Meal Order

team members

洪世彬、林俊佑、吳庭維、張家誠、戴靖婷

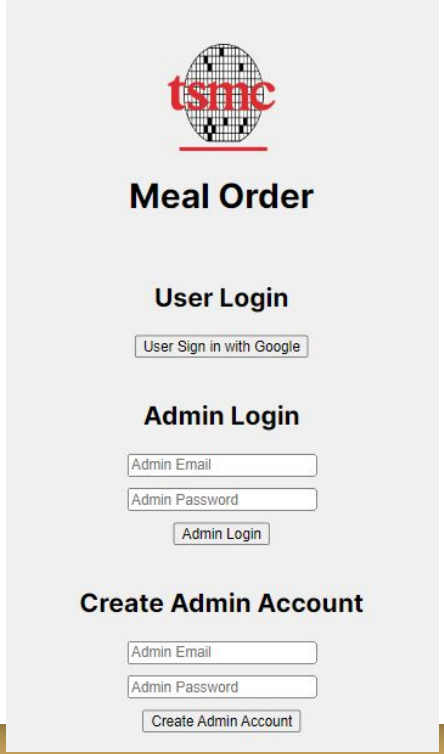
User Requirements

User Requirements

1. 消費者：**點餐與取消**，可以成功點餐/刪除訂單
2. 商家：**更新菜單**，可以上架新菜、下架餐點、管控菜品數量。
3. All：**Mail 通知**，訂餐成功通知/當日通知商家與消費者
4. All：**統計月結餐費**，提供商家與消費者月結餐費

User Requirements - Setup

- All:
  - **登入及用戶設定**



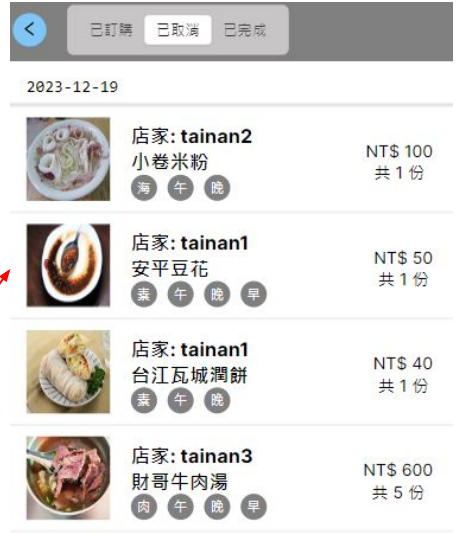
消費者:

User Requirements - 1

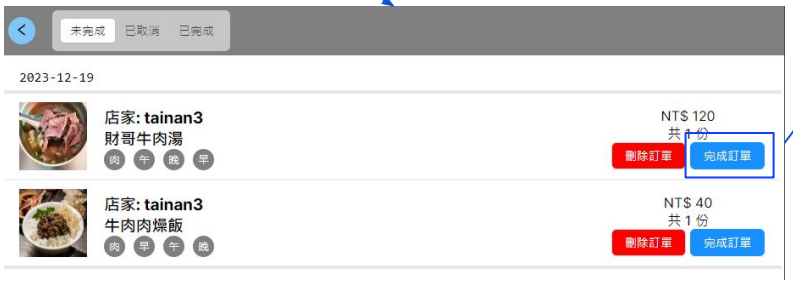
- 消費者:
  - **點餐與取消**，可以成功點餐/刪除訂單

消費者:

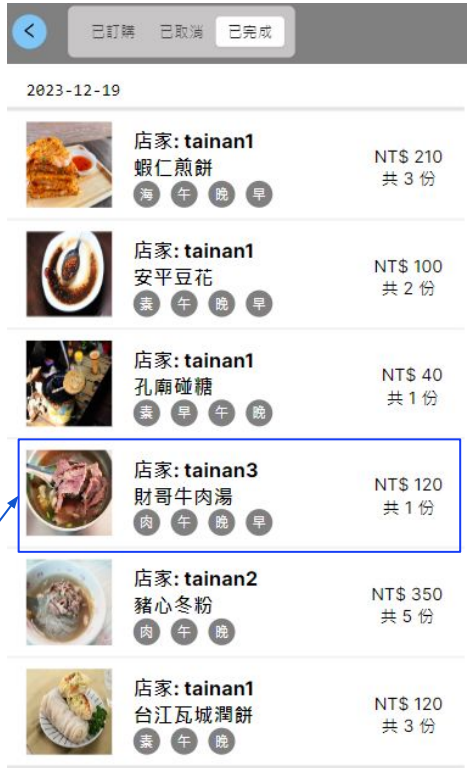
消費者:



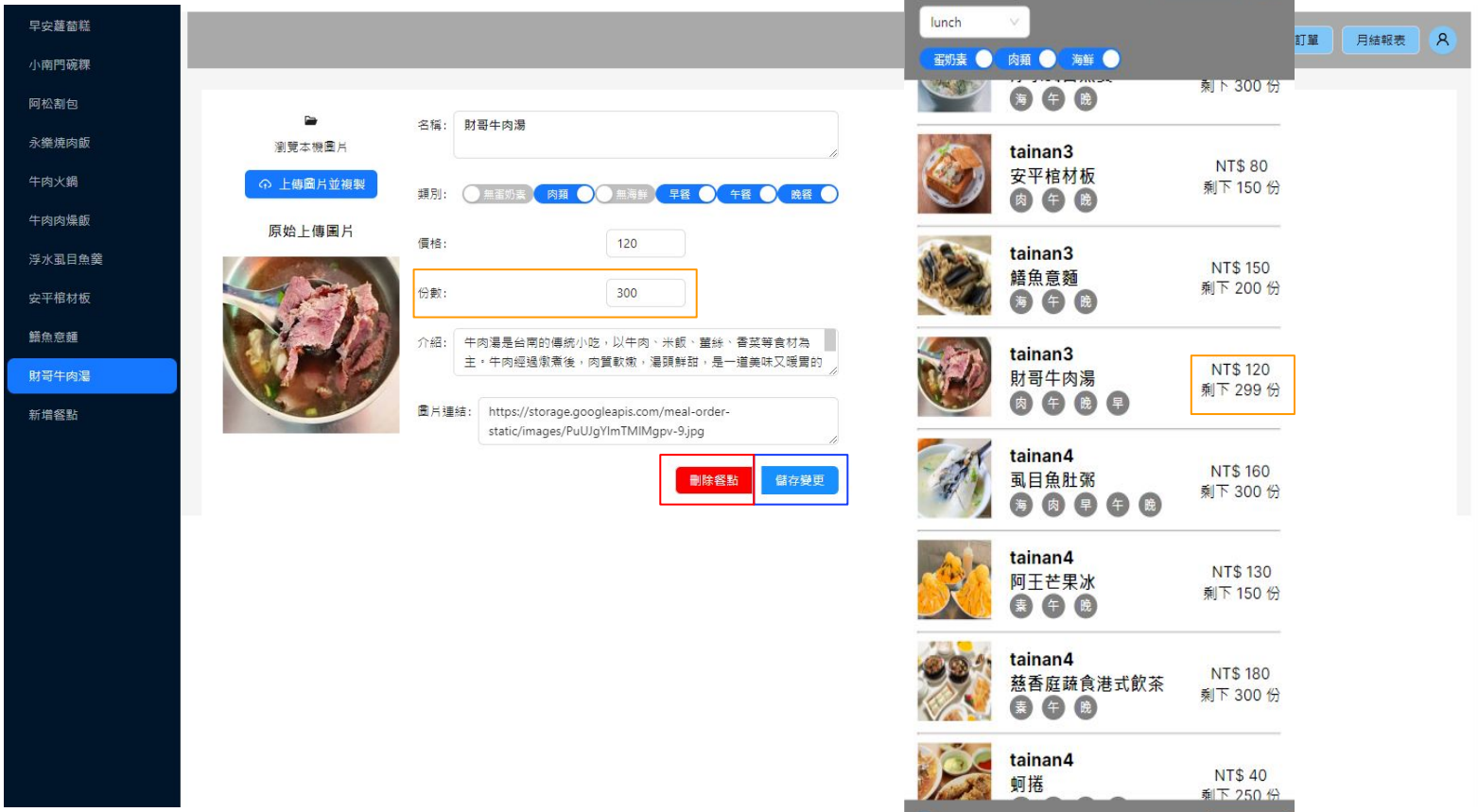
商家:



消費者:



商家:



早安蘿蔔糕
小南門碗粿
阿松割包
永樂燒肉飯
牛肉火鍋
牛肉肉燥飯
浮水虱目魚羹
安平棺材板
鱔魚意麵
財哥牛肉湯
新增餐點
瀏覽本機圖片
上傳圖片並複製
原始上傳圖片
名稱: 財哥牛肉湯
類別: 無蛋奶素 肉類 無海鮮 早餐 午餐 晚餐
價格: 120
份數: 300
介紹: 牛肉湯是台南的傳統小吃，以牛肉、米飯、薑絲、香菜等食材為主。牛肉經過燉煮後，肉質軟嫩，湯頭鮮甜，是一道美味又暖胃的
圖片連結: https://storage.googleapis.com/meal-order-static/images/PuUJgYlmTMIMgpv-9.jpg
刪除餐點
儲存變更
2023/12/19
月結報表
lunch
蛋奶素 肉類 海鮮
訂單
月結報表
剩下 300 份
tainan3
安平棺材板
NT$ 80
剩下 150 份
tainan3
鱔魚意麵
NT$ 150
剩下 200 份
tainan3
財哥牛肉湯
NT$ 120
剩下 299 份
tainan4
虱目魚肚粥
NT$ 160
剩下 300 份
tainan4
阿王芒果冰
NT$ 130
剩下 150 份
tainan4
慈香庭蔬食港式飲茶
NT$ 180
剩下 300 份
tainan4
蚵捲
NT$ 40
AI
檢視訂單

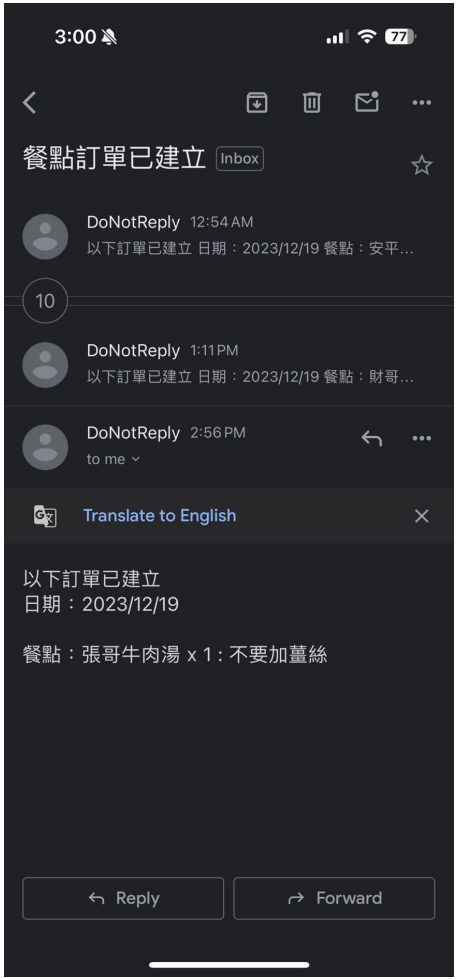
消費者:

User Requirements - 3

- All:
  - **Mail 通知**，訂餐成功通知/當日通知商家與消費者

商家:

消費者:

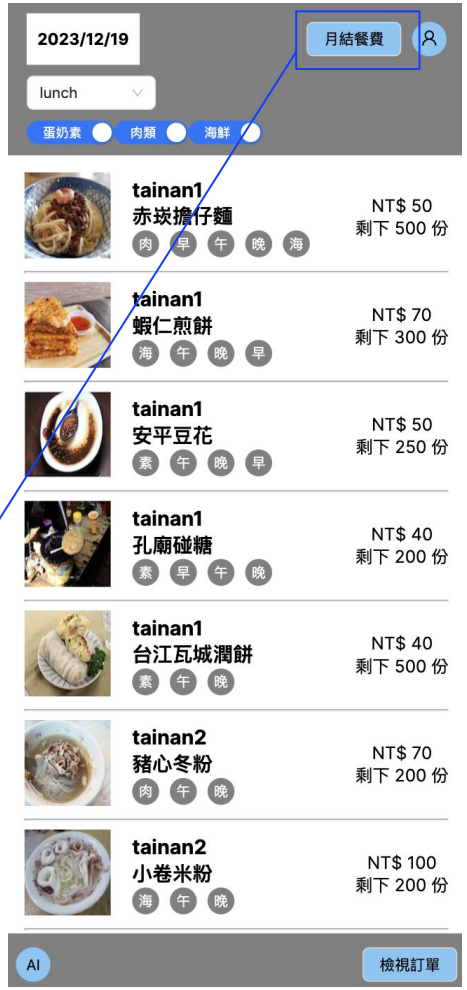


商家: User Requirements - 4

User Requirements - 4

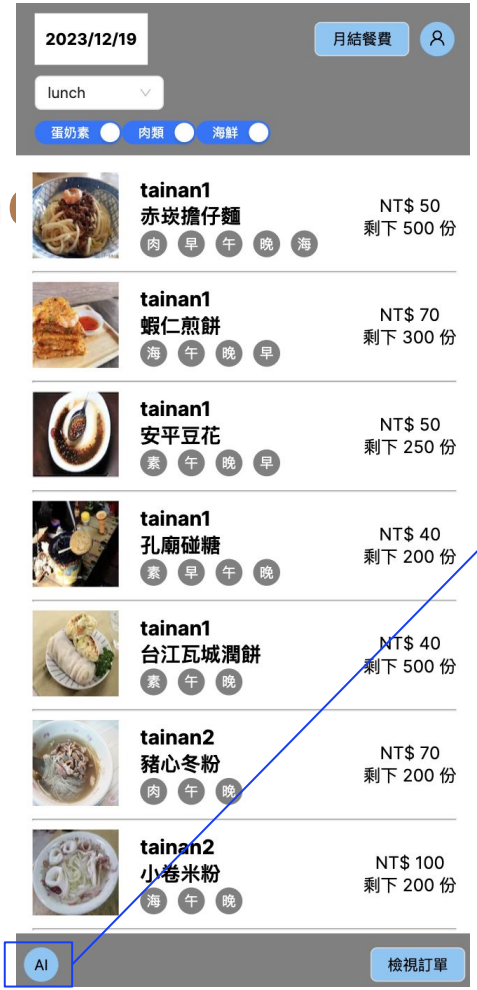
- All:
  - **統計月結餐費**，提供商家與消費者月結餐費

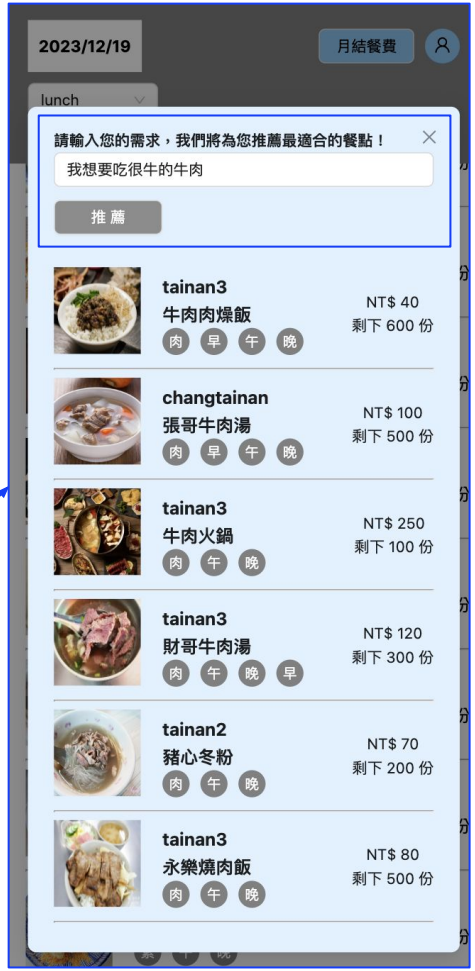
消費者:



User Requirem

- 消費者:
  - **AI 智能推薦餐點**





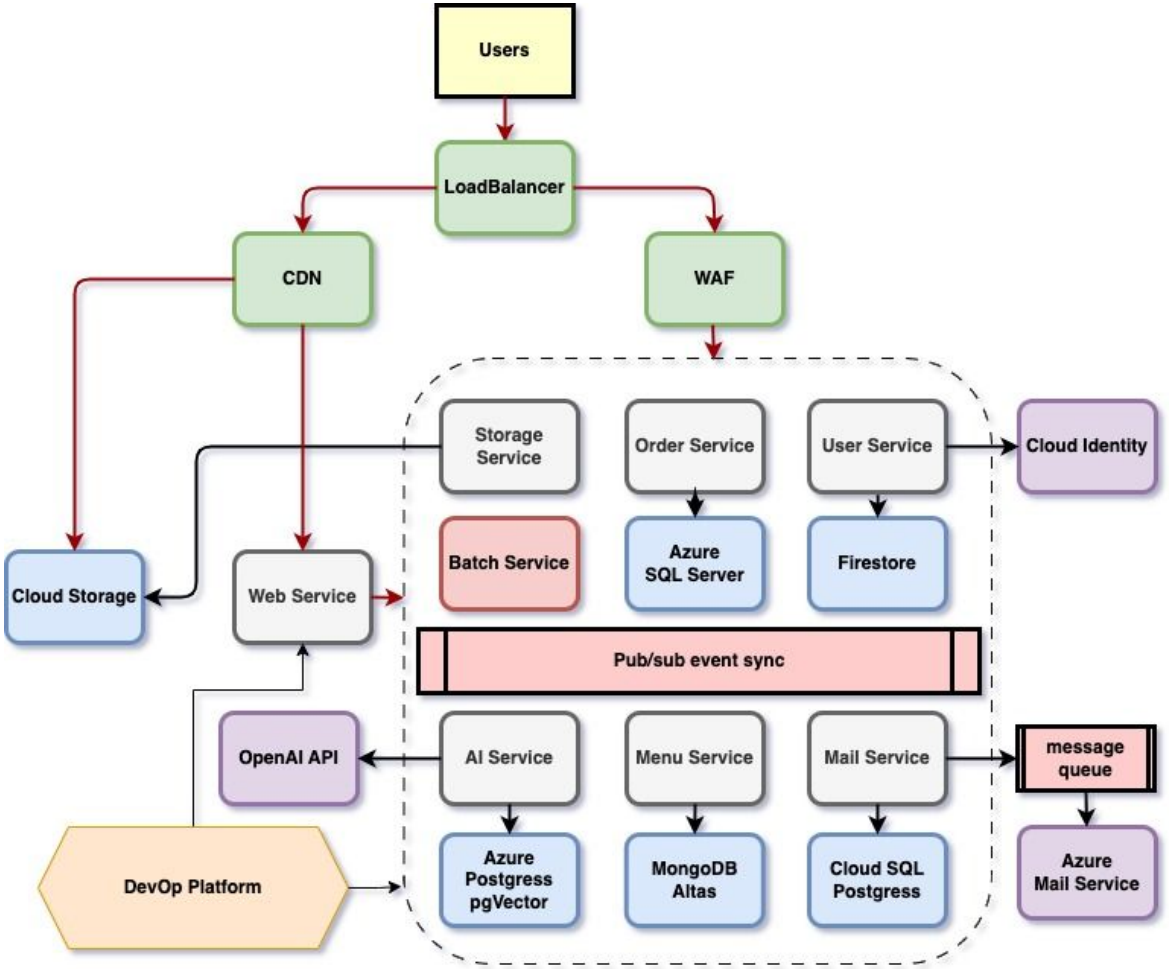
Demo

https://ntuartfest28th.com/

System Architecture

overview

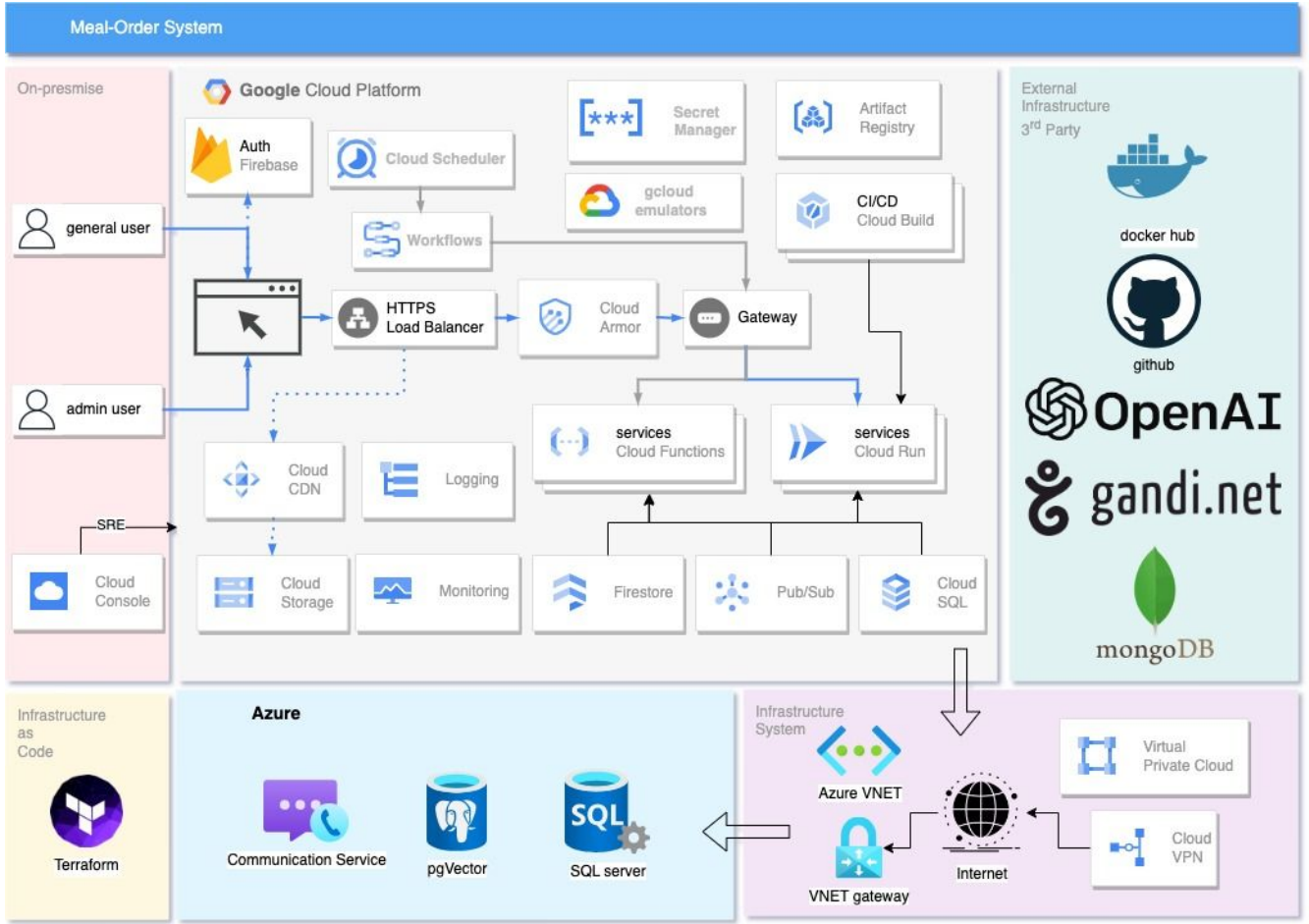
micro service
+
event driven



Infra

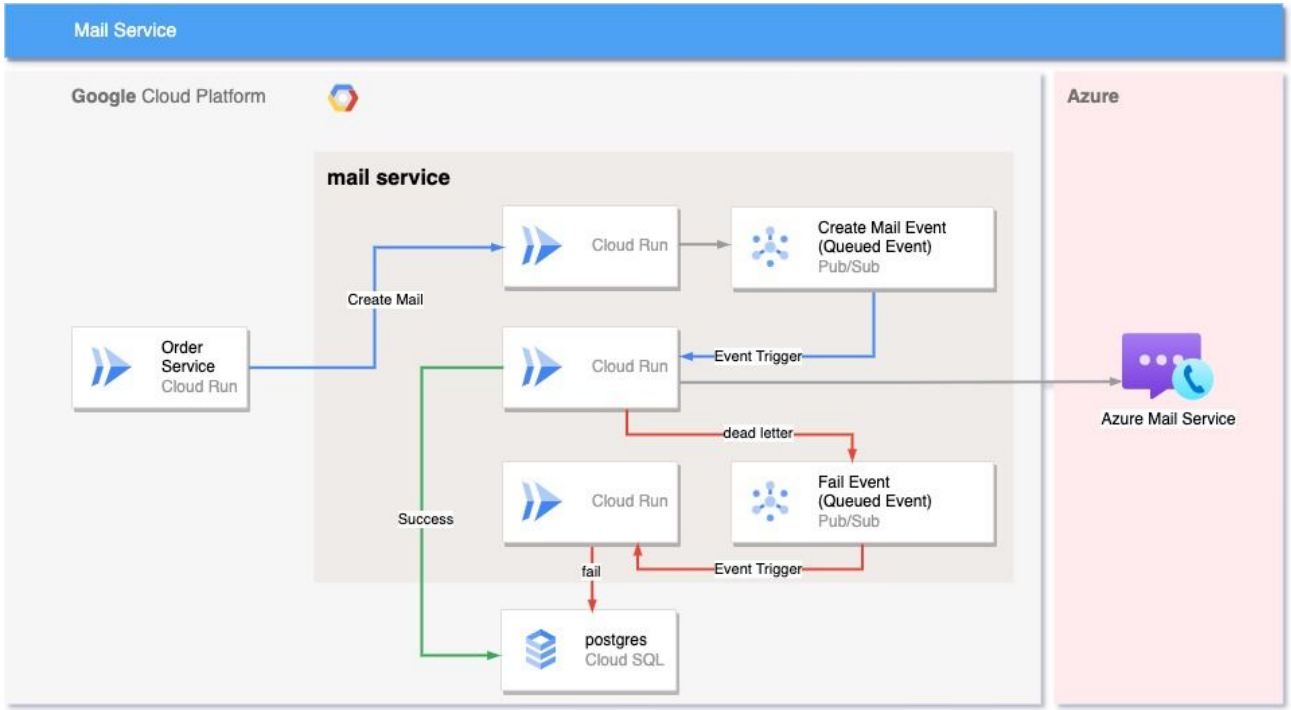
GCP Resouce Created By Terraform (IaC)

