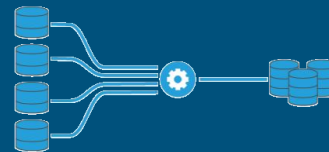


Scalable-ETL-development for The café using AWS cloud



Team 2 Group Project

De-Lon 6

Contents

- Meet the Team
- Problem statement
- Planning
- Technologies used
- Architecture
- Schema
- Extract / Clean
- Transform
- Load
- CI/CD architecture
- CI/CD using codeBuild
- Successful implementation
- CI/CD GitHub Integration - AWS Lambda
- Analysis & Trends (Grafana)
- Monitoring Infrastructure
- Insights / Maximum Viable Product (MVPs)
- What? Where? When?
- Wrap up

Meet The Team

Generation



Marshall Gitanjali Ahmed Bharat Serkan

Problem Statement

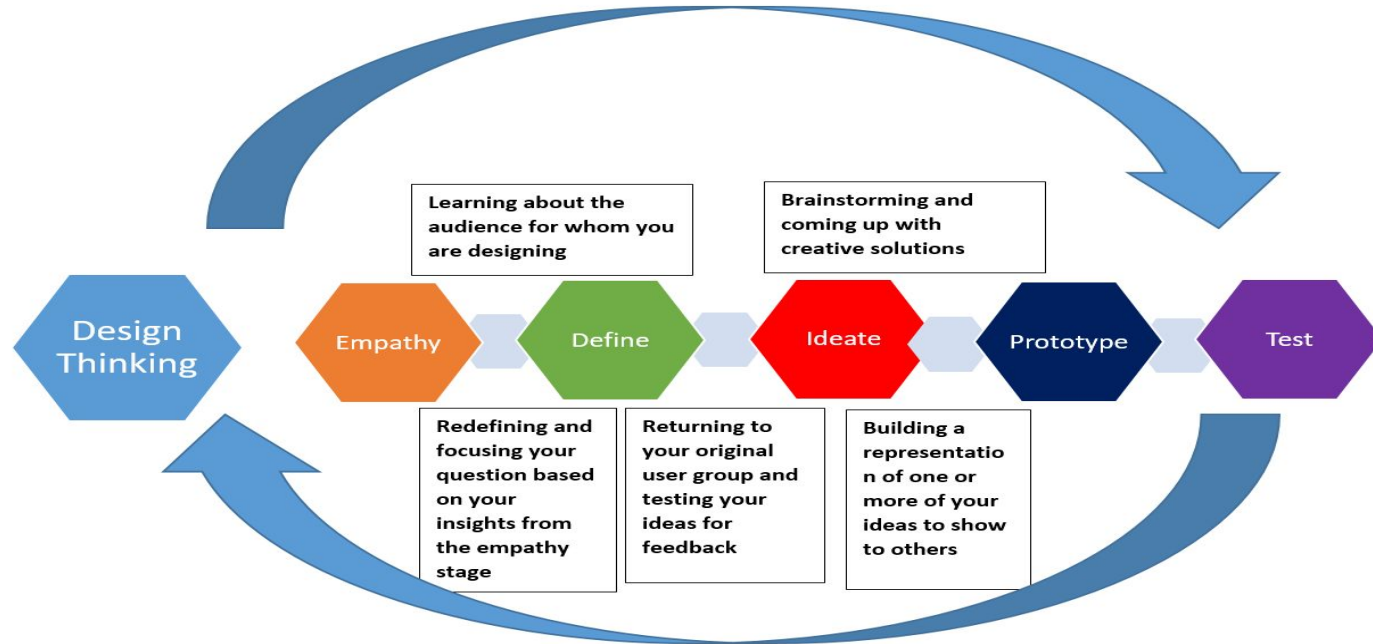
Client: Pop-up café

- Who want us to help them log and track orders
- To track transactions across all outlets for new and returning customers
- Identify the latest trends and make business decisions, to maximise revenue and profits
- Current software has limitations and is time consuming to gather reports from all branches

Our Approach

- Building a fully scalable ETL (Extract, Transform, Load) pipeline to handle large volumes of transaction data from multiple stores
- Use Grafana for data visualization by querying on the transformed data, enabling the client to identify company-wide trends and insights

Planning



Technologies



Python:

Developing our application



AWS Services:

ETL from S3 to Redshift Data warehouse



Grafana:

Application monitoring & Visualisation



GitHub:

