Visibility Timeout and DLQ

1. Visibility Timeout

The purpose of "Visibility Timeout" in the cloud is to ensure reliable message processing by preventing duplicate handling of the same order.

How It Works:

- When Lambda retrieves a message from SQS, the message becomes *invisible* for a set duration, for example for 30 seconds. If Lambda processes the message successfully, it deletes the message from the queue and gets inserted to the dynamoDB database, In the the case of assignment 2 it will be inserted to the Orders table in dynamoDB, and if Lambda fails due to DynamoDB throttling or errors, the message reappears after the timeout for reprocessing.

Why is it useful?:

- Prevents **duplicate orders** from being processed simultaneously.
- Gives Lambda time to recover from failures such as temporary network issues.
- Aligns with the assignment's requirement to *"handle failures reliably."*

---

2. Dead-Letter Queue (DLQ)

Purpose: Isolates messages that repeatedly fail processing for manual review.

How It Works:

- SQS moves messages to the DLQ after **3 failed attempts** which is configurable in the cloud using the maxReceiveCount.
- Some common failures are: Invalid order data, Lambda timeouts, or DynamoDB errors.

Why Useful:

- **Prevents infinite retries** that could clog or cause the system to lag.
- Allows **admins to debug** failed orders.
- Meets the assignment's goal to *"handle failures reliably"* by ensuring no orders are lost.

Combined Workflow Example based on the assignment

1. **Lambda fails** to process an order for example based on my code if the orderId is missing it will fail.
2. SQS retries with a total of **3 attempts**, per maxReceiveCount=3.
3. After the third failure, SQS **moves the message to DLQ**.
4. The Admin should investigate the DLQ, fix the issue, and resubmit the order.

Why This Matters for the Assignment

- **Visibility Timeout**: Ensures each order is processed **exactly once**, critical for e-commerce systems to process orders.
- **DLQ**: Acts as a safety net for **edge cases** such as corrupted data, fulfilling the requirement for reliability.