https://github.com/leon123858/tsmc-meal-order/tree/main/infra



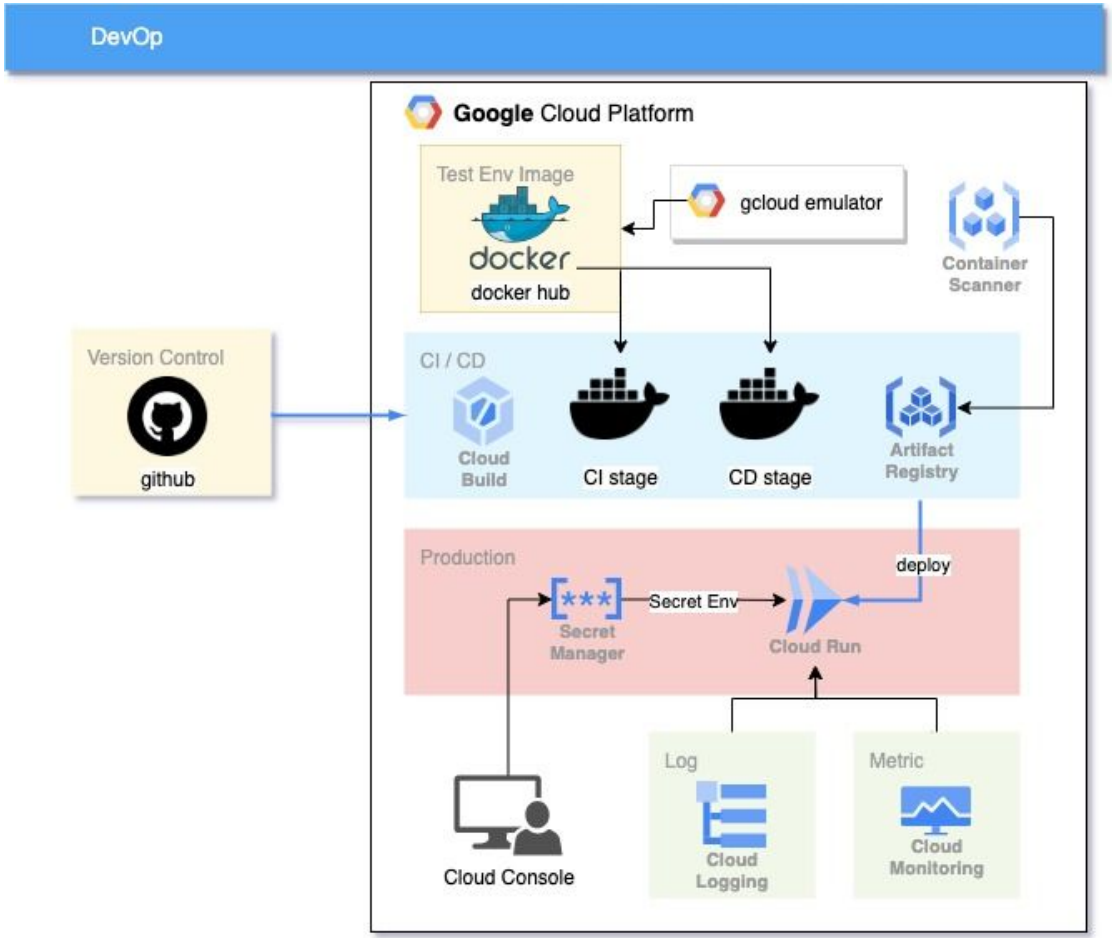
mail service HA structure

message
queue
+
dead letter



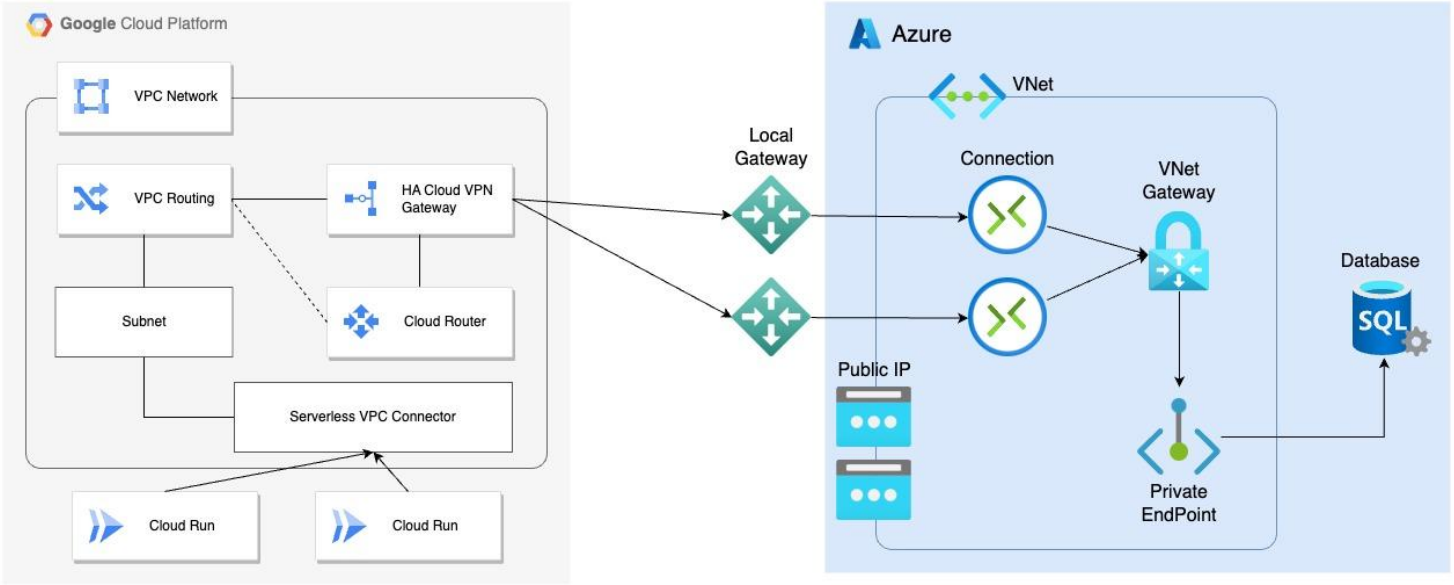
DevOp

CI / CD
pipeline
+
log & monitor



VPN

建立 S2S VPN 來提高資料安全性，並且使用BGP 的自動容錯移轉來確保VPN 的高可用性。



Monitor Metrics

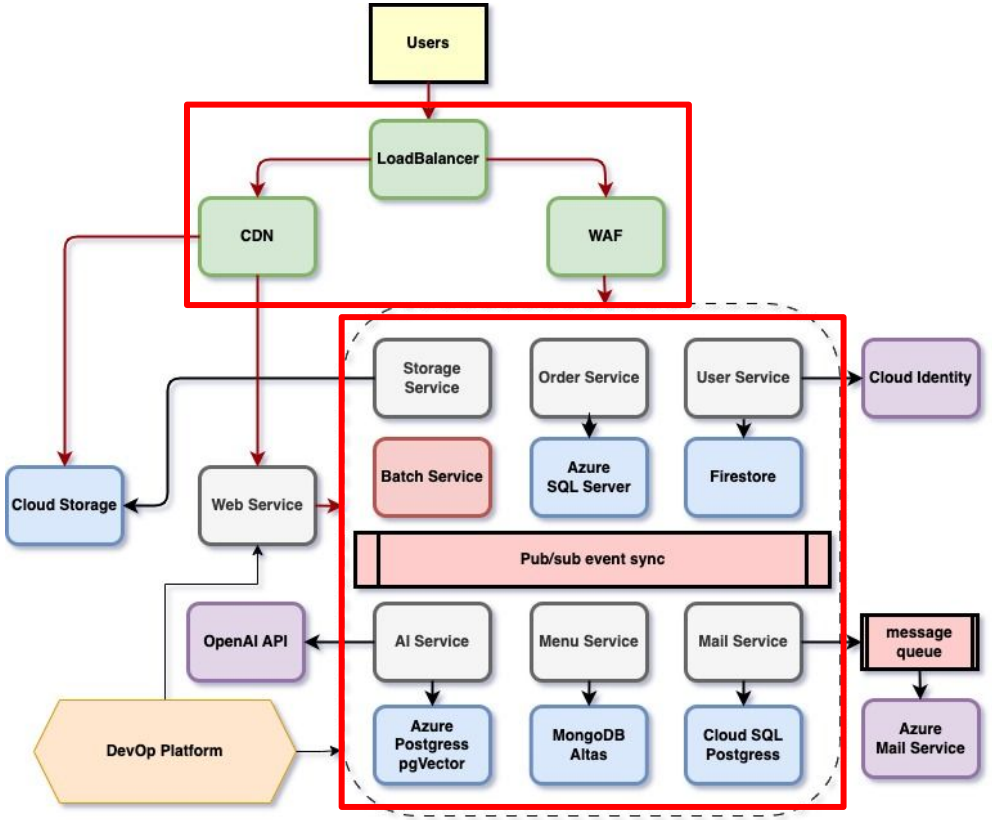
Monitor

監看關鍵

LoadBalancer

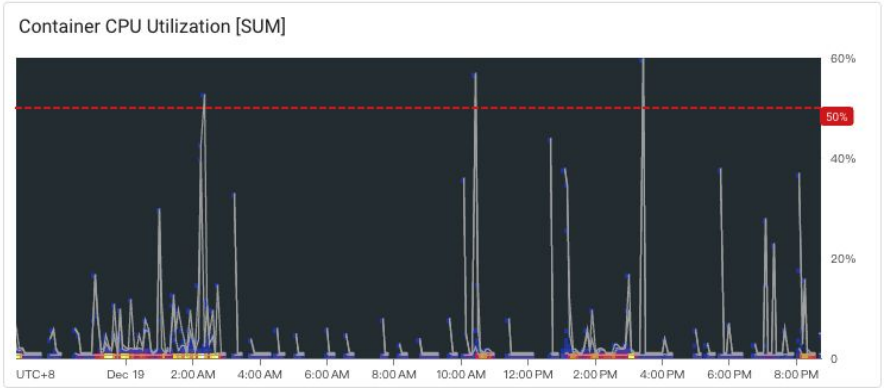
+

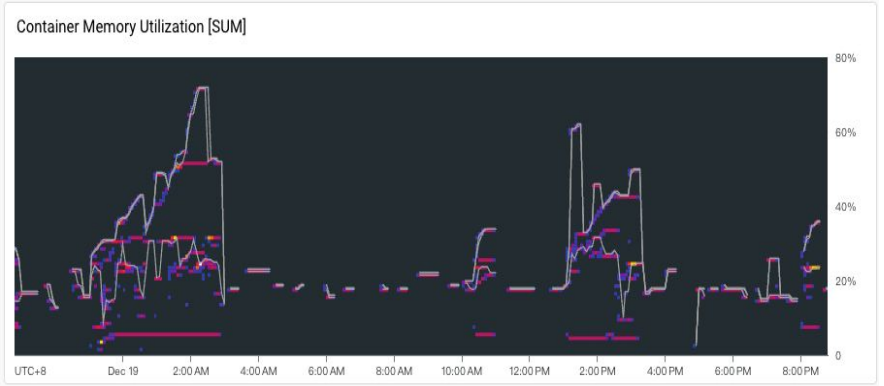
Compute Instance

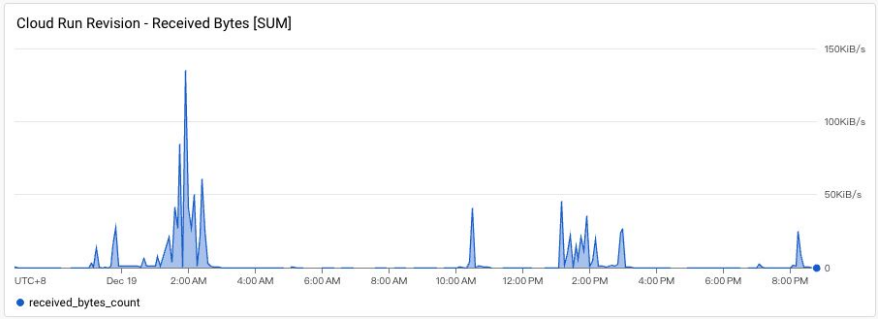


