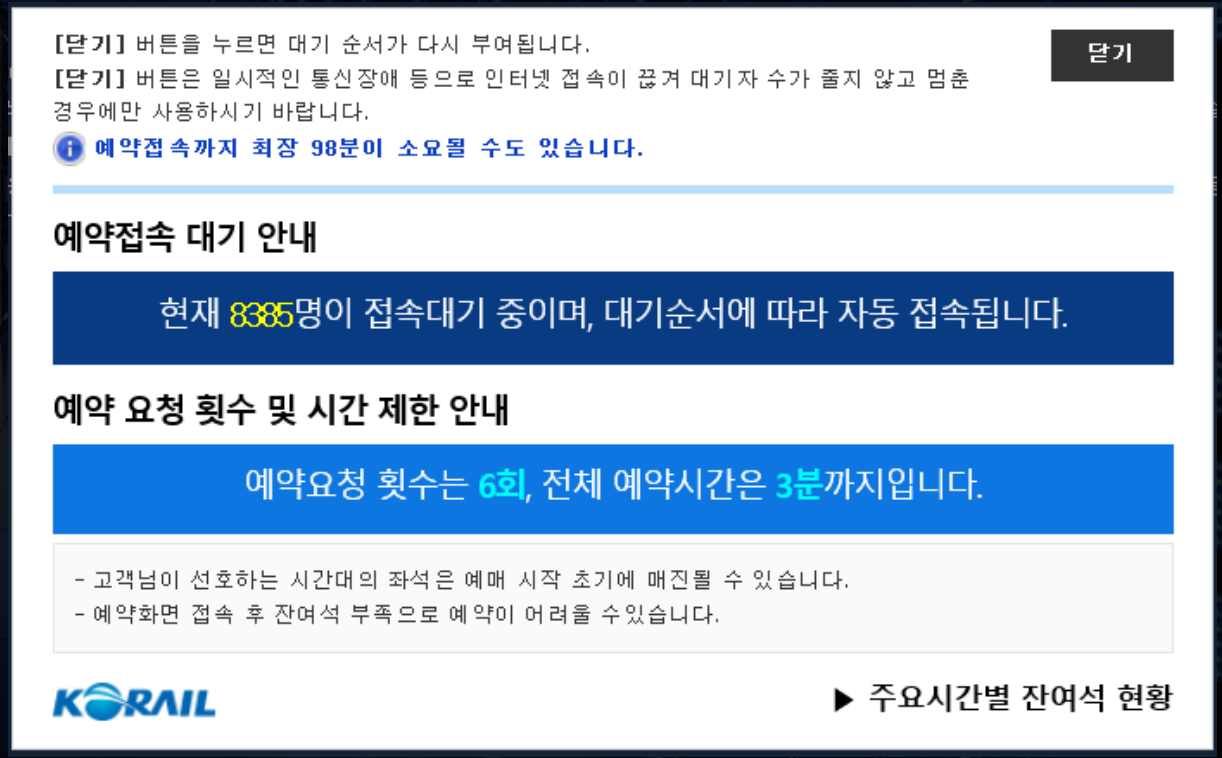

AWS Serverless 기반 아키텍처링 및 개발
“Demo – Serverless 기반 대기열”
RubyWave

# › 목표

## › 목표

- 명절 KTX 예매와 같은 대기열 구성
  - 동시에 X명만 예약 가능
  - 각 사람 당 Y시간 만큼만 예약 가능
  - N초 단위로 대기중인 사람 숫자 갱신
- Serverless서비스로만 구성



# 기본 아키텍처 요구사항

## 클라이언트와 실시간 통신

- 만료시간 알림, 실시간 체크 등 실시간 통신 필요
  - API Gateway의 Websocket 활용

## 유저 데이터 저장

- Websocket Connection ID 등 저장을 위해서 DynamoDB 사용


DynamoDB
Counter Table

DynamoDB
Connection Table

# “동시에 x 명만 예약 가능”

## “동시에 x 명만 예약 가능”

- SQS를 매인 대기열로 정하고 x개의 메시지만 가져와서 처리
  - 문제: SQS를 Lambda로 연동해서 사용할 경우 “한 번에 x개만 가져오기” 기능이 없음
    - Lambda Concurrency를 설정하더라도, Lambda와 SQS를 가져오는 로직은 분리되어 있음
    - 따라서 Message가 In-Flight 상태로 처리 불가능한 상태로 변함
    - 예: 메시지가 15개 있다면, Lambda의 Concurrency를 1로 설정해도 한 개는 처리, 나머지 14개는 In-Flight으로 변함(Pulling 후 처리 불가능 판정)
  - 해결 방법 : FIFO Queue의 MessageGroupId 활용

