Azure DevOps Service Pipeline設計與非正常應用

講者：Edward Kuo



twMVC

歡迎參加我們的每週四固定聚會

About Me

- Edward Kuo
  - Microsoft Regional Director
  - Microsoft Azure MVP
  - Kingston IT Manager
  - DevOps 研討會 講師
  - 企業 DevOps 講師
  - 微軟技術社群 講師



https://profile.edwardkuo.dev/

https://www.facebook.com/EdwardKuo.Teched

我們就是用
Azure DevOps Service

為什麼~

- 時間要花在創造(完成)商業價值(需求)，降低整合異質平台的不便性
- 從需求、開發到發佈，採用同一個工作平台
- 技術的不斷推進、版本不斷更新，平台要能自動升級和擴充
- Server版還是會有維護成本，Service版則能降低平台維護成本
- Cloud DevOps Platform將會是一個趨勢

Azure DevOps 版本

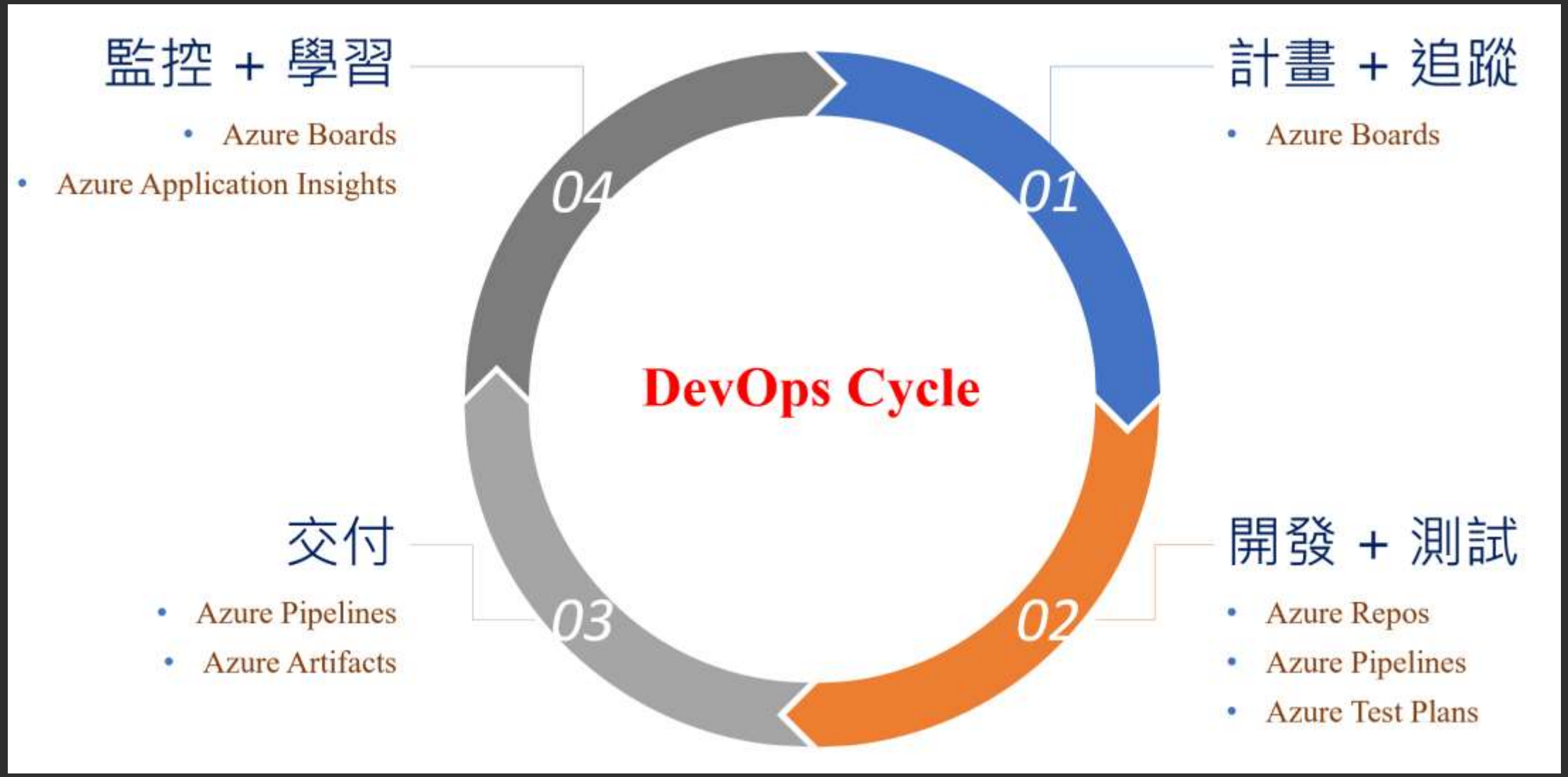
- Azure DevOps Server是基於SQL Server的地端產品
- Azure DevOps Service是基於Azure的SaaS級的雲端服務平台
- Service無法透過SSRS產生DevOps數據報表
- 兩者的迭代速度不同

