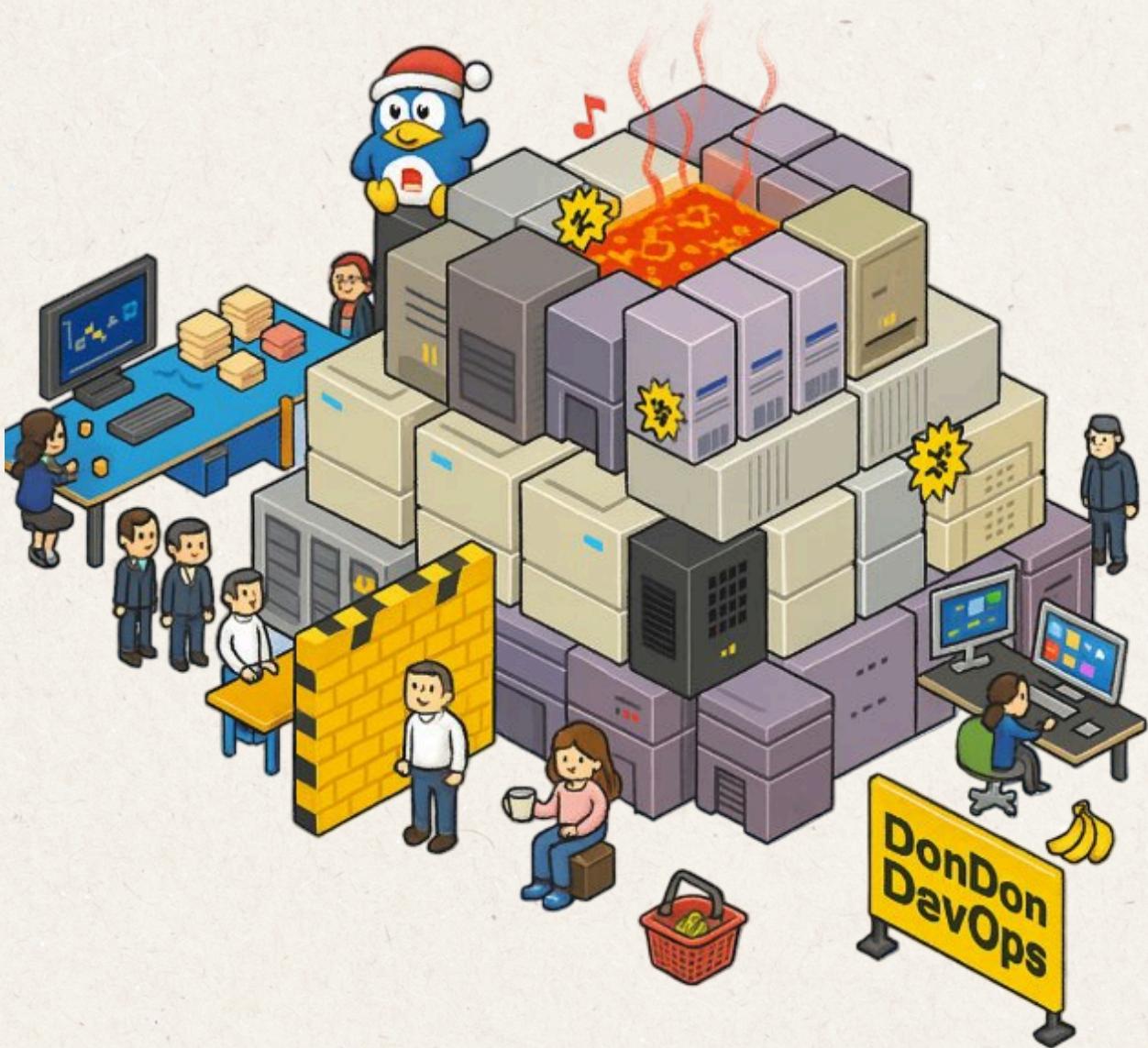




CS302 G1T8 (Team DonDonDonki)

DEMO 2:  
HASHCAT

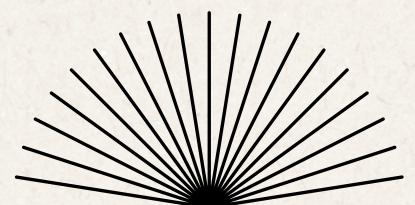


# DONDON DEVOPS

“I’m in a paradise, it’s a wonderful place for me”

PROJECT:  
**DonDonDevOps**

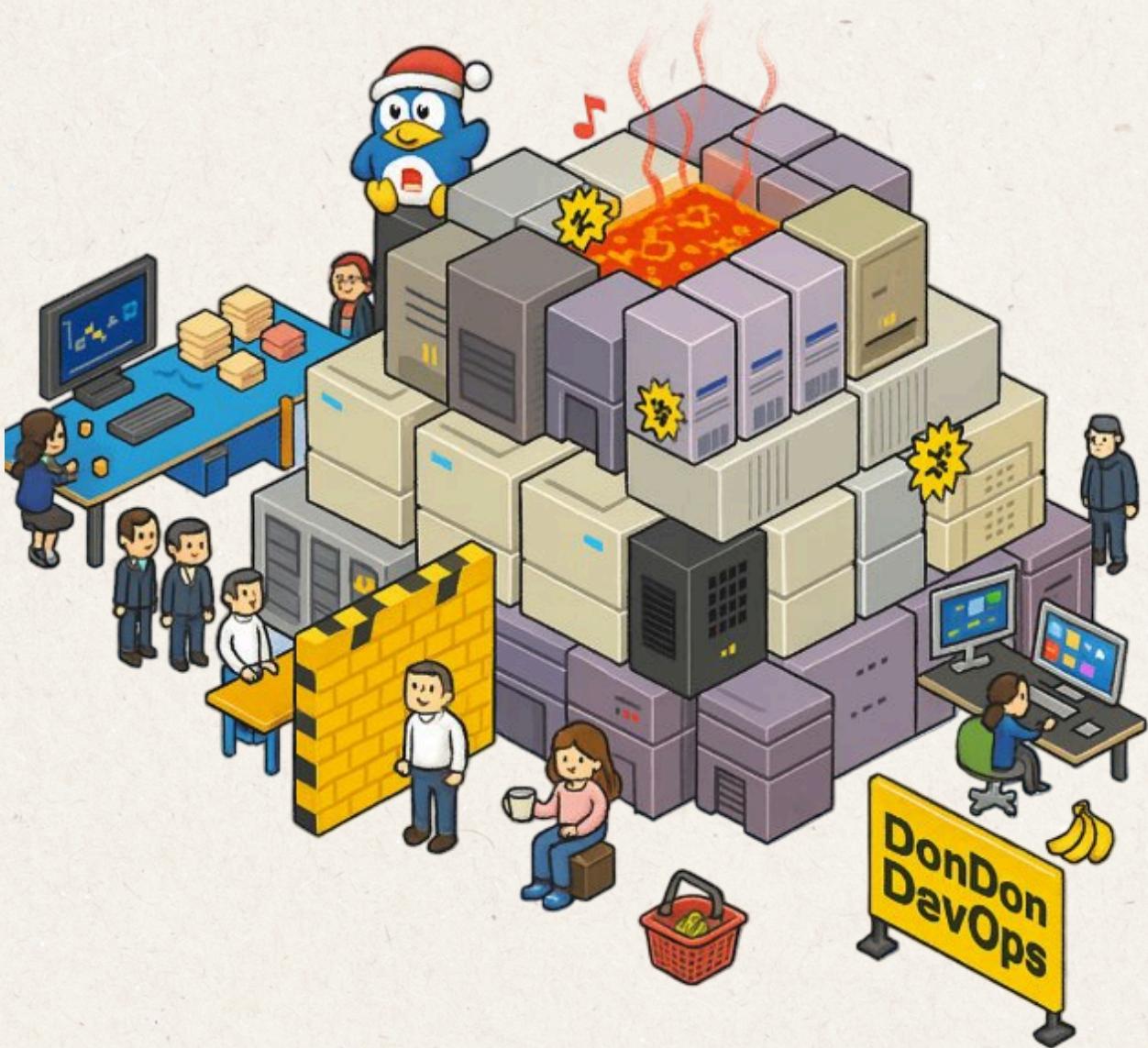
PRESENTED BY:  
**G1T8 Team DonDonDonki** (aka International Chefs):  
Alson Sim Wei Jie, Bill Johnathan, Cody Jeow Teng Xiang, Jeremy Lim Jin Zhai,  
Ng Guo Feng Eric, Sim Kah Hong





