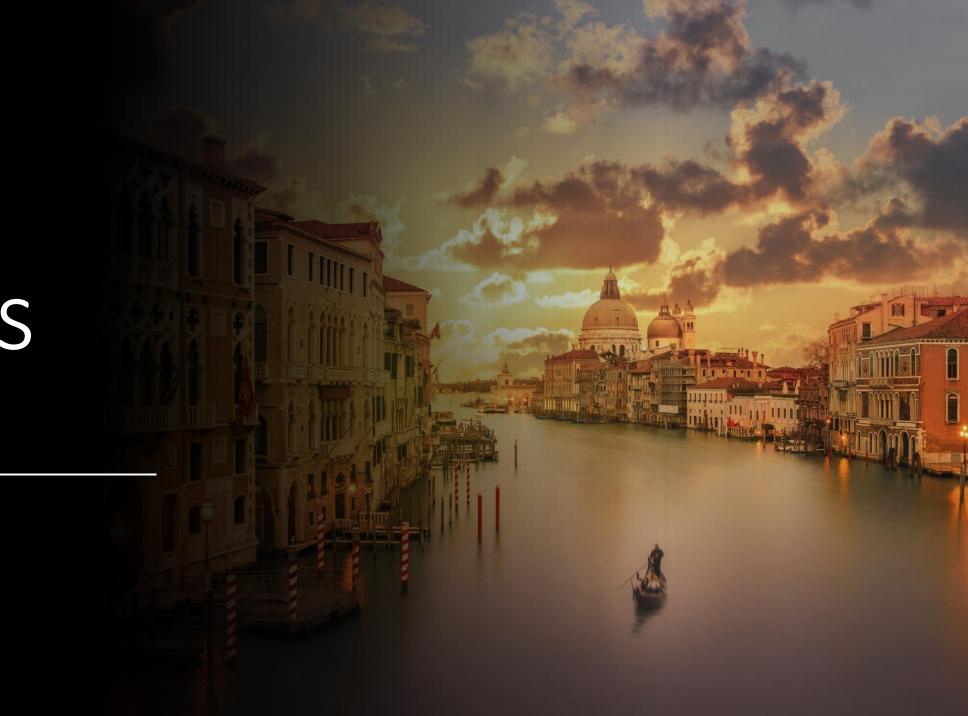


David Tung



About Me...

董大偉 David@isRock.com.tw

- 微軟技術社群區域總監(Regional Director)
- 微軟最有價值專家(AI + Developer MVP)
- LINE API Expert (LINE官方認證API專家)
- 資策會教育訓練中心特約講師、恆逸教育訓練中心特約講師
- 集英信誠資深顧問講師、TSMC敏捷(Agile/DevOps)顧問講師
- Agile Summit 2019, DevOpsDays Taipei 2017,2019 大會講師
- 歷屆台灣微軟TechED, TechDays, Azure Day, DevDay 大會講師
- China TechED大會講師
- Windows Phone / Silverlight / Visual Studio上市發表會巡迴講















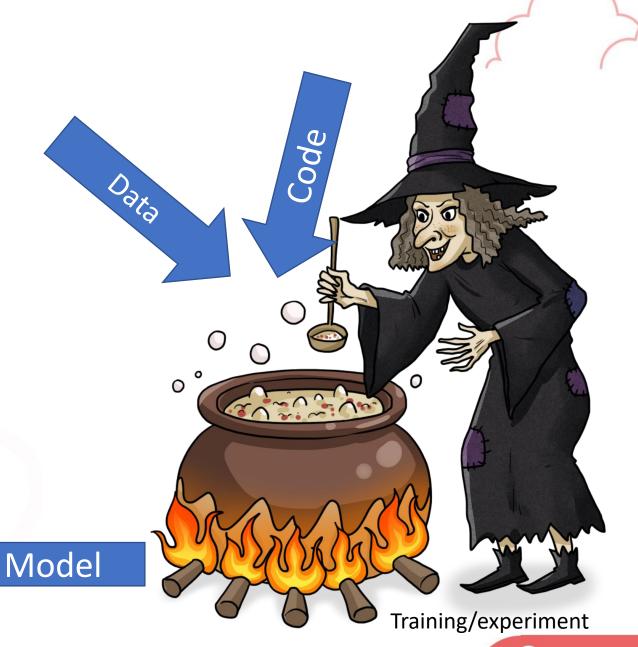


7,645.05 210.95 12,411.80 149.16 207.70 210.95 207.70



- 將DevOps概念延伸到Machine Learning
- 建立可重現(可重用)的服務產生流程
- 在ML生命週期中提供自動化監控與通知
- End-to-End ML生命週期自動化

- 將DevOps概念延伸到Machine Learning
- 建立可重現(可重用)的服務產生流程
- 在ML生命週期中提供自動化監控與通知
- End-to-End ML生命週期自動化