Source control

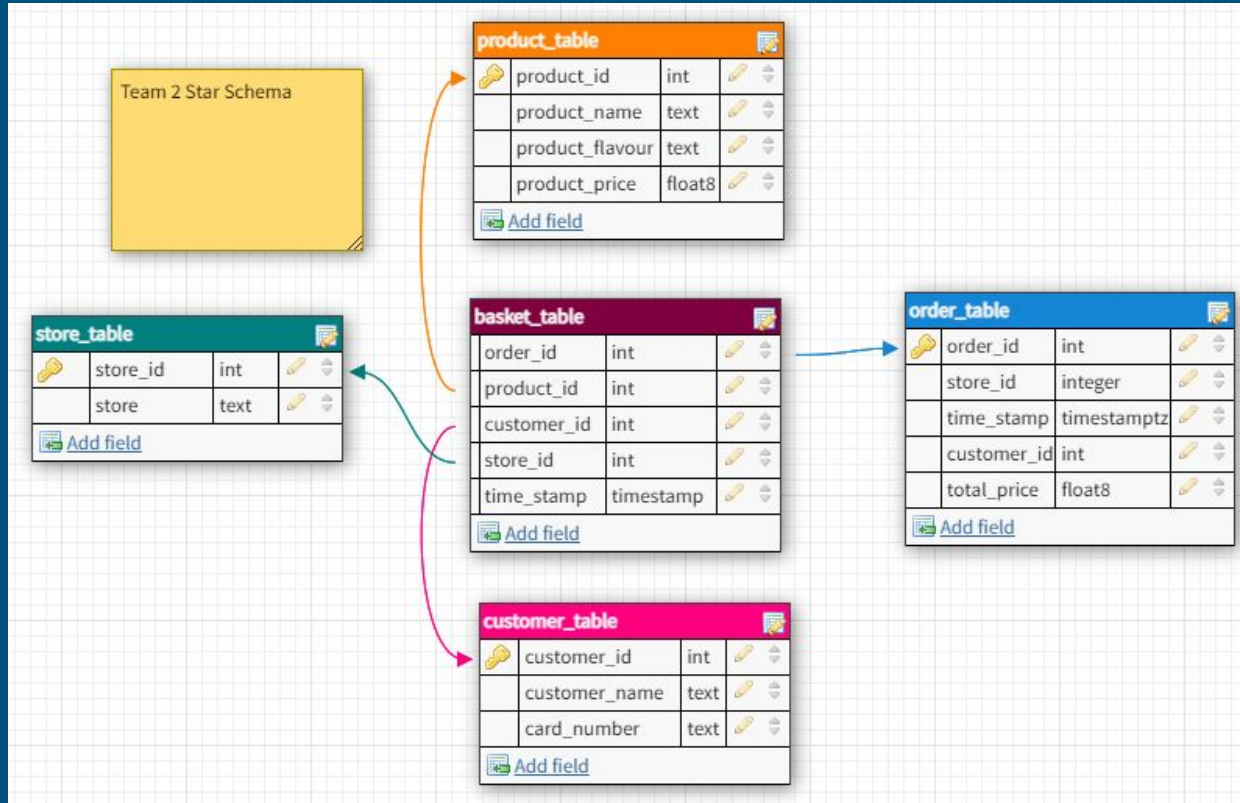


Trello:

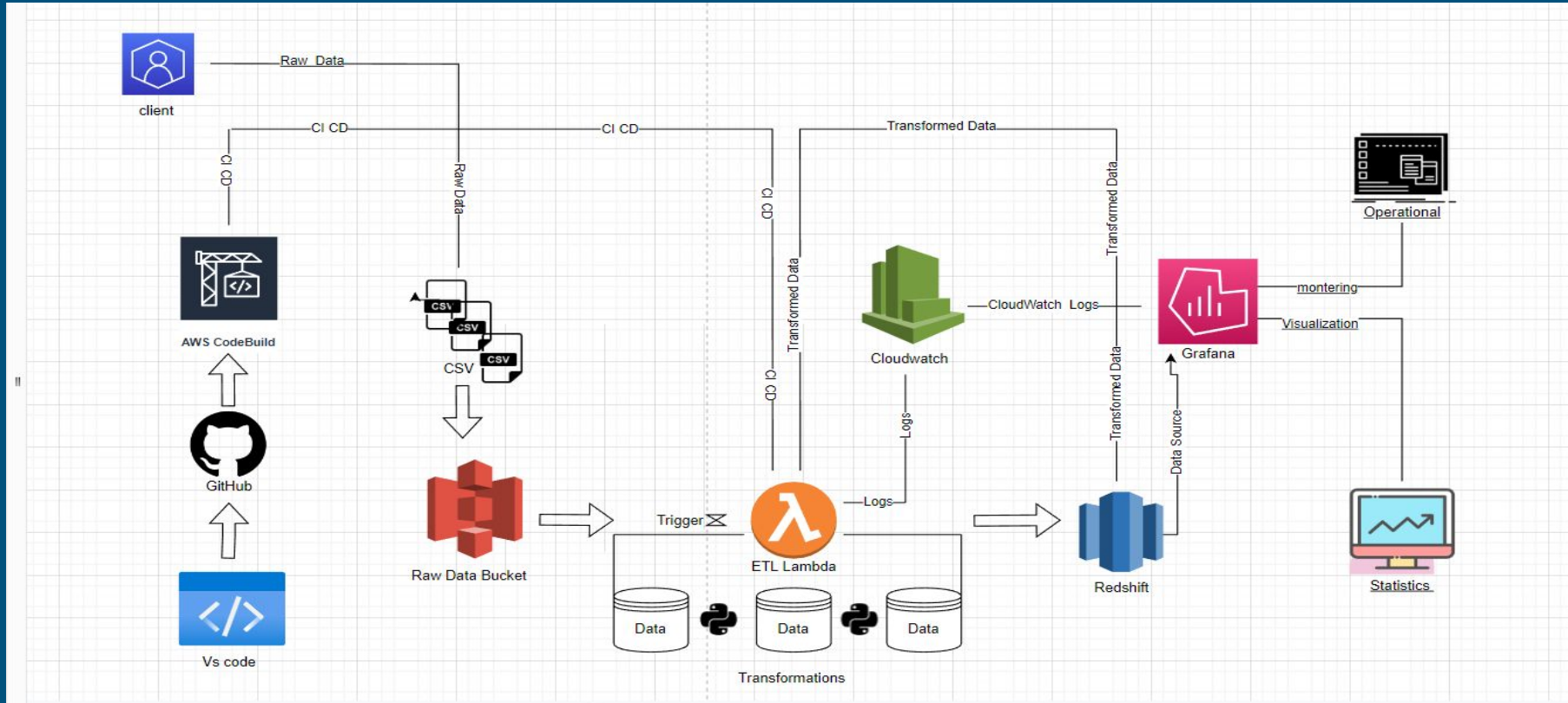
Agile project management principles -
SCRUM sprints