CS302 G1T8 (Team DonDonDonki)

DEMO 2:  
HASHCAT

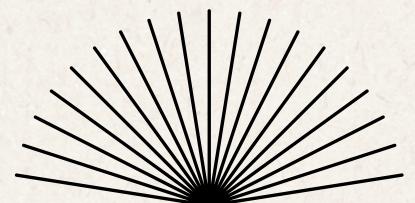


# DONDON DEVOPS

“I’m in a paradise, it’s a wonderful place for me”

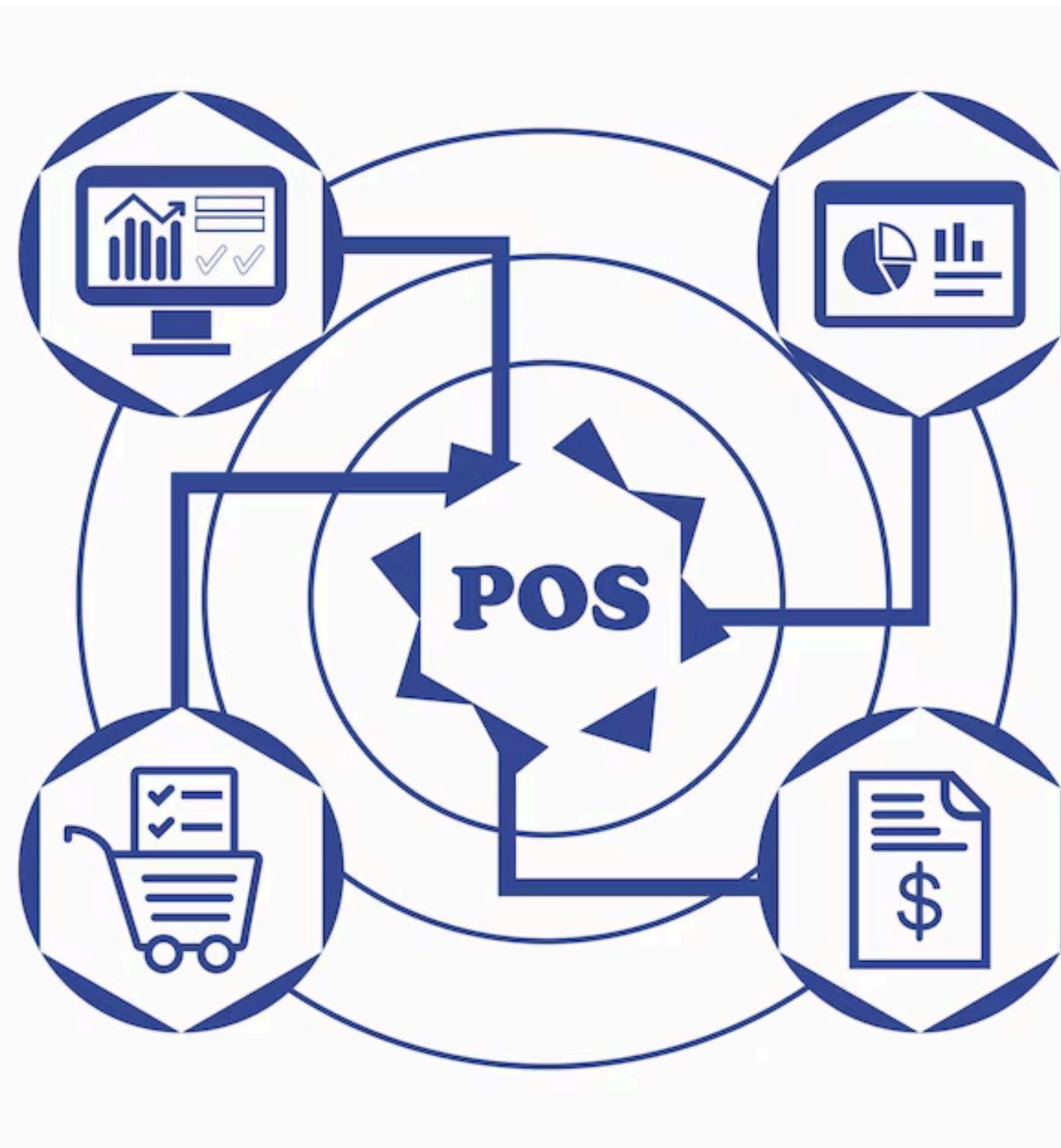
PROJECT:  
**DonDonDevOps**

PRESENTED BY:  
**G1T8 Team DonDonDonki** (aka International Chefs):  
Alson Sim Wei Jie, Bill Johnathan, Cody Jeow Teng Xiang, Jeremy Lim Jin Zhai,  
Ng Guo Feng Eric, Sim Kah Hong



# Overview

Point-of-Sale (POS) Solution



**DonDonDevOps** is a web application that assists with point-of-sale inventory management, enabling grocery stores to sync their real stock with a web interface, automate re-stocking, and **streamline the sale process**, from shelf-to-sale.

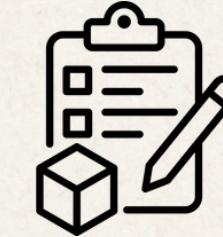
# Core Features

These are critical features that our solution encompasses to contribute to solving user pain points in the context of point-of-sales (POS) transactions.