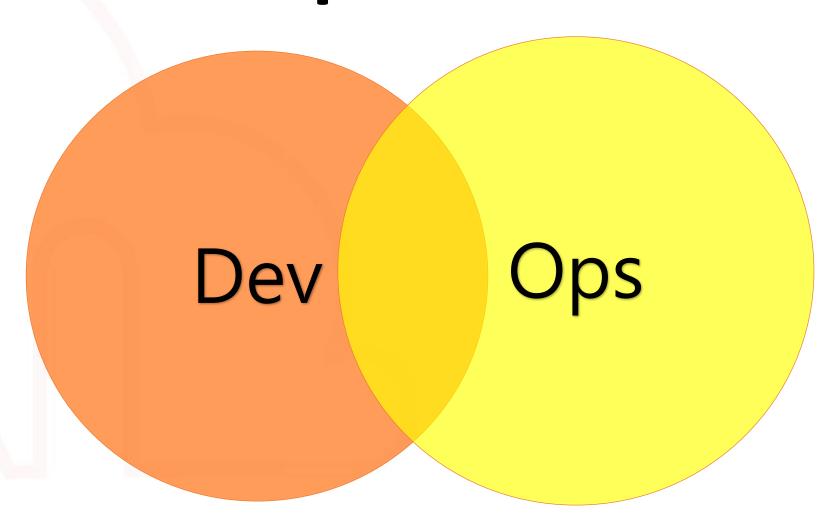




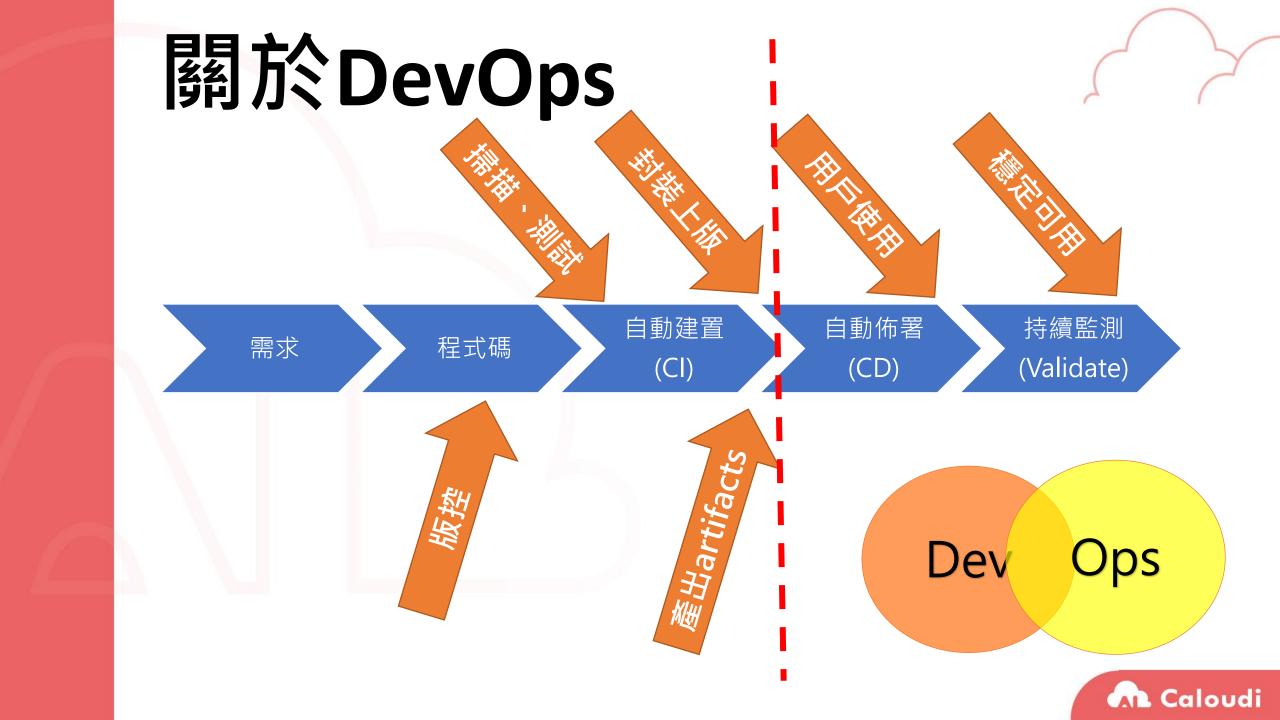
Services

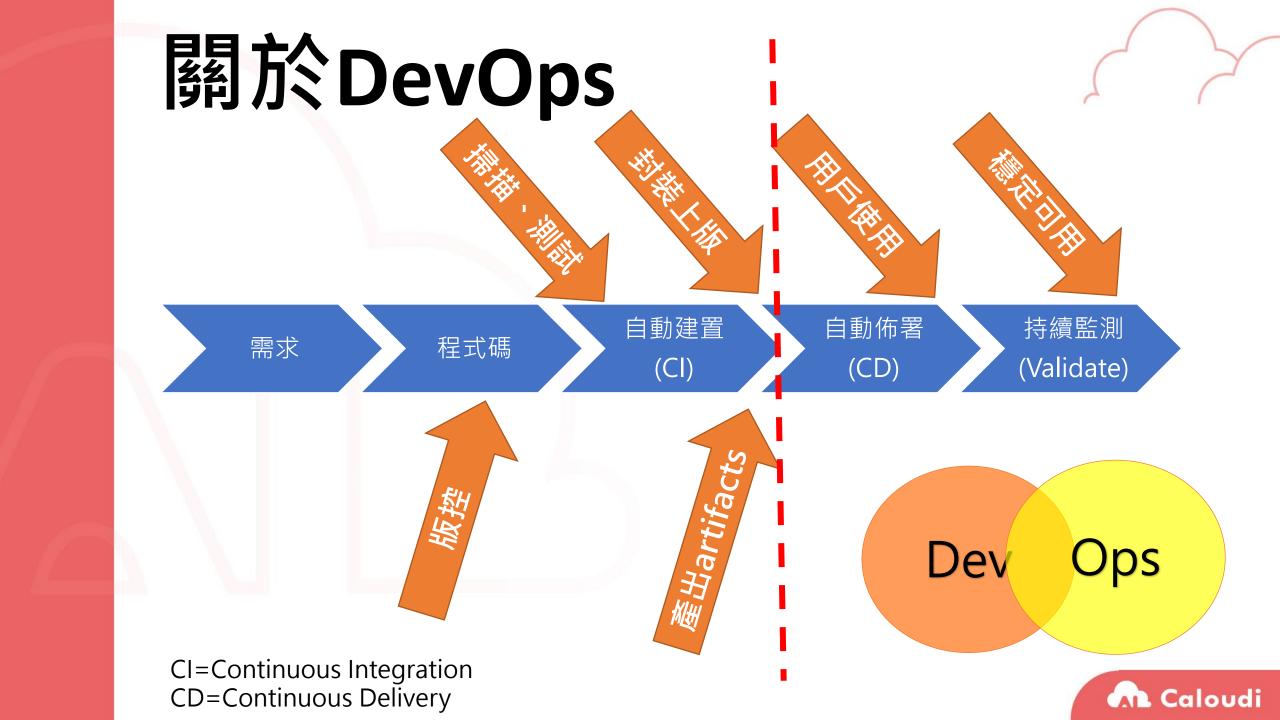




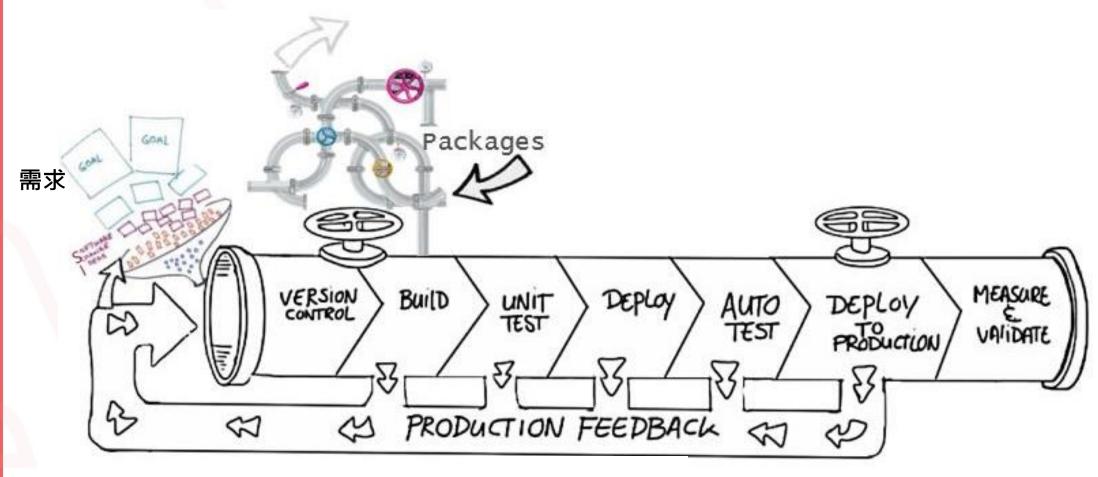






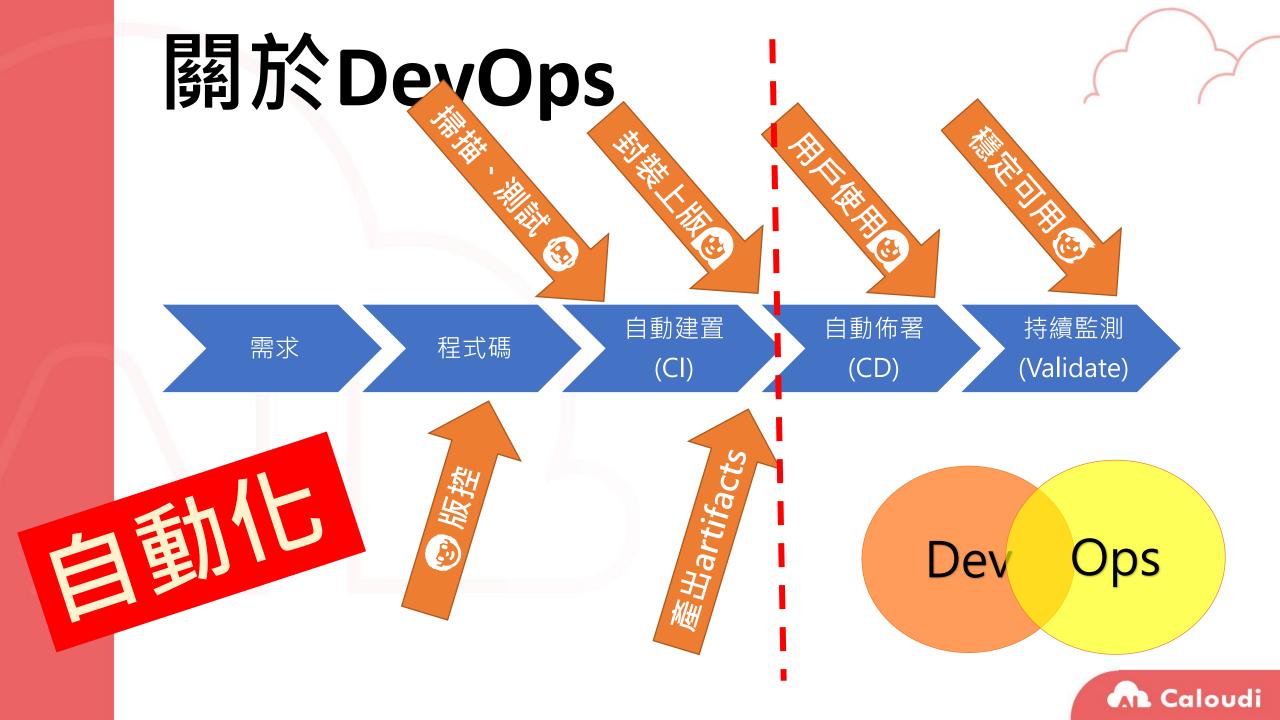








- 開發維運一體化
- 面對更短的生命週期、提供更快速的改良 (更新)迭代,以面對真實世界所需要的快速應變
- 在快速交付的同時間保有高品質、安全性



關於MLOps 版控 版控 持續監控 Services Model experiment



Caloudi

- 版控(Source Control)
 - 資料版本控管
 - 程式碼 training code 版本控管
 - 如有異動,重新自動訓練模型
- 自動化
 - 自動觸發實驗進行
 - 自動建構模型
 - 自動將模型發佈為正式服務(ACI、AKS)
- 持續監測(Continuous Monitoring)
 - 隨時偵測服務可用性
- ML Studio自動化
 - 自動建立ML Studio環境
 - 自動建力訓練用虛擬機



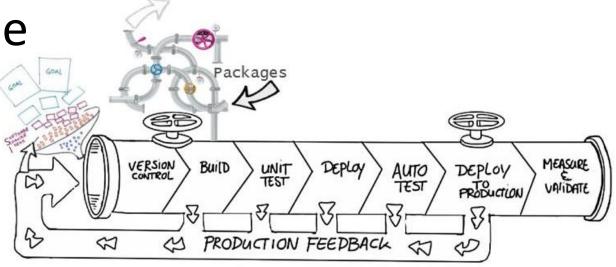


- 我們會使用到的工具
 - AZ CLI/Azure Machine Learning CLI
 - PowerShell/Bash
 - Python SDK(Azure Machine Learning SDK)
 - ML Studio
 - Azure DevOps
 - Azure Repos
 - Azure Pipeline
 - Azure Services
 - Storage/Blob
 - App Insight
 - Azure Function
 - AKS, ACI