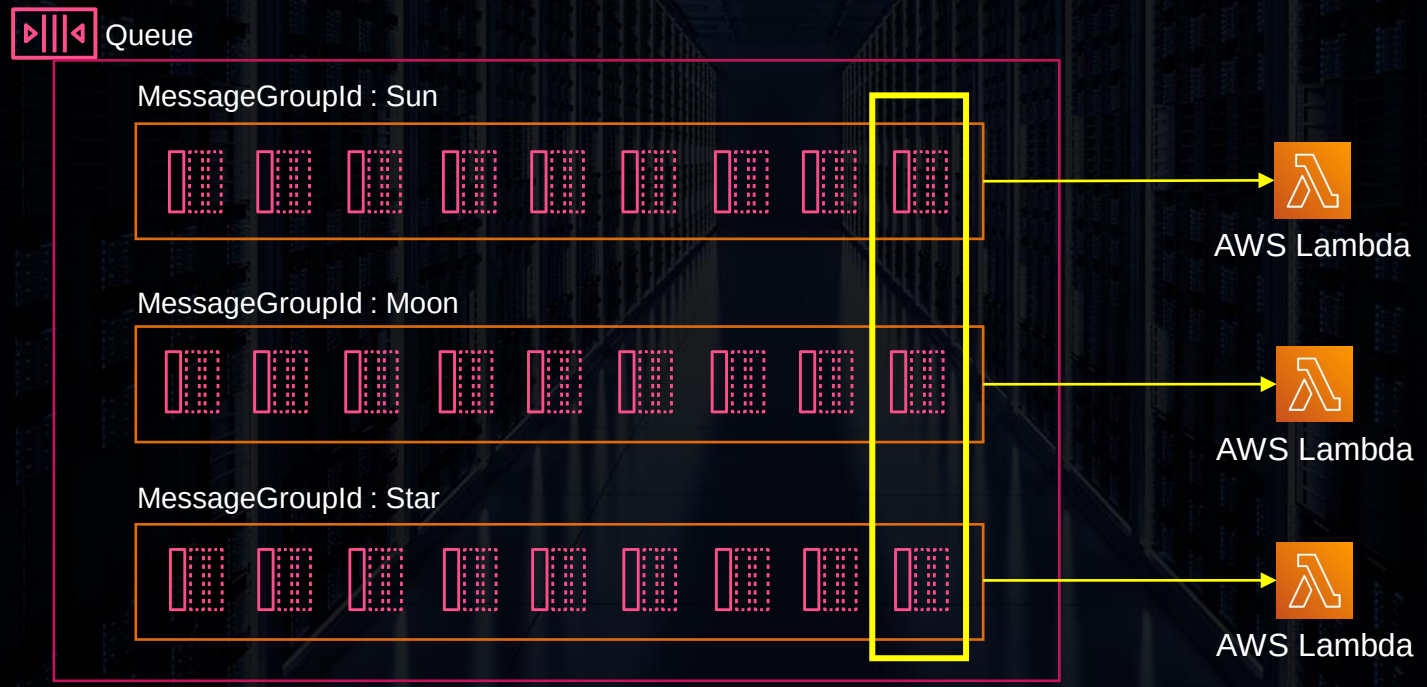
# SQS의 MessageGroupId

## SQS의 MessageGroupId

- FIFO모드에서만 동작하는 일종의 SQS안의 채널
- 같은 MessageGroupId는 동시에 한번만 처리 가능
  - 즉 아무리 Consumer가 많아도 현재 특정 MessageGroupId가 처리중이라면, 나머지 같은 MessageGroupId를 가진 쌓여있는 메시지는 Receive 불가능
- 즉 MessageGroupId의  종류 만큼만 동시에 처리 할 수 있도록 제어 가능