Star schema

Generation



ETL Pipeline Architecture



Extract / Clean

Raw data extracted from CSV file & read as dataframe (Data Ingestion)



EXTRACT



```

2 import pandas as pd
3
4 file = 'chesterfield_10-06-2022_09-00-00.csv'
5 fn = ['timestamp', 'store', 'customer_name',
6       'basket_items', 'total_price', 'cash_or_card', 'card_number']
7 df = pd.read_csv(file, names =fn )
8
9 print(df)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\ahmed\Desktop\testing> & C:\Python310\python.exe c:/Users/ahmed/Desktop/testing/app.py

	timestamp	store	customer_name	basket_items	total_price	cash_or_card	card_number
0	10/06/2022 09:00	Chesterfield	Allen Ford	Regular Flavoured iced latte - Hazelnut - 2.75...	10.20	CASH	NaN
1	10/06/2022 09:02	Chesterfield	Nicole Miles	Regular Flavoured iced latte - Caramel - 2.75...	12.00	CARD	8.435947e+15
2	10/06/2022 09:04	Chesterfield	Arlen Calvert	Regular Latte - 2.15, Regular Flavoured iced l...	4.90	CARD	9.116675e+15
3	10/06/2022 09:06	Chesterfield	Delaine Crosby	Regular Flavoured iced latte - Hazelnut - 2.75	2.75	CARD	6.555526e+15
4	10/06/2022 09:08	Chesterfield	Daniel Pettrey	Large Flavoured iced latte - Vanilla - 3.25	3.25	CASH	NaN
...
263	10/06/2022 16:45	Chesterfield	Traci Abeles	Regular Flavoured iced latte - Hazelnut - 2.75...	10.50	CASH	NaN
264	10/06/2022 16:49	Chesterfield	Dante Jackson	Large Flavoured latte - Hazelnut - 2.85	2.85	CASH	NaN
265	10/06/2022 16:52	Chesterfield	Hector Rosel	Large Flavoured latte - Hazelnut - 2.85, Large...	11.10	CASH	NaN
266	10/06/2022 16:56	Chesterfield	Mary Stanner	Large Flat white - 2.45	2.45	CASH	NaN
267	10/06/2022 16:59	Chesterfield	June Paulson	Large Flavoured latte - Hazelnut - 2.85, Regul...	8.15	CASH	NaN

[268 rows x 7 columns]

PS C:\Users\ahmed\Desktop\testing>

Transform

Dataframe transformed into enriched dataframe with desired schema and respective values



```

1 import pandas as pd
2
3 file = 'chesterfield_10-06-2022_09-00-00.csv'
4 fn = ['timestamp', 'store', 'customer_name',
5       'basket_items', 'total_price', 'cash_or_card', 'card_number']
6 df = pd.read_csv(file, names=fn)
7
8 print(df)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest Powershell for new features and improvements! <https://aka.ms/PowerShell>

```

PS C:\Users\shad\Desktop\testing> & C:\Python38\python.exe c:\Users\shad\Desktop\testing\app.py
timestamp store customer_name basket_items total_price cash_or_card card_number
0 10/06/2022 09:00 Chesterfield Allen Ford Regular Flavoured Iced Latte - Hazelnut - 2.75... 10.50 CASH null
1 10/06/2022 09:00 Chesterfield Nicole Miles Regular Flavoured Iced Latte - Caramel - 2.75... 12.00 CARD 8.455629e+15
2 10/06/2022 09:00 Chesterfield Aylen Chovert Regular Latte - 2.15, Regular Flavoured Iced L... 4.90 CARD 9.138676e+15
3 10/06/2022 09:00 Chesterfield Melissa Crosby Regular Flavoured Iced Latte - Hazelnut - 2.75 2.25 CARD 6.555330e+15
4 10/06/2022 09:00 Chesterfield Daniel Pettray Large Flavoured Iced Latte - Vanilla - 3.25 3.25 CASH null
... ..
203 10/06/2022 10:45 Chesterfield Traci Achilles Regular Flavoured Iced Latte - Hazelnut - 2.75... 10.50 CASH null
204 10/06/2022 10:49 Chesterfield Bartle Jackson Large Flavoured Latte - Hazelnut - 2.85 2.85 CASH null
205 10/06/2022 10:52 Chesterfield Hector Noel Large Flavoured Latte - Hazelnut - 2.85, Large... 11.10 CASH null
206 10/06/2022 10:56 Chesterfield Mary Starner Large Flat white - 2.45 2.45 CASH null
207 10/06/2022 10:59 Chesterfield Jane Jackson Large Flavoured Latte - Hazelnut - 2.85, Regul... 8.15 CASH null

[208 rows x 7 columns]
PS C:\Users\shad\Desktop\testing>