Azure DevOps

Azure Pipeline



- Azure Repos
 - 可存放程式碼(為主)或資料
 - 採用Git版控機制



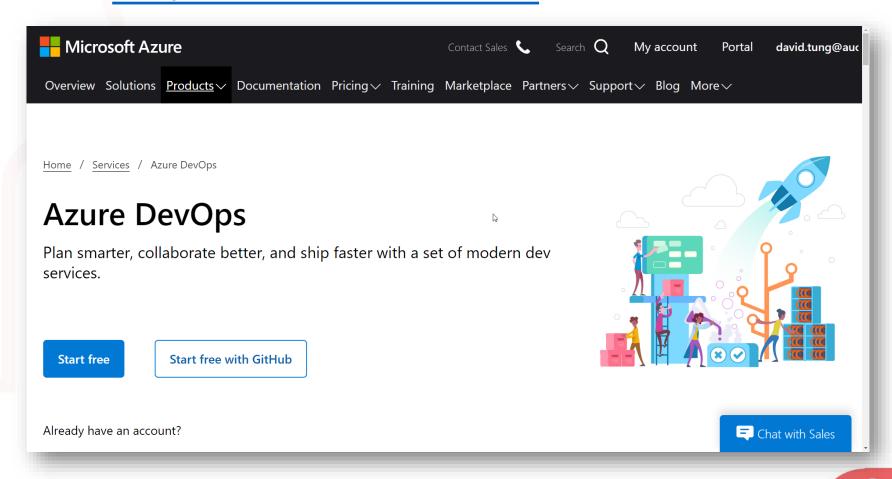


建立Azure DevOps站台



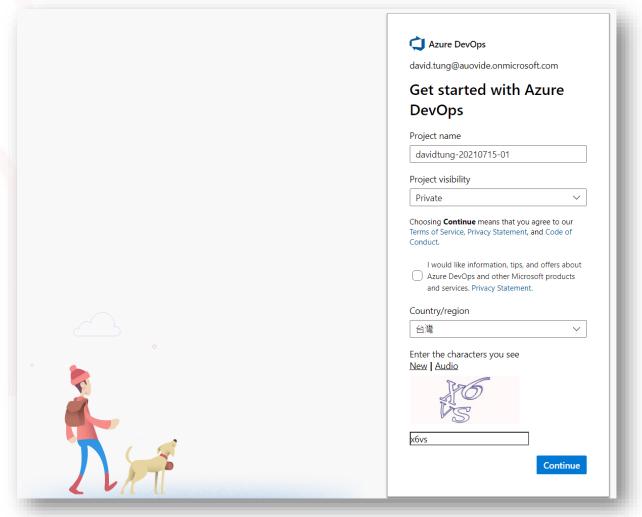
申請Azure DevOps

• 進入 http://Dev.Azure.Com



申請Azure DevOps

• 申請專案

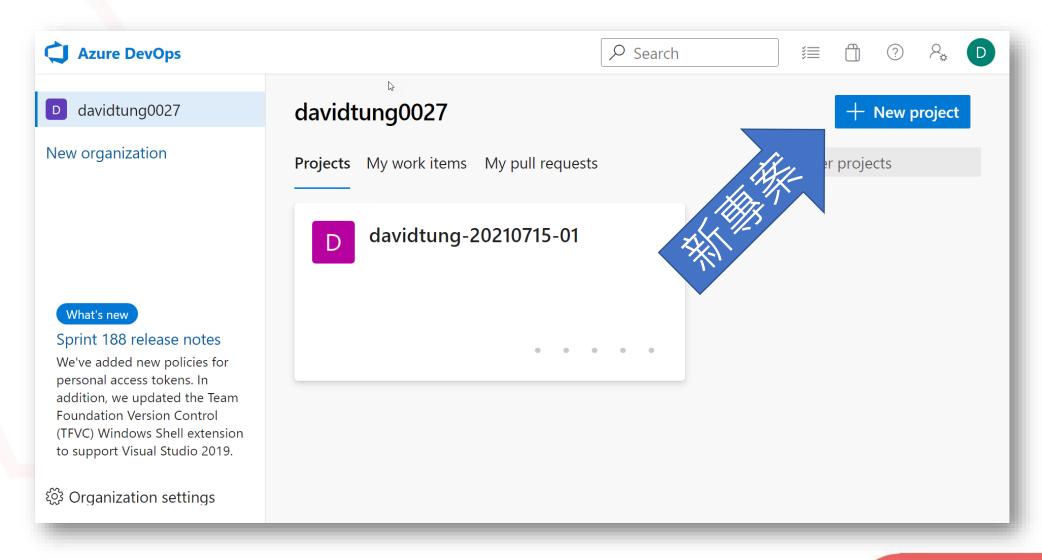






使用Azure DevOps



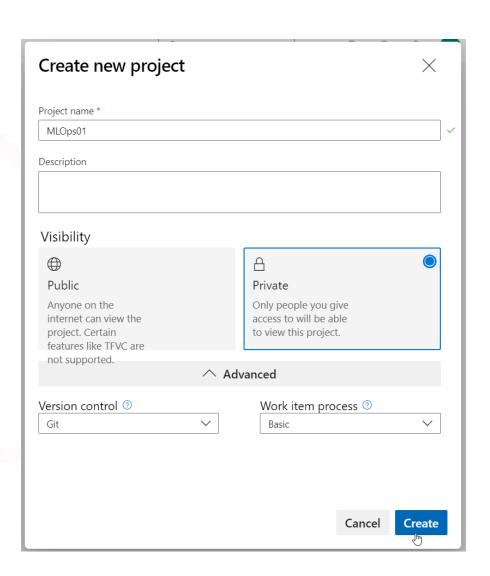


Azure DevOps專案結構



申請Azure DevOps

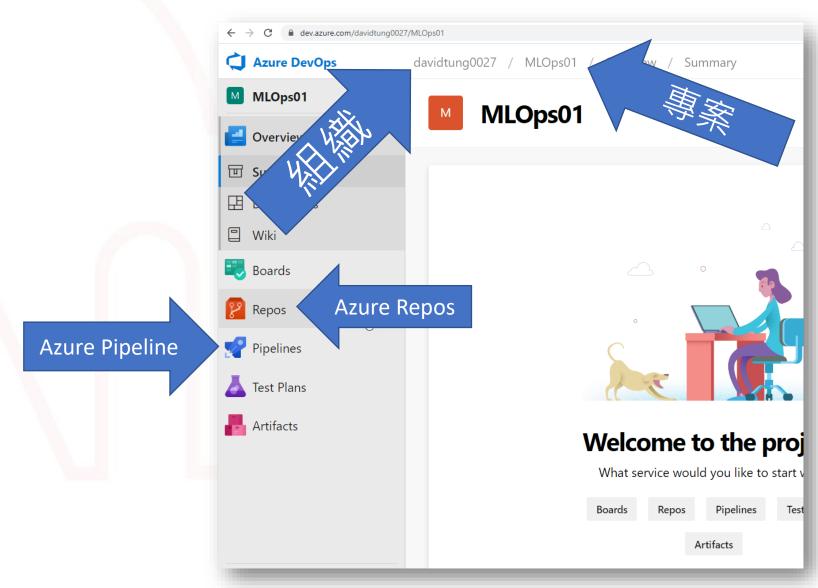
• 建立新專案







使用Azure DevOps





Hands-on:

建立Azure DevOps 組織

步驟:

- ①以學員帳號登入 portal.azure.com
- ②以同一個帳號登入 dev.azure.com
- ③第一次進入可申請組織





自動化流程體驗 (Web App)

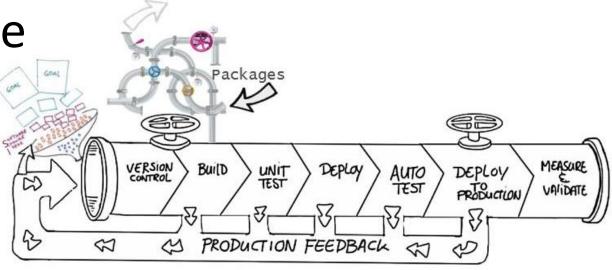


- 1. 關於DevOps
- 2. WebApp自動化生命流程 (ALM)
- 3. 維運與監控
- 4. 持續修改、自動佈署、持續監控



Azure DevOps

Azure Pipeline



- Azure Repos
 - 可存放程式碼(為主)或資料
 - 採用Git版控機制



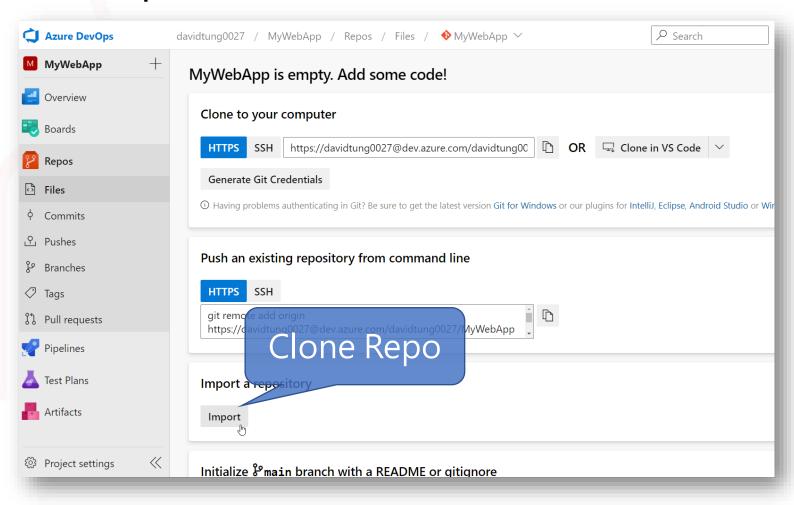




- Azure Repos
 - 可用來存放檔案,並支援git版本控管所有功能,包含Branch、PR、Code review、版本比較、修改歷史紀錄...etc.
 - 可匯入(Clone)任何git repo
 - 一個專案可建立多個repos
 - 可以在Web上直接修改(或建立)程式碼
 - 可以配合任何git client