## Self-Checkout System

Reduce waiting time by introducing self-checkouts



## Inventory Management System

CRUD Features for inventory management of in-store products

# Self-Checkout System

## Problem Statement

Traditional supermarket checkout counters cause **long queues**, especially during peak hours. Limited manpower further constrains the number of open counters.



## Solution

Developed a self-checkout software system that enables customers to complete their purchase independently. This **reduces wait times**, improves store efficiency, and enhances the overall shopping experience.



# Inventory Management

## Problem Statement

Manual inventory tracking often leads to **missed expiries**, and inefficient stock management. Store managers struggle to manage a variety of different products with different demands and shelf-lives.



## Solution

An inventory management system that **tracks** product stock level, expiries, and promotions. **Automatically alerts** managers to expiring products and triggers promotional pricing.

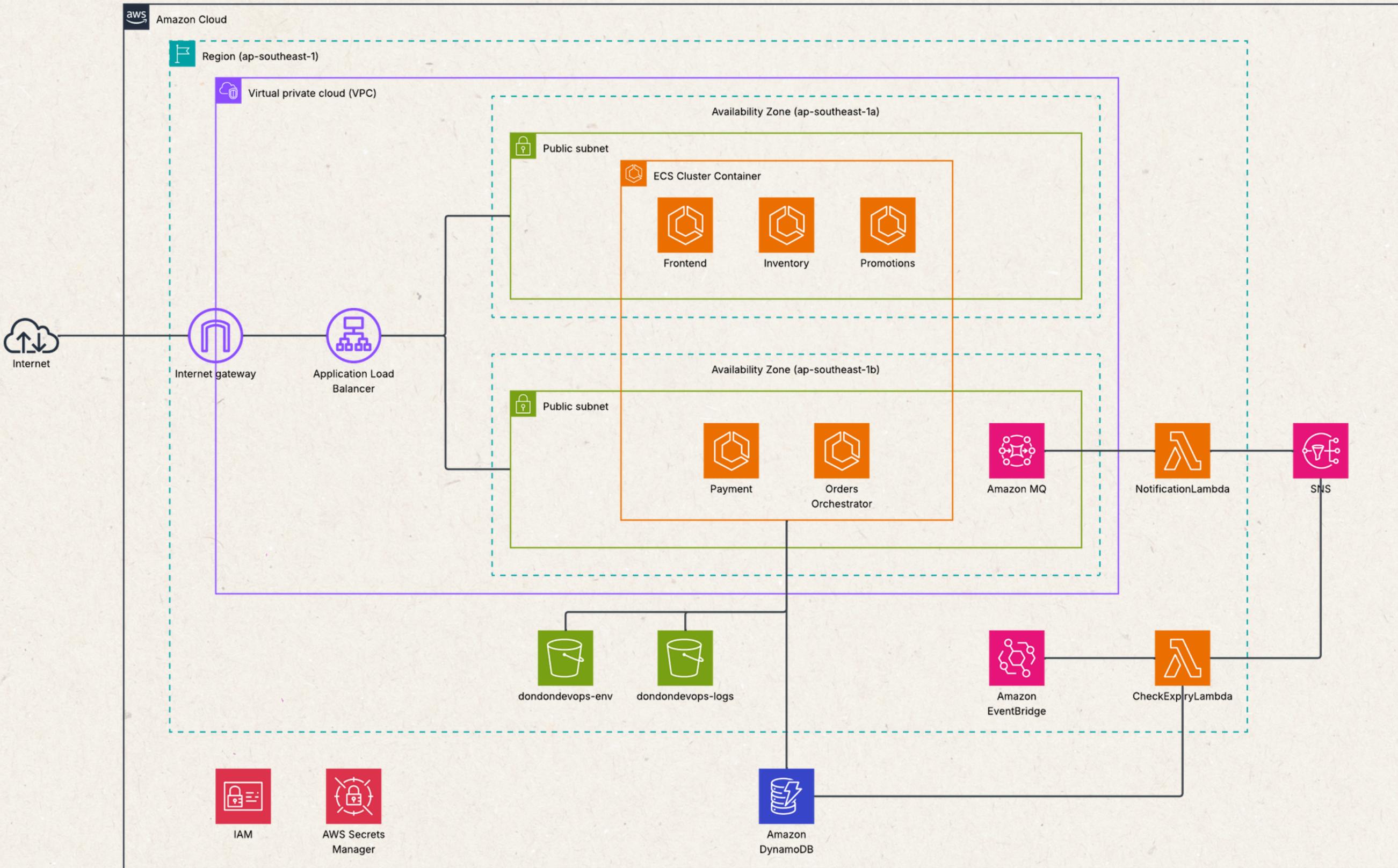
Name	Expiry	Actions
Milk	28/11/2026	
Ice Cream	10/05/2026	
Milk	22/01/2026	
Salmon Fillet	29/05/2026	
Beef Steak	04/06/2026	
Laundry Detergent	31/03/2026	
Croissant	28/06/2026	
Dog Food	01/12/2025	
Ice Cream	01/08/2026	
Olive Oil	20/05/2026	