```

Table

Load

Loading transformed data tables into Redshift using database connection credentials

```
2 import pandas as pd
3
4 file = 'chesterfield_10-06-2022_09-00-00.csv'
5 fn = ['timestamp', 'store', 'customer_name',
6       'basket_items', 'total_price', 'cash_or_card', 'card_number']
7 df = pd.read_csv(file, names=fn)
8
9 print(df)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\ahmed\Desktop\testing> & C:\Python10\python.exe c:\Users\ahmed\Desktop\testing\app.py

	timestamp	store	customer_name	basket_items	total_price	cash_or_card	card_number
0	10/06/2022 09:00	Chesterfield	Allen Ford	Regular Flavoured iced latte - Hazelnut - 2.75...	10.20	CASH	NaN
1	10/06/2022 09:00	Chesterfield	Nicole Riley	Regular Flavoured iced latte - Caramel - 2.75...	12.00	CARD	8-4355479415
2	10/06/2022 09:04	Chesterfield	Arlon Calvert	Regular latte - 2.15, Regular Flavoured iced l...	4.90	CARD	9-1166759415
3	10/06/2022 09:06	Chesterfield	Delaine Crosby	Regular Flavoured iced latte - Hazelnut - 2.75	2.75	CARD	6-555526415
4	10/06/2022 09:08	Chesterfield	Daniel Pettrey	Large Flavoured iced latte - Vanilla - 3.25	3.25	CASH	NaN
