Microsoft STUDY4 Build School

Azure DevOps Service 應用架構與秘辛

Edward Kuo

Edward Kuo

Microsoft Regional Director

Microsoft Azure MVP

Kingston IT Manager

https://profile.edwardkuo.dev/

- 2020 宏碁科技 DevOps 講師
- 2020 啟碁科技 Azure DevOps 講師
- 2020 Azure Monitor with TSMC Form 講師
- 2019 台灣 恩智浦半導體 DevOps 培訓講師
- 2019 .NET Conf Taipei 講師
- 2019 中原大學 資料科學實務 客座講師
- 2019 中原大學 資工系 雲端計算平台實務 客座講師
- 2019 Agile Tour Hsinchu DevOps 講師
- 2019 DevOps Days Taipei 講師
- 2019 DevOps Expo @Trend Micro 講師
- 2019 翻轉營運契機 Azure DevOps 趨勢與實務研討會 講師
- 2019 Insider Dev Tour 台北站 講師
- 2018 DevOps Days Taipei 講師
- 2018 DOIS DevOps 國際峰會 深圳站 講師
- 2018 Insider Dev Tour 台北站 講師
- 2018 .NET Core Conf 台北站 講師
- 2018 Agile Tour Taichung 講師
- Global Azure Bootcamp 台灣 & 廣州 講師
- 微軟 北京 Tech Summit 講師

序

Help Us Something

- 時間花在創造商業價值，降低整合異質平台帶來的不便性
  - 尤其對 .NET 或是微軟技術開發者
- 從需求管理、開發到發佈採用同一個平台工作
- 因應開發技術的不斷更新，自動擴充和支持
  - Service版本才能做到即時性更新
- 彈性且容易整合非Microsoft的平台

Microsoft STUDY4 Build School

Azure DevOps

- Team Foundation Server
  - Azure DevOps Server

    是基於SQL Server的地端產品
- Visual Studio Team Service
  - Azure DevOps Service

    是基於Azure的SaaS級的雲端服務平台

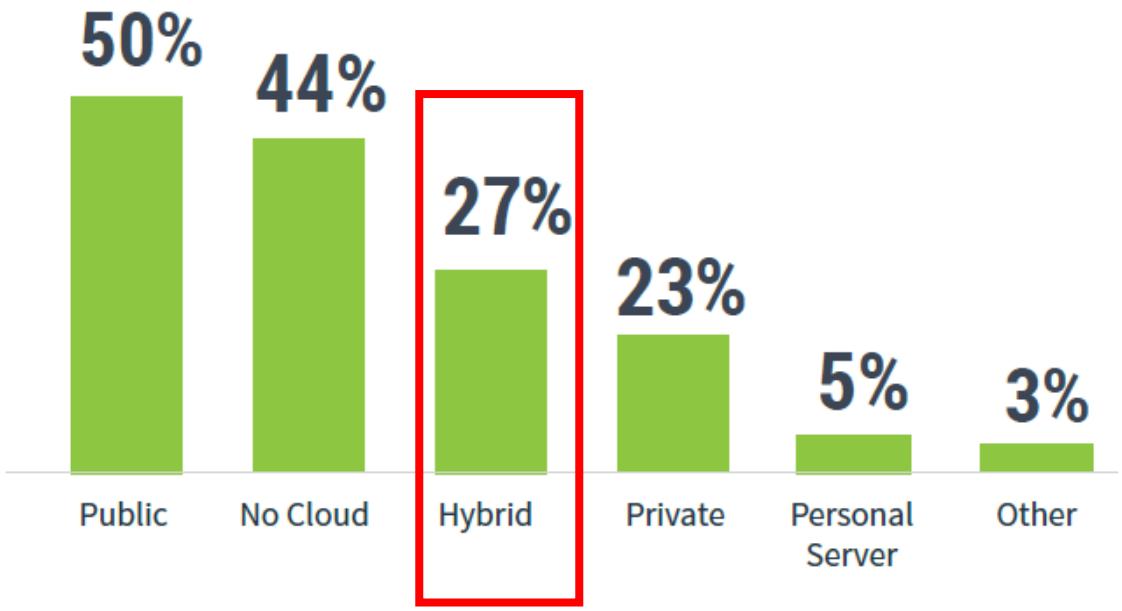
Microsoft STUDY4 Build School

Azure DevOps

- Server & Service 兩者區別
  - Scope and scale data
  - Authentication
  - Users and groups
  - Manage user access
  - Security and data protection
- Service無法透過SSRS做DevOps數據報表
- 兩者迭代速度不同

