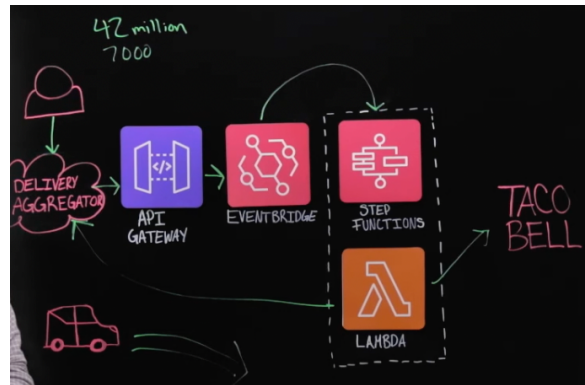


Company Name: **Taco Bell**

Video Link: <https://www.youtube.com/watch?v=sezX7CSbXTg>

Solution: **Order Middleware - Enabling Delivery Orders at Massive Scale**

An exploration of using a **serverless pipeline** from a **customer order** (perhaps on a mobile phone) to the **restaurant**, to the **delivery** driver. This architecture integrates API Gateway, EventBridge, Lambda, Step-Functions and geofencing to fulfill orders on time at scale at low cost.



Customer order the food using **mobile app or web application** (Business logic via Delivery Aggregator). It will send the events of new order made to an **API gateway(http API)**. It forward events to **Event-Bridge**. It takes a routing system and make sure that it's a new order. It will integrate with **STEP Functions** which trigger workflow. It will call a series of **Lambda functions**.

First lambda communicates with restaurants and making sure that they are available or accepting the order or not (**POS [point-of-sale]** technology). If everything is ok, another lambda functions sent an event to Aggregator (mobile app) and let them know that they will make the order.

STEP function sitting and waiting for the acknowledgement from the **restaurants**. Parallely sent notifications to all **drivers** who are ready to deliver the food. In order to solve the waiting issue, using the **GPS technology**, another Lambda can trigger and let restaurant know that the driver is on the way and the distances.

AWS SERVICES mentioned:

1. **Amazon API Gateway** Create, maintain, and secure APIs at any scale. <https://aws.amazon.com/api-gateway/>
2. **Amazon EventBridge**: Build event-driven applications at scale across AWS, existing systems, or SaaS apps. <https://aws.amazon.com/eventbridge/>
3. **AWS Step Functions**: Visual workflows for modern applications. <https://aws.amazon.com/step-functions>
4. **AWS Lambda**: Run code without thinking about servers or clusters. <https://aws.amazon.com/lambda/>