...
263	10/06/2022 16:45	Chesterfield	Traci Ables	Regular Flavoured iced latte - Hazelnut - 2.75...	10.50	CASH	NaN
264	10/06/2022 16:49	Chesterfield	Dante Jackson	Large Flavoured latte - Hazelnut - 2.85	2.45	CASH	NaN
265	10/06/2022 16:52	Chesterfield	Hector Russell	Large Flavoured latte - Hazelnut - 2.85, Large...	11.10	CASH	NaN
266	10/06/2022 16:56	Chesterfield	Mary Starnes	Large Flat white - 2.45	2.45	CASH	NaN
267	10/06/2022 16:59	Chesterfield	June Paulson	Large Flavoured latte - Hazelnut - 2.85, Regul...	8.15	CASH	NaN

[268 rows x 7 columns]

PS C:\Users\ahmed\Desktop\testing>



LOAD



CI/CD Using CodeBuild

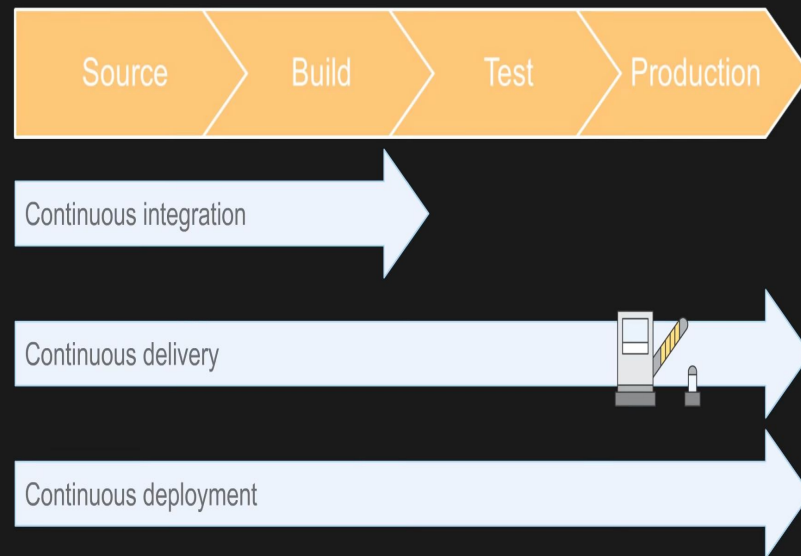
Why CI/CD?

- To deliver a new version of software - series of steps required
- CI/CD automate these steps to improve software delivery throughout the software development life cycle

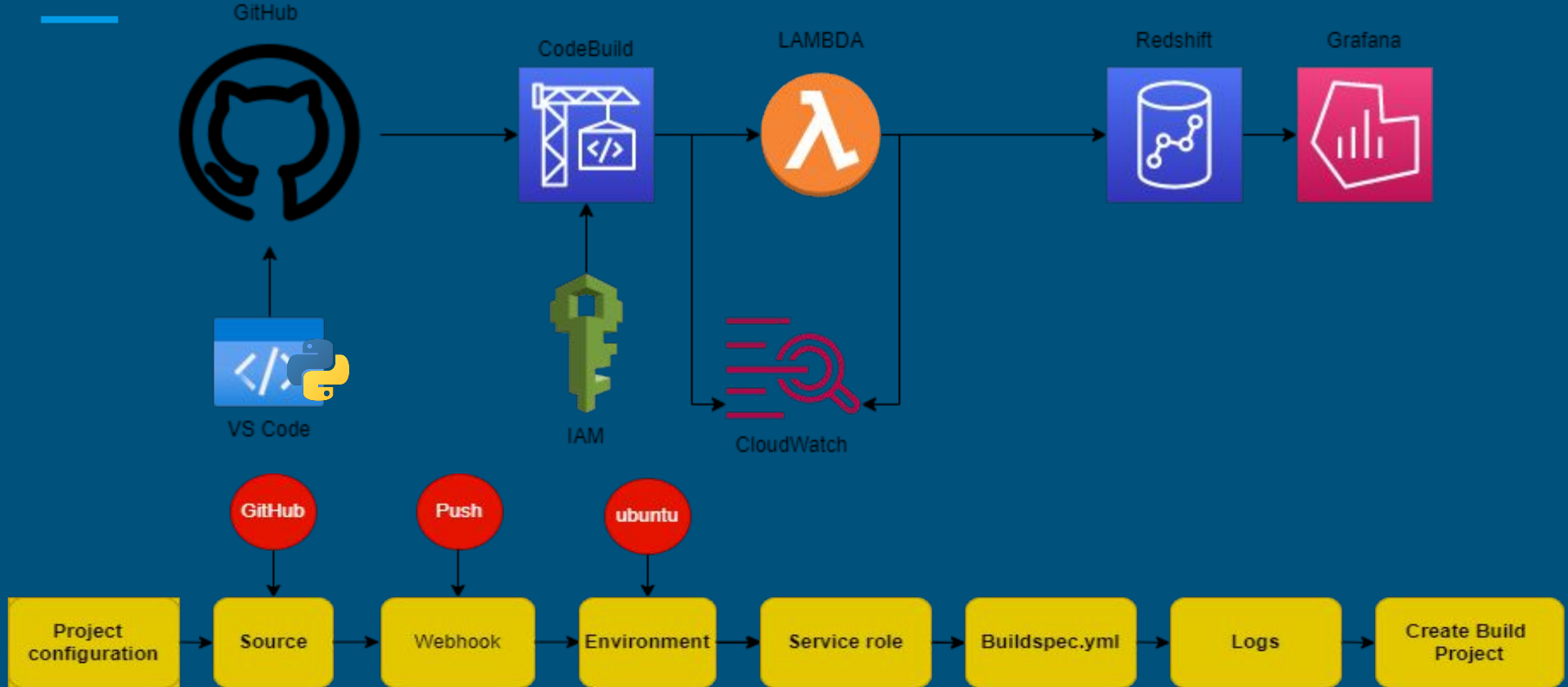
Why CodeBUILD?

- Fully managed continuous integration service
- Readily deployable software packages
- Continuous scaling and concurrent processing of multiple builds
- Quick start by using prepackaged build environments