# Architecture



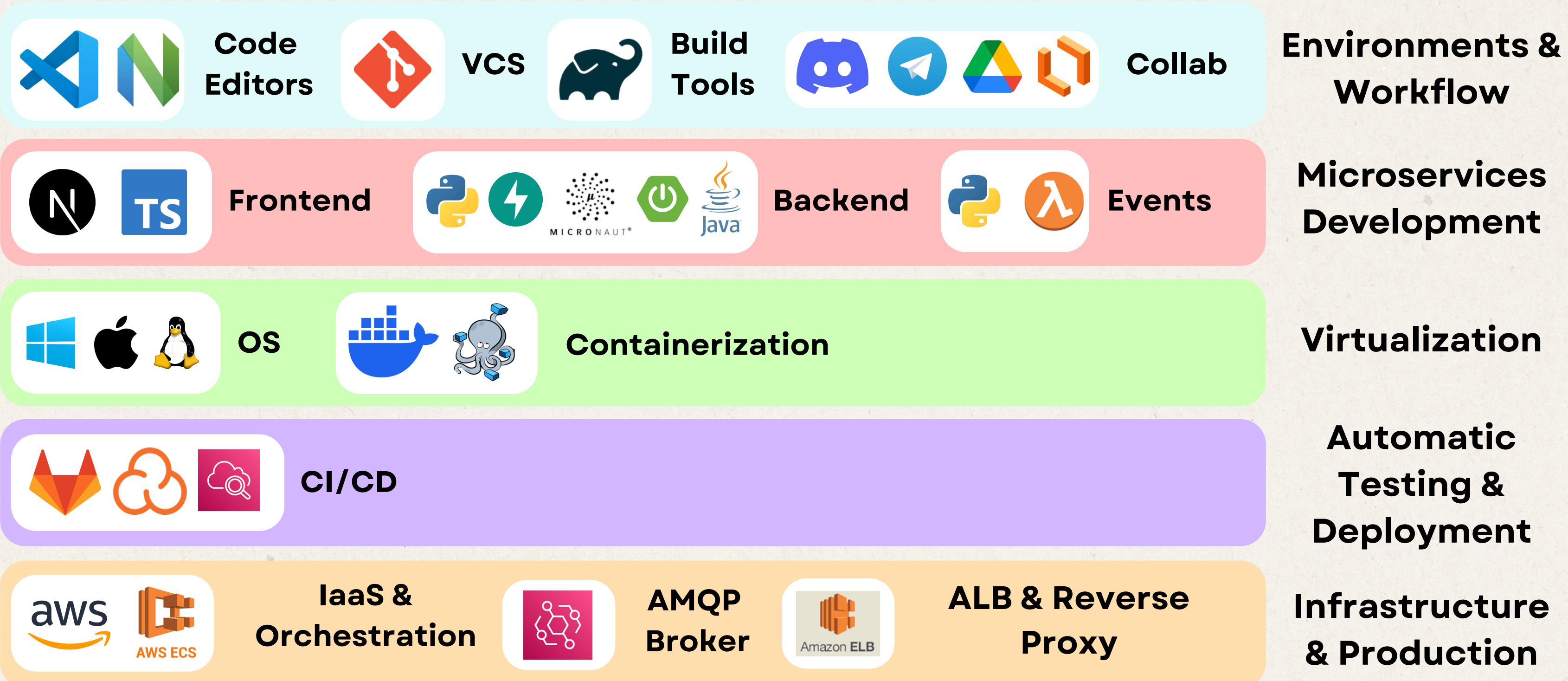
06/21





07/21

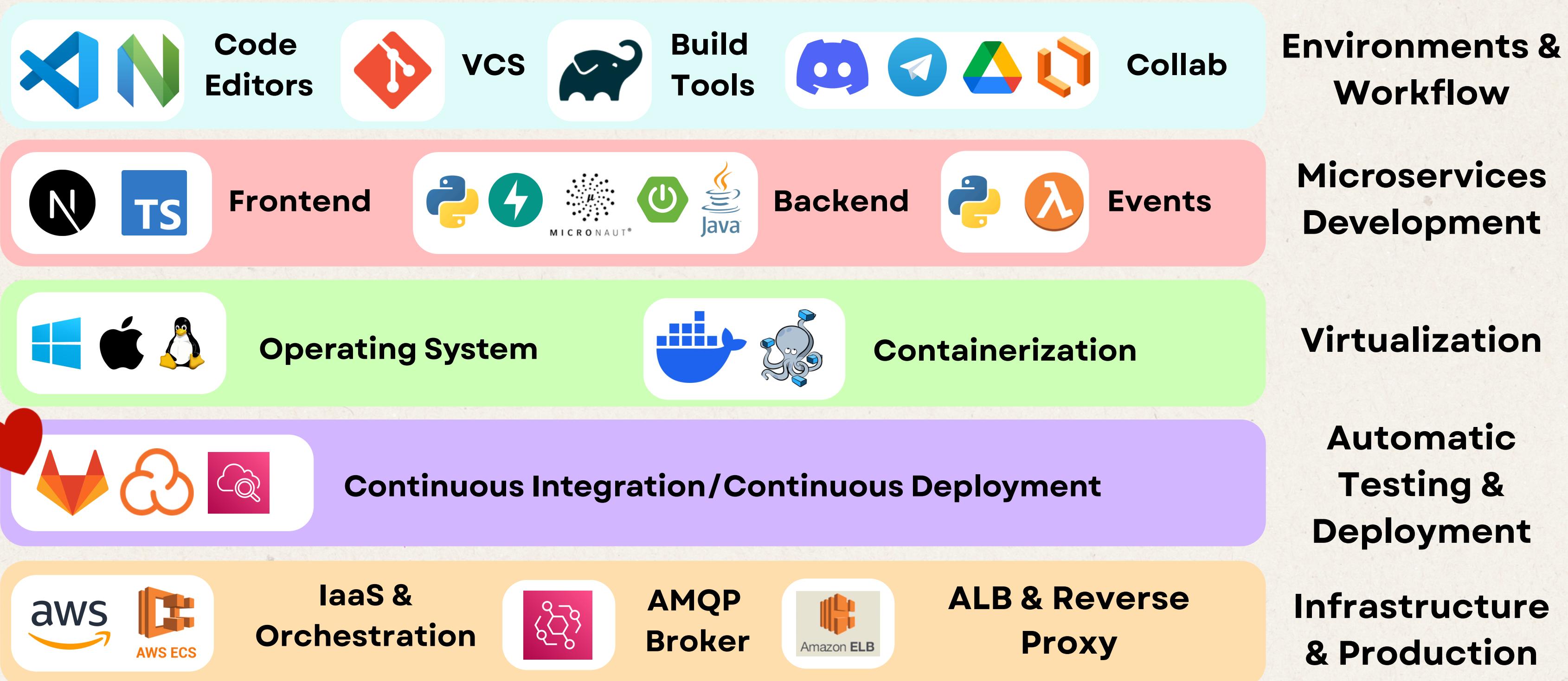
# Technologies (Our DevOps Stack)





07/21

# Technologies (Our DevOps Stack)



# DevOps Practices



## Test-Driven Development

Write Tests First.

Ask Questions Later.

Fail Fast.

We cultivate a strong testing culture.



## Reducing Silos

Encourage collaboration between atomic microservices teams.

Swarm against problems.

Develop in Trunk.



value stream



technology value stream



## Automation-Centricity

Serve automation and ship customer velocity to the value stream as much as possible.

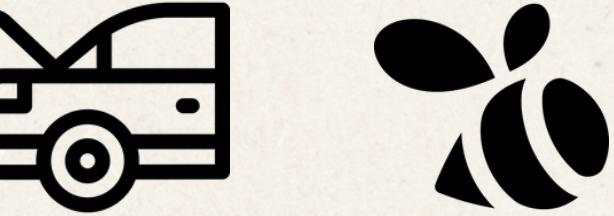
Automation is Flow.

