--<https://goo.gl/YhhBA3>

# 第一天上午

## 講師自介

日商 -> 系統擴增

新創 -> 接案

大型 -> 混和

北京 -> 跨區

自開 -> 數位轉型

## 課程總攬

應用鐵三角 -> 存取, 操作, 分級授權

存取

資料列

交易(oltp)

SQL -> rds

NoSQL -> DynamoDB

分析(olap)

Redshift

影片圖片

常存取 -> S3

不常存取 -> Glacier

操作

VM (想像成是電腦) (毛巾)

EC2

Container (想像成是 用完就丟掉的電腦) (濕紙巾)

Lambda

分級授權(管理)

監控

操作行為 -> CloudTrail

運行狀態 -> CloudWatch

權限

權限分配 -> IAM

|  |  |  |  |
| --- | --- | --- | --- |
| 應用鐵三角 | | | |
| 應用 | 類型 | 項目 | 備註 |
| 存取 | 資料列 | 交易(oltp) | SQL—rds  NoSQL—DynamoDB |
| 分析(olap) | Redshift |
| 影像 | 常存取 | S3 |
| 不常存取 | Glacier |
| 操作 | VM | EC2 | 毛巾 |
| Container | Lambda | 濕紙巾 |
| 分級授權(管理) | 監控 | 操作行為 | CloudTrail |
| 運行狀態 | CloudWatch |
| 權限 | 權限分配 | IAM |

第一天

雲端的戰略味道, EC2(操作)

第二天

雲端的儲存 (S3)

第三天

雲端的權限管理(IAM)

第四天

未來的雲端方向 雲原生

管理集中化 (瀏覽器 集中管理)

**題外話-老師業配 PAPA WHILE**

## 結論

G2B2B2C

雲服務黑土地

新商業模式

專案成本彈性

技術沿用性

加密手段多

營運全球化

特有服務(資源無限, 授權費用結算方式)

學習成本

價錢 (估算, 大家的專業)

多雲選擇 ( 各家雲特色, 雲原生)