# SQS의 MessageGroupId

# SQS의 MessageGroupId

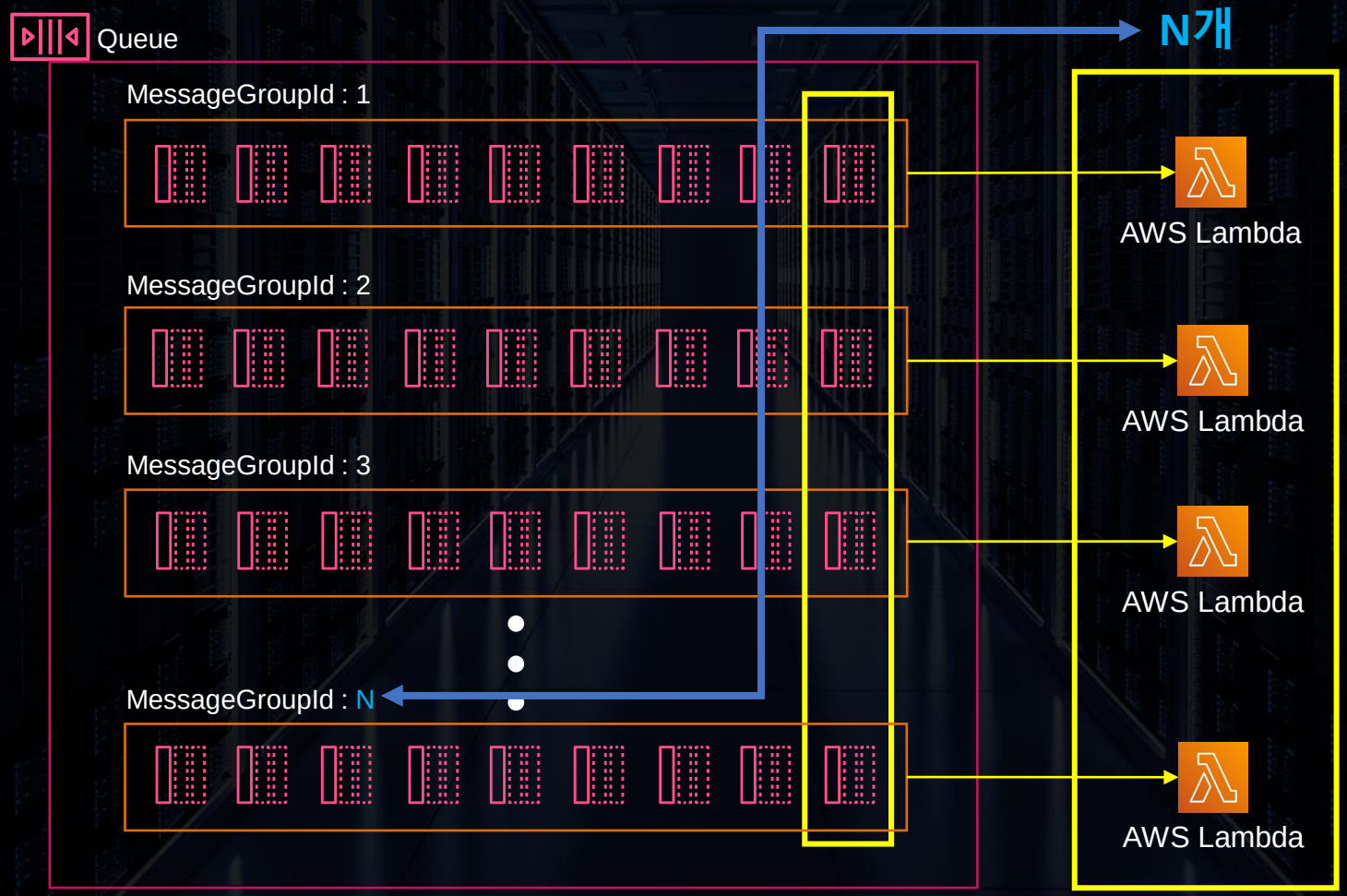


# › “동시에 x 명만 예약 가능”

## › “동시에 x 명만 예약 가능”

- 한번에 처리 가능한 유저 숫자를 정하고, 들어오는 순서대로 MessageGroupId를 부여해서 SQS에 넣기
  - 유저 카운터는 DynamoDB에 저장(Increment)
  - MessageGroupId  = 유저 카운터%(한번에 처리할 숫자)
  - 예: 한번에 처리할 숫자가 5라면, 256번째 유저는 MessageGroupId = 1
- 등록이 끝나면 Queue에서 메시지 삭제(HTTP기반 Lambda 호출) -> 다음 MessageGroupId에 해당하는 대기열 처리 시작

