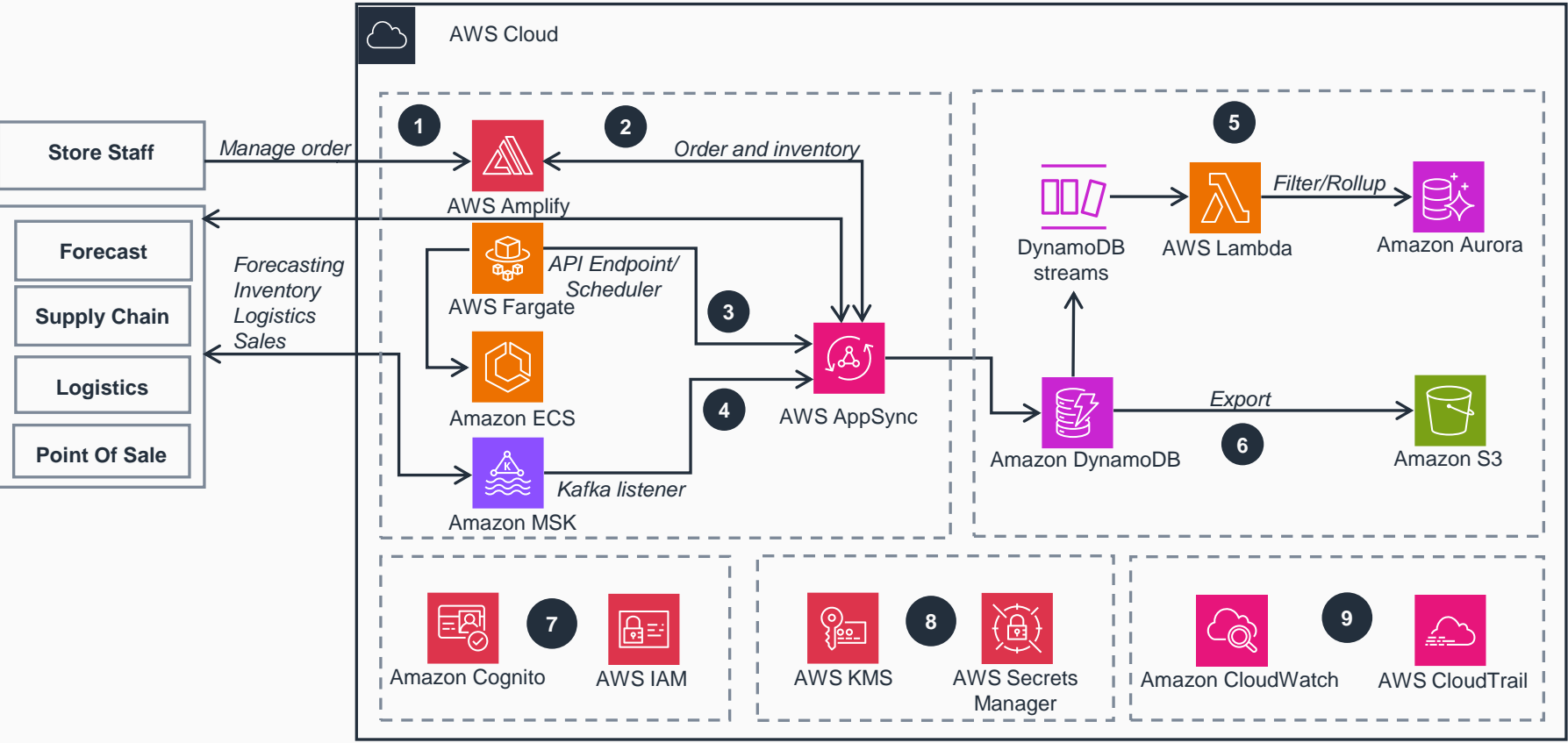


Guidance for Implementing Order and Inventory Management for Quick Service Restaurants on AWS

This architecture diagram demonstrates how you can build a serverless order and inventory management system to enable integration across stores, distribution centers, and logistics systems at scale.



- 1 Store staff use a web or tablet application running on **AWS Amplify** to perform online and offline updates to manage orders.
- 2 **AWS AppSync** supports online and offline client updates, as well as backend updates received from **AWS Fargate** and **Amazon Managed Streaming for Apache Kafka (Amazon MSK)**.
- 3 **Fargate** hosts API endpoints and a scheduler running on **Amazon Elastic Container Service (Amazon ECS)**, along with an **Amazon MSK** Listener component to read from and update **Amazon DynamoDB**.
- 4 **DynamoDB** receives sales, inventory, forecast, and logistics data through **Amazon MSK**.
- 5 **DynamoDB** uses **DynamoDB Streams** to trigger **AWS Lambda** functions based on new order events. The **Lambda** functions then filter and aggregate the data before storing it in the **Amazon Aurora** database.
- 6 **Amazon Simple Storage Service (Amazon S3)** receives historical data from **DynamoDB**, using its export capabilities, for analytics and reference purposes.
- 7 **Amazon Cognito** and **AWS Identity and Access Management (IAM)** are used for authentication and authorization needs.
- 8 **AWS Key Management Service (AWS KMS)** and **AWS Secrets Manager** are used for certificate and secret management.
- 9 **Amazon Cloudwatch** and **AWS CloudTrail** are used for monitoring and audit needs.