SKILLTREE

Architect On Azure DevOps

Our Azure DevOps Ecosystem

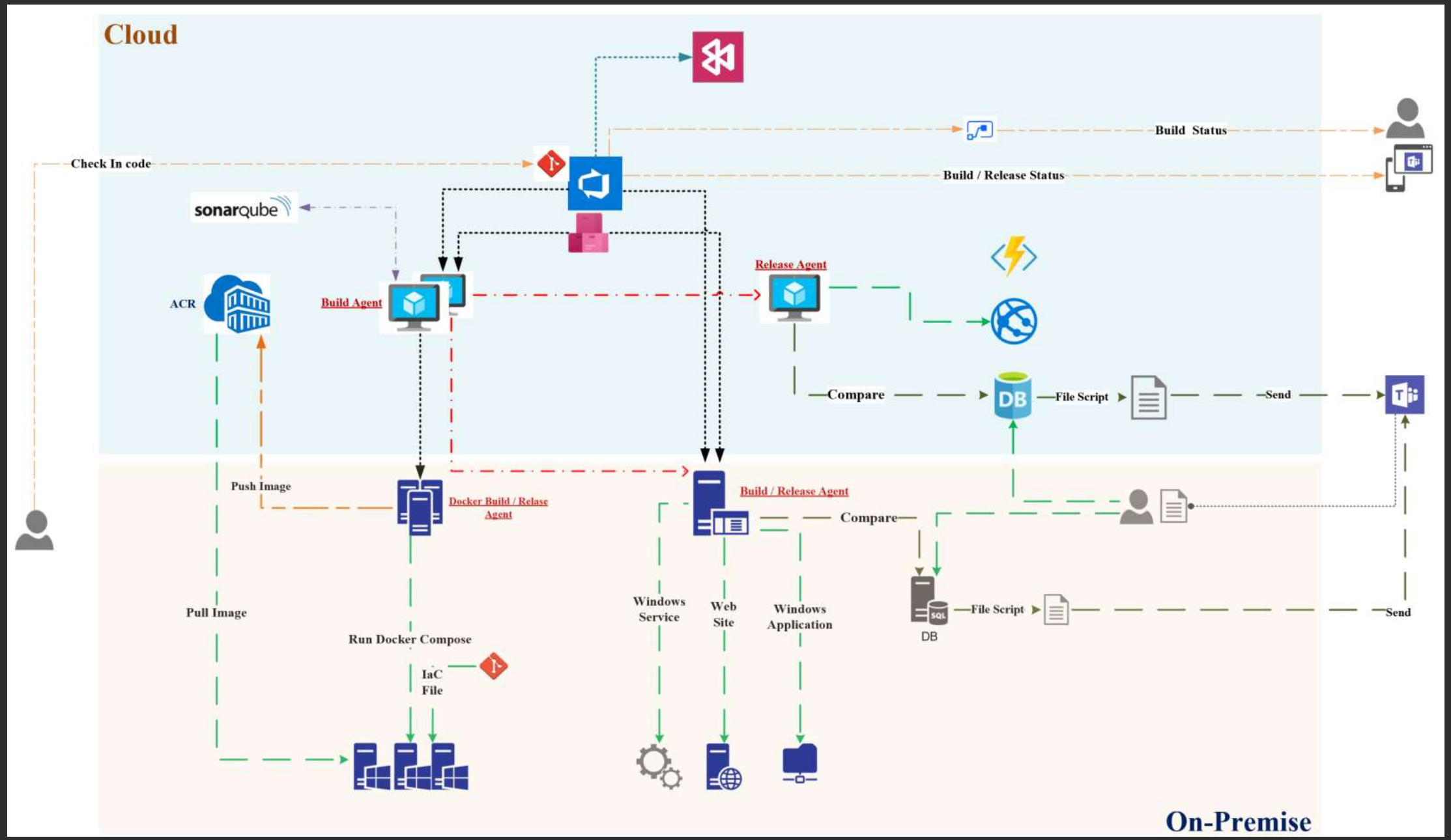
DevOps對應Azure DevOps服務應用



我們的應用場景

- 80%系統是佈署在企業內部
- 20%系統是佈署在Azure
- 整體需要兼顧地與雲的混合應用模式，且還是→製造業

Azure DevOps 我們的生態架構

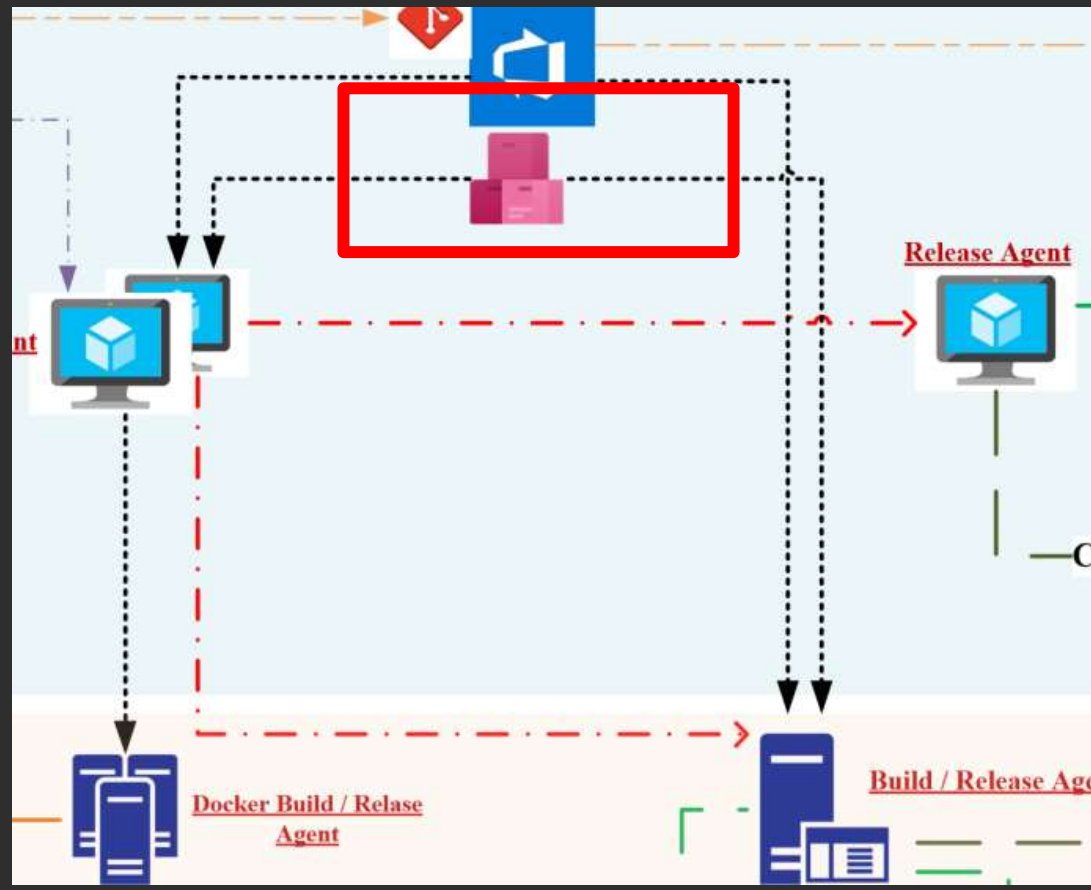


DevOps進化-解耦系統

- 系統拆解
  - 拆解系統讓內部耦合性能最小，增加重複使用與被測試的可能性
- 降低CI時間
  - 系統程式碼越多，每次編譯需要的時間就會越長
- 元件化
  - 在分散式開發與快速迭代下，避免影響他人且加速持續整合和釐清問題

Artifacts Service

- Artifacts Service管理編譯組件
- 建立團隊/企業的Feed
- 第三方元件的.nupkg檔案
- CI/CD建立.nupkg、.NPM元件
- 避免使用過期元件
- 強制設定Package保留週期