AWS經典服務

# 第一天下午

盡可能減少白板使用，增加螢幕使用率

## EC2(13:08上)

應用場景

以前的時候玩電動，準備一台電腦，把電動灌進去

執行軟體的時候，硬體當作載體

本機

虛擬機

本地有虛擬機，雲端當然也有，aws雲端叫做ec2

Ec2就是 虛擬機服務啦

設計思路/核心名詞

人類世界跟資訊世界很像

光華 -> 軟體(windows 電動) -> 硬體規格 -> 選購硬碟 -> 貼紙 -> 防火牆 -> 登入設密碼

映像檔平台 -> 映像檔 -> 虛擬機-> 虛擬硬碟 -> 標籤管理 -> 防火牆 -> 帳密/金鑰

ami marketplace -> ami ->Instance -> EBS -> tag -> Security group -> key pair

第一次實作與進階概念介紹

登入

<https://cxcxc-learning.signin.aws.amazon.com/console>

先介紹 aws 帳號體系的登入方式

Root account -> 今天晚上，幫我辦一個

明天我會帶大家做帳號預算的管理

此帳號權力很大，需要iam管理

iam user

帳密

權限管理的重要性-資深管理員絕對能幫公司架構師，讓同事們按照劇本走

**題外話 - 鄭問**

Aws console

Aws 服務的總覽頁面

Aws 全球化服務

服務分級

Global

Region

AZ

很多夥伴沒意識到->在東京開 ->在新加坡找

明明東京區的服務都停了，費用繼續增加。

EC2 Console

開機器

AMI

1a15c77b

以後公司能把軟體變成ami

Instance type

T2.micro 1core 1g

T2.medium 2core 4g

後面的英文 代表規格

重點在第一個英文字

應用場景

Configure

監控

防刪除

IP, Public, Private

**Userdata**

#!/bin/bash

touch /tibame

EBS

可調整大小

Tag

帳單管理