- Charged by the minute for the compute

Release processes levels



CI/CD Architecture



CI/CD GitHub Integration - AWS Lambda

WEBHOOKS GitHub to CodeBuild

Webhooks / Manage webhook

Settings Recent Deliveries

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

`https://codebuild.eu-west-1.amazonaws.com/webhooks?t=eyJlbmI`

Content type

`application/json`

☒ Pull requests

Pull request opened, closed, reopened, edited, assigned, unassigned, review requested, review request removed, labeled, unlabeled, synchronized, ready for review, converted to draft, locked, unlocked, auto merge enabled, auto merge disabled, milestone, or demilestoned.

☒ Pushes

Git push to a repository.

☒ Active

We will deliver event details when this hook is triggered.

Update webhook

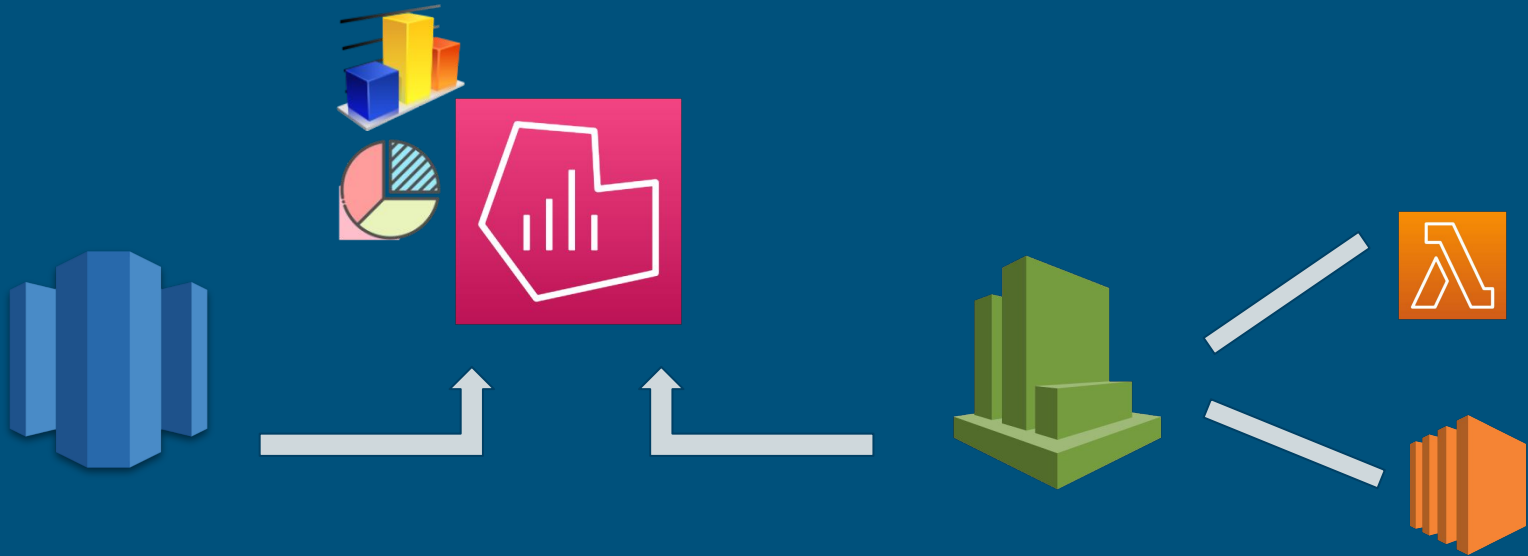
Delete webhook

Successful Implementation

Build status					
Status	Initiator	Build ARN	Resolved source version		
🟢 Succeeded	GitHub-Hookshot/1b8175f	am:aws:codebuild:eu-west-1:506555054152:build/tff-clc:ea782e39-f84e-4752-a53b-3ab26300947b	214e384ee3d08a100a7a595447c28dcdad6f9efa		