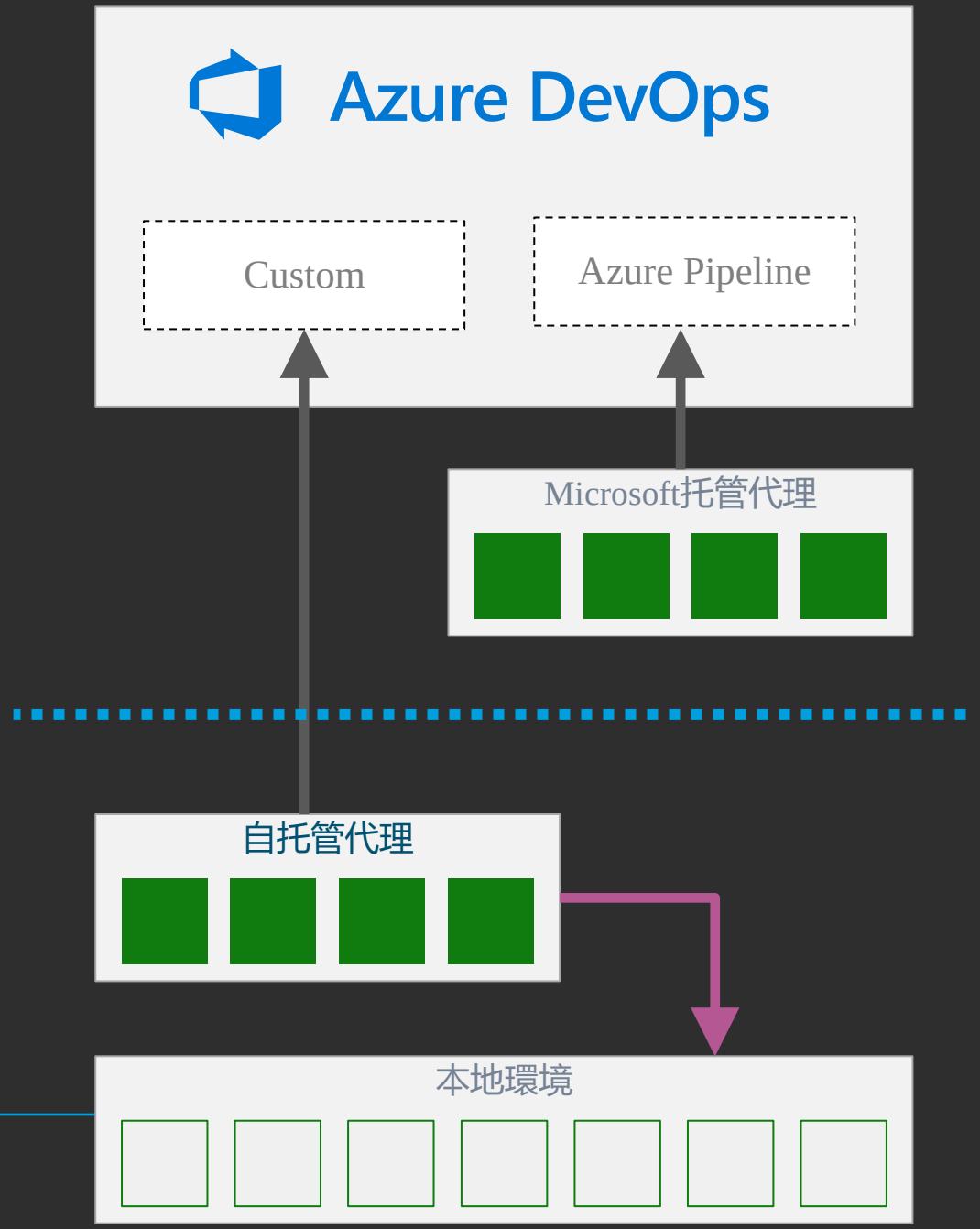
SKILLTREE

Agent Pool Practice

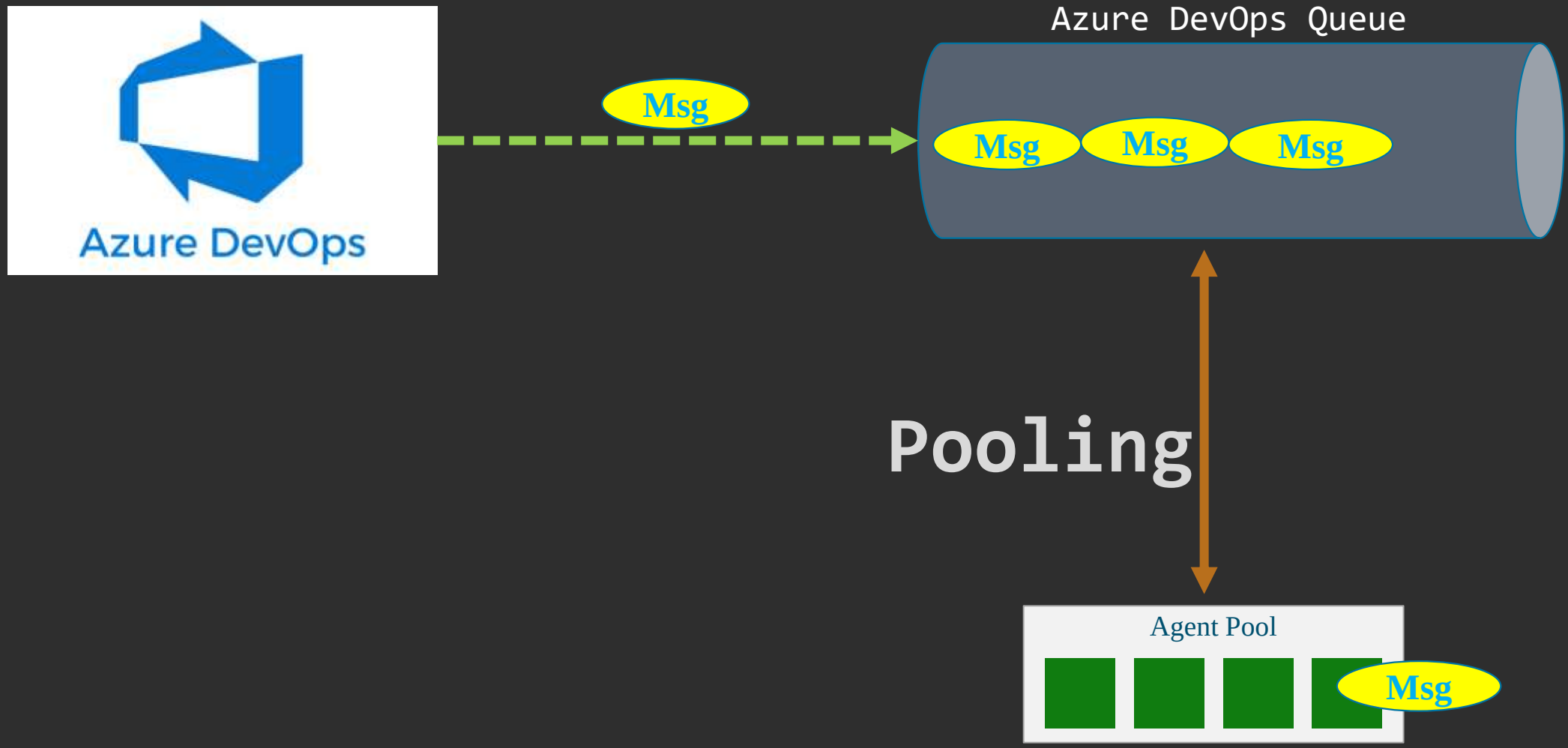
Azure DevOps Agent Design

Agent Pool

- Microsoft Agent Pool
  - Azure Pipeline
  - Hosted macOS
  - Hosted Windows Container
- Custom Agent Pool



Agent & Azure DevOps Service 關係

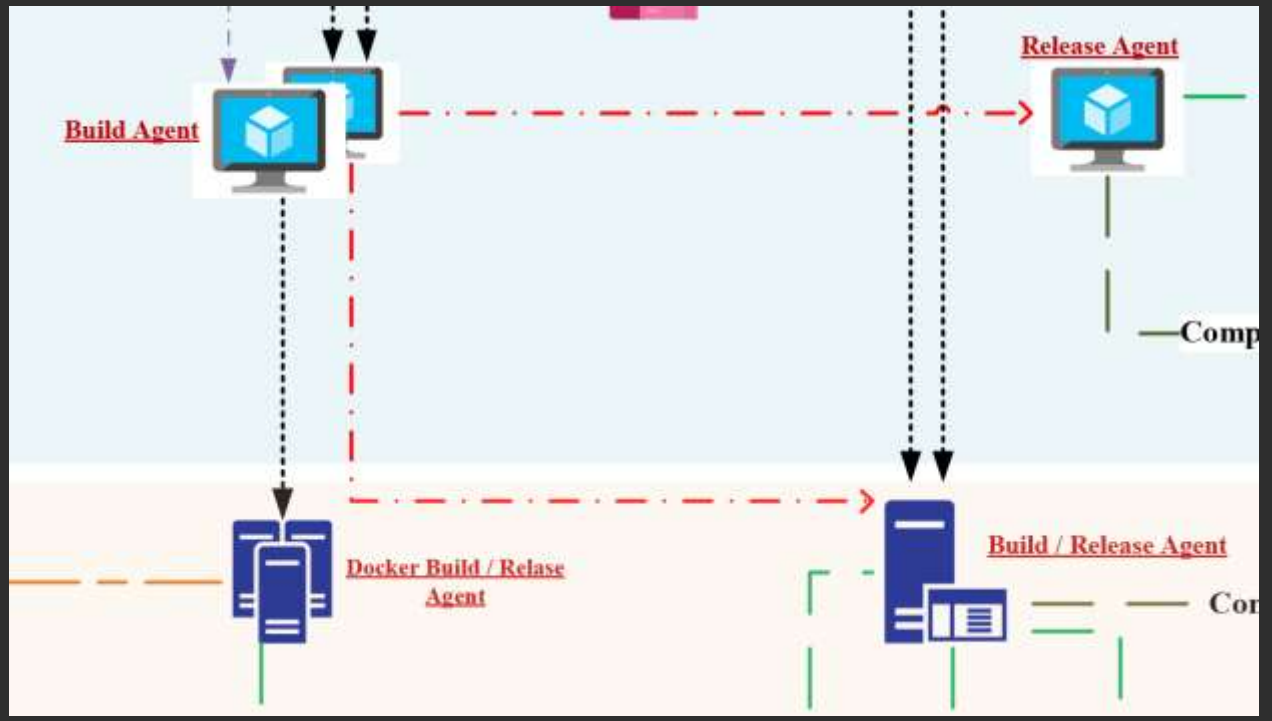


Agent Pool & Agent 關係

- Agent Pool
  - Agent
    - Host
- Agent Pool可以匯集各種不同地的Agent
- 一個Host可以掛載多個Agent
- Pipeline能指定Pool，但不能指定運作的Agent
- Agents被啟動”基本”是隨機

我們的Agent Pool

- 我們區分這些Agent Pool
  - Azure Pipeline
  - Private Cloud Build Pool
  - Private Cloud Release Pool
  - Private Premise Build Pool
  - Private Premise Release Pool
  - Private Premise Docker 1803 Pool
  - Private Premise Docker 2004 Pool



使用Custom Agent理由

- 省錢(只能免費用1800分鐘，現在要申請制)
- Microsoft Agent啟動速度有時太慢 (有時真的很難用)
- 加快Pull Code速度
- 加快Build Container速度(用過Microsoft Host Agent，會有想不開的感覺)
- 避免.NET Core / .NET Framework更新問題，導致舊系統無法Build
- Release的環境在地端

我們Agent配置與設計

- Build和Release Agent要分開不同Pool
- 地端Build和Release Agent的啟動服務帳號要不同
- 如果資源允許，Build和Release Agent最好能放在不同Host
- Build Container與Non-Build Container Agent Pool需要分開
- 雲端Build Agent Pool內的Agent最好設置在不同Region

我們Agent配置與設計

- Build Container使用Private Premise的Build Agent
  - 裝載Container Host版本不同，Build Agent也要放在對應的Host版本
- Release到Azure環境，使用Private Cloud Release Agent
- Release到On-Premise，使用Private Premise Release Agent
- 優先選擇Private Cloud Build Agent

Agent特殊現象

- Agent的版本最好能自動更新
- 無法自動更新時，務必要定期手動離線更新Agent
- Build Agent不更新，會導致新版本.NET或是相關套件無法進行編譯
- 若發生Azure DevOps Service找不到Agent，很大機會是Agent版本過舊

SKILLTREE

Pipeline Design Practice

我們的Pipeline設計

你以為Pipeline很好設計嗎?

