

# 分散式專案文本報告

## 小組成員

r10725012 呂晟維

r10725013 陳志剛

r10725052 楊柏睿

## 情境

某島嶼大學是該國高中生夢寐以求的學府。然而，由於高中生在學習經歷中往往缺乏面試經驗，他們需要初步了解各科系面試的內容，同時也希望在這個探索的過程中能夠發掘出自己的興趣和性向。為了滿足這樣的需求，我們計劃建立一個島嶼大學面試介紹網站，成為一個有價值的資源平台。

這個網站將提供各個科系面試的相關資訊和情境，讓高中生能夠事先瞭解並準備面試所需的知識和技能。網站上的內容將根據每個科系的特點和要求進行詳細介紹，包括面試的常見問題、技術測試、實際案例等等。這些資訊將幫助考生更好地理解面試過程，提前做好準備。

## 網站功能

- Query 所有面試資訊的檔案。
- Preview 單個面試資訊的檔案內容。
- Download 單個面試資訊的檔案。
- Upload 一或多個面試資訊的檔案。
  - 網站介面截圖: 上方區域為所有面試資訊檔案的總表，當我們點擊預覽便能在下方區域顯示檔案內容。我們的測試資料有10筆科系介紹檔案，每個檔案皆有300~1000字。

島嶼大學各系面試資料

Year	Department	Title	Preview	File URL
107	國企系	邁向Elite之路暨院生的面試手冊.doc	預覽	點我下載
108	數學系	數學不會就是不會.txt	預覽	點我下載
108	農藝系	你知道烏龍茶和普洱茶的差異嗎.pdf	預覽	點我下載
109	大氣系	不要再問今天會不會下雨了.txt	預覽	點我下載
109	法律系	賄賂 賄賂 賄賂到底是什麼.txt	預覽	點我下載
109	資訊系	資訊管理學系科系介紹.txt	預覽	點我下載
110	中文系	面試問答 不僅長講話能讀中文永福.txt	預覽	點我下載
110	公衛系	面試絕對不會問參訪台北市長的意圖.txt	預覽	點我下載
110	生科系	想了解更多科系的一定要看.txt	預覽	點我下載
110		sample.txt	預覽	點我下載

QR Code 1

QR Code 2

前端

後端 (上傳)

Archive Preview:

國際企業管理（國企）系是一個提供學生綜合國際商業知識和專業技能的系。該系致力於培養學生成為具有全球視野和國際商業知識的專業人才，在現今全球化的商業環境中，國際企業管理的專業人士扮演著關鍵角色，能夠有效地區隔和聯繫企業之間的國際關係。

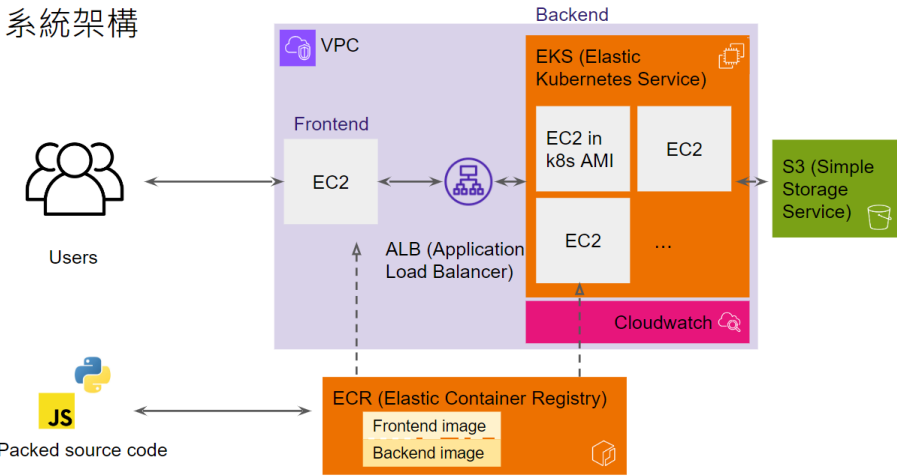
在國企系四年的課程中，學生將學習關於國際貿易、國際市場分析、跨文化管理、國際金融、國際行銷等相關領域的知識。學生也將學習如何進行國際商務談判、制定國際策略和解決國際商業挑戰的能力。此外，學生還將瞭解全球經濟和商業法規等相關議題，並學習如何應對不同國家和地區的文化差異。