Start time	End time	Build number			
Jul 13, 2022 9:39 AM (UTC+1:00)	Jul 13, 2022 9:40 AM (UTC+1:00)	7			
Build logs	Phase details	Reports	Environment variables	Build details	Resource utilization
Name	Status	Context	Duration	Start time	End time
SUBMITTED	🟢 Succeeded	-	<1 sec	Jul 13, 2022 9:39 AM (UTC+1:00)	Jul 13, 2022 9:39 AM (UTC+1:00)
QUEUED	🟢 Succeeded	-	2 secs	Jul 13, 2022 9:39 AM (UTC+1:00)	Jul 13, 2022 9:39 AM (UTC+1:00)
PROVISIONING	🟢 Succeeded	-	27 secs	Jul 13, 2022 9:39 AM (UTC+1:00)	Jul 13, 2022 9:40 AM (UTC+1:00)
DOWNLOAD_SOURCE	🟢 Succeeded	-	5 secs	Jul 13, 2022 9:40 AM (UTC+1:00)	Jul 13, 2022 9:40 AM (UTC+1:00)
INSTALL	🟢 Succeeded	-	<1 sec	Jul 13, 2022 9:40 AM (UTC+1:00)	Jul 13, 2022 9:40 AM (UTC+1:00)
PRE_BUILD	🟢 Succeeded	-	<1 sec	Jul 13, 2022 9:40 AM (UTC+1:00)	Jul 13, 2022 9:40 AM (UTC+1:00)
BUILD	🟢 Succeeded	-	<1 sec	Jul 13, 2022 9:40 AM (UTC+1:00)	Jul 13, 2022 9:40 AM (UTC+1:00)
POST_BUILD	🟢 Succeeded	-	15 secs	Jul 13, 2022 9:40 AM (UTC+1:00)	Jul 13, 2022 9:40 AM (UTC+1:00)
UPLOAD_ARTIFACTS	🟢 Succeeded	-	<1 sec	Jul 13, 2022 9:40 AM (UTC+1:00)	Jul 13, 2022 9:40 AM (UTC+1:00)