Security Group

Keypair

連線

Linux -> SSH

MOBAXterm -> SSH 連線

`

<https://mobaxterm.mobatek.net/download-home-edition.html>

<https://download.mobatek.net/2032020060430358/MobaXterm_Portable_v20.3.zip>

解壓縮zip，啟用mobaxterm

<https://goo.gl/YhhBA3>

下載student.pem

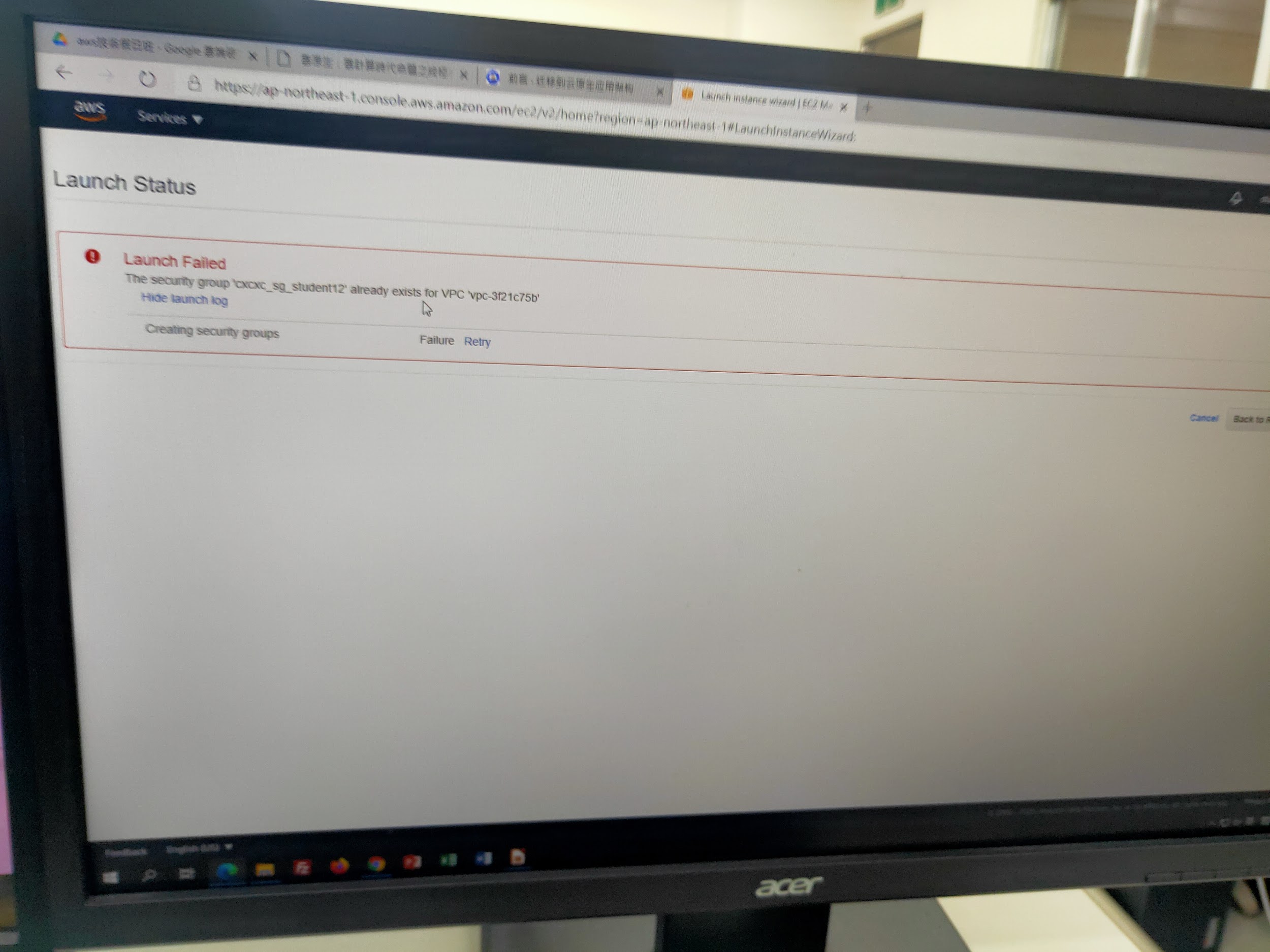
連入之後

打 ls /

發現真的有 tibame檔案

發現Userdata很有用

Windows -> 遠端桌面連線



第二次實作與進階概念介紹

Userdata 強化

在不連入電腦的情況下，把電動灌起來

#!/bin/bash

yum install -y docker

usermod -aG docker ec2-user

service docker start

docker run -d -p 8600:8080 pengbai/docker-supermario

重點在不連入，就可以灌好來

連回機器

找ip

打開mobaxterm

輸入ip 輸入username ec2-user

輸入金鑰

在命令列打上

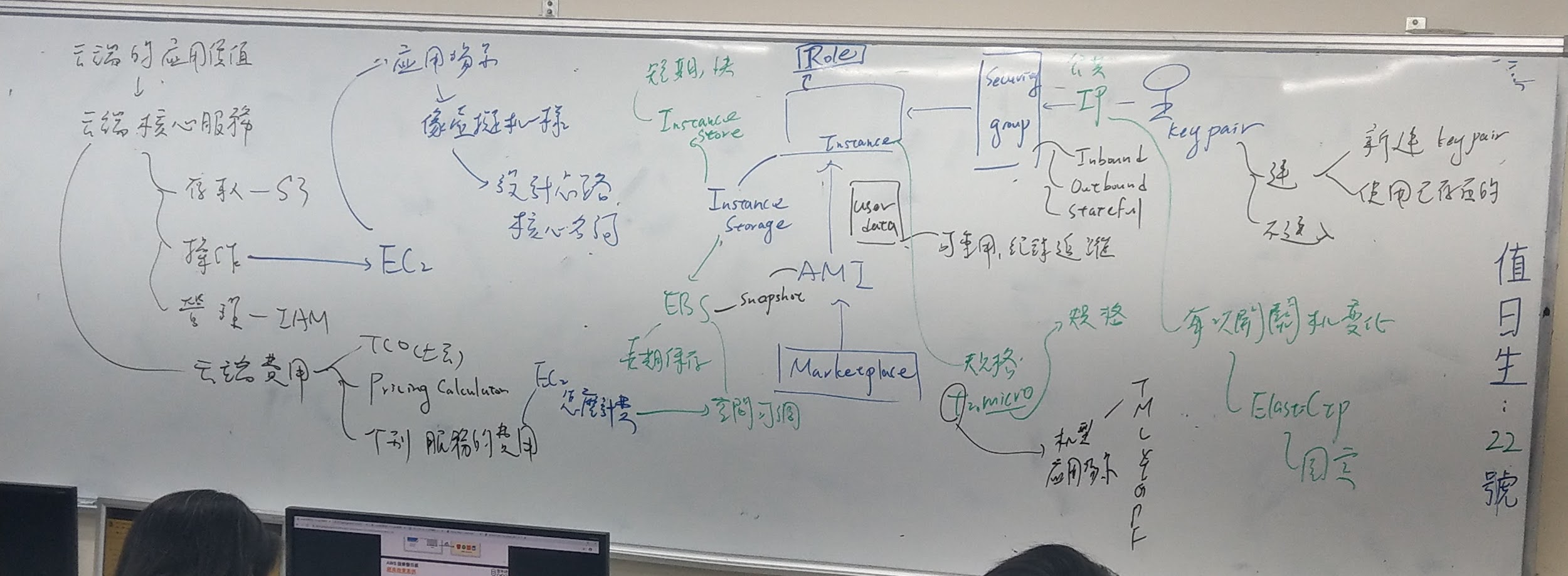
docker ps

刪除container

docker rm -f 前三位的代碼

docker run -d --name dosgame -p 262:262 oldiy/dosgame-web-docker:latest

<https://hackmd.io/qVKecNP0Q4iWQqy2jCwEew#day1-%E9%9B%B2%E7%AB%AF%E7%9A%84%E6%88%B0%E7%95%A5%E5%91%B3%E9%81%93-EC2%E6%93%8D%E4%BD%9C>