Cloud Trend

有越來越多的組織選擇多雲和混合雲解決方案。 這些解決方案除了提高性能外，還提供靈活性，控制性和可用性。多雲和混合雲的使用比例是有所增加。



Microsoft STUDY4 Build School

Architect Base On Azure DevOps

Our Azure DevOps Ecosystem

DevOps Map Azure DevOps

計畫 + 追蹤

- Azure Boards

開發 + 測試

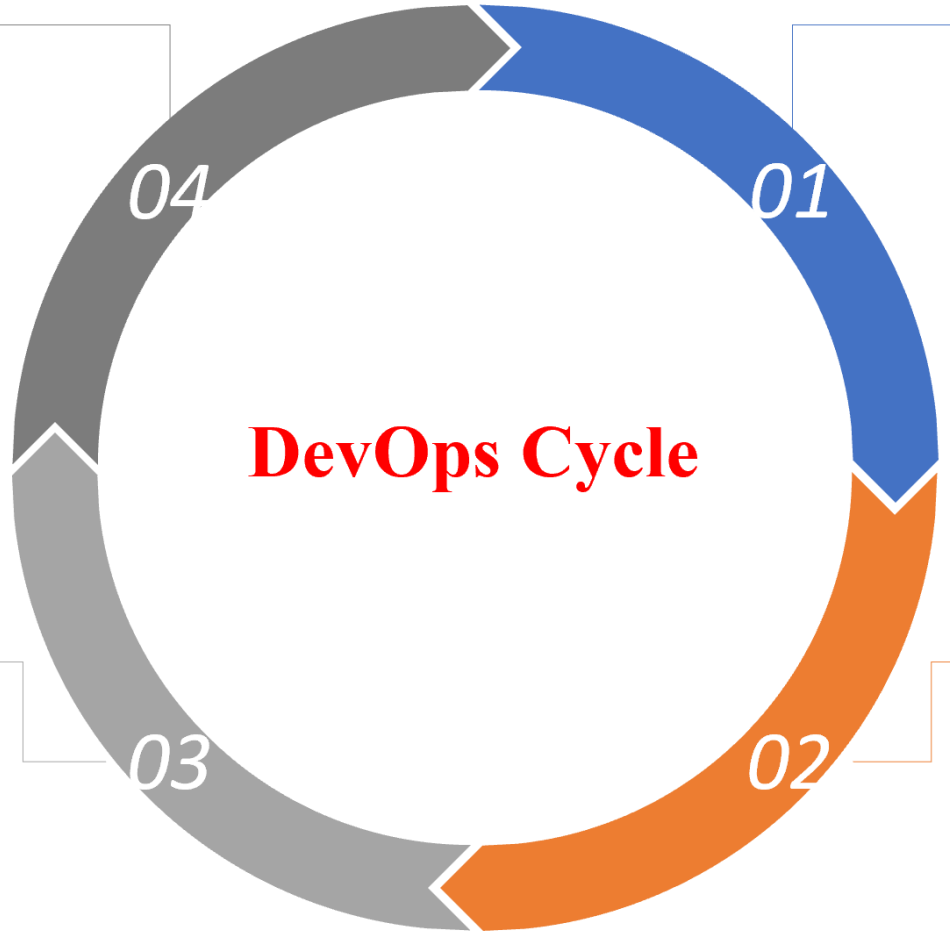
- Azure Repos
- Azure Pipelines
- Azure Test Plans