這個科系的畢業生有許多就業機會，可以在國際企業、跨國公司、貿易組織、國際金融機構和政府部門等各種組織中工作。他們可以擔任國際業務經理、進出口專員、國際市場分析師、國際品牌經理、國際業務發展等職位。此外，畢業生也有機會在不同國家和地區工作，體驗不同的商業環境和文化。

# 使用哪些AWS套件達到CI/CD

CI/CD功能	AWS服務
Docker Container Auto Scaling	EKS , ECR
Application Request Load Balancing	ALB
Network Throughput Record	Cloudwatch
Container Fault Detection and Auto Recovery	ASG

# 系統架構



## A. 資料儲存

我的使用的資料庫為 noSQL 的 AWS S3，將每個文檔以<檔名-檔案>的 key-value 儲存管理。起初我們的設定為每台機器可以 local cache 被請求過的檔案在本地的檔案系統中，當緩存文件數量超過60%，會自動刪除最舊的文件，讓資料庫與後端間的負荷不要太大。

報告後我們統整老師和助教的意見，做出以下改動，讓緩存機制更兼具邏輯與效率。

- 廢除緩存 60% 文件數量的限制，改為無上限。但新增以下功能。
- File consistency handling: 不允許修改檔案，只允許上傳同名檔案覆蓋原本檔案 (versioning)。但上傳檔案時只會覆蓋一台後端以及S3中的檔案，此時其他的後端的 local cache 會產生不一致的情形。我們允許 local cache 擁有5分鐘的有效性，一但超過5分鐘便會自動刪除本地緩存，達到 eventually consistence 。
- Hot spot handling: (1)管理者可額外設定不能被刪除的重要檔案。(2)當有檔案某台後端上在2分鐘內的請求次數超過10次，視為熱門檔案。此時後端會向 Load balancer 發出一次該熱門檔案的請求，讓此熱門檔案擴散出去。(不使用redis的原因是，我們已經手刻一個功能類似的緩存機制了)

## B. 服務請求流程

前端請求會透過 Load Balancer 傳遞至各個後端，因為根據2023 年教育部統計資料，高三畢業的升學學生約有 15 萬人，當面試季節來到，如此規模的人都要跟我們後端請求資料時就需要自動擴增後端來處理大量的流量，而面試季節過了之後就會自動縮減後端資源。後端會用 Cache 來優化服務，因為跟 S3 請求的這個動作的成本較高。

經過壓力測試實驗，我們發現在一開始前端並不存在瓶頸，因此將 Load balancer 設計在後端的 EKS cluster 之前。若正式上線使用時，前端成為瓶頸，則可將前端也加入 EKS cluster 中並加入依 CPU 使用率自動擴展功能，也將 Application Load balancer 改設在前端之前，依照不同的請求 route 流量給後端。

