to delete then the message will again become visible and picked up by other instances.
- It helps to prevent multiple components to process same message at same time.
- AWS SQS is used for buffering events, messages to enable loose coupling and asynchronous process in between layers of applications.



For example, we have application in EC2 that pulls the message A out of queue. When the message is retrieved by sending acknowledge then the queue will mark message A across all the partitions as invisible and will be locked so that it won't be retrieved twice.

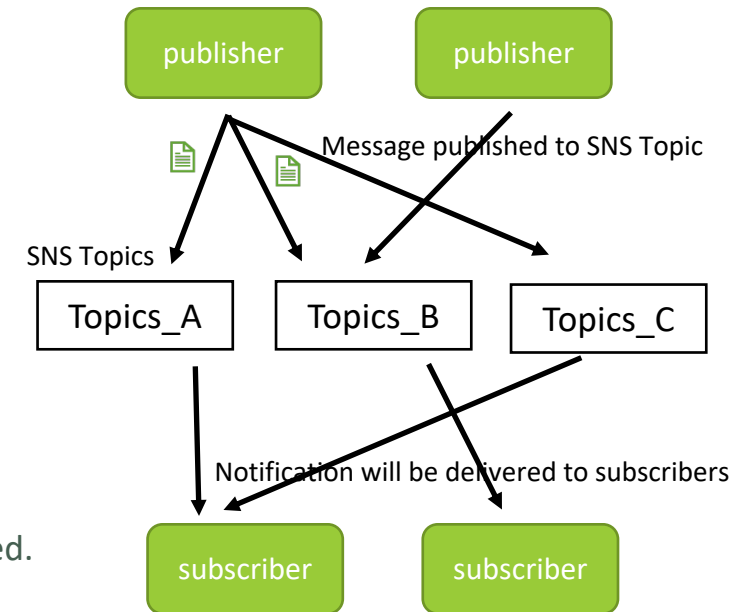
Amazon Simple Notification Service (SNS)

- It is a highly available, durable, secure and fully managed publish/subscribe messaging services.
- It is used to decouple microservices, distributed systems, and serverless applications. It provides topics for high-throughput, push-based, many-to-many messaging.
- This model is different than queue services. Queue service is based on putting and pulling or polling and asking for message, buffered in the services.
- SNS is a pub/sub model where topics are created in SNS. When we publish message to the topics then those message will be delivered or subscribed via by several methods such as:
 - Email (Plain or JSON)
 - HTTPS
 - SMS
 - Amazon SQS
 - Mobile Push
- When message are published into the topics then notification are delivered to subscribers.
- The order is not guaranteed. The message limit is 64 KB.
- The message are not stored. It means when the messages is delivered to subscriber then it won't stored.

Self Assignment: SQS Vs SNS

Demo: <https://aws.amazon.com/getting-started/hands-on/filter-messages-published-to-topics/>

Architecture: https://www.youtube.com/watch?v=c_WNBmEc6EE



Amazon Simple Email Service (SES)

- It is cloud-based email sending services.
- It is ideal for sending bulk emails at scale. For e.g. sending transactional emails when order is completed and they get receipt, sending marketing email to several users, social networking email when user invite their multiple friends etc.
- It is different email sending via Simple Notification Services. SNS is meant for sending email but not for bulk emails and there is no way to tracking what happen when the email is sent.
- SES is concerned with what happened when it leaves the outbox. All of complexities for sending email is handled by SES.
- We can send email using SMTP, single API call.
- We get our sandbox environment for experiment and our application is sending email appropriately, once we know everything is working then we can transition into production environment.
- SES is not available in every region.
- We can track the delivery status.
- We can track like: how may are opened, marked as spam, bounced, and deleted.
- There are sending limit in SES. This is put in place to ensure that Internet Service Provider (ISP) is not putting us in Blacklist. i.e. when there are large amount of email coming from same location then they might flag as spam.
- There is a Sending Quota. i.e. Maximum number of email in 24 hour period. Sandbox is 200/24hrs and prod is 10,000/24 hrs.
- There is Maximum Send Rate. i.e. Maximum number of emails per second. Sandbox is 1 email/ sec and prod is 5 emails/sec.
- The quota will be lifted automatically over period of time.