- 企業每個系統應用情境不同，Pipeline設計都不太可能相同
- 因為自動化的好處，人人都希望什麼都可以在Pipeline內做到
- 企業的場景與文化往往決定Pipeline設計複雜度
- 開發者總愛提出各種"創意"，挑戰Pipeline設計
  - 後面的講者都是挖這個坑的人

Build Pipeline設計理念

- Pipeline不是建立後讓它會跑就好
- CI 僅處理Build & Unit Test Process
- CI 編譯時間能越短越好
- CI 設計盡量與Agent的Host相依性低
- CI 設計要能隨時可以切換Agent Pool
- 用Azure Library管理Application內變數或共用參數

Release Pipeline設計理念

- Release僅處理應用程式佈署或是進行環境佈署
- 敏感性參數的安全性，用Azure Library管理
- 假使用一個PowerShell Script能完成，也要區分Task處理

Pipeline Design Flow



Menu

手動執行模擬



Process

建立 Flow



Test

測試CI



Review

觀察執行時間

https://mvc.tw

Pipeline Design Flow

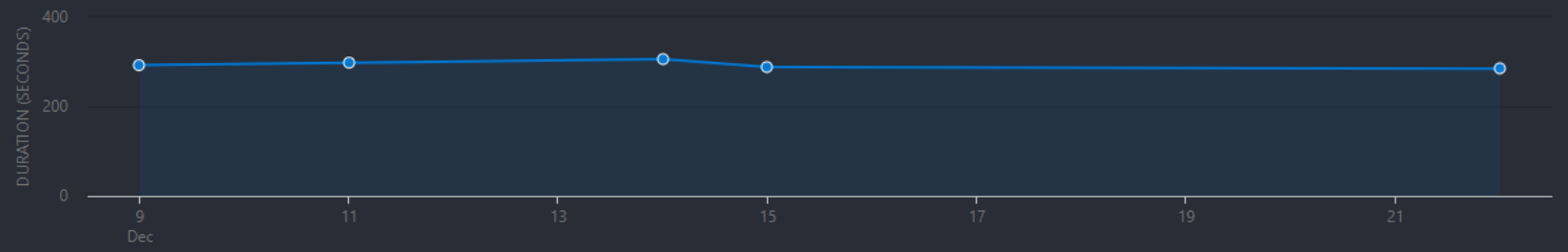
Pipeline Duration

80th percentile

4m 57s ▼2.7%

10 succeeded pipeline runs

22.93% of total run time is taken by step - *npm run*

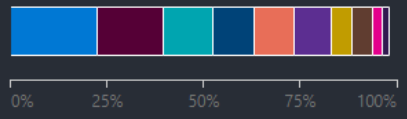


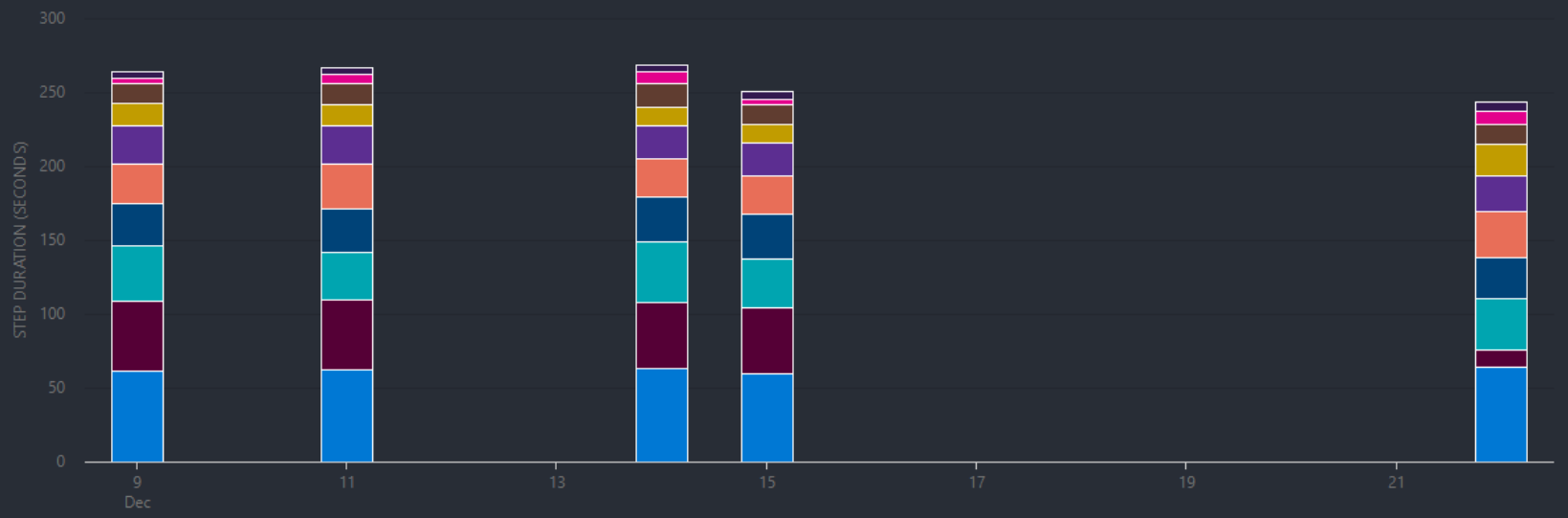
Top 10 Steps By Duration

80th percentile

- Default\npm run
- Default\Download Build Artifacts
- Default\npm CI
- Default\Push images
- Default\Publish Artifact: EASWeb
- Default\Checkout Hawkeye.Docker@Develop to s
- Default\Initialize job
- Default\dotnet build
- Default\dotnet restore
- Default\Build an image

Top 10 steps by duration account for 98.36% of the execution time.





Build Pipeline Common Rule

- Library做DEV、UAT和PRD參數置換
- Build後的結果，必須上傳到Pipeline Artifacts
- Container
  - 先Build Application，再Build Container
- 不同版本Host編譯的Container Image不能共用
- 不同Branch都要有Build Pipeline



edwardkuo OK

.vs	OK
CI Pipeline	Add file
Release Pipeline	Add file
LICENSE	Initial commit
README.md	Initial commit