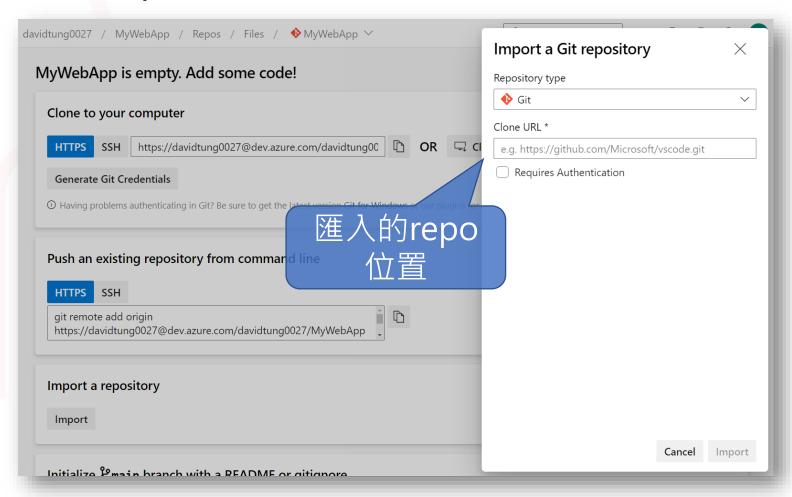


Azure repos

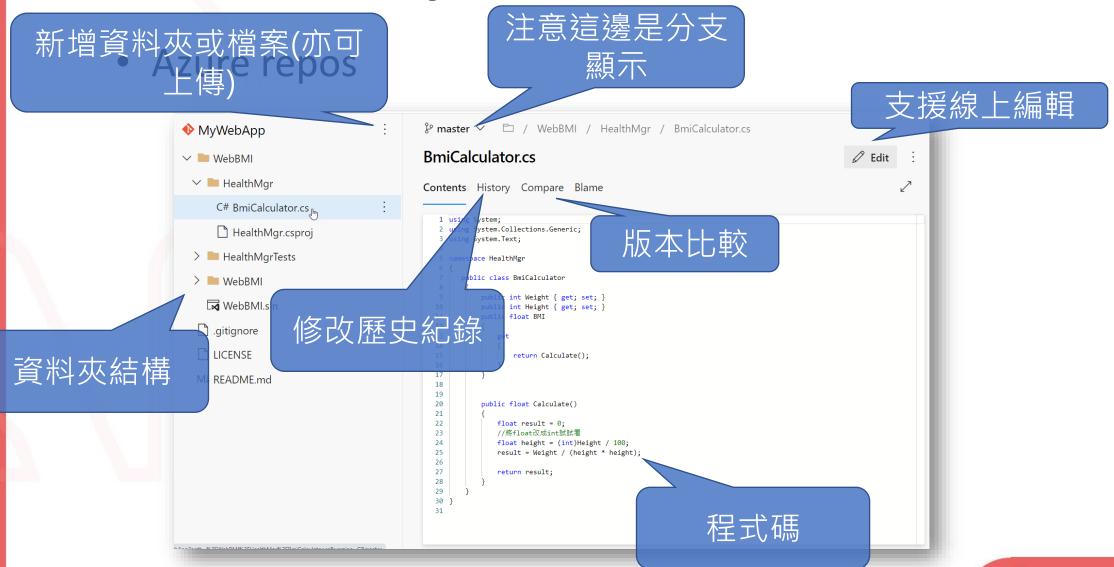




Azure repos









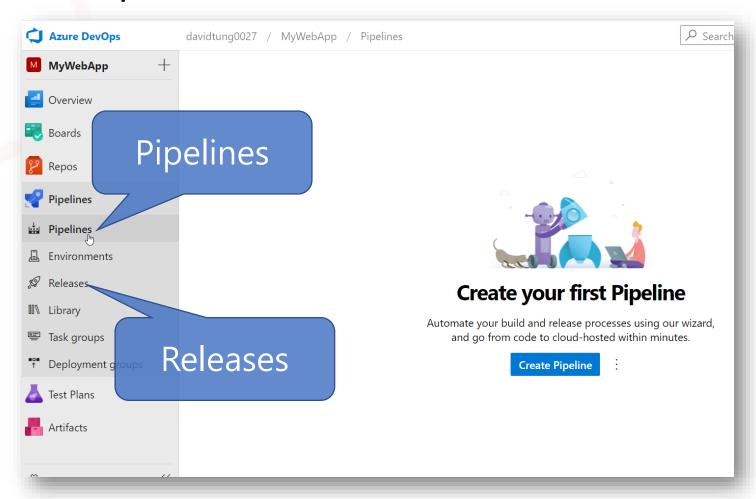
Azure DevOps Pipeline介紹



- 1. 名詞介紹
- 2. 運行位置
- 3. 觸發條件
- 4. 環境變數
- 5. Tasks

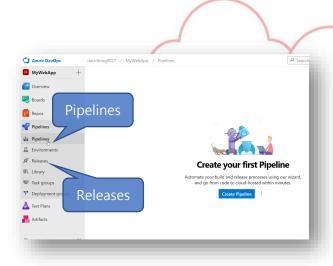


Azure Pipelines



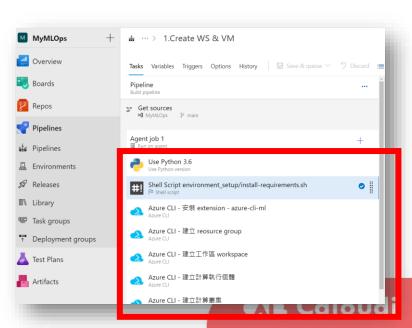


- Azure Pipelines
 - Pipelines
 - CI作業的自動化流程管線
 - 可被repos的異動所觸發
 - 可在管線中設計各種自動化流程 (程式碼建置、掃描、安全性偵測、測試...etc.)
 - 最終產出artifacts
 - Releases
 - CD作業的自動化流程管線
 - 一般用於佈署artifacts到各種環境 (DEV、QA、Staging、Production...etc.)
 - 支援環境(environment)建立與人員簽核(Approve)

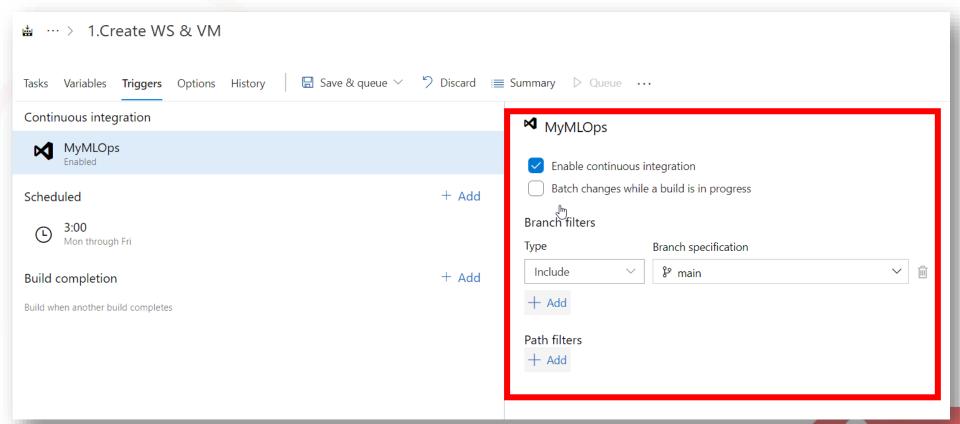




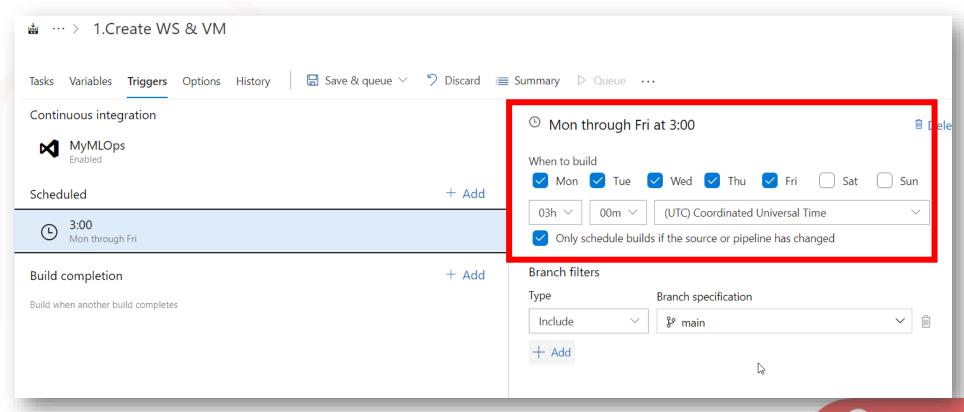
- Azure Pipelines
 - Tasks
 - Pipeline(Release)中的每一個作業
 - 有上千個內建功能套件 (Use Python, Bash, Docker, ...etc.)
 - 亦可由marketplace找到新套件



- Azure Pipelines
 - 觸發條件(CI)

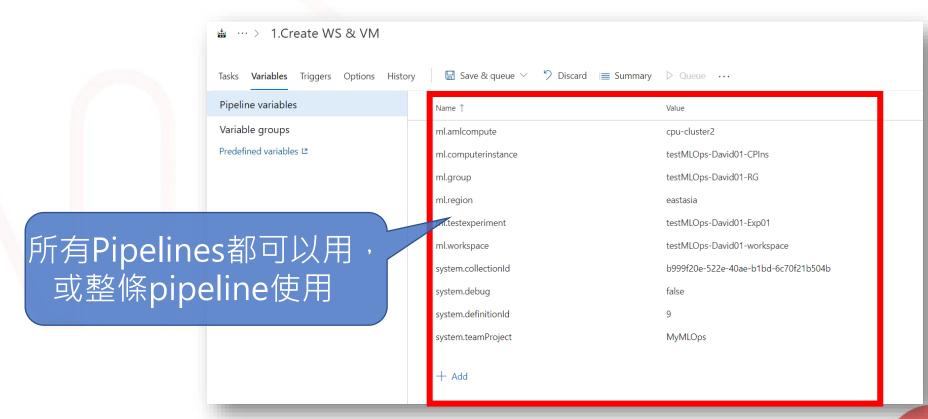


- Azure Pipelines
 - 觸發條件(Schedule)

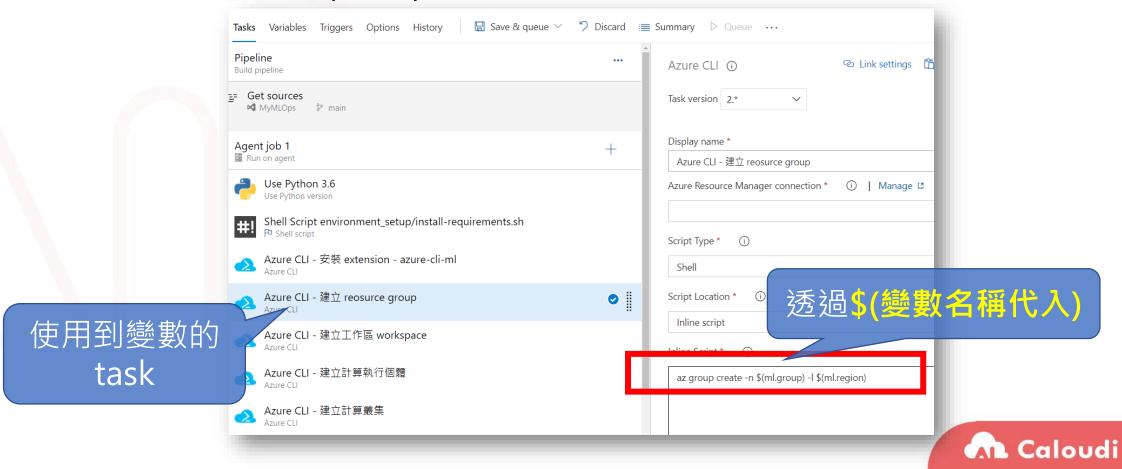




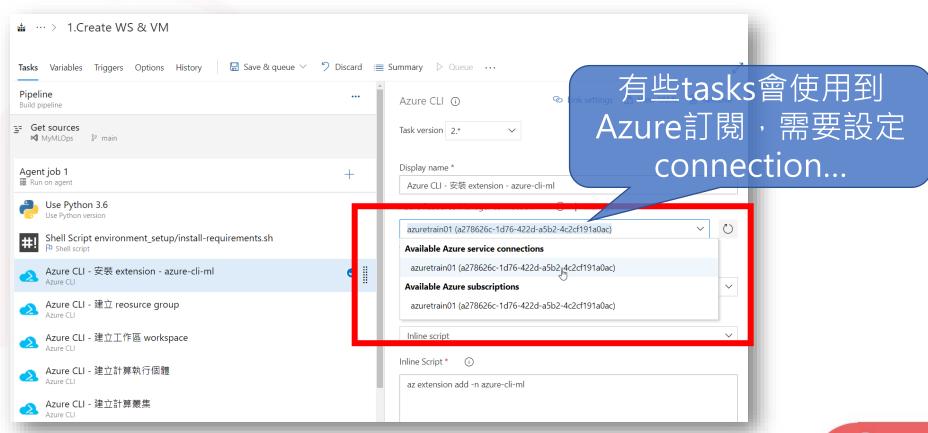
- Azure Pipelines
 - Variables(變數)