交付

- Azure Pipelines
- Azure Artifacts

監控 + 學習

- Azure Boards
- Azure Application Insights

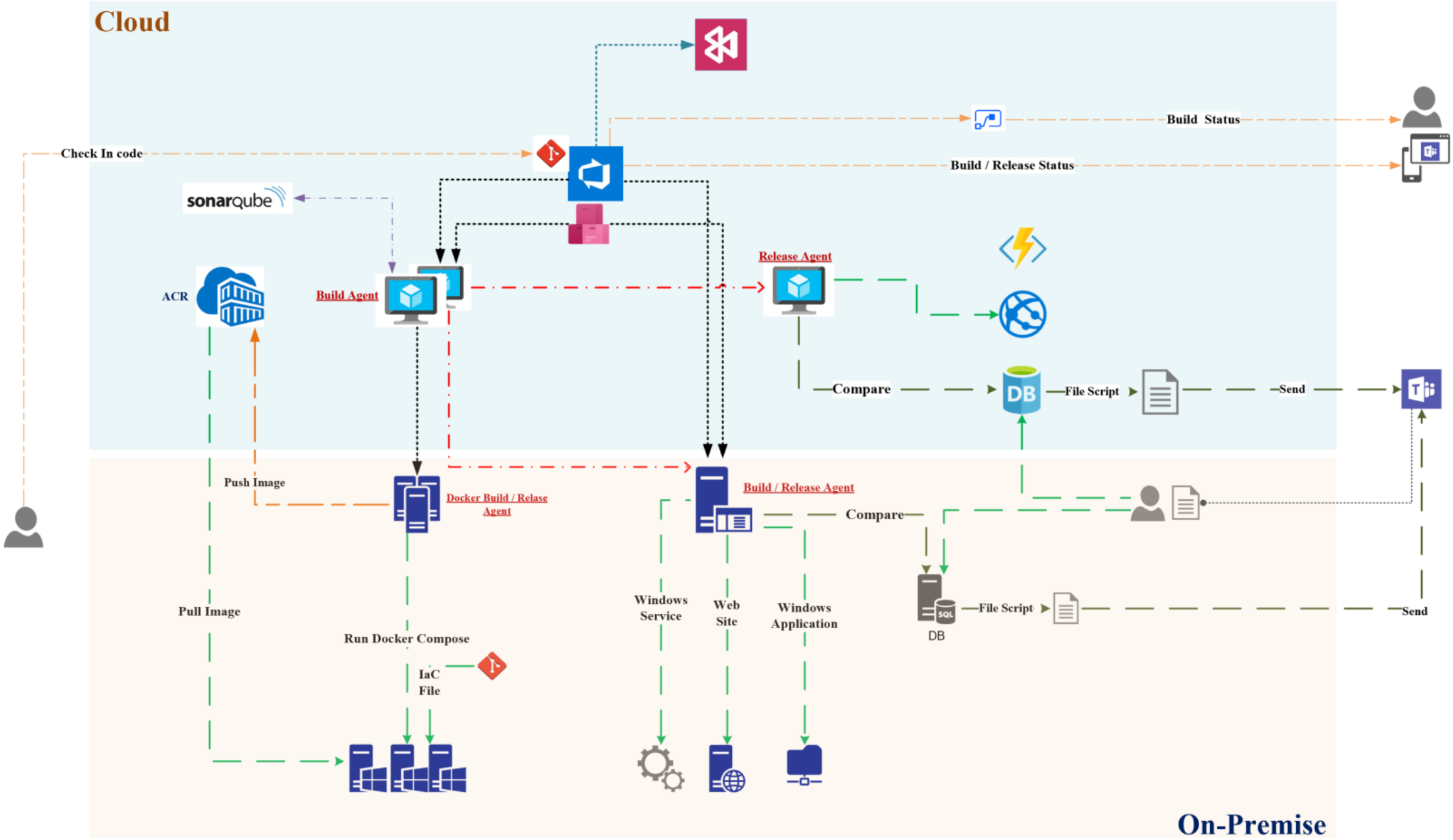


Microsoft STUDY4 Build School

Design Azure DevOps Architect

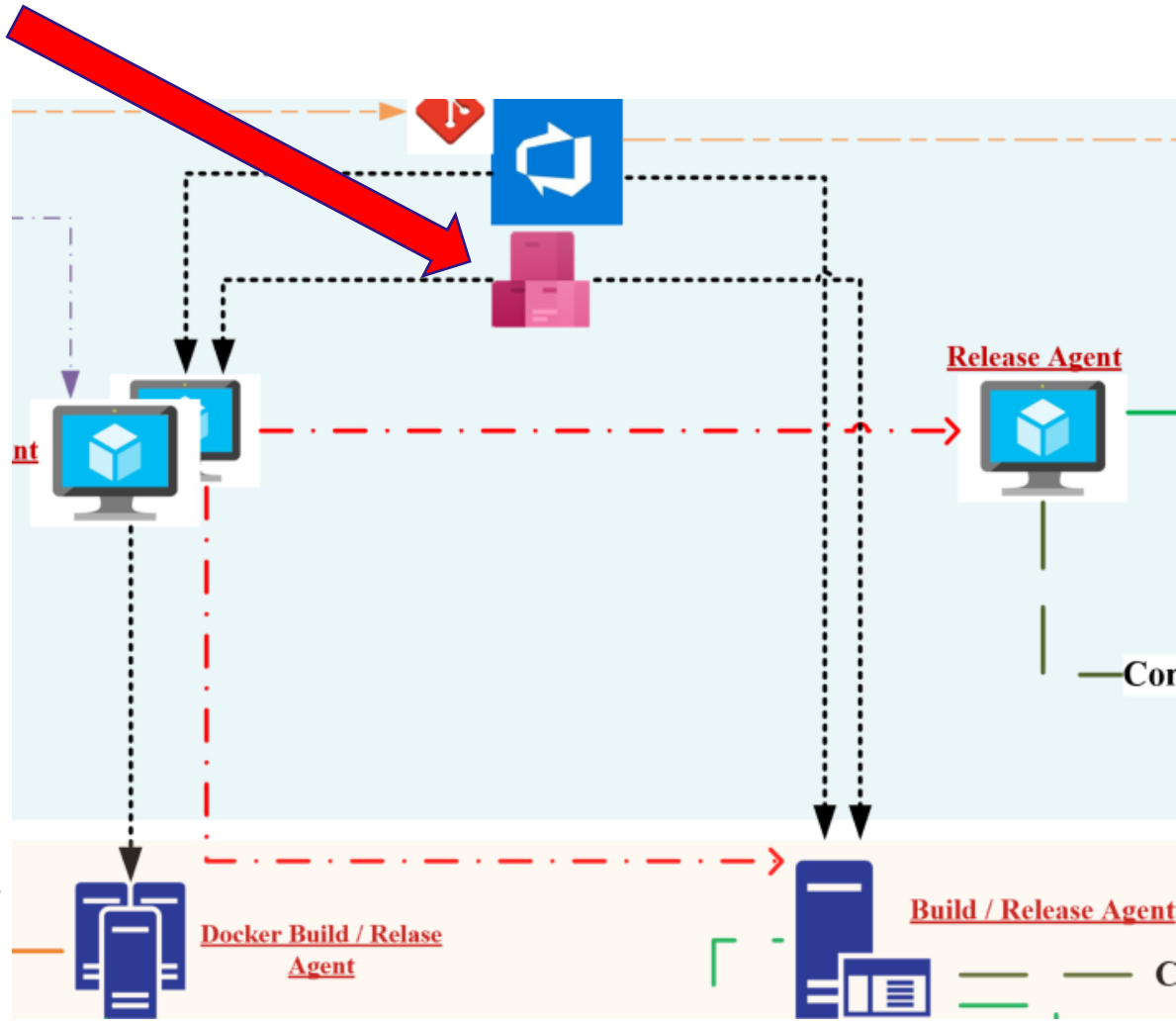
- 80% 系統是佈署在企業內部
- 20% 系統是佈署在Azure Services
- 必須能符合企業應用情境模式進行佈署

Design Hybrid DevOps Architect


Cloud
Check In code
Build Status
Build / Release Status
sonarqube
ACR
Build Agent
Release Agent
Compare
DB
File Script
Send
Push Image
Docker Build / Relase Agent
Build / Release Agent
Compare
Pull Image
Run Docker Compose
IaC File
Windows Service
Web Site
Windows Application
DB
File Script
Send
On-Premise