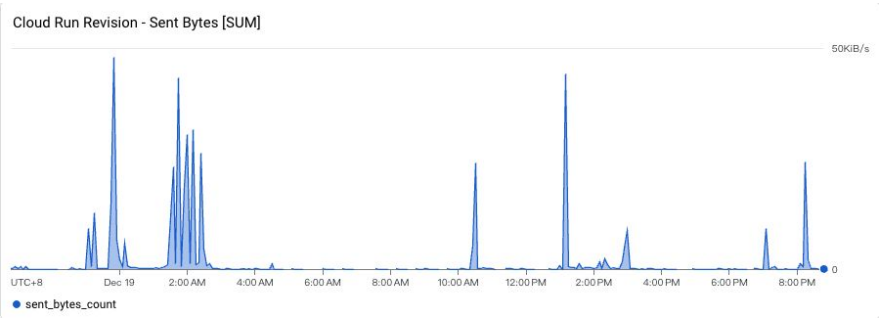
Monitor Instance

監看 CPU, Mem, throughput



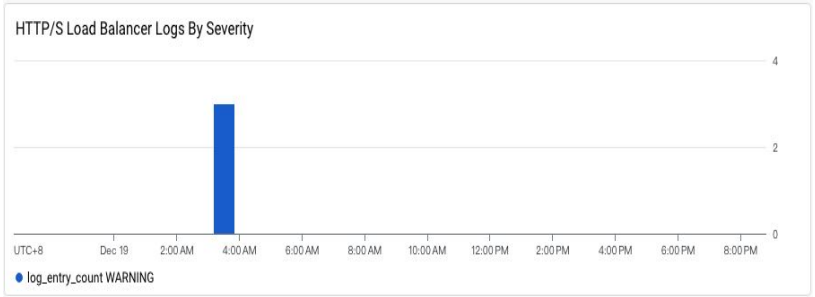


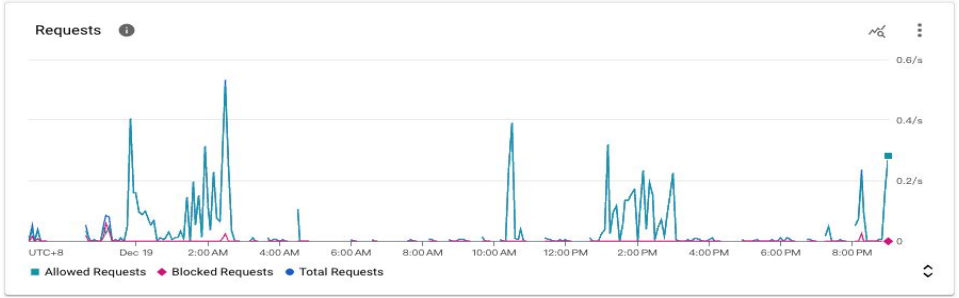


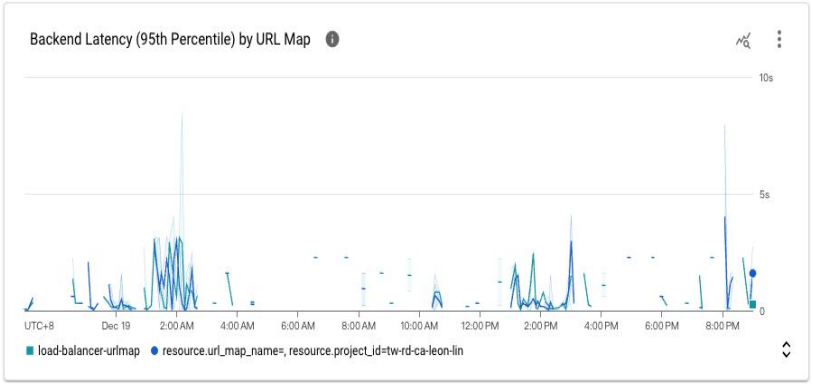


Monitor LB

確認入口流量 + 後端狀態





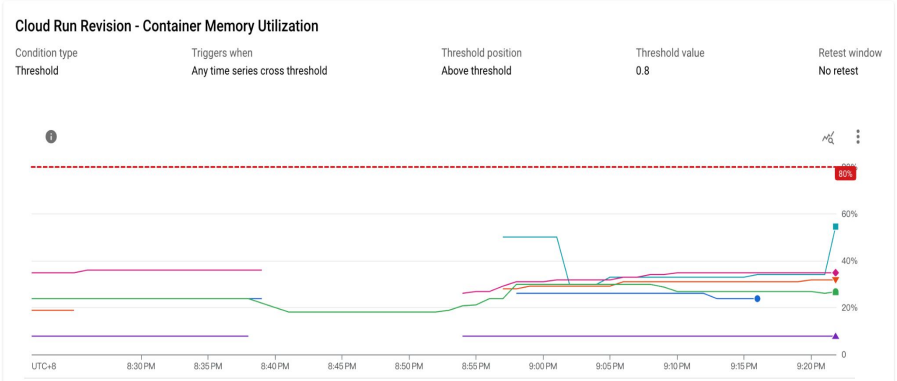


Backend Target	Backend	5xx Error Ratio ↓	Backend Latency (95th Percentile)	Total Latency (95th Percentile)	Health	Location
menu-backend	menu-neg	0%	2,997ms	2,997ms	N/A	asia-east1
order-backend	order-neg	0%	3,099ms	3,099ms	N/A	asia-east1
storage-backend	storage-neg	0%	392ms	392ms	N/A	asia-east1
user-backend	user-neg	0%	96ms	97ms	N/A	asia-east1
web-backend	web-neg	0%	1,373ms	1,373ms	N/A	asia-east1
ai-backend	ai-neg	–	–	–	N/A	asia-east1
mail-backend	mail-neg	–	–	–	N/A	asia-east1
load-balancer-bucket	meal-order-static	–	–	–	N/A	global

Monitor Alert 爆預算 & 後端服務中斷 email 警告專案持有者

Budget name	Budget period	Budget type	Applies to	Trigger alerts at	Spend and budget amount
tw-rd-ca-leon-lin	Monthly	Specified amo...	1 project and 6 credit types	80% and 100%	$41.90 / $50.00 Includes -$2.73 credit