- Azure Pipelines
 - Variables(變數)



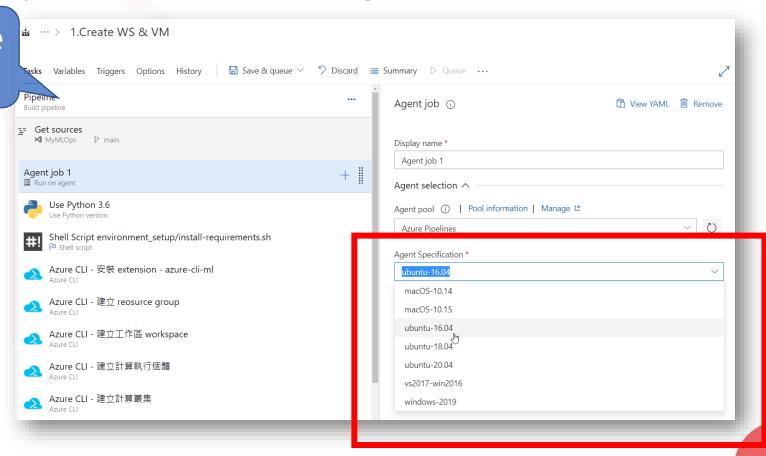
- Azure Pipelines
 - 使用到Azure 訂閱的 Task





- Azure Pipelines
 - Pipeline的運作環境 Agent

Agent 是具體pipeline 中tasks運行的所在

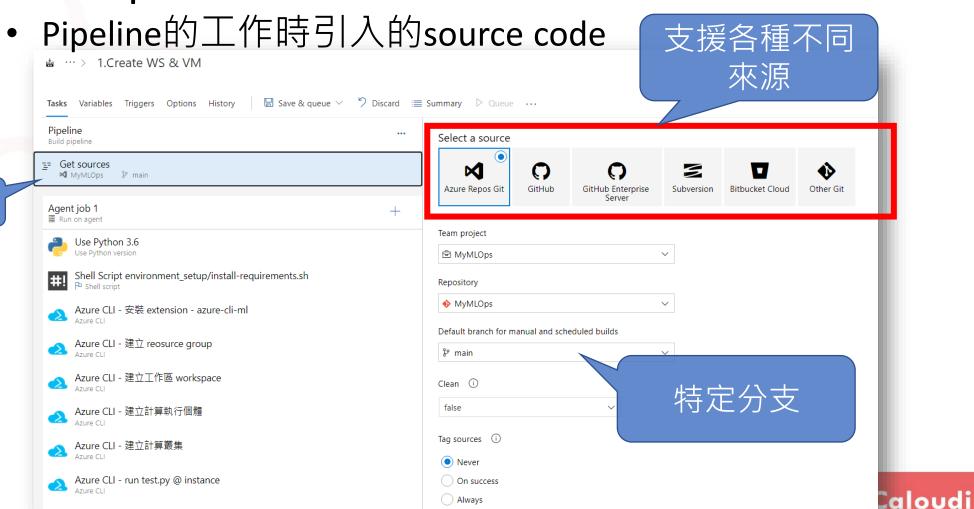






Azure Pipelines

這裡設定



Lab 0: WebApp網站自動化

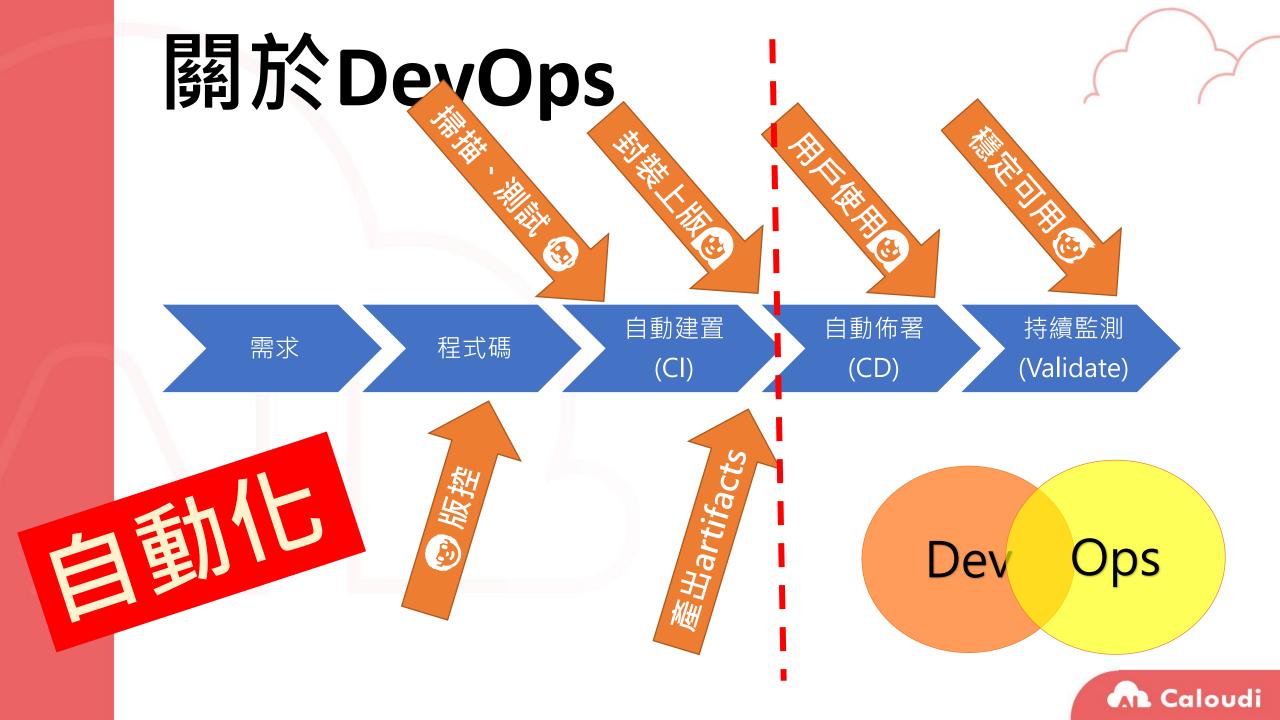


目標:建立一個自動化上版的網站佈署(CI)流程

- ① 建立MyWebApp專案
- ② Clone source code
- ③ 建立Pipeline,並關聯佈署網站
- ④ 設為CI,儲存並手動觸發(Queue)
- ⑤ 修改程式碼,是否自動觸發建置?
- ⑥ 是否上版?
- ⑦ 修改程式碼,是否自動上版?