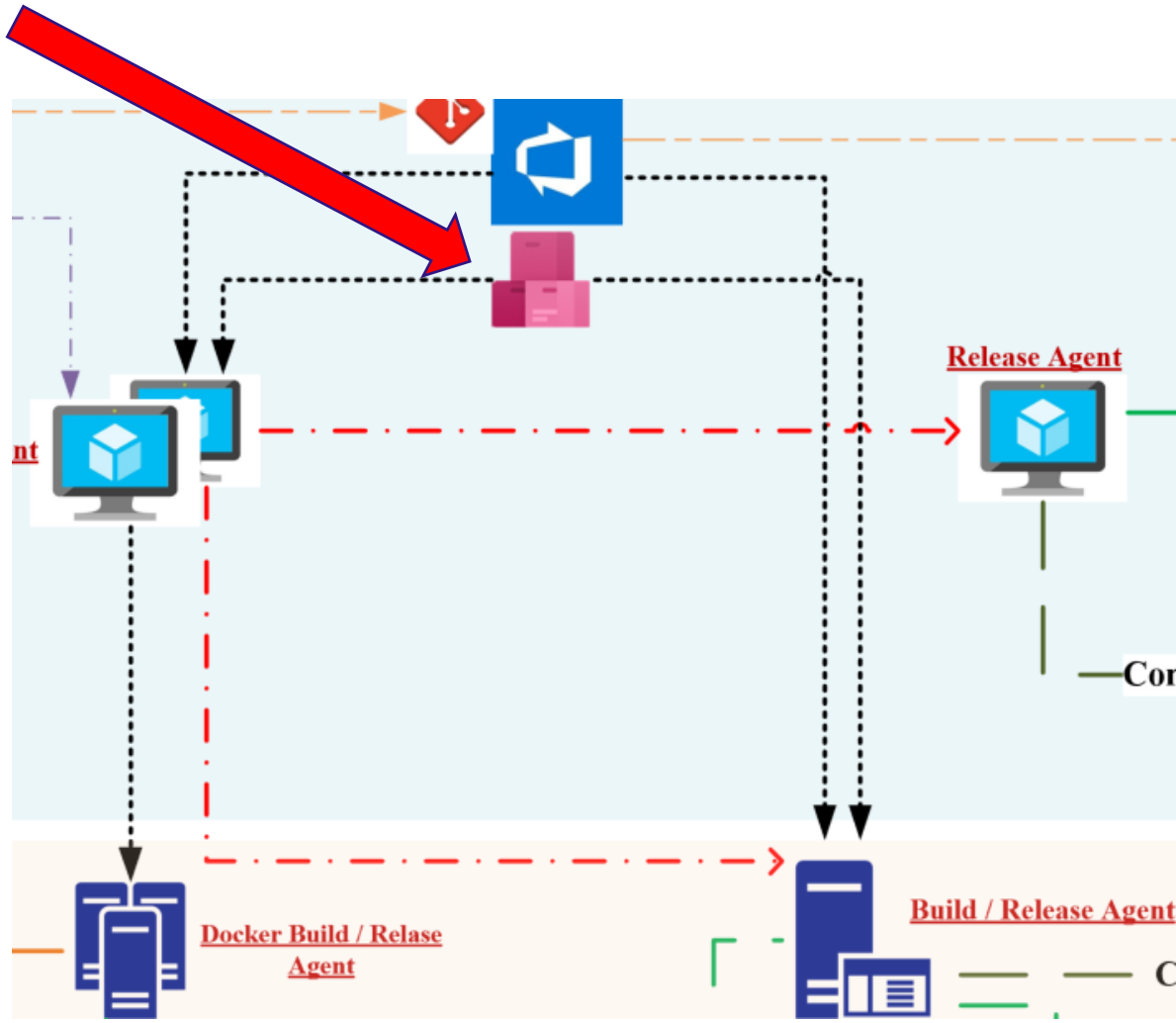
Artifacts Service

- 解耦
  - 系統拆解
    - 拆分到耦合性最小元件，透過共用元件或是核心元件，並重複使用與增加被測試可能性
  - 降低CI時間
    - 系統程式碼越多，每次編譯拉下程式碼重新編譯時間就會越長
  - 元件化
    - 在分散開發與快速迭代下，不影響他人，耦合性降低，持續整合也會加快且更好釐清問題



