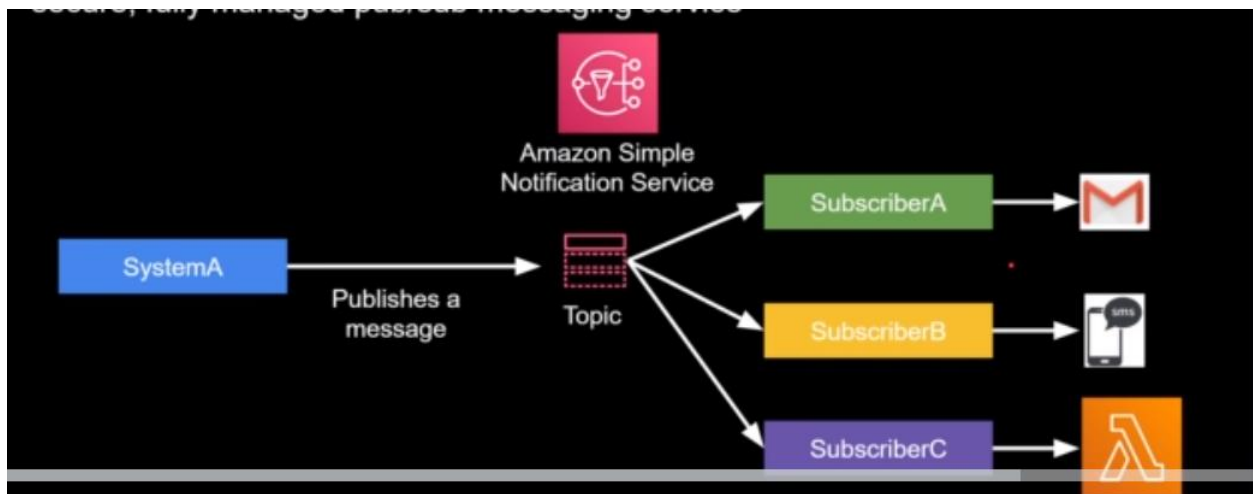


Designing using SNS, SQS Lambda

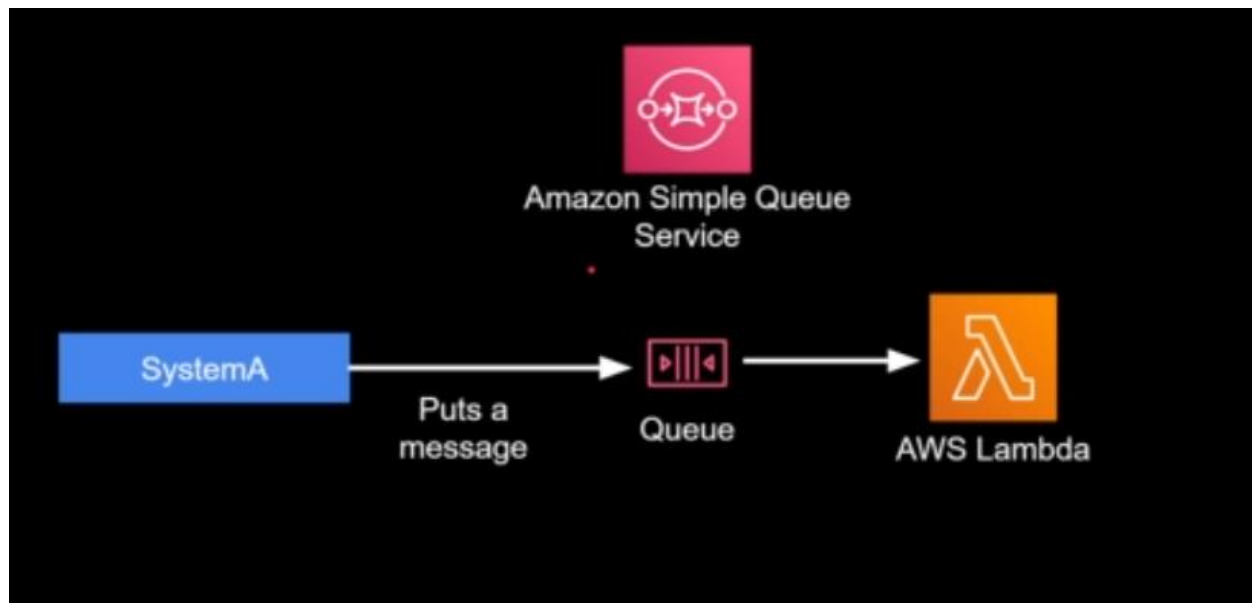
SNS: Simple notification service

Fully Managed pub/sub messaging service



1. Create Topic
2. Create subscribers and send message to different subscribers based on fields in message using Message Filtering
3. Fan Out Architecture: Can send same message to different subscribers.

