Components

1) Node server
2) Dynamo DB
3) SQS queue
4) Schedules
5) Lambda function
6) Payment Webhook

Amazon Flash sale Architecture :-

Node.js app -> User will place an order using "/order" API

-> Based on the inventory stock count it will place / reject the order and store the details in Dynamo DB.

-> If payment successful the the order details will go to SQS Queue

-> Now for every successful order in the SQS queue a different application will fetch those orders from SQS queue and will process the post-order processing, which we should not bother at this point, as our main concern is to book as many order as possible, we should not worry about post-order process.

SO HAPPY CASE IS SORTED

Handling failed and Pending Cases

1) Failed case is straight forward, after payment request failed just mark the status 5 (Failed)

2) Pending Case : If payment request is pending
↓
A scheduler will run every 2 mins, to find pending orders
↓
if pending order create_at > 2 mins
↳ Mark status -4 (Rejected timeout)
↳ Restore inventory

3) Payment Gateway Webhook / Callback
!
!-- On payment Success :
|-- If orderstatus 1 (Pending) → Push to SQS
|- If orderstatus 4 (timed-out) → initiate a refund
|- if already processed → ignore

--- On Payment Failed :
↳ Mark it failed and restore inventory