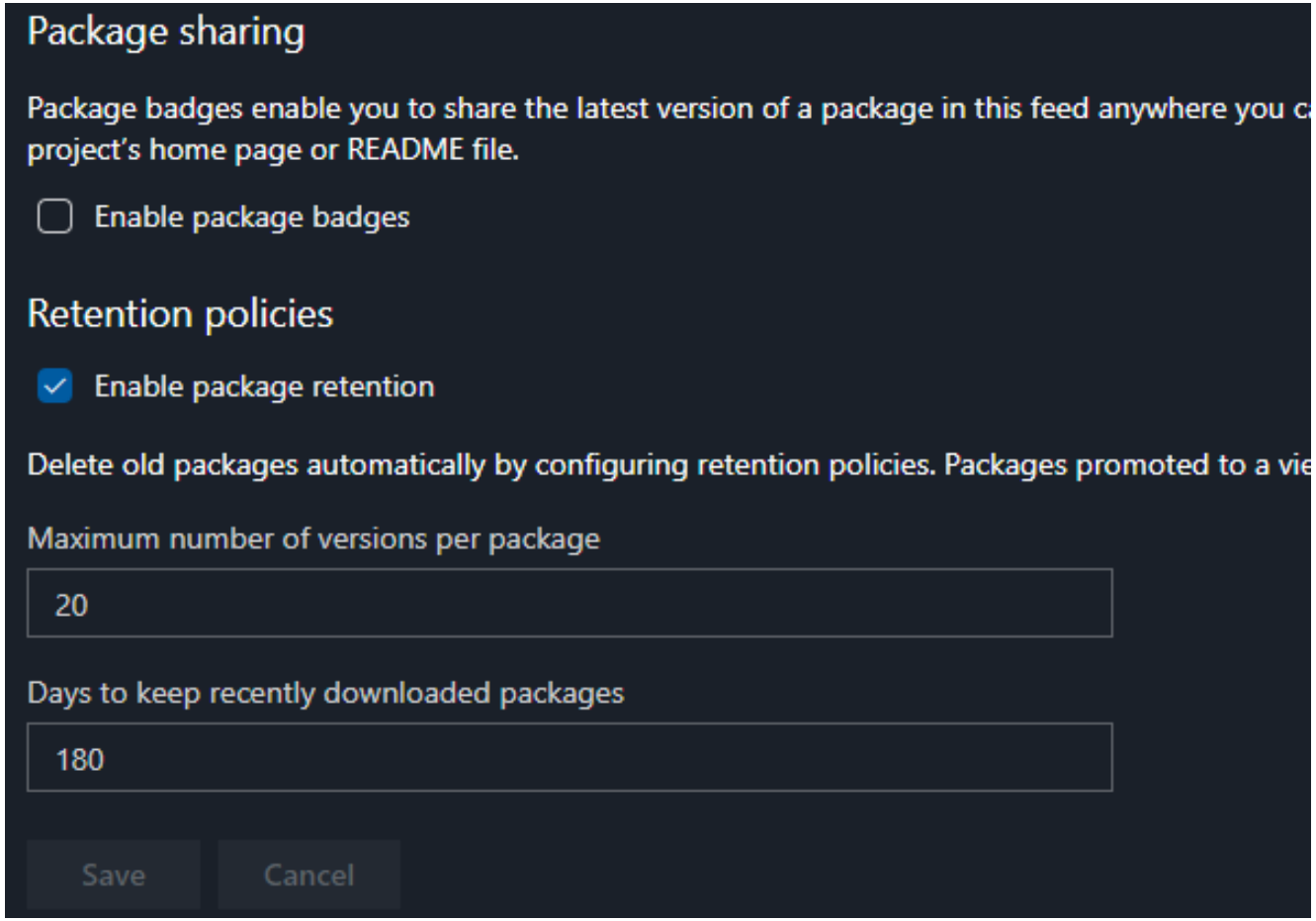
Artifacts Service

- 建立團隊/企業的Feed
- 能放上第三方元件的nupkg 檔案
- 透過CI / CD 建立 nupkg 檔案
- 只需要使用NuGet Push Task