FINALIZING	🟢 Succeeded	-	2 secs	Jul 13, 2022 9:40 AM (UTC+1:00)	Jul 13, 2022 9:40 AM (UTC+1:00)
COMPLETED	🟢 Succeeded	-	-	Jul 13, 2022 9:40 AM (UTC+1:00)	-

Analysis & Trends

Grafana configured with Redshift for database access & Cloudwatch for Lambda, EC2 monitoring

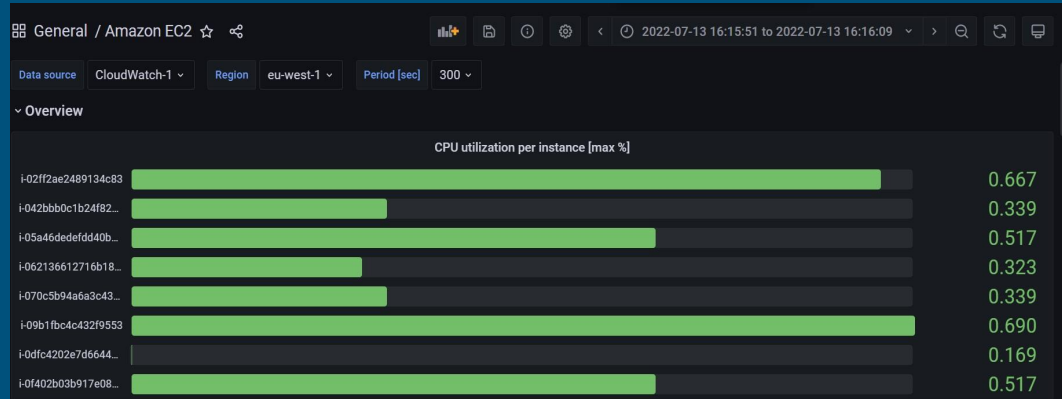


Monitoring Infrastructure



**Lambda
Durations**

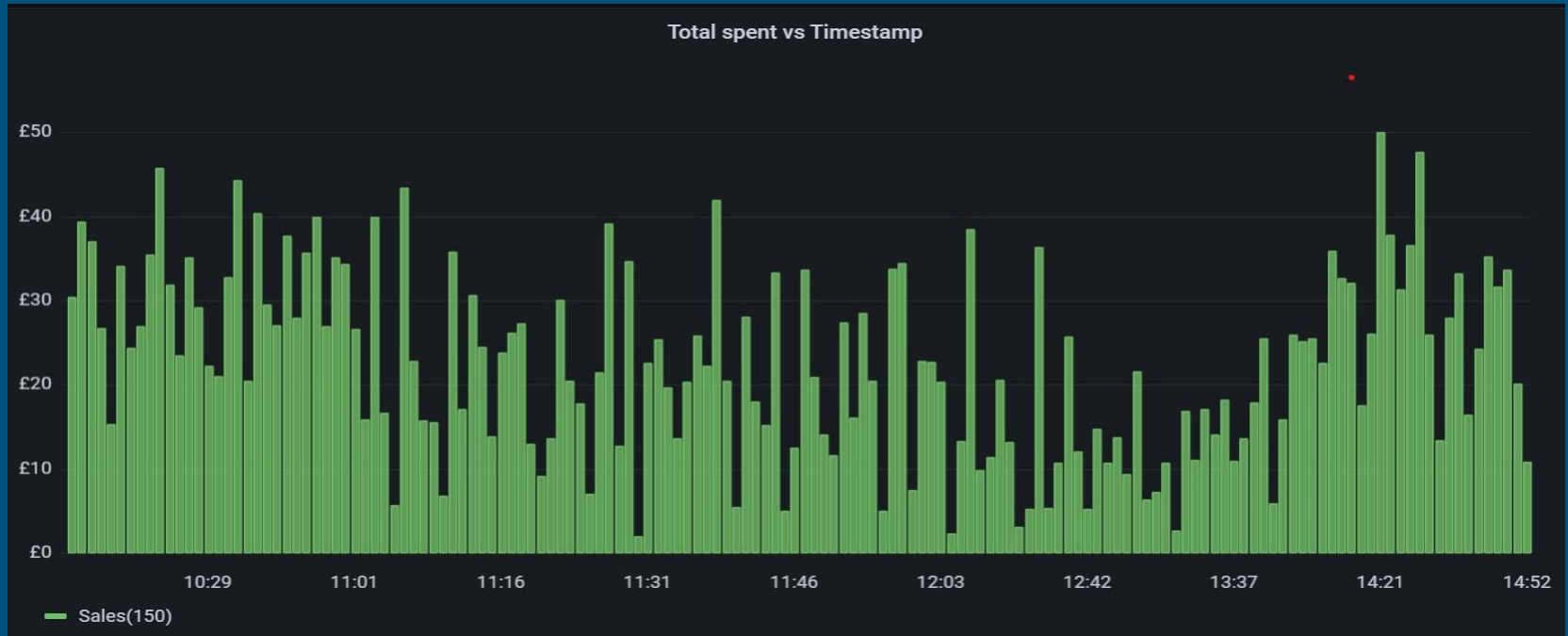
**EC2
Metrics**



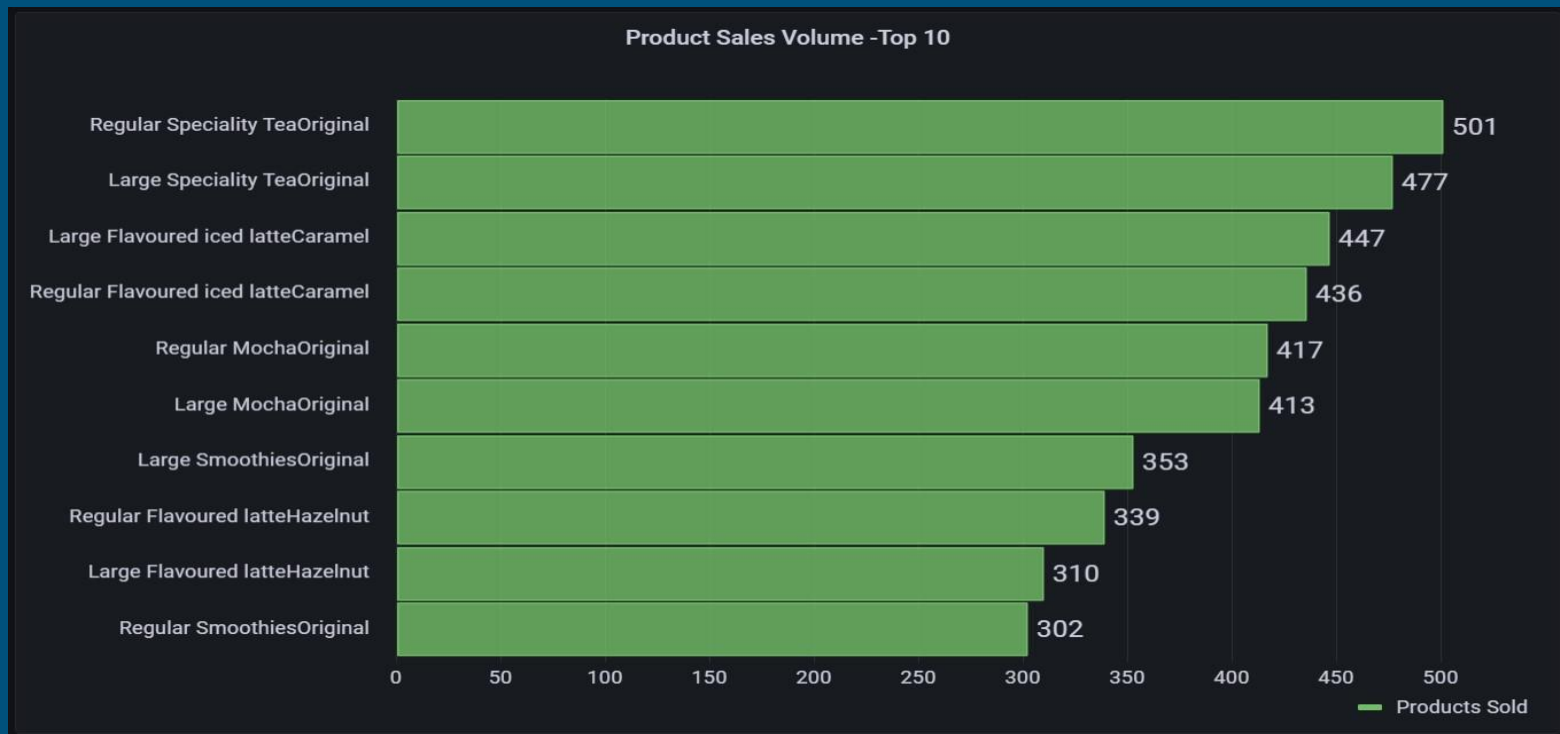
Insights from multiple variables



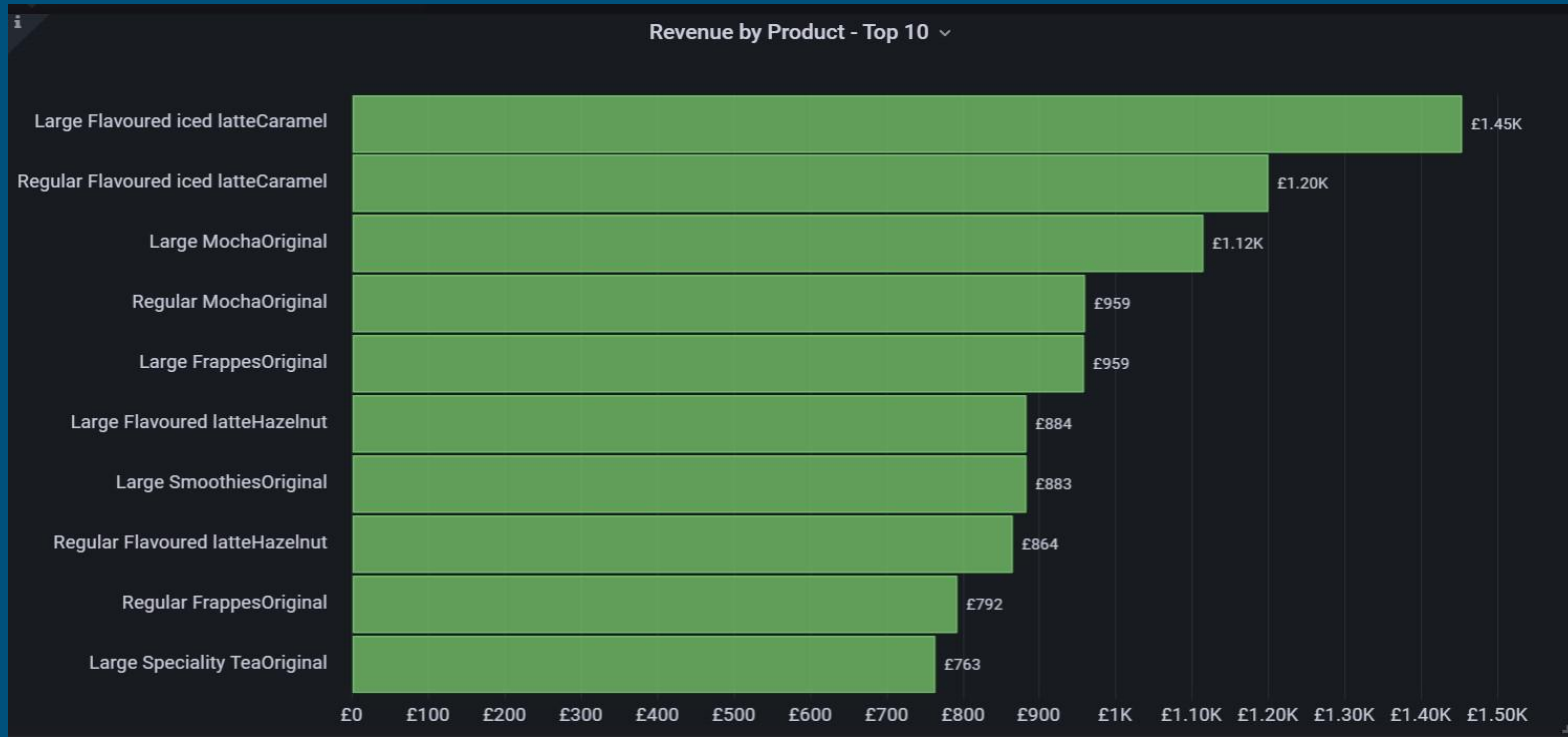
Sales trend over a period of 5 hrs



Highest Selling Product by Volume



Maximum Viable Product (MVP)



What? Where? When?

What - what items were selling well



Where - best performing store



When - peak business hours



Thanks...for listening

Applause to Generation team and Infinity Works from Team 2



Thanks to

- Jakub and Rachel

For all your support and motivation

Special thanks to

- Bala and Darren

For being there for us & helping us through out

- Entire Generation team

Working at backend to make this programme a smooth experience