Name	5xx Error Ratio	Backend Latency (95th Percentile)	Total Latency (95th Percentile)	Forwarding Rules	Health Status
load-balancer-urlmap	0%	1,338ms	1,356ms	34.149.240.181	7 backend services (0 instance groups, 7 network endpoint groups) 1 backend bucket



Summary

Incidents firing	Incidents acknowledged	Alert policies
0	0	1

View all

Unit Test

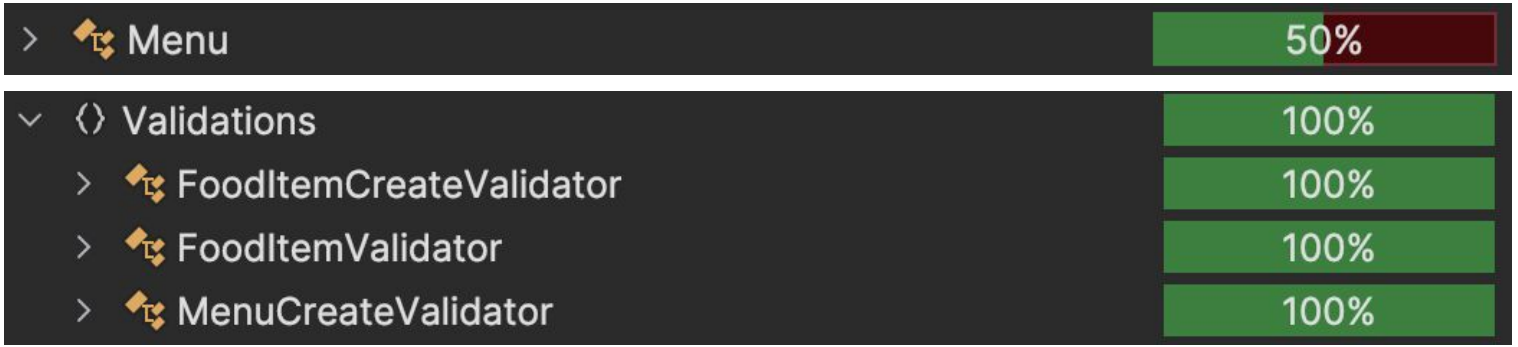
Unit Test - User

在 Service Layer 達到 78.2% 的 Test Coverage。

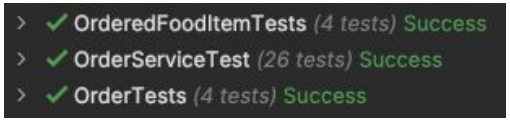
```
Running script: go test ./... -p=1 -cover=true
    github.com/leon123858/tsmc-meal-order/user  [no test files]
    github.com/leon123858/tsmc-meal-order/user/controller   [no test files]
    github.com/leon123858/tsmc-meal-order/user/docs [no test files]
    github.com/leon123858/tsmc-meal-order/user/model    [no test files]
    github.com/leon123858/tsmc-meal-order/user/service  0.273s  coverage: 78.2% of statements
    github.com/leon123858/tsmc-meal-order/user/utils    [no test files]
```

Unit Test - Menu

在 Domain Layer 達到平均 90% 的 Test Coverage。



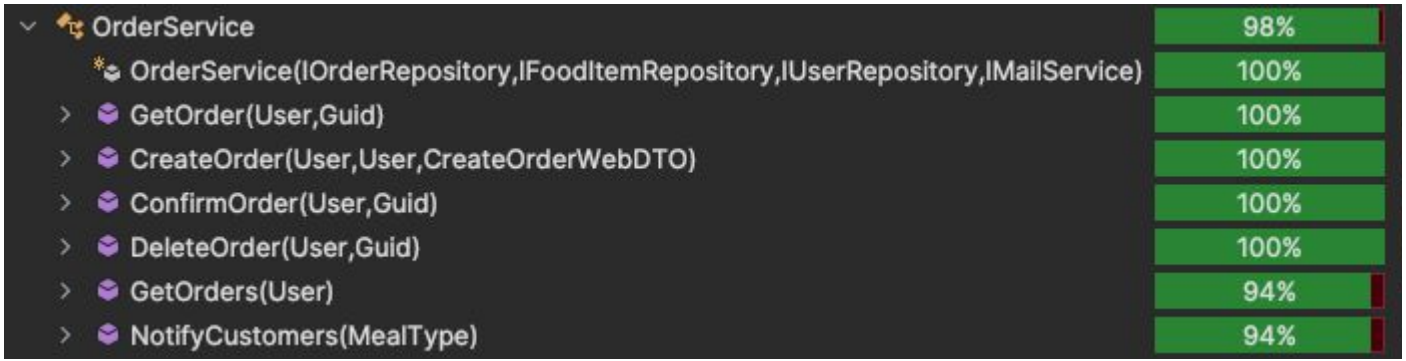
總共撰寫了 34 個 Unit Test。



Unit Test - Order

在 Domain Layer 達到平均 93% 的 Test Coverage。

在 Application Service Layer 達到平均 98% 的 Test Coverage。



Unit Test - CI / CD

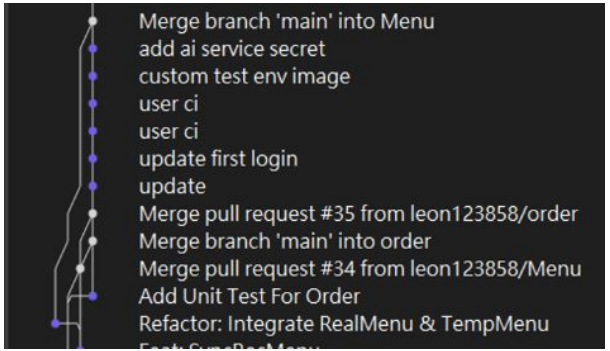
有將 Unit Test 加入 CI / CD 流程中，會在 Cloud Build 建置時確認是否通過測試。

```
Running script: go test ./... -p=1 -cover=true
    github.com/leon123858/tsmc-meal-order/user  [no test files]
    github.com/leon123858/tsmc-meal-order/user/controller   [no test files]
    github.com/leon123858/tsmc-meal-order/user/docs [no test files]
    github.com/leon123858/tsmc-meal-order/user/model    [no test files]
    github.com/leon123858/tsmc-meal-order/user/service  0.273s  coverage: 78.2% of statements
    github.com/leon123858/tsmc-meal-order/user/utils    [no test files]

Starting test execution, please wait...
A total of 1 test files matched the specified pattern.

Passed!  - Failed:     0, Passed:    41, Skipped:     0, Total:    41, Duration: 680 ms - core_test.dll (net7.0)
```

12 Factors

12-Factors

1. **Codebase :** version control & single source of truth



2. **Dependencies :** specific package versions

```
ItemGroup>
 <PackageReference Include="AutoMapper.Extensions.Microsoft.DependencyInjection" Version="12.0.1" />
 <PackageReference Include="FluentValidation.DependencyInjectionExtensions" Version="11.8.0" />
 <PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="7.0.13" />
 <PackageReference Include="Microsoft.VisualStudio.Azure.Containers.Tools.Targets" Version="1.17.0" /
 <PackageReference Include="MongoDB.Driver" Version="2.22.0" />
 <PackageReference Include="Newtonsoft.Json" Version="13.0.3" />
 <PackageReference Include="Swashbuckle.AspNetCore" Version="6.5.0" />
```