README.md

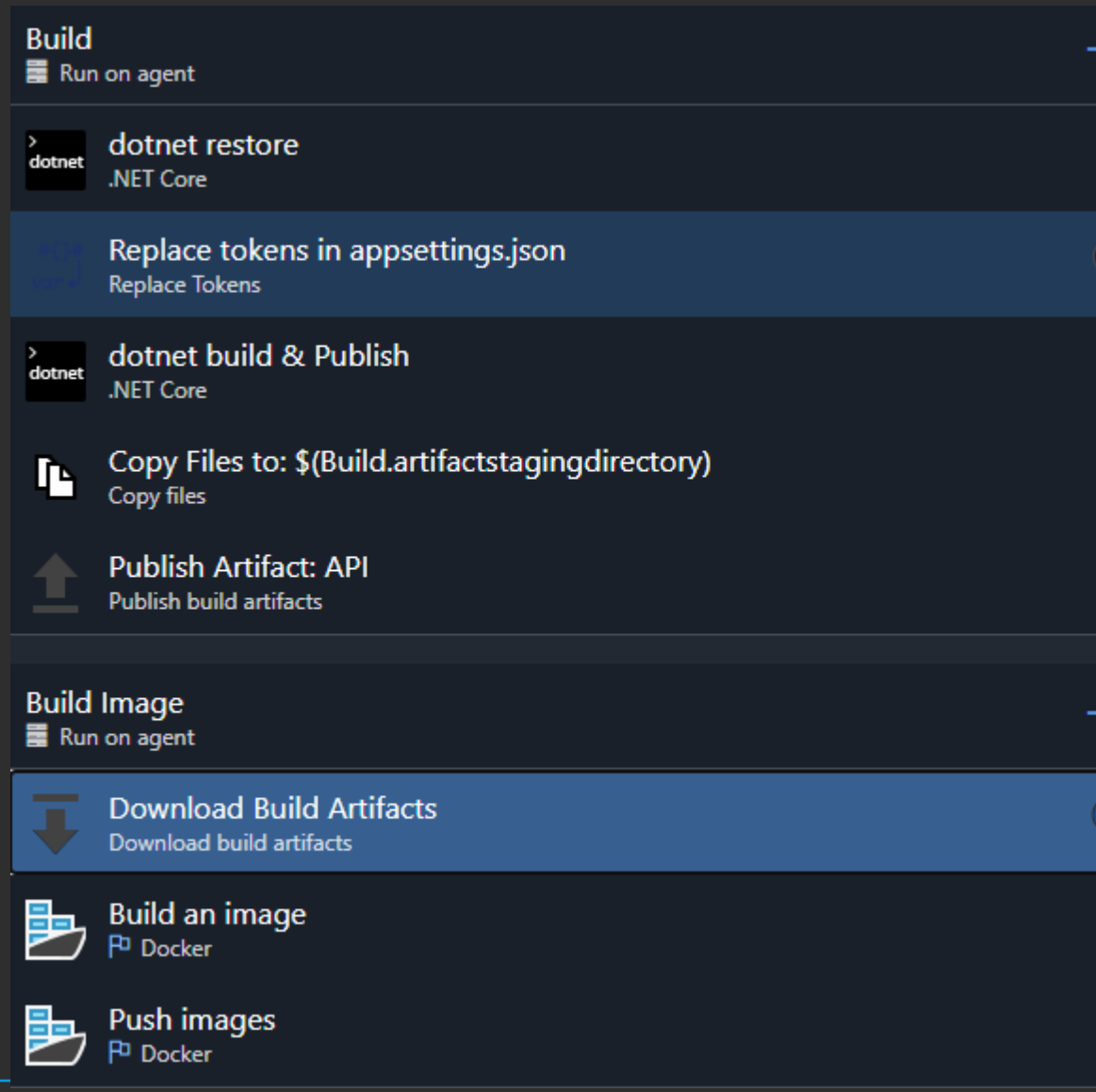
AzureDevOpsPipelineSample

Build Pipeline Task Flow

https://github.com/edwardkuo/AzureDevOpsPipelineSample

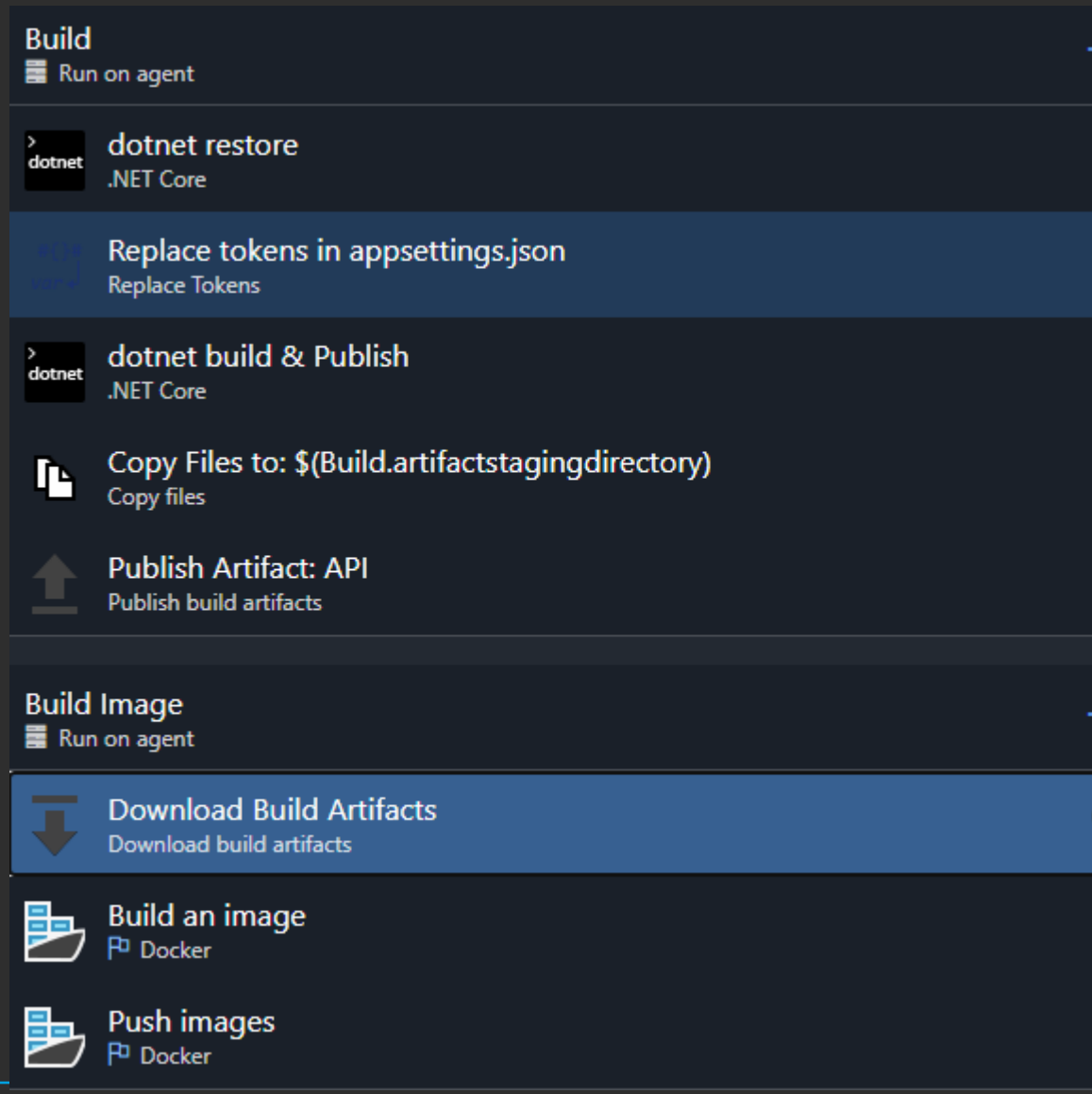
Build .NET Core & Container

- 設計順序，Fist Agent
  - .NET Core : 還原Package
  - Replace Parameter : 更換appsettings.json的設定參數
  - SonarQube: Scan Code
  - .NET Core : 編譯
  - Publish Artifact :上傳編譯後的檔案



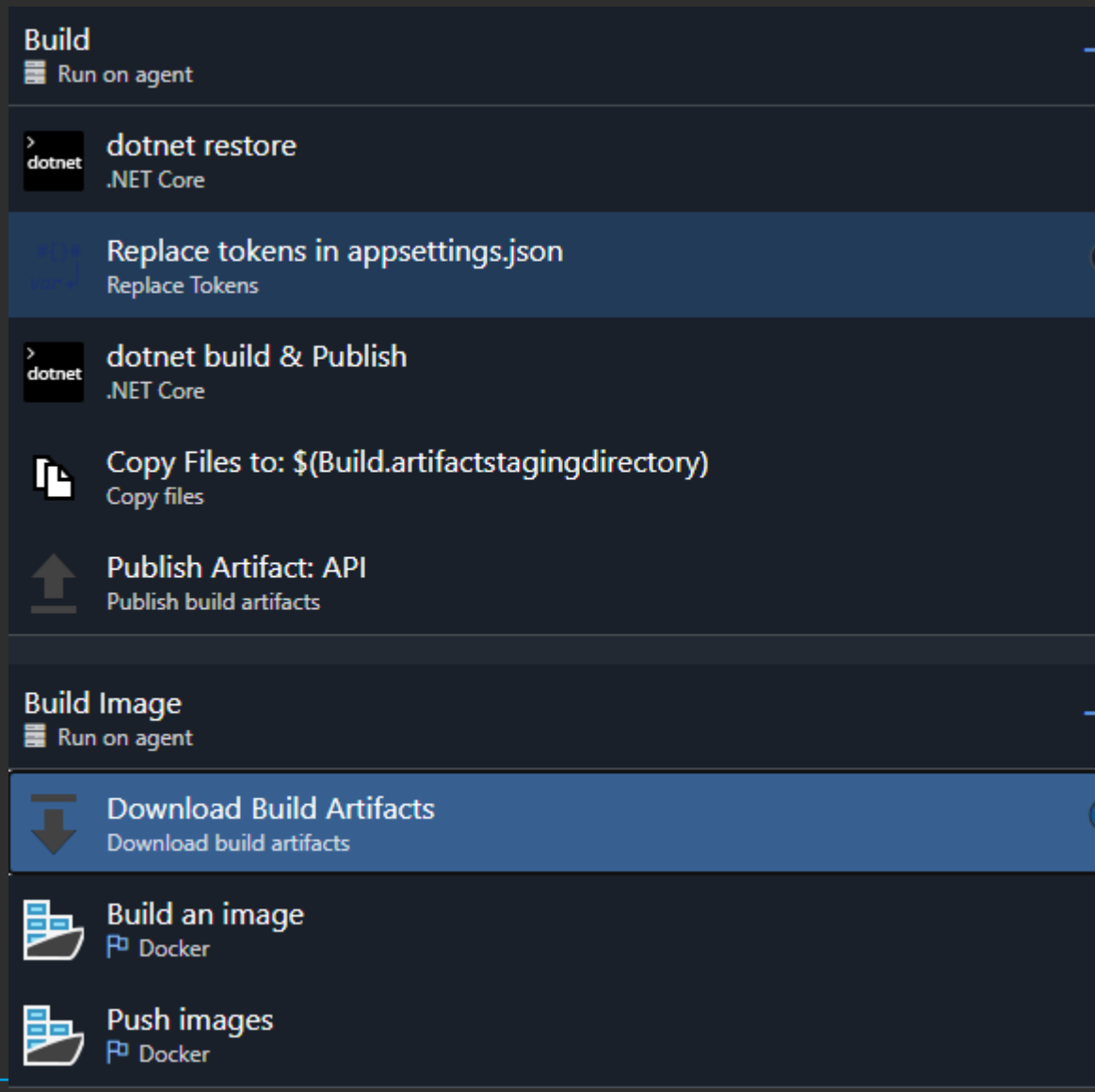
Build .NET Core & Container

- 設計順序，Second Agent
  - Download build artifacts : 下載編譯檔案和Dockerfile
  - Build Docker : 檔案Copy到Container並編譯Docker
  - Push Docker : Push Container到ACR