# SQS의 MessageGroupID





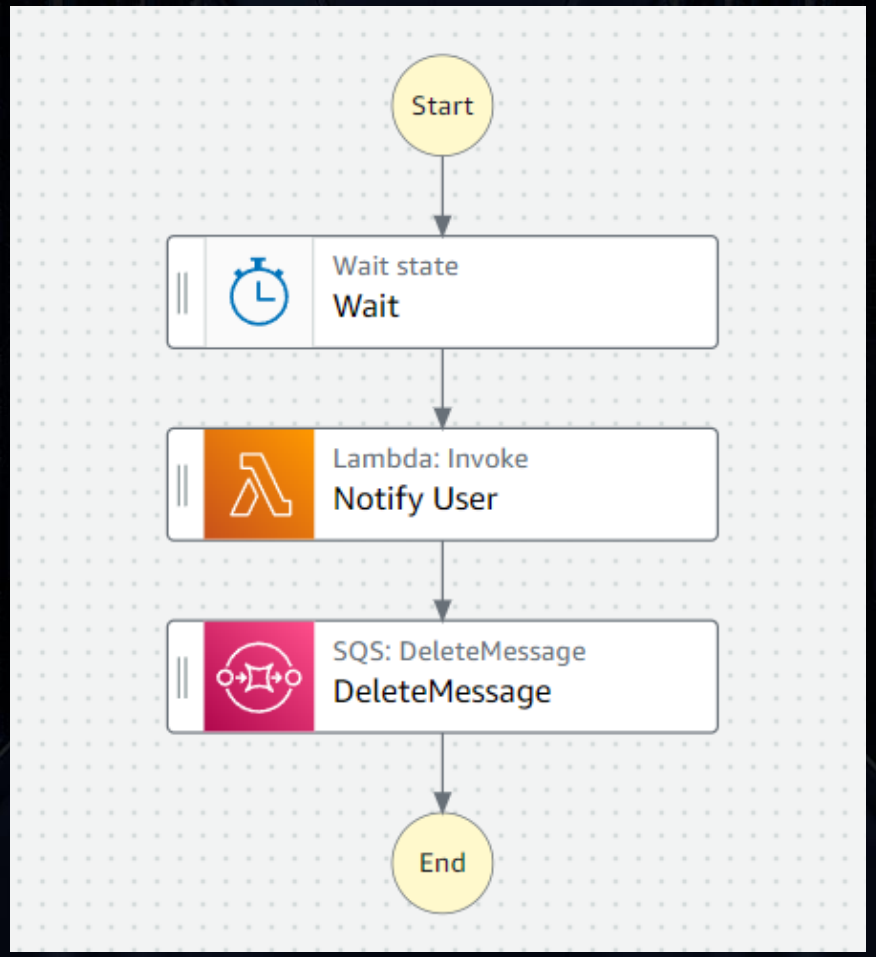

## Demo – Serverless 기반 대기열

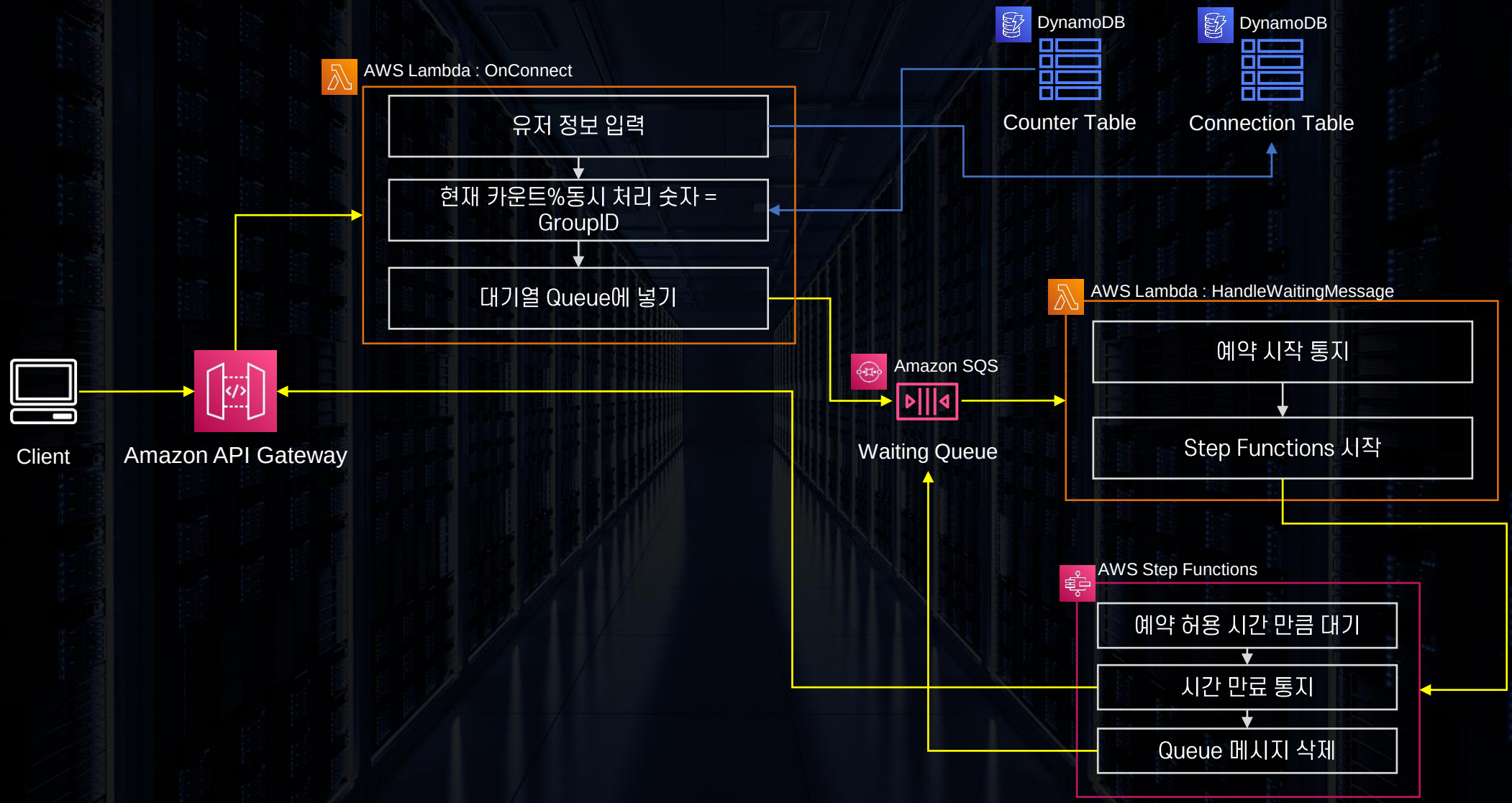
# › “한 사람당 Y시간 만큼 예약 가능”

## › “한 사람당 Y시간 만큼 예약 가능”

- 처음 Queue메세지를 처리할 때(대기열이 돌아왔을 때) 시작 통지
- 일정 기간 동안 기다린 후, 서버에서 알림 전달
- Lambda의 실행 시간은 최대 15분 : 15분 이상 기다려야 한다면?
  - Step Functions
- Step Functions의 역할
  - Y초동안 대기
  - 이후 유저에게 만료 알림(Websocket)
  - 이후 SQS 메시지 삭제

# › “한 사람당 Y시간 만큼 예약 가능”








## Demo – Serverless 기반 대기열

# SQS와 Lambda

## SQS와 Lambda

- SQS에서 Lambda를 직접 트리거할 경우 Lambda가 종료될 때 메시지 자동 삭제
  - 즉 직접 삭제하지 않더라도 삭제됨
- 문제점
  - 우리는 Lambda에서 메시지를 삭제하지 않고 StepFunctions에서 대기 후 삭제 예정
  - 하지만 Stepfunction실행 후 Lambda가 종료되면 메시지가 삭제됨
  - 해결 방법:  sqs:DeleteMessage Deny!