SQS: Simple Queue Service : Message Queuing service



Standard Vs FIFO

Standard

- Order is not guaranteed
- Messages may be delivered more than once
- Nearly unlimited messages processed per second
- \$0.40/million requests/month after free tier

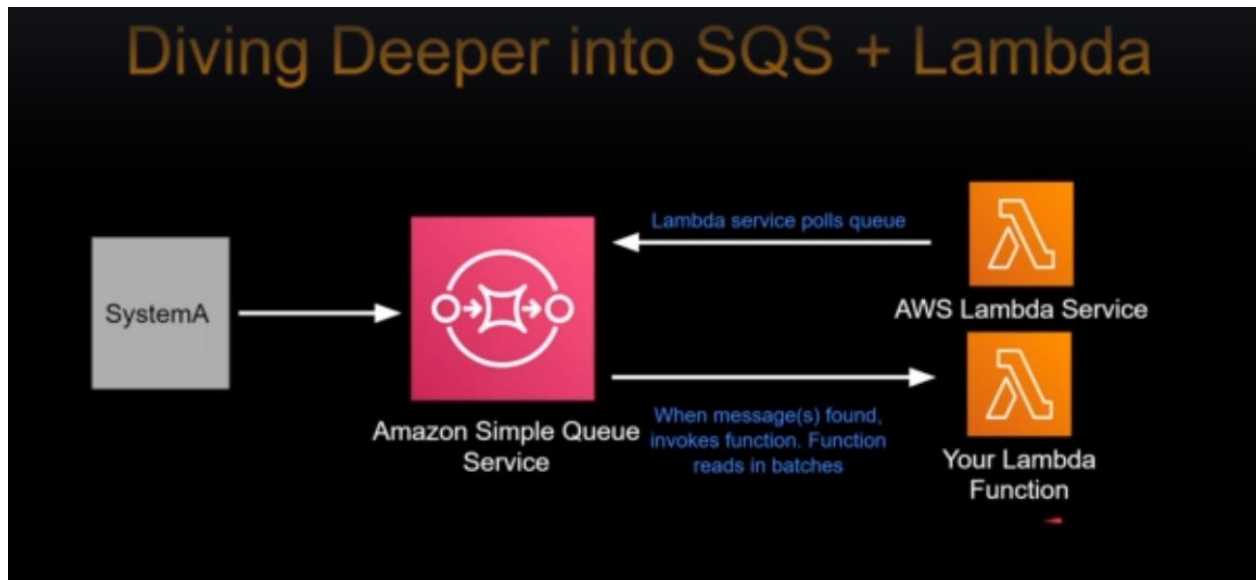
FIFO

- Order is strictly preserved
- Messages follow exactly once processing, dedup configuration avoids duplicate message delivery
- 300 messages/second. With batching supported upto 3000 message/second
- \$0.50/million requests/month after free tier

Difference between SNS and SQS:

1. Can retrieve failed messages from dead letter queue. In SNS after retry attempts, message gets discarded.
2. SQS can convert synchronous architecture to asynchronous.

3. In SQS, 1 message can't be delivered to multiple consumer as after being processed, message gets deleted.



Working of SQS and Lambda:

1. Lambda service polls for messages in SQS.
2. When messages are found, Lambda function invokes messages in batches.
3. Batch Size Limit is 10
4. $5 < \text{Polling Limit} < \text{Lambda concurrency limit}$
5. As per the polling limit, Lambda service will invoke that many copy of lambda functions and messages equal to queue size will get processed.
6. If all the messages in the batch is processed successfully, all messages are deleted from sqs queue.
7. If all the messages in the batch are failed, Lambda service will send all the messages back to SQS queue and a new lambda will be invoked.
8. If a batch has both successful and failed processed messages then Lambda service will still send back all the messages to be processed again. This causes duplicity as some messages can be reprocessed.
9. To avoid this: Instead of Lambda service deleting messages, use lambda code to delete messages i.e. your function can successfully delete the messages that has been processed successfully. So only unprocessed messages will be reverted back to sqs.

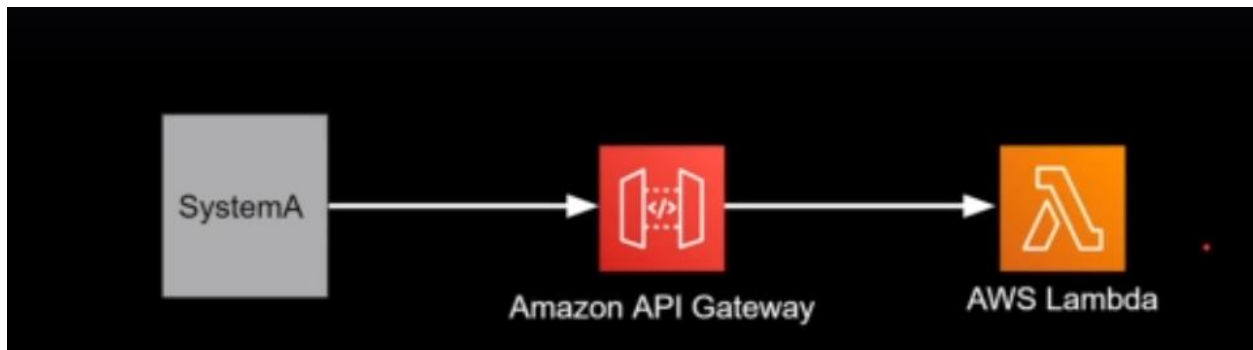
Architecture 1: Convert Sync to Async for High traffic