# DevOps Practices



## Test-Driven Development

Fast Feedback



## Reducing Silos

Continuous Learning &  
Experimentation  
(in a Collaborative Setting)



value stream



technology value stream



## Automation-Centricity

Faster Flow

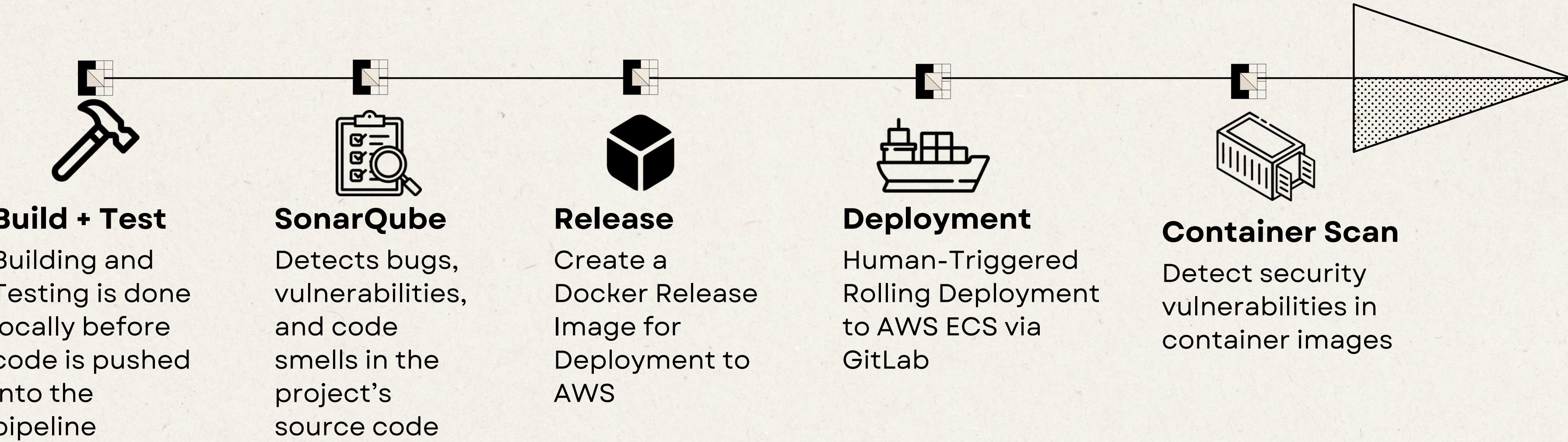


09/21

# Pipeline Design



Continuous Integration / Continuous Deployment (CI/CD)  
via GitLab

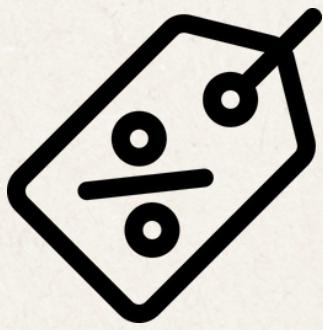


# Key Scenarios



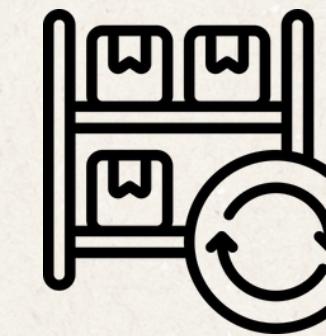
## Cart Checkout

Adding product items into a cart and paying via Stripe API



## Expiry Alert

Product items that are about to expire will have notifications. Owner can decide what to do next.



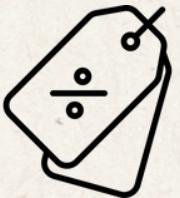
## Stock Alerts

If stock for an item is low, we publish a notification to replenish it, and discounts automatically applied to clear stock

# Cart Checkout



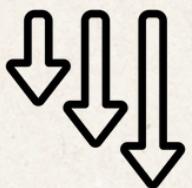
## Specifications



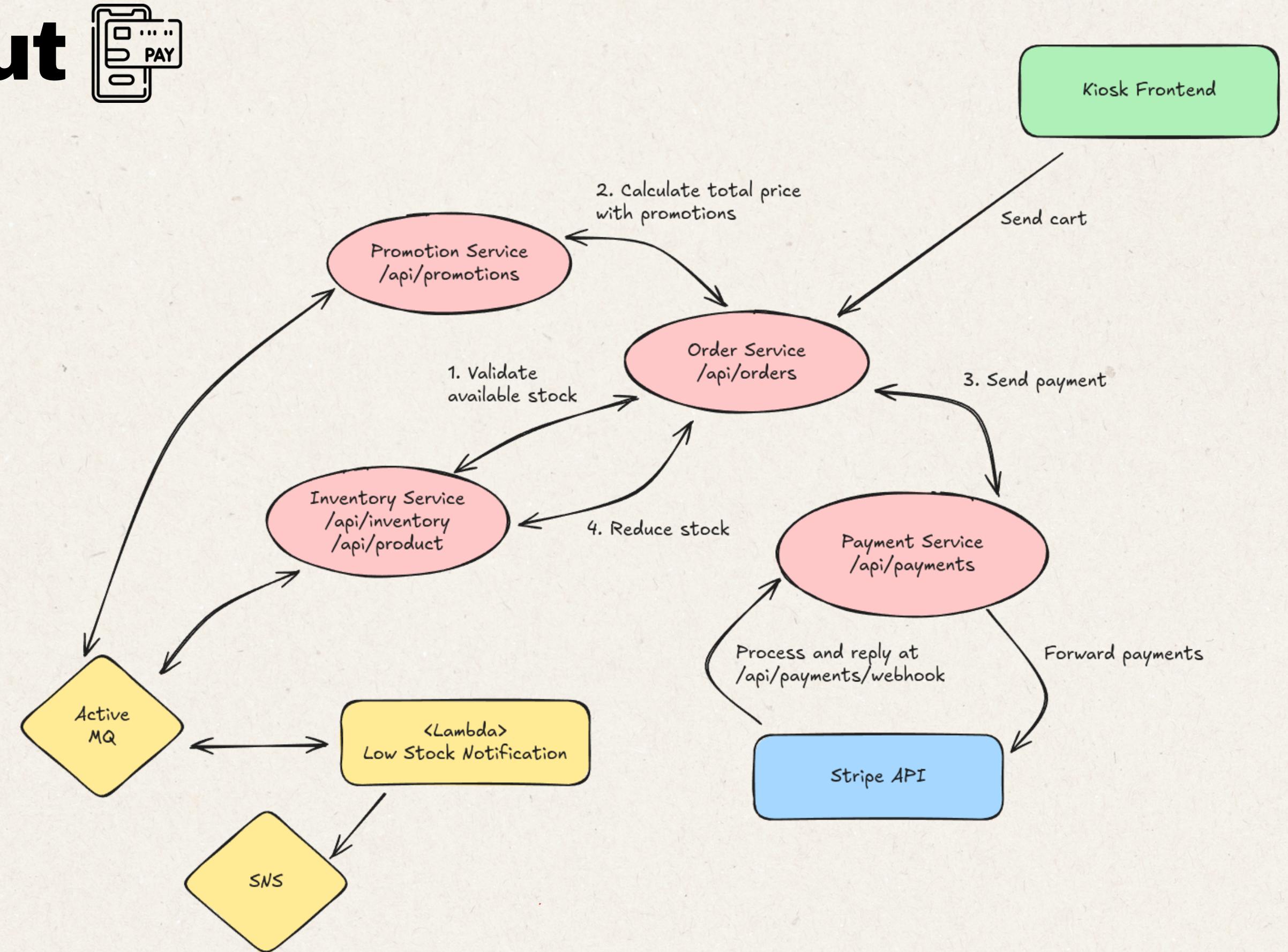
Product promotions automatically applied