### EC2 價格

以VM 角度的思考點，有三種經典的付費方式

隨需-> 臨時項目用

保留-> 長期項目，平均折扣3-4折 -> 長期的可以轉賣到AWS二手平台

競標-> 容錯性項目，機器持續競飆 -> 若被別人標走，我們的機器就砍掉了，平均2折左右

### Security Group

核心名詞

Inbound

Outbound

Stateful

成功進去，直接出來

成功出來，直接進去

在AWS體系中，其實還有Stateless的防火牆 - NACL

但那在我們VPC的範疇裡

# 第二天

## 

## 帳號的權限管理方式

實作流程

Email 申請帳號，身分為root account

快速用root account 做掉一些安全措施

更改登入連結

增強root account 帳號密碼複雜度

(Optional) 申請軟體mfa登入

創建IAM User -> 雲端管理員帳號

允許 iam user 可以看帳單

Iam user 登入

設定billing與通知的時機。

教育帳號

## VPC

應用場景

每一個專案，都會有自己的網頁伺服器，應用伺服器，資料庫。

不會希望個別專案混來混去，互相可連線。

過去的本地解法

切網路環境

切多個網段

公網段與私網段

本地怎麼做，雲端怎麼做

雲端推出一個服務，VPC

去反應 本地網路環境的設計

設計思路

大樓 VPC

大門 Internet Gateway

房間 Subnet

樓層告示牌 Route table

保全系統 NACL

核心名詞

VPC

Internet Gateway

NACL

Subnet

Route table

實作流程

六大步驟

創建vpc

創建IGW

` ` 創建Public Subnet

創建Route table，更改Route table的規則

更改Public Subnet 參照的Route table

創建PrivateSubnet

兩大步驟，驗證

在公私網段各開一台機器

用瀏覽器連入機器做驗證

VPC Summary

VPN Connection

Peering Connection

VPC Endpoint

## IAM 初探

應用場景

管帳號的啦

設計思路

法條 -> 憲法-> 某國人民 -> 王曉明 -> 走路 -> 身分證

開車 -> 駕照

外國人訪問 -> 簽證 -> 守憲法

Policy -> Permission -> IAM Group -> IAM User -> Web(password)

-> program(credential)

外部帳號 -> Role -> 我方帳號

EC2 -> Role -> S3 (任兩個aws 服務 溝通，都要透過role)