# 自動擴展規劃 & 壓力測試

## A. 實驗以找出瓶頸

我們沒有刻意設計系統的瓶頸，為了找出真正的瓶頸，我們設計了兩組實驗。

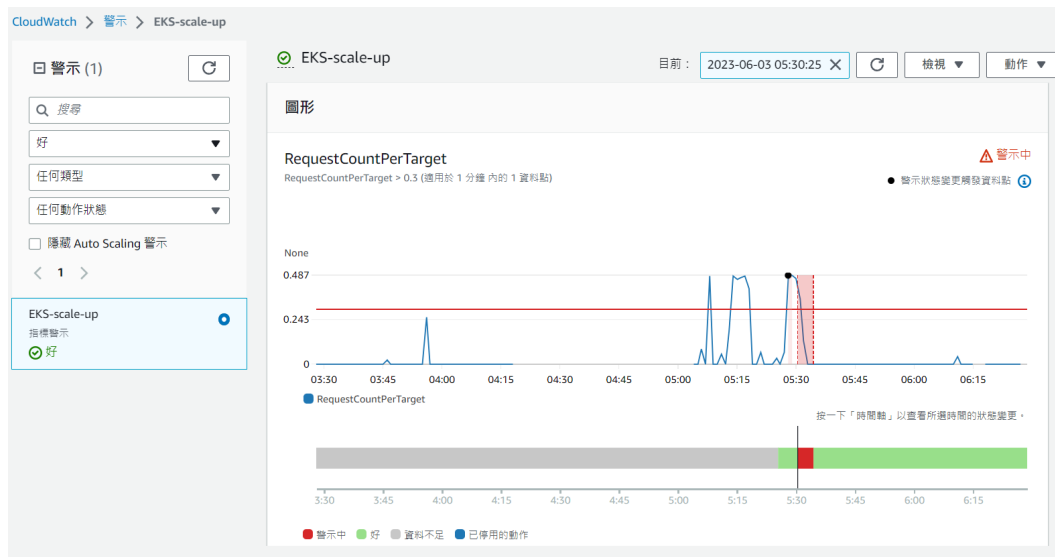
- 實驗一: 測試後端至aws S3下載所有測試文本的時間 → 6.7 seconds
- 實驗二: 測試前端至後端下載所有測試文本的時間 → 0.3 seconds (∵相同VPC)

由實驗結果很明顯的得出，系統瓶頸在實驗一。推測原因為前端與後端處於相同的 VPC，而 S3 是全球性的儲存空間，我們教育版的帳號也只能使用較低階的 S3。

得知了瓶頸所在，那麼觸發瓶頸的時機為何？從實驗結果推論，當下載請求不斷發送時，會堆積在與 S3 資料庫的連線上。因此我們為後端加上本地緩存機制 (local cache)，誠如系統架構章節的說明。

## B. 壓力測試

根據前面提到的敘述，我們認為當面試季節來到，大規模的人都要跟我們後端請求資料時就需要自動擴增後端來處理大量的流量，因此我們的壓力測試會每 0.1 秒發出 1000 個請求，用以模擬短時間內大規模的請求。



## 實作時遭遇的困難

### A. Docker Container 打包歷時太久

解決方案:

- 更換成 Python 的基礎映像檔 (Base Image)，而非 Ubuntu image，藉此輕量級的基礎映像檔來減少額外套件的安裝，有效節省打包時間。
- 優化依賴管理，不使用 aws configure 的 Linux 版指令集工具，改為使用 ENV 環境變數來存放 access\_key\_id 等帳戶相關的資訊。

### B. EKS (Elastic Kubernetes Service) 無法使用 EC2 AMI 作為 image

- 使用 Amazon 官方提供的 EKS Linux image，讓 scale-out 的新機器能連線回 k8s 並加入 NodeGroup。

- 但這個優化也帶來了一寫困擾。比如我們須依格式撰寫啟動範本的腳本，而不能直接撰寫 bash script，才能在官方 image 中部屬並執行我們後端的容器服務。(腳本難撰寫，從原先 5 行指令 => 增加到30行才得以達到相同的成效，如下圖)

```
使用書資料

Content-Type: multipart/mixed; boundary=""
MIME-Version: 1.0

--//
Content-Type: text/cloud-config; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="cloud-config.txt"

#cloud-config
cloud_final_modules:
- [scripts-user, always]

--//
Content-Type: text/x-shellscript; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="userdata.txt"

#!/bin/bash
sudo yum install docker -y
sudo yum install amazon-cloudwatch-agent -y
sudo systemctl start docker

export AWS_ACCESS_KEY_ID=ASIAQSP745PEC2866UQ0
export AWS_SECRET_ACCESS_KEY=h5c5fg750g1512g9bCd2Yz8I40dePurLruaXr
export
AWS_SESSION_TOKEN=Fw6ZK1vYKdsELv//////////wEadH169SIxc0XyOfnFLLAU2B0504H4TTu1ouPU8nuu1dof5nR3V4s/W23a8/xYQ1Q1Q+ESe6FzLFuad7W4uK17uV9cALZCnr84myF8hebeKhTfhw313glCmfXORzu7XD7uIDpkoUuBFeRbtf7ZCPs8z7H8tz
cFOW1g8ryxh9xykXqZhsRub5aT1lmFLZ19W20wr1m/wCofrPHDGL7yH7Hp9wQbAymq0YHw5sMXX5FamHT1R5qsAodw1BLI218+Csd2e3x83n7LrZeEb0yKPK76qM9H129ghVYg0u107s1ryOn4AK/E1+955Qua+xbx5TLnw8JKAUXre1K94qvZay3yw8=

sudo aws s3 cp s3://2023ds-config/config.json /home/ec2-user/config.json
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -s -c file://home/ec2-user/config.json

sudo docker login -u AWS -p $(aws ecr get-login-password --region us-east-1) 063325400008.dkr.ecr.us-east-1.amazonaws.com

sudo docker pull 063325400008.dkr.ecr.us-east-1.amazonaws.com/backend:latest

sudo docker run -d -p 5000:5000 -t 063325400008.dkr.ecr.us-east-1.amazonaws.com/backend:latest

--//
```