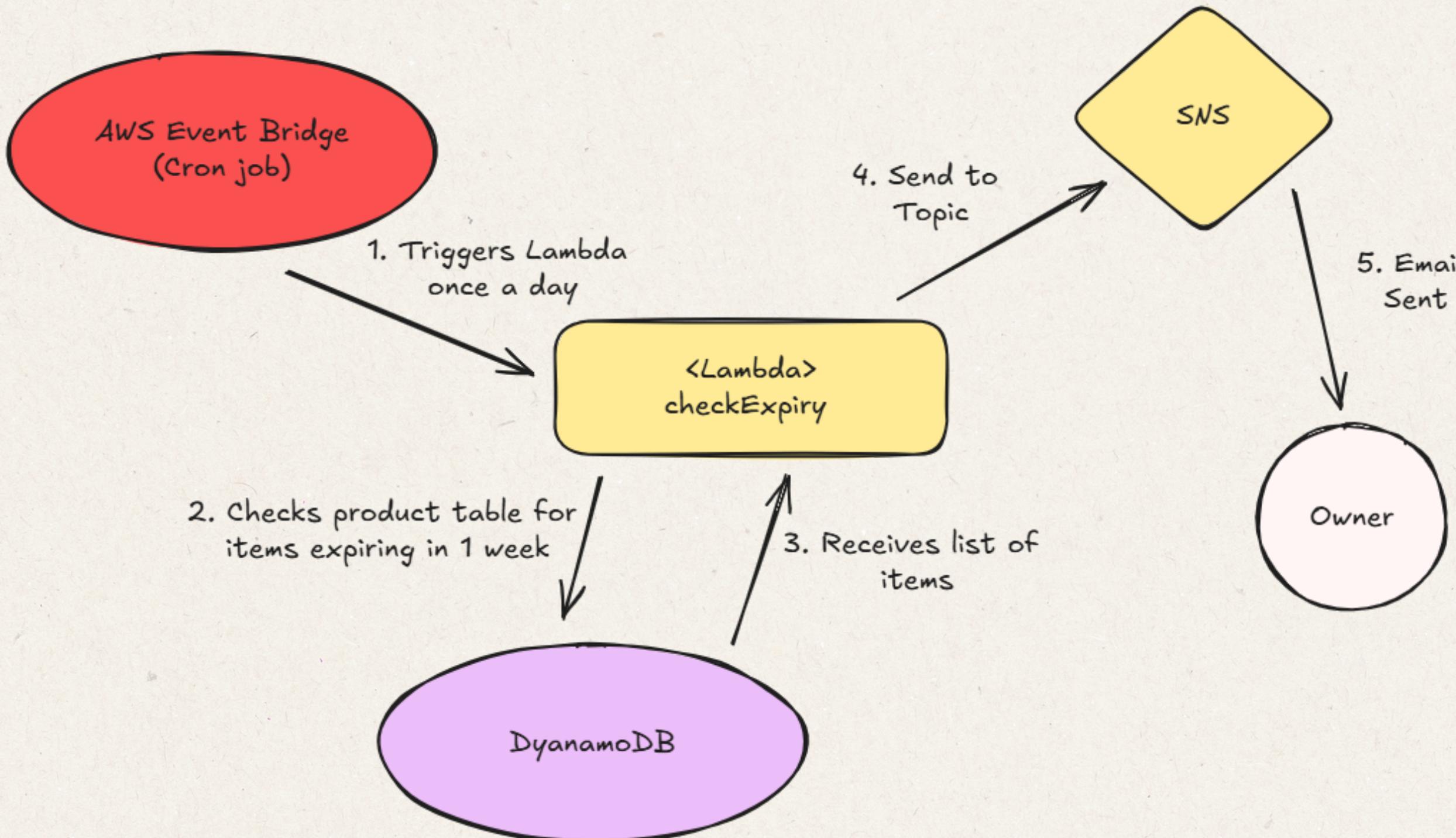
Support payment by credit card



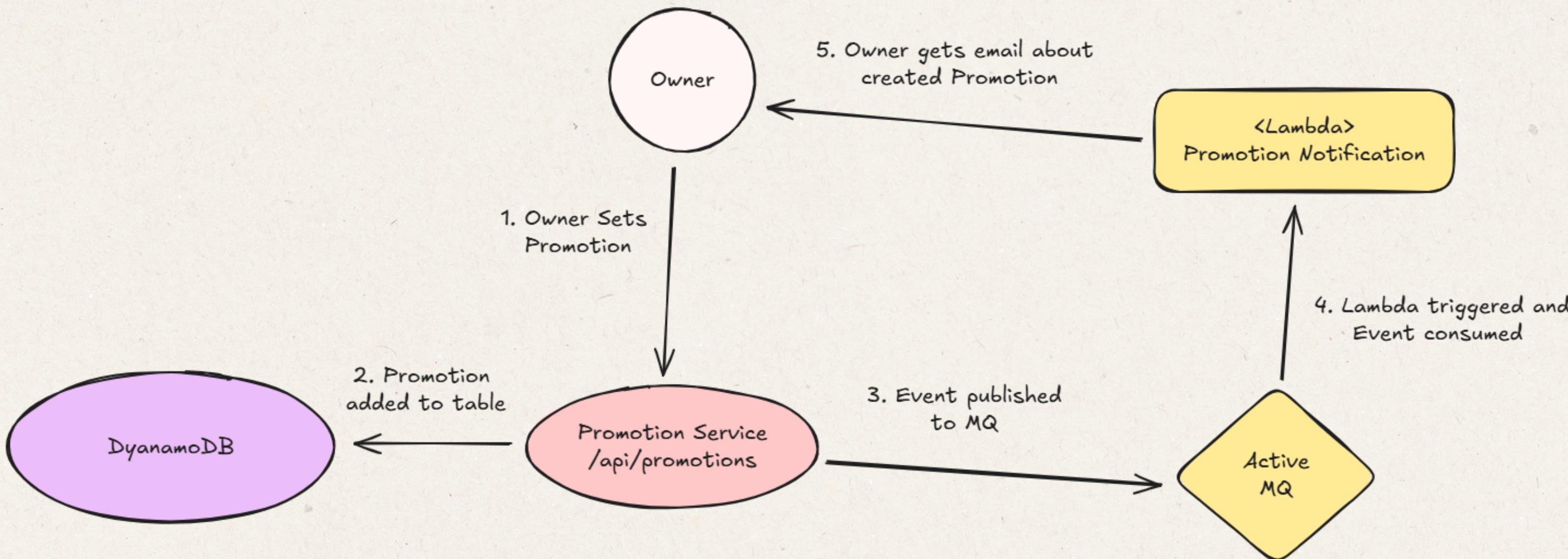
Reduce stock count of the product sold



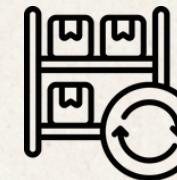
# Expiry Alert



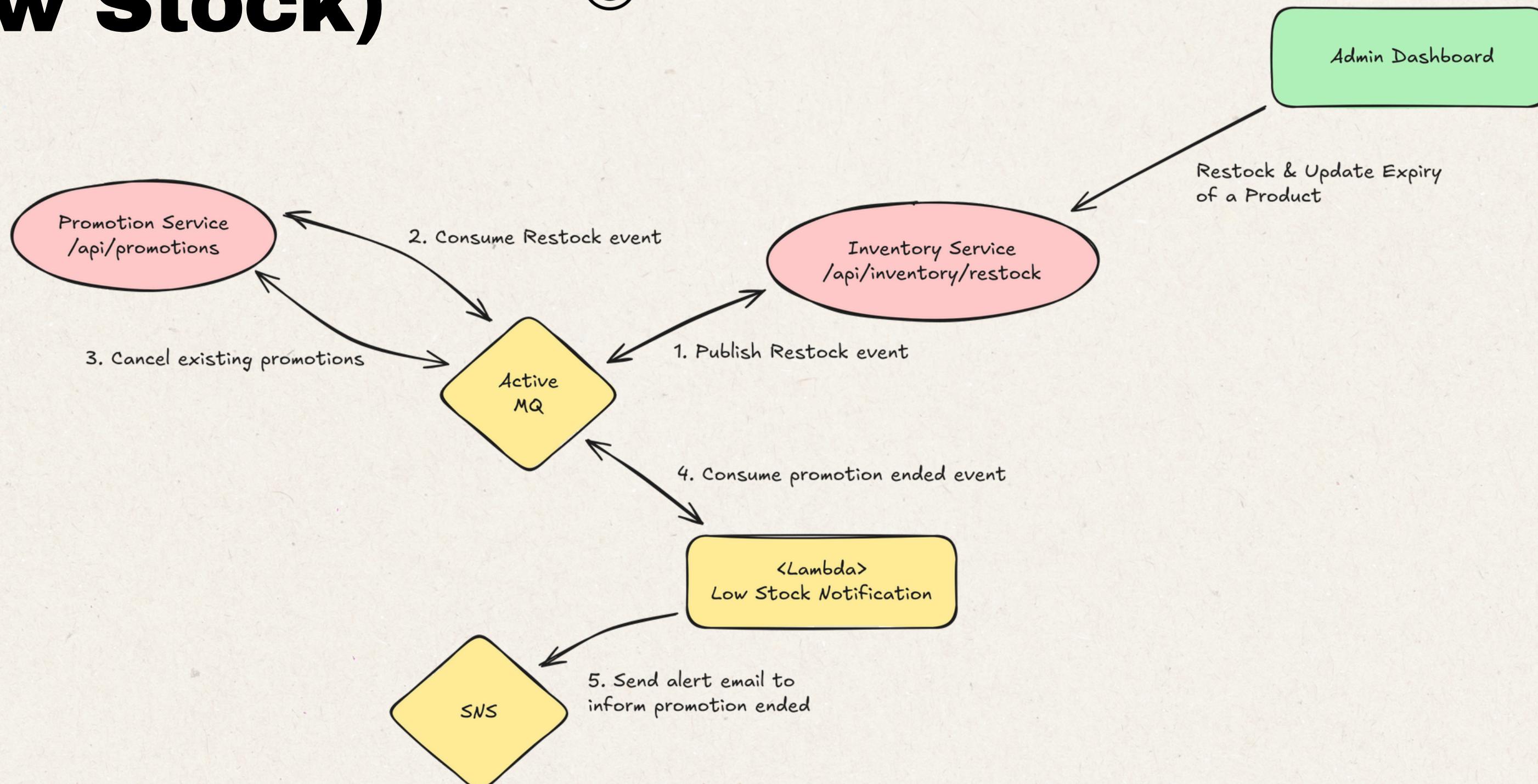
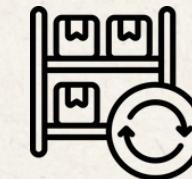
# Expiry Alert



# Low-Stock Alert



# Re-Stock Alert (Low Stock)



# DonDonDemo Time !



## Welcome

Choose an option to continue



### Our Menu

Browse and order your favorites

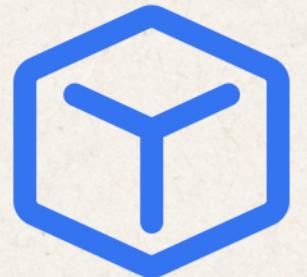


### Admin Portal

Manage inventory and orders

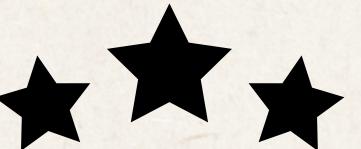


# Self-Directed Research Findings



## Docker Development Containers

Standardized, isolated, and consistent development environments to eliminate “it works on my machines” issues

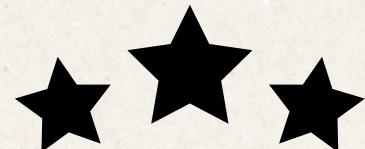


## SonarQube QA Pipeline

Enhances code quality of continuous delivery (CD) and security as a part of a DevSecOps and Continuous Inspection approach