When Lambda reads a batch, the messages stay in the queue but are hidden for the length of the queue's visibility timeout. If your function successfully processes the batch, Lambda deletes the messages from the queue. By default, if your function encounters an error while processing a batch, all messages in that batch become visible in the queue again. For this reason, your function code must be able to process the same message multiple times without unintended side effects. You can modify this reprocessing behavior by including batch item failures in your function response.

# SQS와 Lambda

```
Version: '2012-10-17'
Statement:
  - Effect: Allow
    Action:
      - s3:*
      - sqs:*
      - logs:*
      - states:*
      - execute-api:*
    Resource: '*'
  - Effect: Deny
    Action:
      - sqs:DeleteMessage
    Resource:
      Fn::GetAtt:
        - WaitingQueue
        - Arn
```

# > SQS와 Lambda

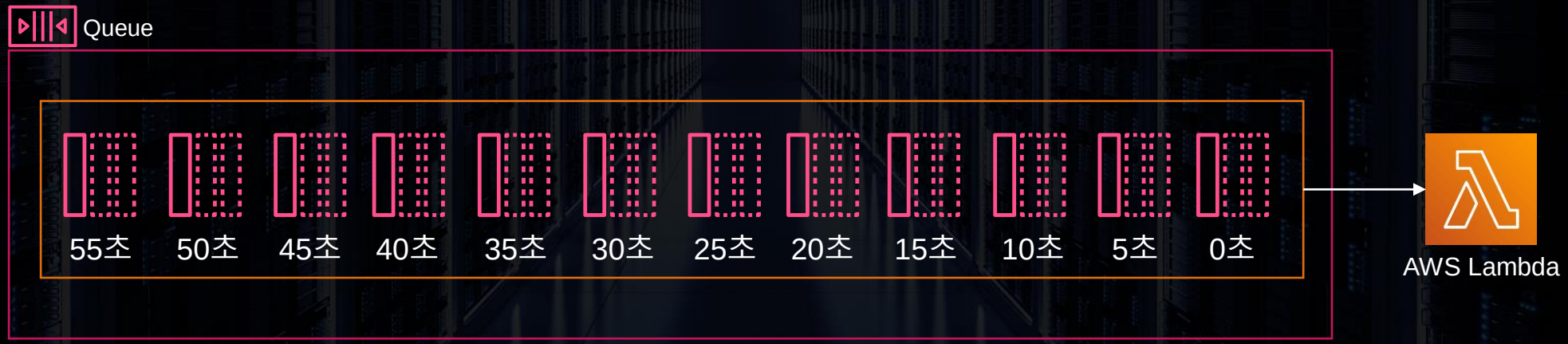
```
Error:
UPDATE_FAILED: SqsUnderscorehandleUnderscorewaitingUnderscoremessageEventSourceMappingSQSWaiti
ngQueue (AWS::Lambda::EventSourceMapping)
Resource handler returned message: "Invalid request provided: The provided execution role does
 not have permissions to call DeleteMessage on SQS (Service: Lambda, Status Code: 400, Request
 ID: 6d2b5e91-1d7b-405a-9f9c-c81f5bb2c2fd, Extended Request ID: null)" (RequestToken: e563f052
-5f51-b089-85da-34d441d41f12, HandlerErrorCode: InvalidRequest)
```