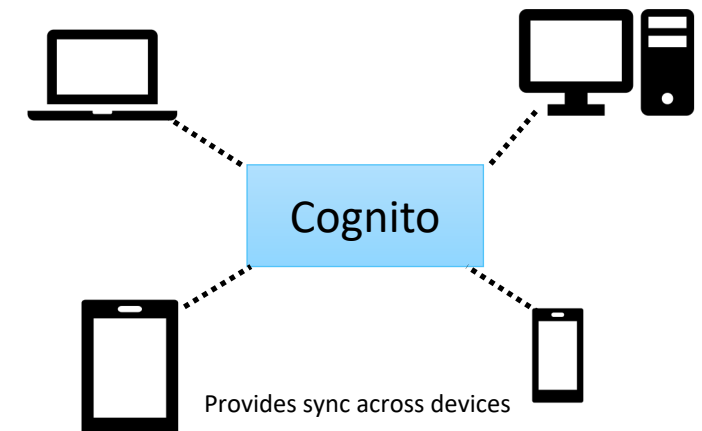
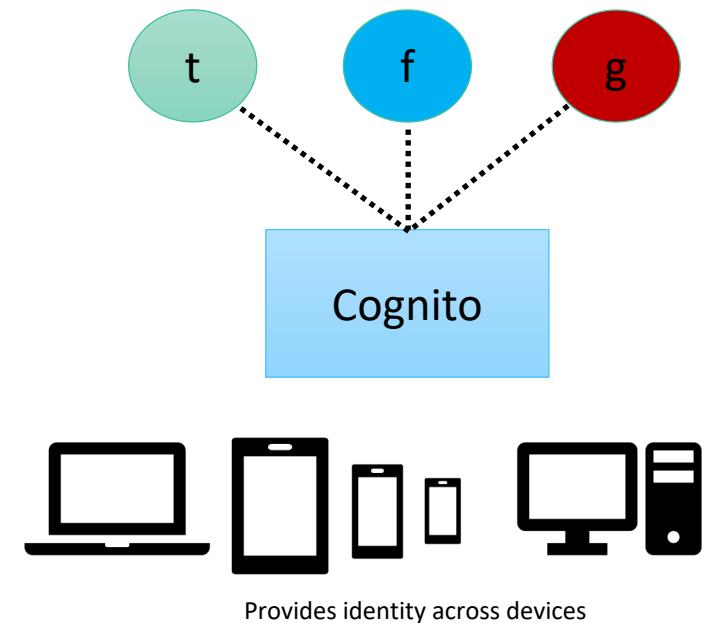
Amazon Cognito

Amazon Cognito Identity

- It helps on identifying by integration with major authentication providers such as google, twitter, facebook etc.
- It allows to implements single-sign-on or login with facebook, twitter on any device or platform.
- It also helps to manage unique identities across various platform and devices and helps implements temporary keys.

Amazon Cognito Sync

- Amazon Cognito is a tools that allows to sync data between various application and platforms. For e.g. Let say have different app data and that application is on desktop, mobile. Cognito will help to sync data between those devices. Like saving the status of the game we are playing, maintain the same preferences.
- It works across devices and operating systems.
- It also enable to work offline by storing data in local SQLite database. So, whether or not user is online or not, they can store their data locally then when they are online Amazon SDK can resync data back to Cognito service.



Amazon Pinpoint

- This service is previously known as Amazon Mobile Analytics.
- This service helps to measure the usage of mobile applications. For e.g. we have mobile apps, then we want to understand what the user is doing, what are the trends, what's the retention/ either they coming back, understand user behaviors. i.e. to monitor the application, how to make apps better. Instead of writing a service to do analytics, we can leverage Amazon Pinpoint.
- It is easy to integrate via AWS mobile SDK: i.e. iOS, Android, Fire OS, Unity.
- It also offers REST API.
- Once we start sending information to Pinpoint then we can see reports within 60 minutes.
- We own the collected data.
- Pinpoint collects three types of events:
 - System events: apps started, apps shutdown etc.
 - In-app purchases: apps purchased
 - Custom events: we define based on our need.