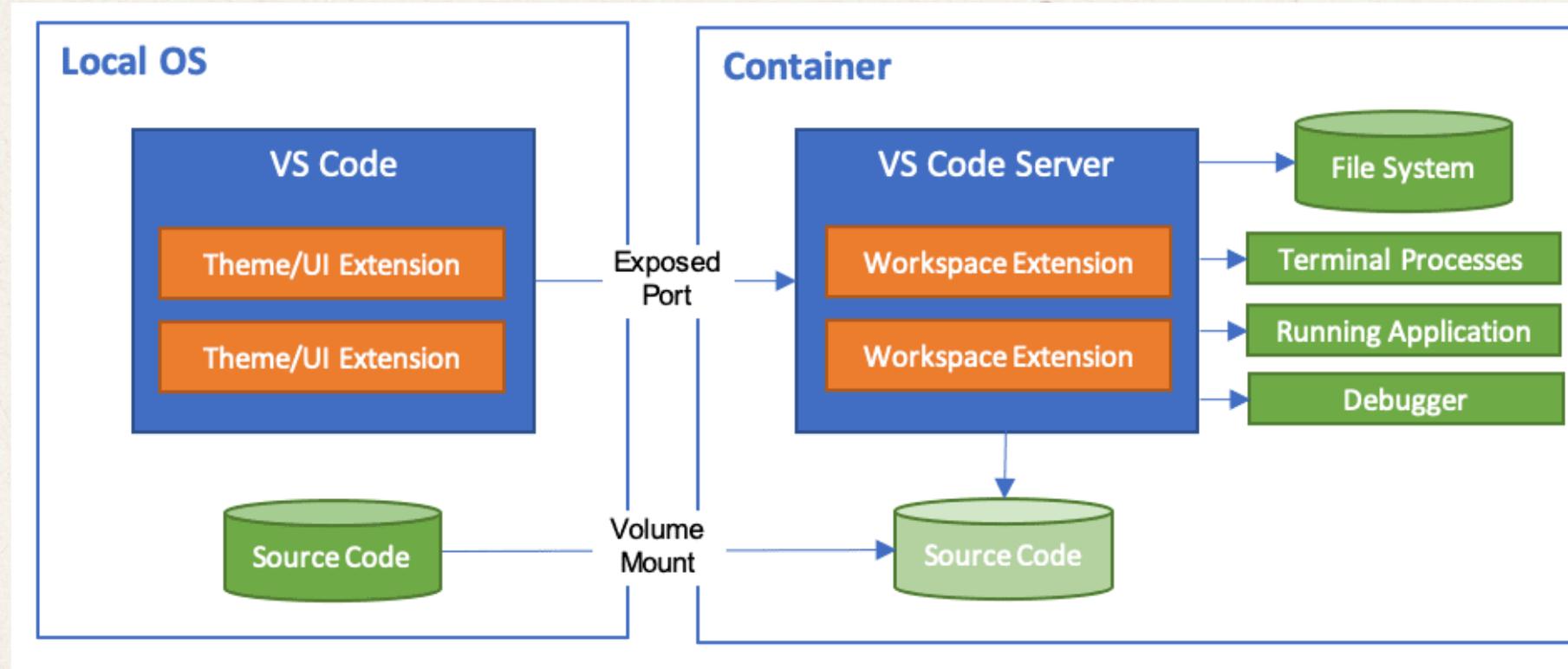
aws Lambda



## CI/CD Pipeline for Serverless Code

Performing CI/CD for a strong testing culture even on production-only components.

# Standardized Development Environment



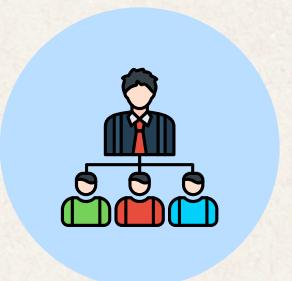
## Benefits

- Consistency and Reproducibility
- Reduced Environment Bugs
- Rapid Onboarding
- Isolation

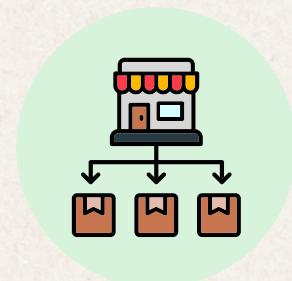
## Demerits

- Higher maintenance compared to local development
- Debugging Challenges
- Longer build times and iteration speed

## Use-cases:



Large Organization



Specialised Dependencies



Remote Development



# Continuous Inspection / Continuous Quality (CI/CQ)

Activity

- Alon Sim assigned to @Xskullibur 2 days ago
- Alon Sim added 1 commit 2 days ago
  - d541aa8e - test: add unit tests for ProductPromotionService methods
- Compare with previous version
- Alon Sim @Xskullibur 2 days ago
- SonarQube Cloud Code Analysis
- Quality Gate passed**
- Issues
  - 11 New issues
  - 0 Accepted issues
- Measures
  - 0 Security Hotspots
  - 95.5% Coverage on New Code
  - 0.0% Duplication on New Code
- See analysis details on SonarQube Cloud
- Alon Sim mentioned in commit 9593a79
- Alon Sim merged 2 days ago

All activity ↑

The screenshot shows the SonarQube Cloud Activity feed. It displays a list of recent events, including a commit from Alon Sim and a mention in a commit. Below the activity feed, there are five project cards. The first project, 'frontend', has a yellow star icon and is labeled 'New' and 'Private'. It shows a 'Quality Gate passed' status with 11 new issues. The second project, 'order-orchestrator', also has a yellow star icon and is labeled 'New' and 'Private'. It shows a 'Not computed' status with 0 security hotspots, 95.5% coverage on new code, and 0.0% duplication. The third project, 'inventory', has a yellow star icon and is labeled 'Private'. It shows a 'Failed' status with 7 reliability issues, 25 maintainability issues, and 16.2% coverage. The fourth project, 'payment', has a yellow star icon and is labeled 'Private'. It shows a 'Failed' status with 2 reliability issues, 19 maintainability issues, and 100% coverage. The fifth project, 'promotions', has a yellow star icon and is labeled 'Private'. It shows a 'Passed' status with 0 security issues, 0 reliability issues, 15 maintainability issues, 100% hotspots reviewed, 81.5% coverage, and 0.0% duplication.

## Value Add

- Consistent Standards Across Services
- Early Detection of Issues
- Quality Gates in Pipelines
- Team Visibility

## Downfalls

- Customizability and features locked behind a paid tier
- Performance impact on CI Pipelines
- False Positives



# CI/CD Pipeline for Serverless Code

The screenshot shows two panels of a CI/CD pipeline editor. The top panel displays the pipeline configuration in a YAML file:

```
stages:
- test
- deploy
variables:
  PIP_CACHE_DIR: "$CI_PROJECT_DIR/.cache/pip"
test_lambda:
  image: python:3.11
  stage: test
  before_script:
    - pip install --upgrade pip
    - pip install -r requirements.txt
  script:
    - pytest -v tests
  only:
    - main
deploy_lambda:
  image: python:3.11
  stage: deploy
  before_script:
    - apt-get update
    - pip install --upgrade pip
    - pwd
    - ls -la
    - zip -r function
  script:
    - aws lambda update-function-code
  only:
    - main
```

The bottom panel shows the pipeline execution details for a commit:

CS302 (2025) / g1-team8 / Lambdas / NotificationLambda / Pipelines / #2152621603

### Update ci.yaml

Passed JEOW TENG XIANG, CODY \_ created pipeline for commit b75ba870 1 hour ago, finished 1 hour ago

For main

latest branch 2 jobs 1.58 1 minute 35 seconds, queued for 1 seconds

Pipeline Jobs 2 Tests 0

A visual representation of the pipeline stages: test (with test\_lambda step) and deploy (with deploy\_lambda step).

## Value Add

- Improved code quality for the lambda, something that is generally hard to test
- Improves code reliability through automated testing and validation before deployment
- Allows rapid iteration, experimentation, & continuous learning without risking production stability

## Downfalls

- Limited local debugging
  - Testing serverless code locally may not reflect production environments exactly
- Vendor lock-in
  - Heavy reliance on AWS-specific tooling can reduce portability
- Testing challenges
  - Integration with managed AWS services (e.g., DynamoDB, S3) often requires mock setups or sandbox environments

# Q&A

Or K&A (Kopi & Answer ?) ☕