啟動範本

EC2 > AMI > ami-0b94943bd76cb55c2			
ami-0b94943bd76cb55c2 的映像摘要			
		EC2 Image Builder	動作 ▼ 從 AMI 啟動執行個體
AMI ID	映像類型	平台詳細資訊	標裝置類型
ami-0b94943bd76cb55c2	machine	Linux/UNIX	EBS
AMI 名稱	擁有者帳戶 ID	架構	用量操作
amazon-eks-node-1.27-v20230526	602401143452	x86_64	RunInstances
根裝置名稱	狀態	來源	虛擬化類型
/dev/xvda	可用	amazon/amazon-eks-node-1.27-v20230526	hvm
開機模式	狀態原因	建立日期	核心 ID
-	-	Wed May 31 2023 02:36:25 GMT+0800 (台北標準時間)	-
區塊型儲存設備	描述	產品代碼	RAM 磁碟 ID
/dev/xvda=snap-030a22dd30861dc70:20:true:gp2	EKS Kubernetes Worker AMI with AmazonLinux2 image, (k8s: 1.27.1, containerd: 1.6.4)	-	-
棄用時間			
Sat May 31 2025 02:36:25 GMT+0800 (台北標準時間)			

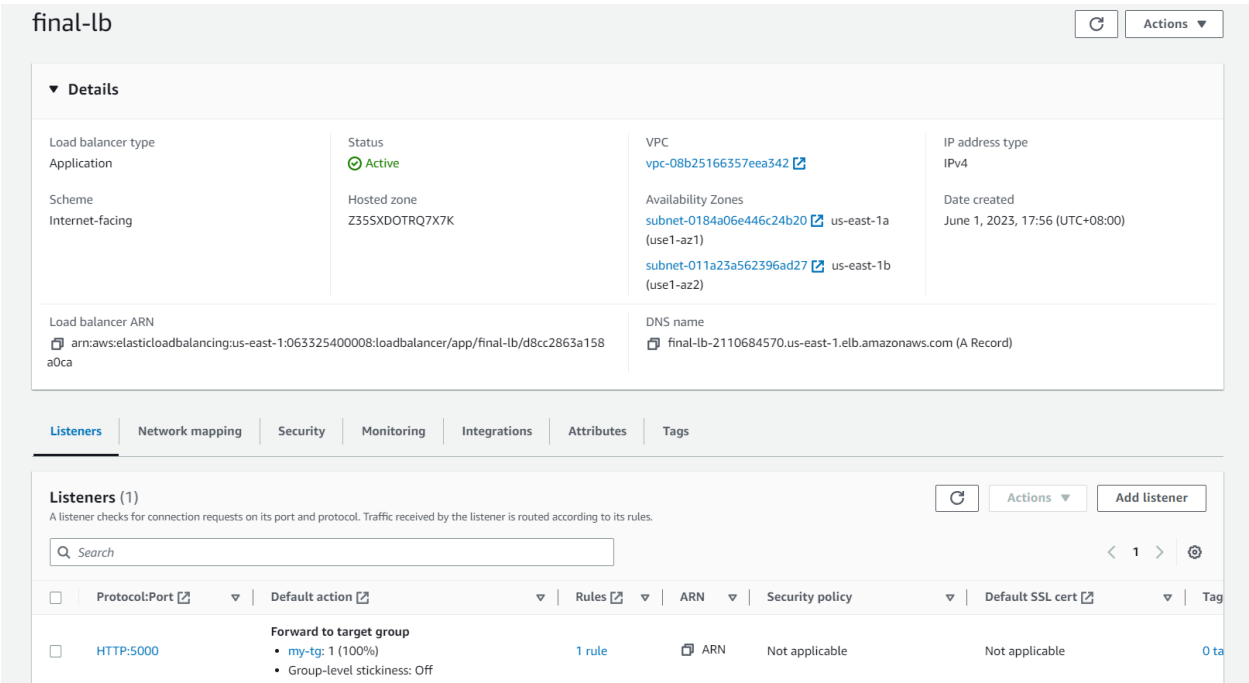
以 Kubernetes Service 提供的 EKS Linux node image 作為基底映像