Build .NET Core & Container

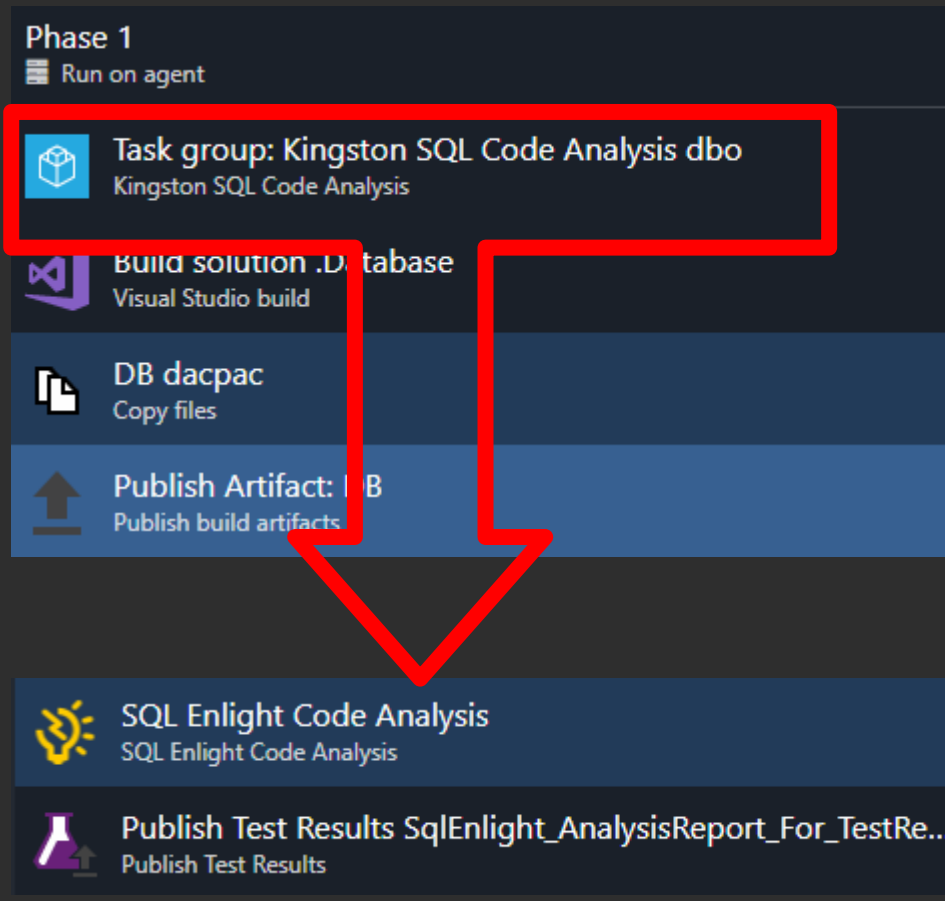
- 分批建置Container Application
  - Application Agent
  - Docker Agent
- 不需要Container時，可以單獨發布
- 能單獨對編譯的Application除錯
- Container編譯一定要帶入版本號



MVC
https://mvc.tw

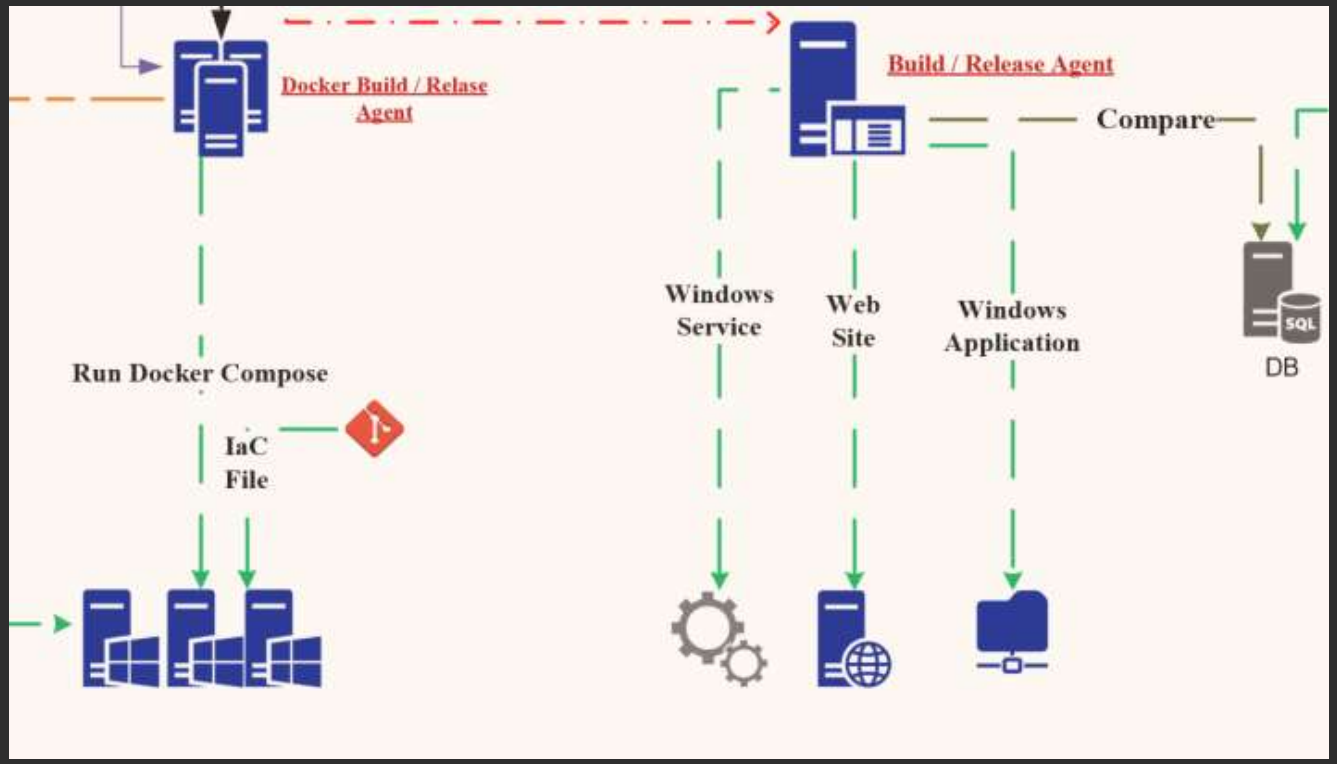
Build SQL Project

- 設計順序
  - SQL Enlight Code Analysis : Scan SQL
  - Publish Test Results : 上傳Test Result
  - MS Build : 編譯
  - Copy File : Copy .DACPAC到Artifact
  - Publish Artifact :上傳編譯後的檔案
- 前提必須用SQL Project，對資料庫程式進行版控