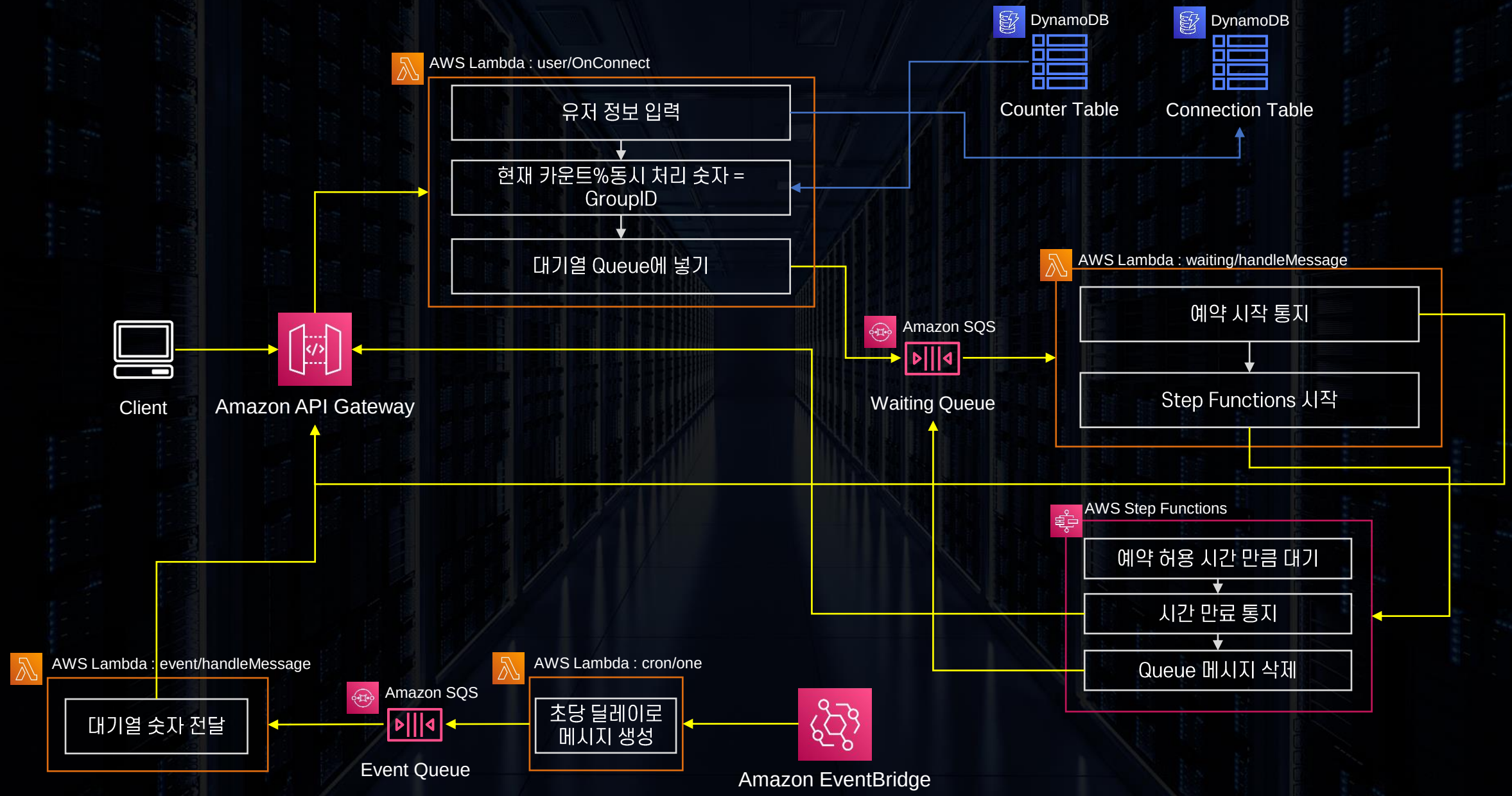
# › “5초 단위로 대기 중인 유저 숫자 알림”

## › “5초 단위로 대기 중인 유저 숫자 알림”

- EventBridge Cron 이벤트로 일정 시간마다 대기열 숫자(SQS ApproximateNumberOfMessages) 전달
- 문제점 : EventBridge Cron 이벤트는 최소 단위가 1분단위
  - 해결방법: SQS의 DelaySeconds 활용
- DelaySeconds: x초 후에 메시지 처리 가능
  - 즉 1분 마다 5,10,15,20,25,30,35,40,45,50,60 초 딜레이를 가진 메시지 12개를 Queue에 넣어 처리
  - 해당 메시지가 처리될 때 마다 모든 유저에게 현재 대기열 숫자 업데이트

# SQS의 DelaySeconds








# Demo – Serverless 기반 대기열


rubywave.io

spark@rubywave.io