關於MLOps 版控 版控 Services Model experiment





透過 AZ CLI 建立 ML Studio 環境



- 1. 關於AZ CLI
- 2. 可以運行的位置 Cloud Shell, Local PC, ML Studio VM, Build Agent...etc.
- 3. 指令結構
- 4. Extensions
- 5. 透過 AZ CLI 建立 ML Studio 環境



什麼是Azure CLI

• Azure CLI 是一種跨平臺命令列工具,可連接到 Azure 並在 Azure 資源上執行系統管理命令

• 可以在 Linux、Mac 或 Windows 電腦本機上安裝 Azure CLI。 您也可以透過 Azure Cloud Shell 在瀏覽 器中使用,或從 Docker 容器內執行。



什麼是Azure CLI



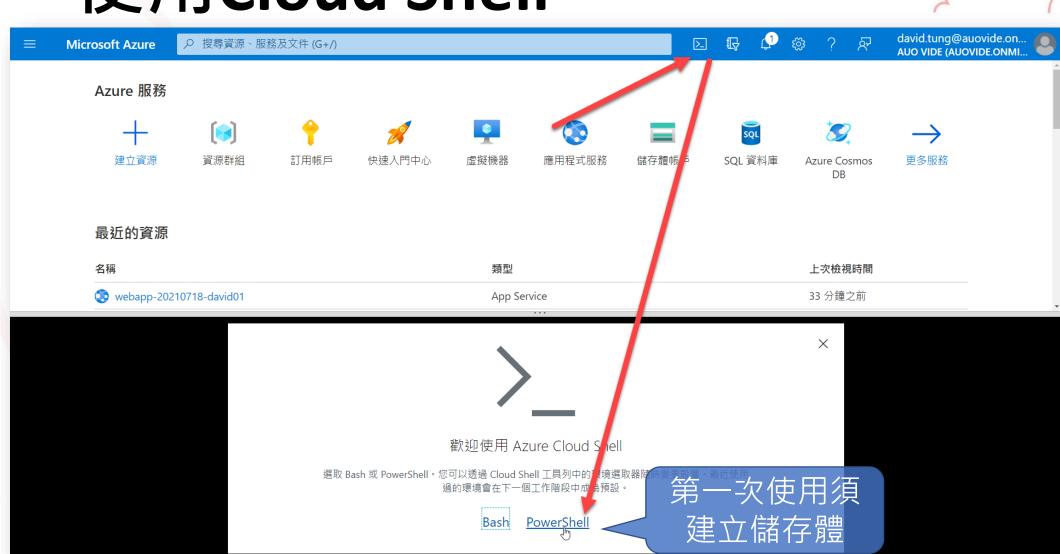
重要指令

- az --version
- az login (本地端才需要, Cloud shell不用)
- az group create -n 名稱 -l 區域
- az account show
- az account set -s "訂閱名稱"
- az extension add -n azure-cli-ml (安裝 ml extension)
- az group delete -n 名稱



什麼是Cloud Shell

- Azure上可運行command line指令的Web環境
- 支援powershell或bash
- 具體儲存於storage
- 可透過portal.azure.com直接使用
- 可透過shell.azure.com獨立使用



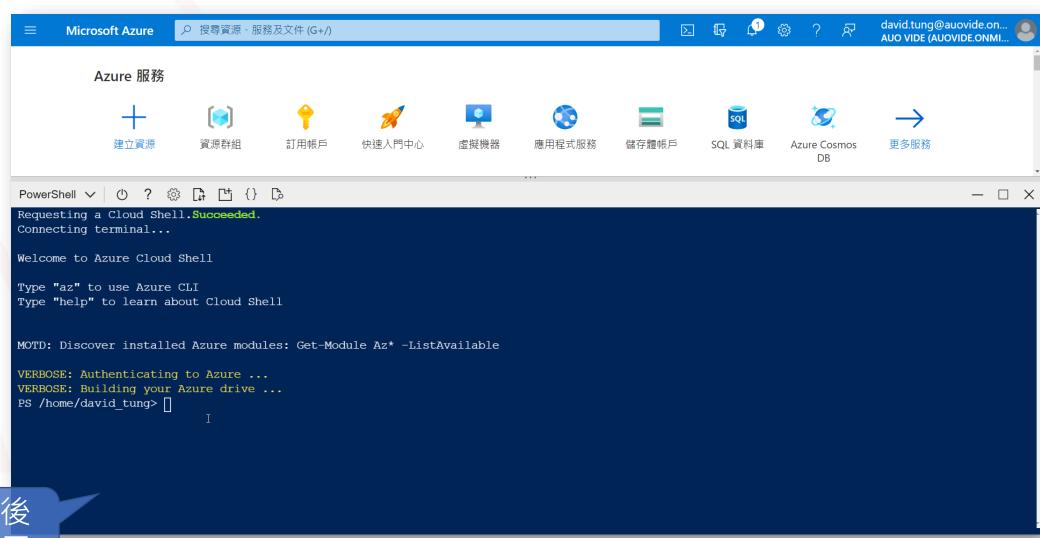












安裝好後 即可使用



```
剛進去試圖執行ml相
             少 ? ኞ ቩ 性 {} ಡ
                                        關指令會失敗
PowerShell V
PS /home/david tung> az ml
'ml' is misspelled or not recognized by the system.
TRY THIS:
az extension add --name anextension
Add extension by name
az extension list-available
List all publicly available extensions
https://docs.microsoft.com/en-US/cli/azure/extension#az extension add
Read more about the command in reference docs
PS /home/david tung>
```

使用Cloud Shell 請執行... az extension add -n azure-cli-ml

```
PowerShell ✓ () ? ☼ ြ [+ [+] {} [-]
PS /home/david tung> az extension add -n azure-cli-ml
PS /home/david tung> az group create -n test20210718david -l eastasia
  "id": "/subscriptions/a278626c-1d76-422d-a5b2-4c2cf191a0ac/resourceGroups/test20210718david",
  "location": "eastasia",
                          建立 resource group
  "managedBy": null,
  "name": "test20210718da
  "properties": {
                                                                                          建立 ml studio
    "provisioningState": "Succeeded"
                                                                                             workspace
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
PS /home/david tung> az ml workspace create -q test20210718david -w tesstdavidws001 -l eastasia --exist-ok --yes
Deploying StorageAccount with name tesstdaystorage50d74d709.
Deploying KeyVault with name tesstdavkeyvault1eede983.
Deploying AppInsights with name tesstdavinsightsffce71f6.
Deployed AppInsights with name tesstdavinsightsffce71f6. Took 5.1 seconds.
Deployed KeyVault with name tesstdavkeyvaultleede983. Took 19.88 seconds.
Deployed StorageAccount with name tesstdaystorage50d74d709. Took 24.62 seconds.
Deploying Workspace with name tesstdavidws001.
Deployed Workspace with name tesstdavidws001. Took 23.82 seconds.
  "allowPublicAccessWhenBehindVnet": false,
  "applicationInsights": "/subscriptions/a278626c-1d76-422d-a5b2-4c2cf191a0ac/resourcegroups/test20210718david/providers/m
ents/tesstdavinsightsffce71f6",
  "creationTime": "2021-07-18T12:39:21.6773873+00:00",
  "description": "",
```



Hands-on:

步驟:

- ①以學員帳號登入 portal.azure.com
- ②建立cloudshell環境(PowerShell)
- ③透過CLI指令建立 ms studio workspace

```
az extension add -n azure-cli-ml
```

```
az group create -n test20210718david -l eastasia
```

az ml workspace create -g test20210718david -w tesstdavidws001 -l eastasia --exist-ok --yes





Hands-on:

建立ms studi中的computer power

步驟:

```
az ml computetarget create computeinstance -g
test20210718david -w tesstdavidws001 -n cp001 -s
Standard_DS3_v2
```

```
az ml computetarget create amlcompute -g
test20210718david -w tesstdavidws001 -n
clusterpower -s Standard_DS3_v2 --min-nodes 0 --
max-nodes 2 --idle-seconds-before-scaledown 120
```



Hands on: 驗證

A. 可否成功建立所有資源?



B. 可否删除特定RG中資源?

```
PS /home/david_tung>
PS /home/david_tung> az group delete -n test20210718david

Are you sure you want to perform this operation? (y/n): y

- Running ..
```









在 Pipeline 中自動建立ML Studio環境

②使用PowerSell/Bash/AZ CLI Tasks之間的差別?