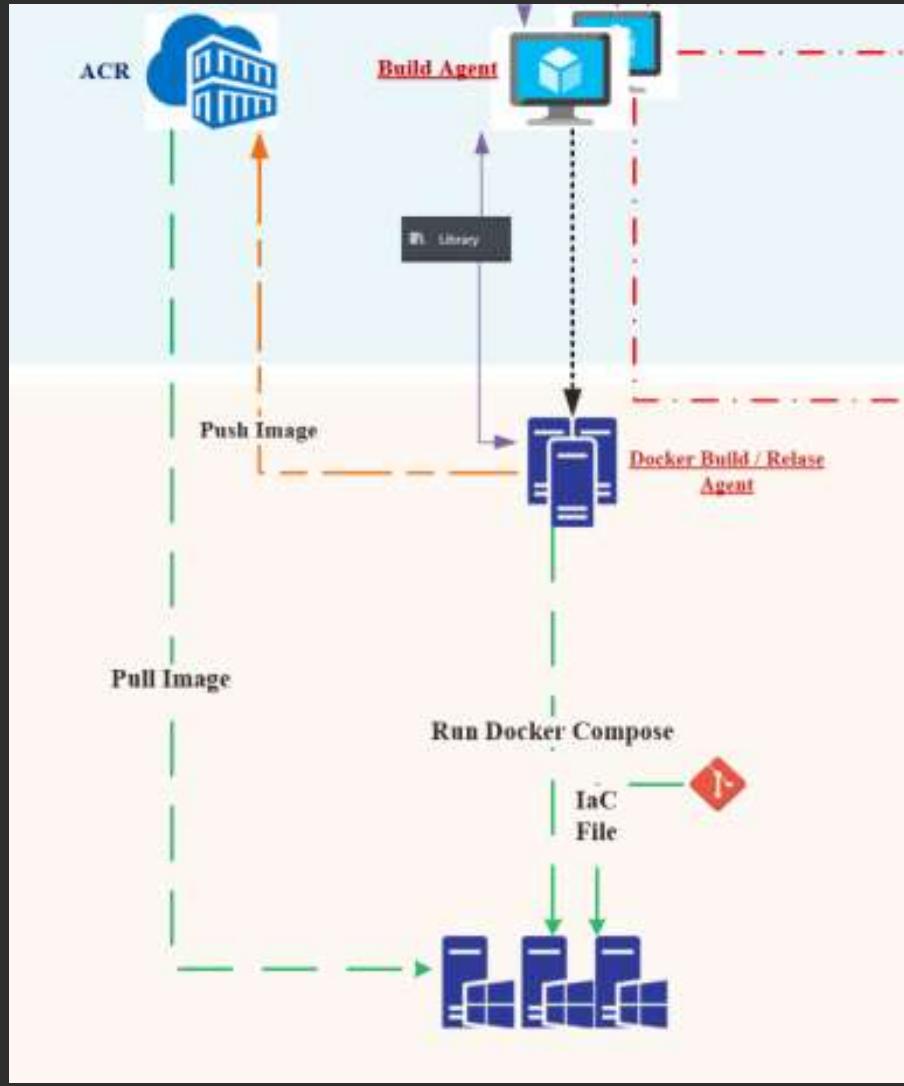
Release Pipeline Common Rule

- 所有編譯的檔案，一律從Build Artifact取得
- 正式環境部署皆需要人員Approve
- 非Container的系統佈署，大多用Copy File方式處理
- 佈署用的參數存放在Library
- Release Agent不放在目的端的Host
- Docker Compose File用額外的Repos管理



Release Container

- 設計順序
  - 從IaC Repos下載該Container的Compose.yml
  - 置換Compose file的Docker版號
  - docker-compose Pull : 下載 Container
  - docker-compose Down : 停掉Container
  - docker-compose Up : 啟動Container
- 從UAT到PRD，僅Container轉移Host，並用不同IaC File啟動Container



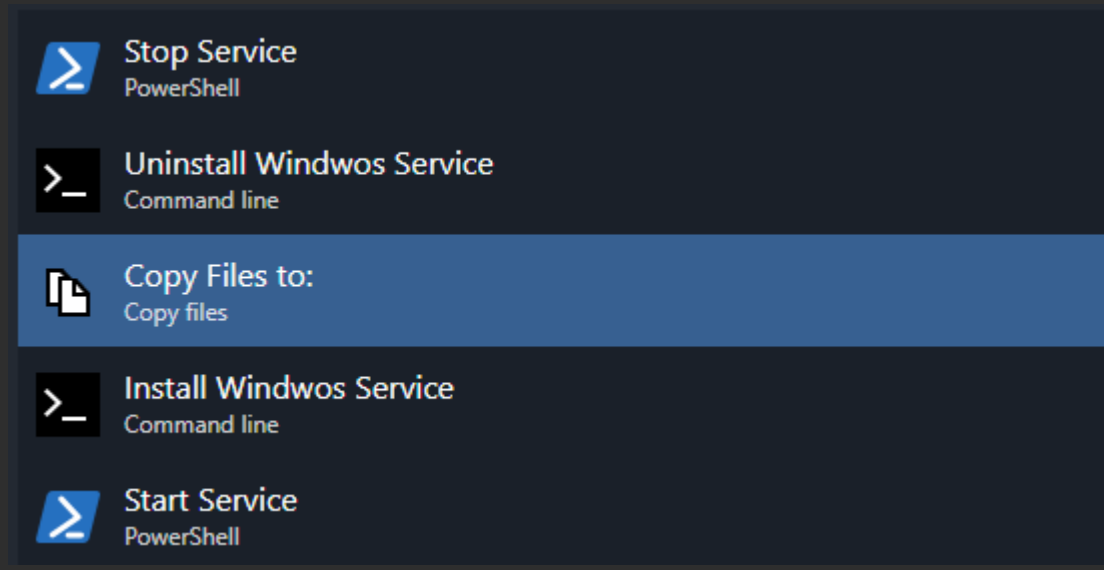
Release Container

- Release的Container必須要有版號，才能進行管理與快速復原
- $(Release.Artifacts.PipelineName.BuildNumber) 取得CI版本
- 置換Compose內的Image版本參數
- Compose的image版本號碼是用Buildnumber

```yaml
version: '3.8'
services:
  pro-easweb:
    image: [redacted]release-qas/platform/easweb:#{BuildNumber}#
    restart: always
    ports:
      - 7014:80
    environment:
      - ASPNETCORE_ENVIRONMENT=Release
networks:
  default:
    external:
      name: "nat"
```

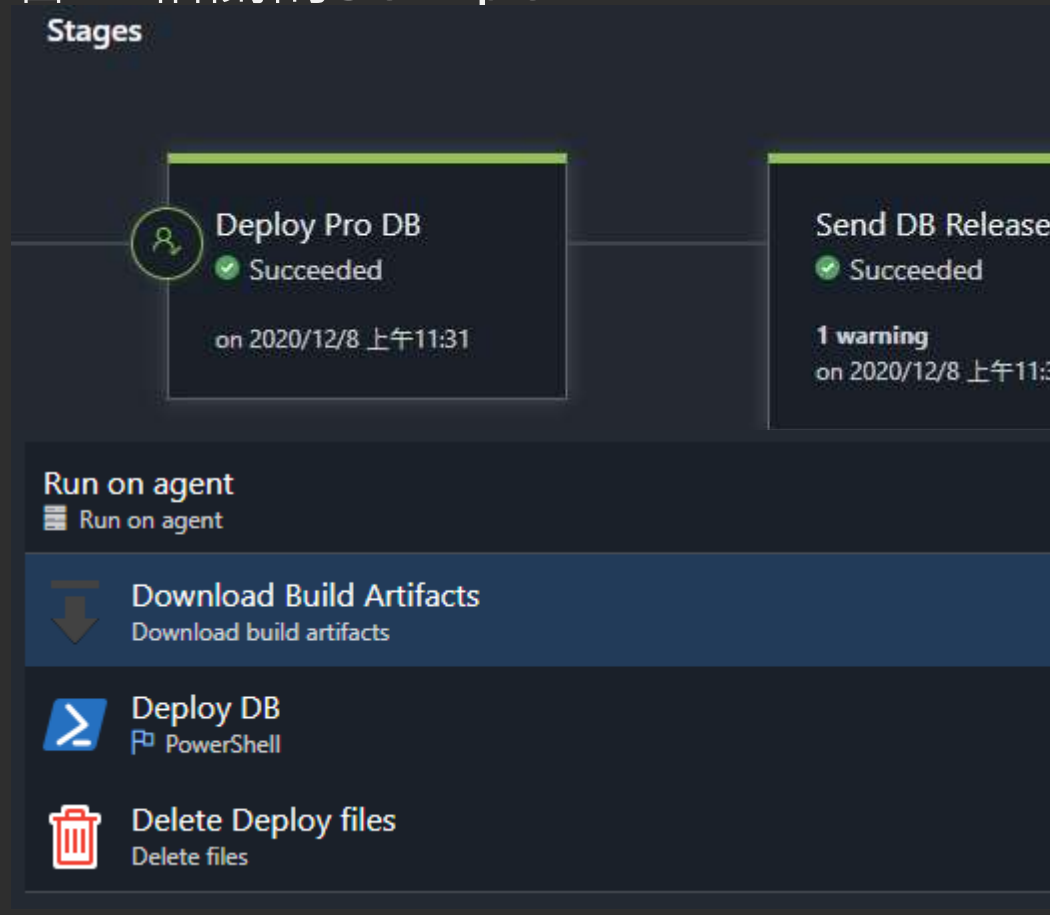
Release Windows Service

- 設計順序
  - PowerShell : 停止WS在遠距Host
  - Command Line : 卸載WS
  - Copy File : 複製新的Application
  - PowerShell : 安裝WS到遠距Host
  - Command Line : 啟動WS
- 無專門Task，必須透過Command處理



Release Database

- 設計順序
  - Download Build Artifacts : 下載需要佈署DB相關的Script
    - Deploy PowerShell Script
    - SQL Publish XML File
  - PowerShell : 執行佈署Script
  - Delete files : 刪除佈署DB相關的Script
  - Send email : 差異化檔案用Mail寄出
  - Agent Host要安裝sqlpackage.exe