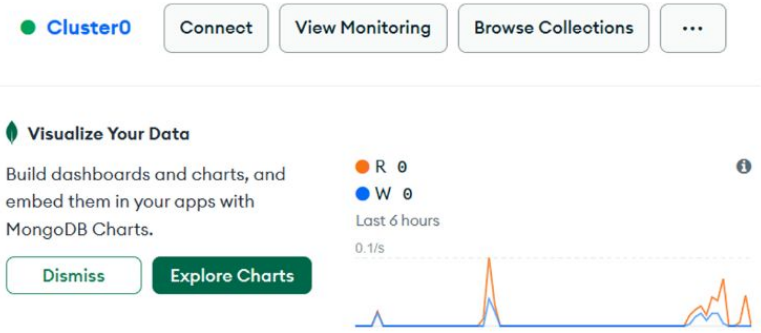
12-Factors

3. **Config :** Keep configs in the cloud platform, instead of the version control tools

環境變數 (3)

名稱	值
SERVER_PORT	:8080
ASPNETCORE_ENVIRONMENT	Production
MONGO_PASSWORD	密鑰：menu-secret-MONGO_PASSWORD:latest

4. **Backing Services :** Loosely coupling with other cloud services or databases



12-Factors

5. **Build, release, run :** Cloud Build & Cloud Run on GCP

12-Factors

6. **Processes :**
    a. processes have no dependency on local memory (within container)
    b. no shared local memory between processes
    c. (a+b) => stateless

7. **Port binding :**
    a. Unified port protocol between all apps, such as 443: HTTPS

12-Factors

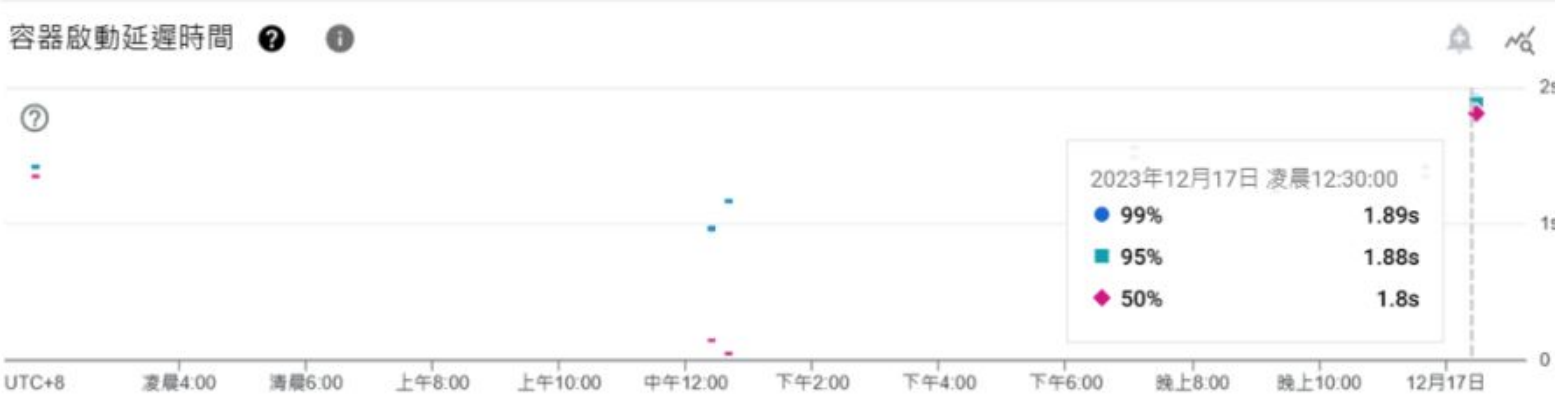
8. **Concurrency :** Auto-scaling handled by GCP Cloud Run autoscaler

   In addition to the rate of incoming requests or events, the number of instances scheduled is impacted by:

   a. The CPU utilization of existing instances when they are processing requests or events over a one minute window, targeting to keep scheduled instances to a 60% CPU utilization.
   b. The current request concurrency, compared to the maximum concurrency over a one minute window.
   c. The maximum number of instances setting
   d. The minimum number of instances setting
   e. The Cloud Run autoscaler evaluates these every 5 seconds.

12-Factors

9. **Disposability :** handled by serverless container



12-Factors

	Traditional app	Twelve-factor app
Time between deploys	Weeks	Hours
Code authors vs code deployers	Different people	Same people
Dev vs production environments	Divergent	As similar as possible

10. Dev/Prod Parity :
configs for different envs.

```
"MenuDatabase": {
  "DatabaseName": "MenuDatabase",
  "RealCollectionName": "MenusBackup",
  "TempCollectionName": "Menus",
  "UserName": "bevis",
  "Password": "default",
  "Cluser": "cluster0.ujzhjoz.mongodb.net/"
},
"WebApi": {
  "UserUrl": "https://user-kt6w747drq-de.a.run.app/api/user",
  "RecUrl": "https://ai-kt6w747drq-de.a.run.app/api/ai"
},
"Logging": {
  "LogLevel": {
    "Default": "Information",
    "Microsoft.AspNetCore": "Warning"
  }
},
"AllowedHosts": "*"
```

```
"MenuDatabase": {
  "DatabaseName": "MenuDatabase",
  "RealCollectionName": "MenusBackup",
  "TempCollectionName": "Menus",
  "UserName": "bevis",
  "Password": "default",
  "Cluser": "cluster0.ujzhjoz.mongodb.net/"
},
"WebApi": {
  "UserUrl": "https://user-kt6w747drq-de.a.run.app/api/user",
  "RecUrl": "https://ai-kt6w747drq-de.a.run.app/api/ai"
},
"Logging": {
  "LogLevel": {
    "Default": "Information",
    "Microsoft.AspNetCore": "Warning"
  }
},
"Kestrel": {
  "Endpoints": {
    "Http": {
      "Url": "http://0.0.0.0:8080"
    }
  }
},
"AllowedHosts": "*"
```

12-Factors

11. **Logs (or, Telemetry) :** GCP is responsible for the storage of log files

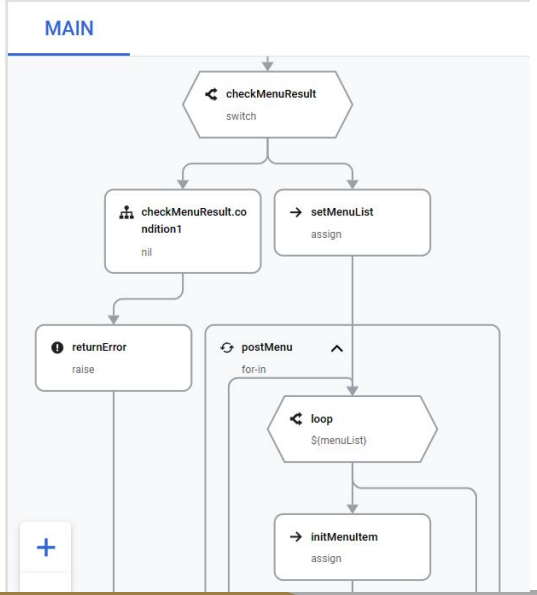
12-Factors

12. **Admin Process :** Utilize GCP WorkFlows to maintain the system

程式碼 編輯

```
- syncMenu:
    call: http.post
    args:
      url: "https://menu-kt6w747drq-de.a.run.app/api/menu/sync"
      headers:
        Content-Type: application/json
    result: syncResult
- logSync:
    call: sys.log
    args:
      text: ${json.encode_to_string(syncResult.body)}
- getMenu:
    call: http.get
    args:
      url: "https://menu-kt6w747drq-de.a.run.app/api/menu"
    result: menuResult
- checkMenuResult:
    switch:
      - condition: ${menuResult.body["result"] == false}
        next: returnError
    next: setMenuList
- setMenuList:
    assign:
      - menuList: ${menuResult.body.data}
```

視覺化內容



Thank you!

https://ntuartfest28th.com/