Policy 重中之重

結構

## S3

應用場景

NAS -> 空間有限

儲存空間無限 -> 工作流程就變化

設計思路

Object

bucket

第一次實作

登入秉鴻的帳號

https://cxcxc-learning.signin.aws.amazon.com/console

S3 Console

Iii-tutorial-v2 桶子

上傳

用兩種方式下載

第二次實作 - 權限的概念

Qwiklab網站，辦帳號，介紹他的介面

到 Lab Introduction to S3

Lab1 - 創建桶子，上傳下載檔案

Lab2 - Block Public Access, ACL - 單物件存取

解除Block Public Access

` 而後對物件做make public

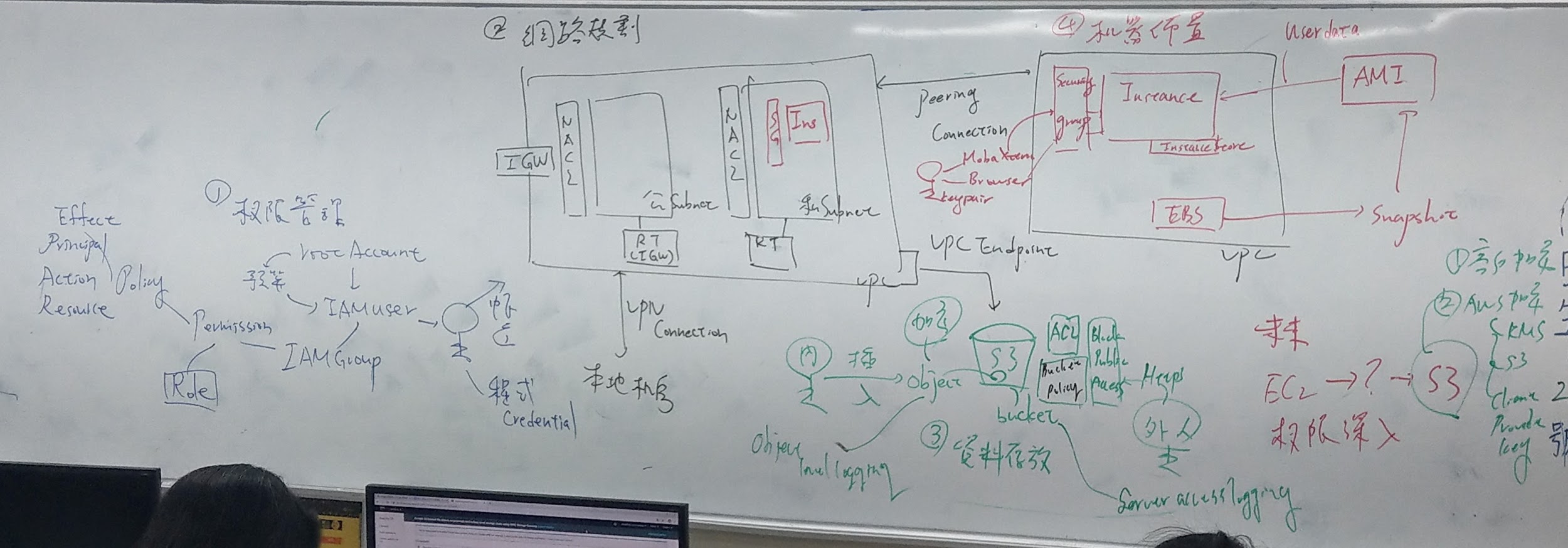
物件的https連結生效

再上傳一個檔案

Lab3 - Bucket Policy 多物件自動權限調整

Bucket Policy Example 複製下來改

Lab4 - Version



# 第三天

## AWS Cli 介紹與實作

### 應用情境

自動化方便

### 實作流程

#### 下載安裝aws cli

<https://aws.amazon.com/tw/cli/>

#### 打開命令列

#### 操作

aws --version

aws s3 ls

發現不能操作, unable to locate credential

重新跟大家回憶，IAM的結構

打指令或程式要靠credential

我們已經先準備好一套credential檔案給大家(IAM 進階會介紹如何生成)