Release Database

- 差異化比較用的參數，用DB.Publish.xml管理
- 執行差異化比較的Script，用PowerShell執行

```
#SQLpackage Path
$SQLPackagePath='sqlpackage.exe'

#Dacpac & release script Path
$DacpacPath=$args[0]
$DBDifferentScript='D:\DB.sql'

#DB Server
$DBServer='#{DB Server}#'
$DBName='#{DB Name}#'
$DBUserName='#{Account}#'
$DBPwd='#{Password}#'

#SQLPackage Command
$CompareCmd='/Profile:'+$args[1]

$SQLPackageCmd=' /action:Script /q:true /sourcefile:'+$DacpacPath+' /TargetServerName:'+$DBServer+' /t
$SQLPackagePath=$SQLPackagePath+' --% '+$SQLPackageCmd+' '+$CompareCmd

"Start Generate Release SQL Script file"

"Waiting some second !!"
#$SQLPackagePath
```


tw MVC
https://mvc.tw

常用的Marketplace套件

- SQL Enlight Code Quality Task
  - 掃描SQL語法是否有符合SQL撰寫規範
- SonarQube Task
  - 整合SonarQube服務，進行Code掃描
- Trigger Build Task
  - 現有Pipeline流程，驅動其他CI Pipeline流程

常用的Marketplace套件

- Regex Replace Task
  - Regular Expression替換掉Code資訊，如: AssemblyInfo.cs
- Replace Token Task
  - 透過 #{變數名稱}#，將資訊注入到File，常搭配Azure DevOps Library使用
- Send Mail Task
  - 發送信件通知，也可以在信件內夾帶附件

SKILLTREE

官方未說的應用方式

使用模式僅參考，使用前要深思

Schedule Controller

- 把Release當作系統 / 維運的定時排程器
  - 不需要使用Windows Task Schedule
  - 可以有很完整歷程記錄
  - 可以設定執行排程的流程
  - 用Azure DevOps Service，還可以透過雲端去手動啟動地端排程

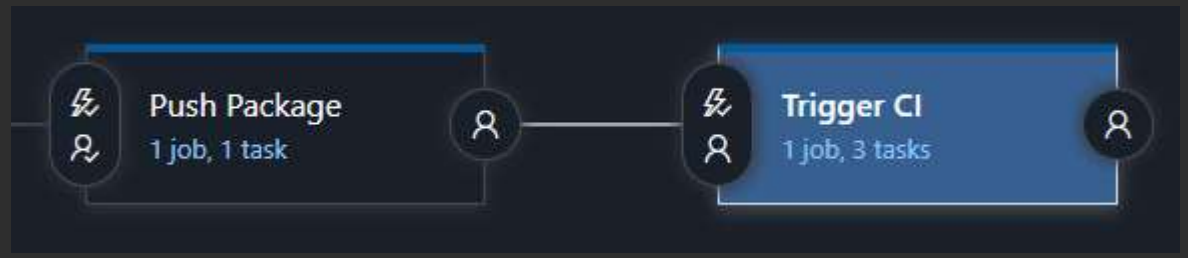


Backup All Repos File

- 基於ISO 27000的規範
- 需要將雲端Repos File 備份到地端的三個步驟
  - PowerShell & Git 指令
  - Azure DevOps API
  - Release排程功能
- GitHub Code

Trigger CI on Release Stage

- 當Package被Release後，必須自動驅動某系統的CI，讓系統立即取得最新的Package版本使用



- 該系統在CI要在NuGet Package添加必須取得最新版本Package指令

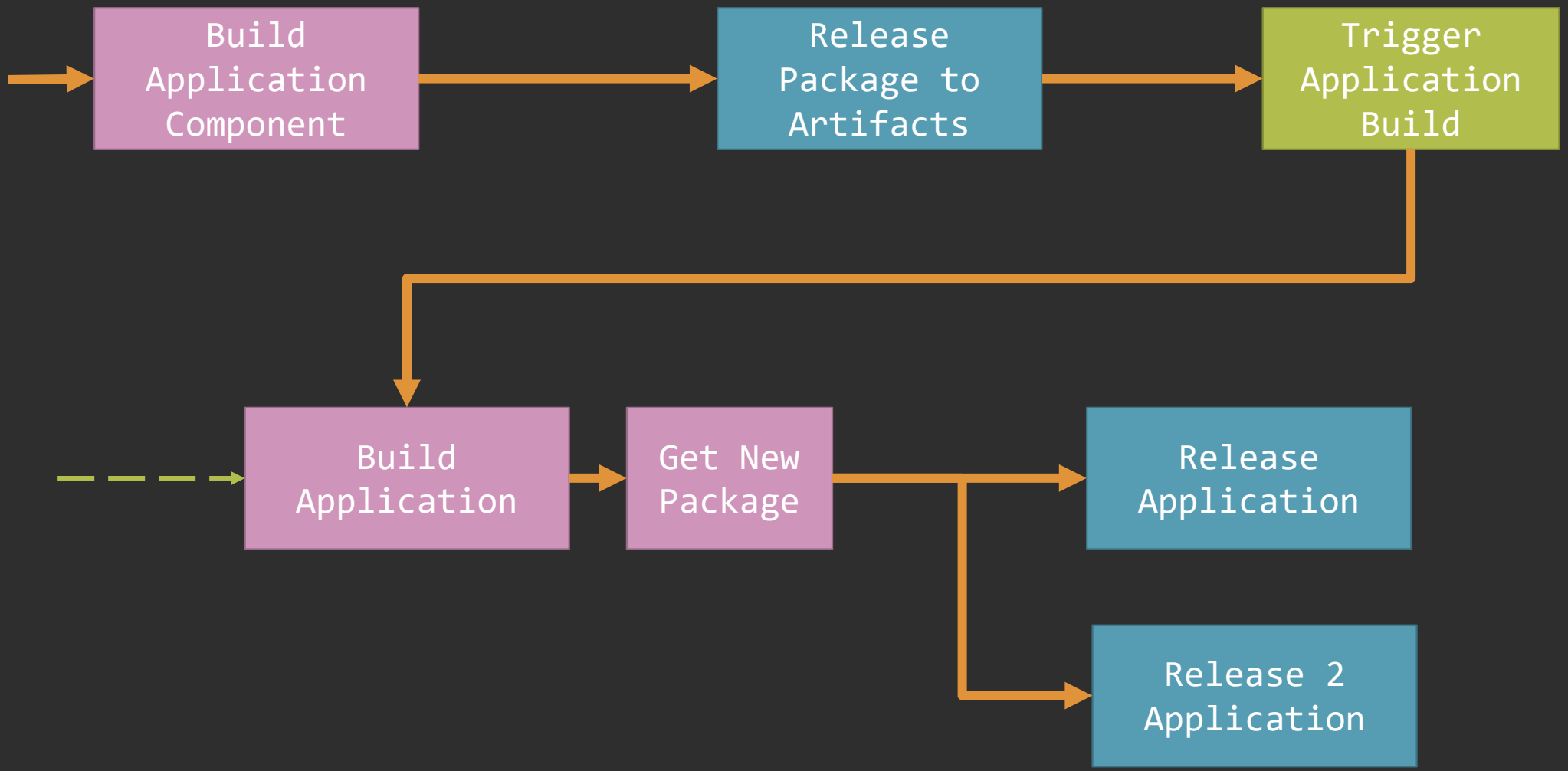
Use NuGet Task

Command : Custom

Command and arguments:

update “Project.sln" -Id “NuGet Package Name" -ConfigFile "nuget.config"

Trigger CI on Release Stage



SKILLTREE

Summary

Our Challenge

- 你以為看到就是真的
  - 不容易明白的錯誤訊息
- 介面與功能常常改變
- 不一定支援全部的開發語言，但.NET世界，99.8%都具備
- 有大量的Pipeline需要被管理
- Pipeline失敗不一定是Application問題

Use YAML?

- 好處
  - CI 用YAML讓有相同流程能快速建立Pipeline
  - Pipeline能進入版控世界
  - 支援CD，但功能尚不完善不建議使用
- 挑戰
  - 語法需要熟悉，遇到第三方Task的YAML，不容易撰寫
  - 入門門檻較高

Azure DevOps CLI & API

- CLI
  - 使用CLI可以擴展Azure DevOps，從命令行管理許多Azure DevOps服務
- API
  - 透過REST API提供對服務的創建，檢索，更新或刪除

Pipeline是該誰設計

- 企業很少有專屬設計(開發)Pipeline人員
- 流程相同盡量用Task Group取代
- Pipeline使用還需一些額外技能
  - Command Line / PowerShell
  - YAML
  - SQL Command



Hello Azure DevOps

是DevOps工具
也是一個應用平台

https://github.com/edwardkuo/AzureDevOpsPipelineSample

Blog 是記錄知識的最佳平台



https://dotblogs.com.tw

SkillTree 為了確保內容與實務不會脫節，我們都是聘請企業顧問等級並且目前依然在職場的業界講師，我們不把時間浪費在述說歷史與沿革，我們並不是教您考取證照，而是教您如何上場殺敵，拳拳到肉的內容才是您花錢想要聽到的，而這也剛好是我們擅長的。



https://skilltree.my

天瓏資訊圖書