少使用AZ CLI預設會進行 AZ Login, 運行在Build Agent

了PowerShell或Bash不會進行AZ Login,一樣運行在Build Agent

了PowerShell或Bash都可以運行在Linux環境的Agent

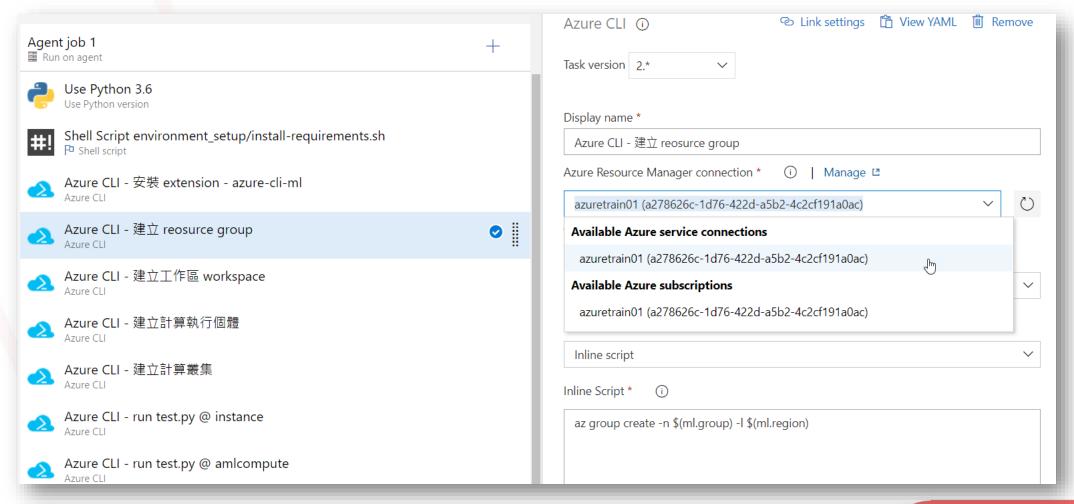
了Bash不能運行在Windows環境的Agent





在pipeline中如何使用AZ CLI task

設定訂閱



透過AZ CLI將Python送上ML Studio執行

```
az ml run submit-script
  --resource-group $(ml.group)
  --workspace-name $(ml.workspace)
  --target $(ml.computerinstance)
  -e $(ml.testexperiment) test.py
```

環境上執行

具體要執行的 python



Lab 1:

目標:



透過Azure Pipeline自動化建立ML Studio環境

問題:

- 1. 為何每次都要安裝 extension?
- 2. 為何需要使用 python 3.6 task
- 3. Python code具體運行的位置在哪?
- 4. 一樣是運行cmd line, 為何使用 az CLI task而非 bash/powershell?
- 5. 那 task 2那個bash又是幹什麼的?
- 6. 使用Pipeline來做這些自動化有何好處?









透過pipeline自動化進行Experiment

建立好workspace了,如何透過pipelin自動化 運行一個實驗呢?

我們需要

- 1.對程式碼能夠版控
- 2.能把training code送上ML Studio運行
- 3.要能指定運行的vm(computer power) ELE: Python SD
- 4.要能夠在ML studio中看到結果
- 5.誰能做到這些事情?



關於Python SDK

全名是: Azure Machine Learning SDK for Python可以用來:

- 以Python語言操作ML Studio
- 操作工作區、存取資料集、運行實驗、產生模型、發佈模型成為服務...etc.

那AZ CLI可以 做這些嗎?





透過Python SDK將實驗(Python Code)送上ML Studio執行

```
# get-started/run-hello.py
from azureml.core import Workspace, Experiment, Environment, Scr
iptRunConfig
                               從config.json設定
                                檔來決定ws環境
ws = Workspace.from config()
experiment = Experiment(workspace=ws, name='day 會被視為相依
experiment-hello')
config = ScriptRunConfig(source_directory='./src', sc
ript='hello.py', compute target='cpu-cluster2')
```

具體要執行的 python

```
run = experiment.submit(config)
aml_url = run.get_portal_url()
print(aml_url)
```

具體在哪一個 環境上執行



Lab 2:

目標:



透過pipeline自動化進行Experiment

問題:

- 1. Run-pytorch.py運行在哪裡?
- 2. train.py運行在哪裡?
- 3. 為何要透過powershell task來建立 config.json檔案?
- 4. 具體是哪一個指令將python檔案打包送上去ML studio執行的?





關於Auto-ML



在Build Agent上執行Python Code(Python SDK)

看起來,Python SDK很好用,那我可以...

- 1. 在Local使用Python SDK嗎?
- 2. 聽說有個厲害的東西叫做AutoML?

透過 Python SDK 執行AutoML

#set-up Experiment Name experiment = Experiment(workspace=ws, name="TestAutoMLExp")

...略...

#local_run
local_run = experiment.submit(automl_config, show_output=True)

Submit這個 實驗

這啥?在哪



透過 Python SDK 執行AutoML

```
#剛才被略掉的code
#data source(from github)
                                                                        訓練資料
data = "https://raw.githubusercontent.com/MicrosoftLearning/mslearn-
dp100/main/data/diabetes.csv"
dataset = Dataset.Tabular.from delimited files(data)
training data, validation data = dataset.random split(percentage=0.7, seed=223)
label column name = "Diabetic"
automl settings = {
    "n cross validations": 3,
   "primary_metric": "AUC_weighted",
   "experiment timeout hours": 0.25, # This is a time limit for testing purposes, remove
it for real use cases, this will drastically limit ability to find the best model possible
   "verbosity": logging.INFO,
   "enable stack ensemble": False,
automl_config = AutoMLConfig(
   task="classification",
   debug log="automl errors.log",
   training_data=training_data,
   label column name=label column name,
   **automl settings,
```

AutoML 參數

AutoML 整個設定



透過 Python SDK 執行AutoML

```
#get best_model
                                                   取得
best_run, best_model = local_run.get_output()
                                               best model
#show model name
model_name = best_run.properties['model_name']
print("model_name:" + model_name)
description = 'AutoML forecast example'
tags = None
#register model
model = local_run.register_model(model_name =
model_name, description = description, tags = tags)
                                在ML Studio
#show model id for deploy
                                 上註冊模型
print(model.id)
```



Lab 3:

目標:

透過 Python SDK 執行AutoML

問題:

- 1. 具體AutoML運算,是執行在哪?
- 2. 訓練與驗證的資料是從哪來的?
- 3. 訓練好的模型被註冊了,是註冊到哪去?
- 4. 我們可以對註冊好的模型繼續做什麼?







對了,那資料呢???



Data異動時,自動觸發下一回合訓練

現在我知道了...

- 1. 可以透過AZ CLI操作ML Studio
- 2. 也可以透過Python SDK來操作ML Studio
- 3. 更可以利用 Azure DevOps Pipeline把所有動作 串起來自動化, 這樣...
 - 1. 每當程式碼有改變,就自動觸發訓練

那資料改變了 怎麼辦???





Lab 4:

目標:

Data異動時,自動觸發Pipeline進行下一回合訓練

問題:

1. 具體是誰去執行Azure DevOps pipeline的?



從Model 到 Web Services





從 Model 到 Web Services

空有Model只能在實驗室裡面用...

如果要對外提供服務

- 1. 我們得把模型包裝成便於呼叫使用
- 2. 由於時常需要更新,最好打包容易
- 3. 過程當然一切要自動化
- 4 應該要採用外部用戶最方便的介面

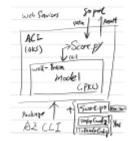
該用哪些 技術呢?



從 Model 到 Web Services

- Web Services 是執行階段 Model的運行實體
- / 以ACI或AKS來實作
- 了可透過GUI(ML Studio), AZ CLI, Python SDK建立端點建立時...

 - 字 需要DeploymentConfig.yml檔案作為設定
 - ② 需要inferenceConfig.yml檔案作為設定
 - ② 需要scoring-env.yml檔案作為score.py運行環境設定





Lab 5:

目標:

Lab 5: 發佈已註冊的model成 為Web Services

1. 接續Lab 4
2. 在 ML Studio中 找到已經 計冊的 model
3. 摸寫 Pipeline「4. deploy AutoML Model 2 service a

Data異動時,自動觸發Pipeline進行下一回合訓練問題:

- 1. 可以發佈成ACI或AKS,兩者有何不同?
- 2. Score.py檔的作用是什麼?

關於MLOps 版控 版控 持續監控 Services Model experiment



Caloudi

關於MLOps