<https://cxcxc-learning.signin.aws.amazon.com/console>

帳密:student

到tutorial-material桶子下載cred

在本機註冊aws cli

輸入aws configure 進行註冊

Access key id

Secret Access Key

Default Region

要輸入那種工程師喜歡的pattern啦

ap-northeast-1

網路上查得到

<https://docs.aws.amazon.com/general/latest/gr/rande.html>

Output

Text - 簡單展示

Json - 跟其他工程師或其他程師做交換

Table - 報表 格式工整

註冊完，即可使用

aws s3 ls

另外還發現 自己家電腦有個隱藏資料夾，裡面有credential

### aws cli 與s3互動的常用指令

瀏覽所有的桶子

aws cli list all s3 bucket

<https://docs.aws.amazon.com/cli/latest/reference/s3/ls.html>

aws s3 ls

瀏覽特定桶子

aws cli list specific s3 bucket

aws s3 ls s3://iii-tutorial-v2/

瀏覽特定桶子內的資料夾

aws s3 ls s3://iii-tutorial-v2/student99/

aws s3 ls s3://iii-tutorial-v2/student99

有斜線跟沒斜線是有差的

上傳物件到桶子內

aws cli upload object to s3 bucket

<https://docs.aws.amazon.com/cli/latest/reference/s3/cp.html>

aws s3 cp cxcxc.txt s3://iii-tutorial-v2/student**99**/

下載

aws s3 cp s3://iii-tutorial-v2/tstuden99/cxcxc.txt lbh.txt

刪除

aws s3 rm 物件位置

aws s3 rm s3://iii-tutorial-v2/student99/cxcxc.txt

## 綜合練習

流程

1.網頁設計人員->抓網頁 用aws console上傳 至 iii-tutorial-v2

<https://www.toutiao.com/a6886985974862217736/>

2.開一台EC2 Instance

機型挑選 /userdata

#!/bin/bash

yum install -y httpd

service httpd start

aws s3 cp s3://iii-tutorial-v2/student99/cxcxc.html /var/www/html/

3.userdata 1.安裝apache網頁伺服器

2.啟動網頁伺服器

3.使用aws cli 下載

4.設tag以利專案管理

5.設定security group 開inbound -80port

6.用瀏覽器連線

<http://18.179.31.93/cxcxc.html>

我們發現userdata沒有aws configure，卻可以從s3拿東西下來

原因就在於 IAM studentRole

秉鴻連進該台ec2，看一下內部環境

## IAM 深入

Review核心名詞

<https://tibame.signin.aws.amazon.com/console>

帳密

student

### 第一次實作

創建iam group

創建 iam user

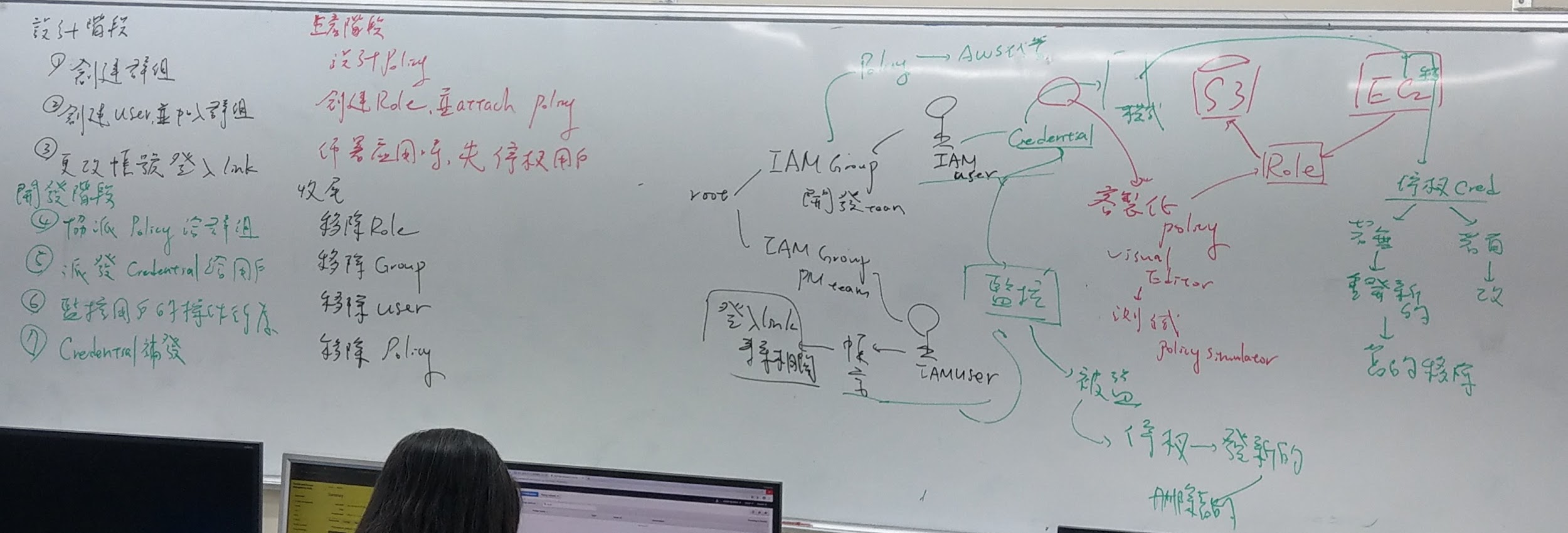
附加 iam policy 至群組內



刪除群組

刪除用戶

### 第二次實作



<https://jamstack.org/>

# 第四天

## Colab體驗

透過google 提供的colab服務，來理解正在發生的未來

並知曉這個未來帶來的改變。

## 第一代開發方法論

每個人用自己的筆電去寫程式

優點

離線也能開發

大家自主性強

缺點

大家的筆電環境 不一致，程式碼未必能共享

A同事的環境是PYTHON3.5

B同事的環境是PYTHON3.8

產品程式碼合併之後，會有不可預期的現象。

程式不能運行

根本性問題

環境不一致

## 第二代開發方法論 - 遠端集中規範的開發環境

大家透過瀏覽器連線到 遠端電腦，寫程式

優點

我們可以**有效的規範環境**，大家的程式碼能夠有效共用

產品程式碼合併之後，不會出現太多不可預期的現象

缺點

必須在有網路的情況下做使用

## 第二代開發方法論的底層實踐工具

### 用實體機/VM

不可行的原因