Disadvantages of Sync Architecture:

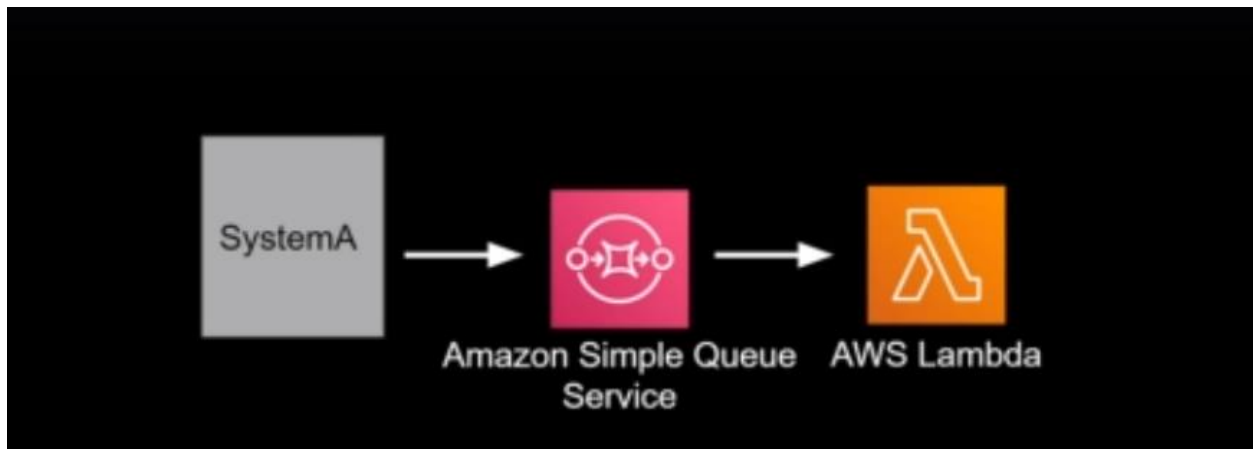
1. In Sync architecture, all components need to scale together.
2. Scaling is as high as scaling of lowest scalable component.

3. If one component fails whole call fails.

Sync Architecture

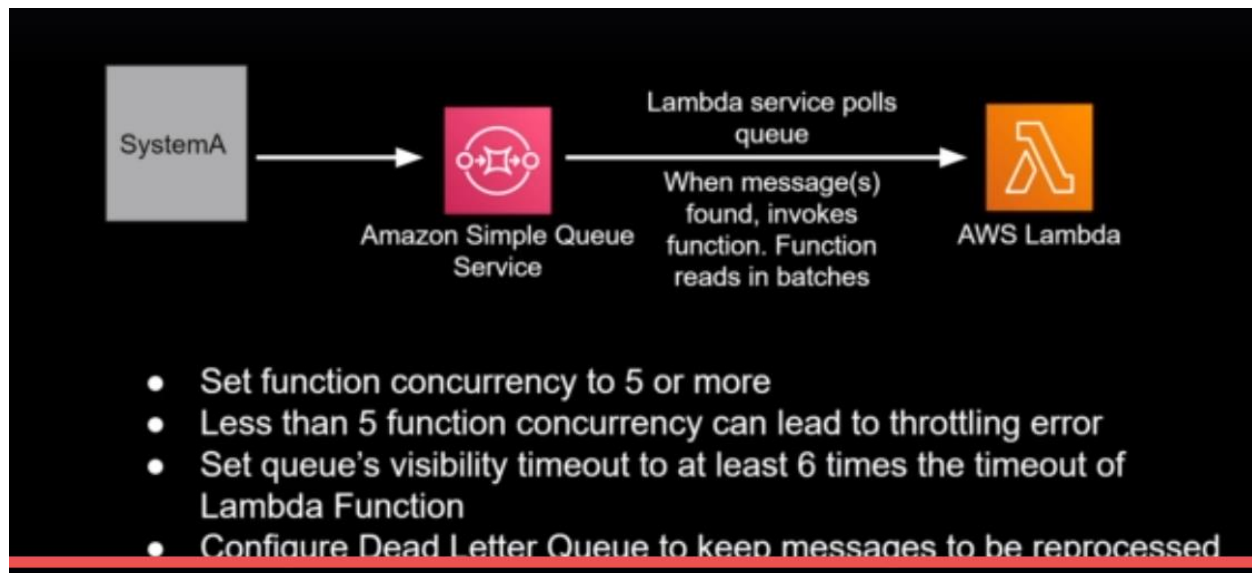


Async Architecture



Advantages of Async Architecture:

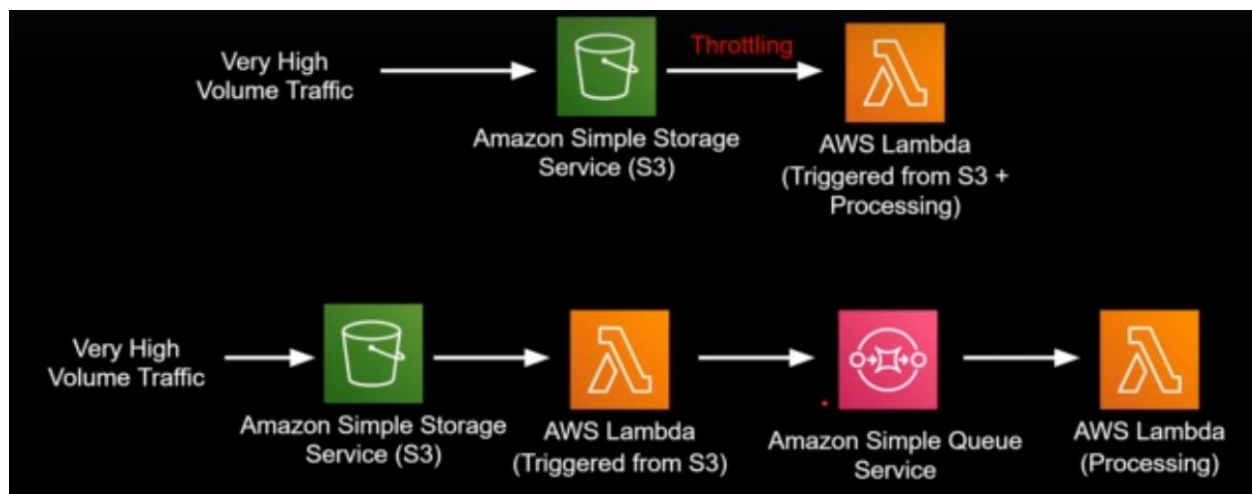
1. All components scale separately.
2. Instead of System A calling Lambda directly, System A sends messages to Queue.
3. Can control traffic to downstream



GET APIs can be sync and POST API can be asyn.

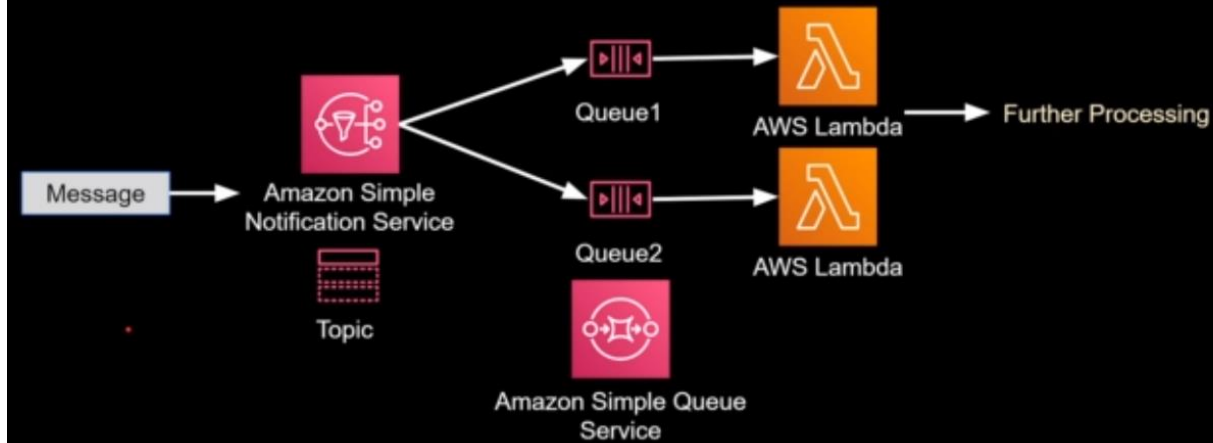
USECASE 2:

High Volume S3 Processing



Usecase 3:

Reliable Fanout Architecture



SNS Message Filtering:

To process each message differently

For regular topic, anytime a message comes, it gets delivered to all the subscriptions.

If we include message filtering then based on message content, you can selectively trigger each subscription.

Diving Deeper into SQS + Lambda