## 專案的程式碼

- 前後端的源碼請參見 Github repo: <https://github.com/weberlu88/2023-Spring-NTU-DS-project>。
- AWS 的設定檔如上述文件的附圖和附錄截圖。

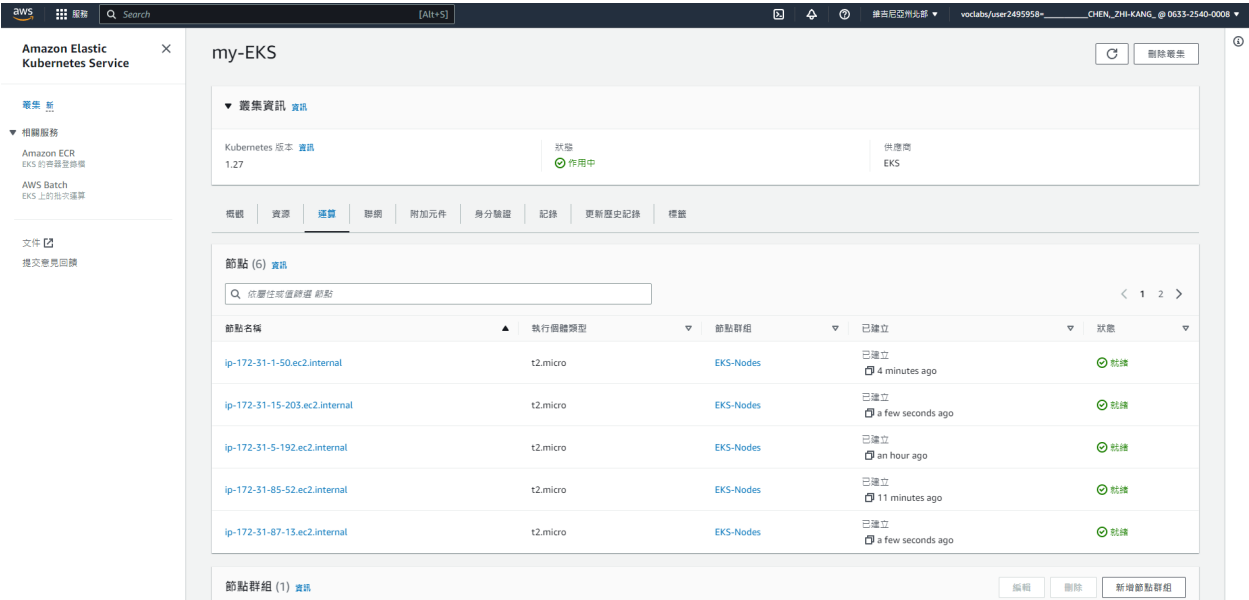
附錄：截圖

Load balancer



Load balancer

EKS cluster (含有一個node group)