必須保留資源給工程師使用 -> 出資源的人不能接受，因為用戶太少了，賺不了錢

工程師可改底層環境 -> 第二代方法論的底層思路就破功了

### 用新的虛擬化方式-Docker

Colab

## Docker 體驗

### 流程

登入aws帳號

啟動EC2 Instance, 並用userdata 安裝docker

使用docker指令，啟動類似虛擬機的東西 - 容器

Python

Game

觀察容器的特性

快速啟動

不可變動

<https://cxcxc-learning.signin.aws.amazon.com/console>

帳密: student

**Userdata**

#!/bin/bash

yum install -y docker

usermod -aG docker ec2-user

service docker start

**Docker使用指令**

EC2內沒有 Python3，我們要如何執行python3的腳本

先連入ec2內

打python

輸入 print(‘hello’)

發現會被拒絕

檢視Container

docker ps

啟用Container

docker run -it --rm python:3 python

輸入print(‘hello’)

檢視Container

公司應用的快速跨環境部屬

在ec2上執行這句話

docker run -d --name dosgame -p 262:262 oldiy/dosgame-web-docker:latest

用瀏覽器訪問

在有安裝docker 的mac上執行這句話

docker run -d --name dosgame -p 262:262 oldiy/dosgame-web-docker:latest

用瀏覽器訪問

在有安裝docker 的windows上執行這句話

docker run -d --name dosgame -p 262:262 oldiy/dosgame-web-docker:latest

用瀏覽器訪問

在aws雲端裡面， 有安裝docker 的windows上執行這句話

docker run -d --name dosgame -p 262:262 oldiy/dosgame-web-docker:latest

用瀏覽器訪問

## 雲原生架構概念

## 資訊架構歷史演進

## Event-Driven Architecture

按摩店理論

V1 - 老闆請師傅 - 實體機

V2 - 老闆請少位師傅，不夠就對外聘，但要等 - 虛擬機

V3 - 產業聚落，老闆完全不請師傅，師傅們自行集中待命，要用瞬間啟動 - 容器時代

按摩店細部流程

消費者(有念頭) - 櫃台 - 師傅

網路封包(event) - 功能集散 - 運算Container

Event - Api gateway - Lambda

## Lambda

### 設計思路與核心名詞

消費者 > 櫃台 -> 師傅(雙手，按摩邏輯) -> 員工身分 -> 倉庫

Event -> Trigger -> Lambda (Runtime, function code ) -> role -> s3

### 實作流程

**Event的兩大種類 - 外來 或 內生**

先下載教材程式碼

Qwiklab - introduction to lambda

[introduction to lambda](https://www.qwiklabs.com/focuses/10541?catalog_rank=%7B%22rank%22%3A1%2C%22num_filters%22%3A0%2C%22has_search%22%3Atrue%7D&parent=catalog&search_id=7611028)

### 結論

未來的項目開發新趨勢

程式碼編寫, 部屬, 事後監控 三合一

## 

## API Gateway

### 概念

### 實作

[Introduction to api gateway](https://www.qwiklabs.com/focuses/10383?catalog_rank=%7B%22rank%22%3A1%2C%22num_filters%22%3A0%2C%22has_search%22%3Atrue%7D&parent=catalog&search_id=7611376)

創建第一個Lambda function => NodeJS12

創建API Gateway

訪問api gateway

創建第二個Lambda function => Python

串接api gateway

### 結論

微服務

多系統集成

<https://docs.aws.amazon.com/zh_tw/vm-import/latest/userguide/vmie_prereqs.html#prepare-vm-image>