所有的 EC2 機器，後5個是 EKS node Group 的中機器

執行個體 (12) 資訊

Q 依屬性或標籤 (case-sensitive) 尋找 執行個體

<input type="checkbox"/>	Name	執行個體 ID	執行個體狀態	執行個體類型	狀態檢查	警報狀態	可用區域	公有 IPv4 DNS	公有 IPv4 地址
<input type="checkbox"/>	test	i-02aa191b06da4752b	執行中	t2.micro	2/2 項檢查通過	無警報	us-east-1c	ec2-18-213-4-4.comput...	18.213.4.4
<input type="checkbox"/>	myNode	i-017355d7d6f964170	執行中	t2.micro	2/2 項檢查通過	無警報	us-east-1b	ec2-3-88-181-155.com...	3.88.181.155
<input type="checkbox"/>	bk-node	i-0233eab4ab93ed359	執行中	t2.micro	2/2 項檢查通過	無警報	us-east-1b	ec2-3-83-191-206.com...	3.83.191.206
<input type="checkbox"/>	-	i-04760a385ff6c4bf9	執行中	t2.micro	2/2 項檢查通過	無警報	us-east-1b	ec2-52-23-196-163.co...	52.23.196.163
<input type="checkbox"/>	-	i-0c261ddc285913857	執行中	t2.micro	2/2 項檢查通過	無警報	us-east-1b	ec2-54-172-135-172.co...	54.172.135.17
<input type="checkbox"/>	-	i-00ab28a7ff7e7e55	執行中	t2.micro	2/2 項檢查通過	無警報	us-east-1a	-	-
<input type="checkbox"/>	-	i-0ca93559b9ee18eed	執行中	t2.micro	2/2 項檢查通過	無警報	us-east-1a	ec2-3-238-189-82.com...	3.238.189.82
<input type="checkbox"/>	-	i-0206e7fe254e25586	已終止	t2.micro	-	無警報	us-east-1b	-	-
<input type="checkbox"/>	-	i-088e6d5f2c65e9963	已終止	t2.micro	-	無警報	us-east-1b	-	-
<input type="checkbox"/>	-	i-0b577156b15cd790	已終止	t2.micro	-	無警報	us-east-1b	-	-
<input type="checkbox"/>	-	i-0092f2488537462c6	已終止	t2.micro	-	無警報	us-east-1b	-	-
<input type="checkbox"/>	-	i-0ecfd9f273ea6d60f	已終止	t2.micro	-	無警報	us-east-1a	-	-

EKS node 的詳細設定資訊

Amazon Elastic Kubernetes Service

EKS > 叢集 > my-EKS > 節點組別: EKS-Nodes

### EKS-Nodes

節點組別組態 資訊

Kubernetes 版本 1.27	AMI 類型 資訊 AL2_x86_64	啟動範本 <a href="#">bk-template</a>	狀態 作用中
AMI 發行版本 資訊 1.27.1-20230526	執行個體類型 t2.micro	啟動範本版本 1	磁碟大小 已在啟動範本中指定

詳細資訊 | 節點 | 操作狀態問題 0 | Kubernetes 標籤 | 更新組態 | Kubernetes Taint | 更新歷史記錄 | 標籤

#### 詳細資訊

節點組別 ARN <a href="#">arn:aws:eks:us-east-1:063325400008:nodegroup/my-EKS/EKS-Nodes/04c3fce-6985-5ad3-0105-eab83a57aee2</a> 已建立 an hour ago	自動調整根據節點名稱 <a href="#">eks-EKS-Nodes-04c3fce-6985-5ad3-0105-eab83a57aee2</a> 節點 IAM 角色 ARN <a href="#">arn:aws:iam::063325400008:role/LabRole</a>	容量類型 On-Demand 所需大小 5 節點 最小值 2 節點 最大值 5 節點	子網路 <a href="#">subnet-0184a06e446c24b20</a> <a href="#">subnet-011a23a562396ad27</a> 指定磁碟存取節點 標籤 EC2 金鑰對 vockey 安全群組 <a href="#">sg-01082ac3f222f84a</a>
---	--	---	---