Artifacts Service

- 避免企業使用過期元件
- 設定Package 保留週期
  - 設定最多Package版本數量
  - 設定保留天數



Microsoft STUDY4 Build School

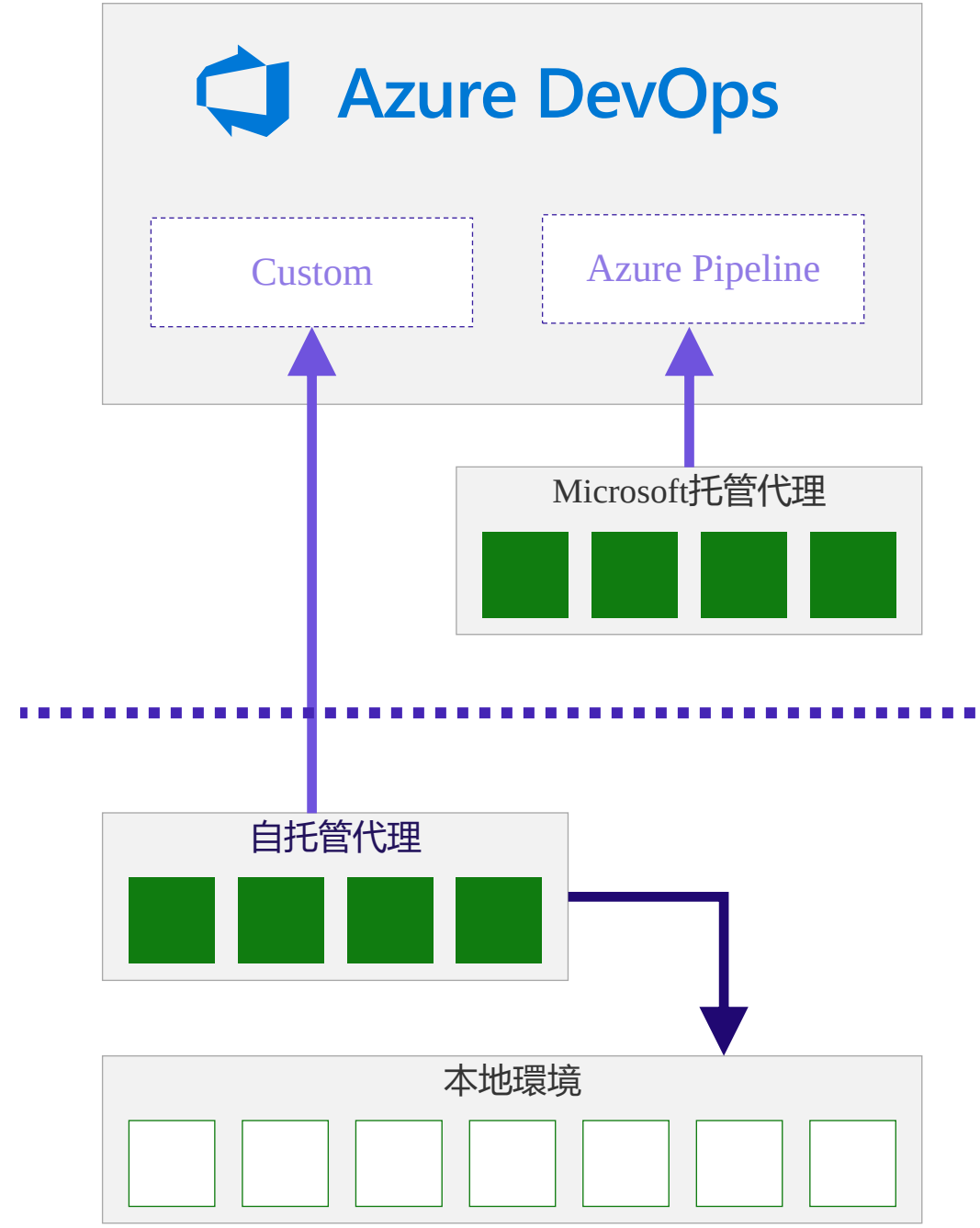
Agent Pool Practice

Azure DevOps Agent Pool

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

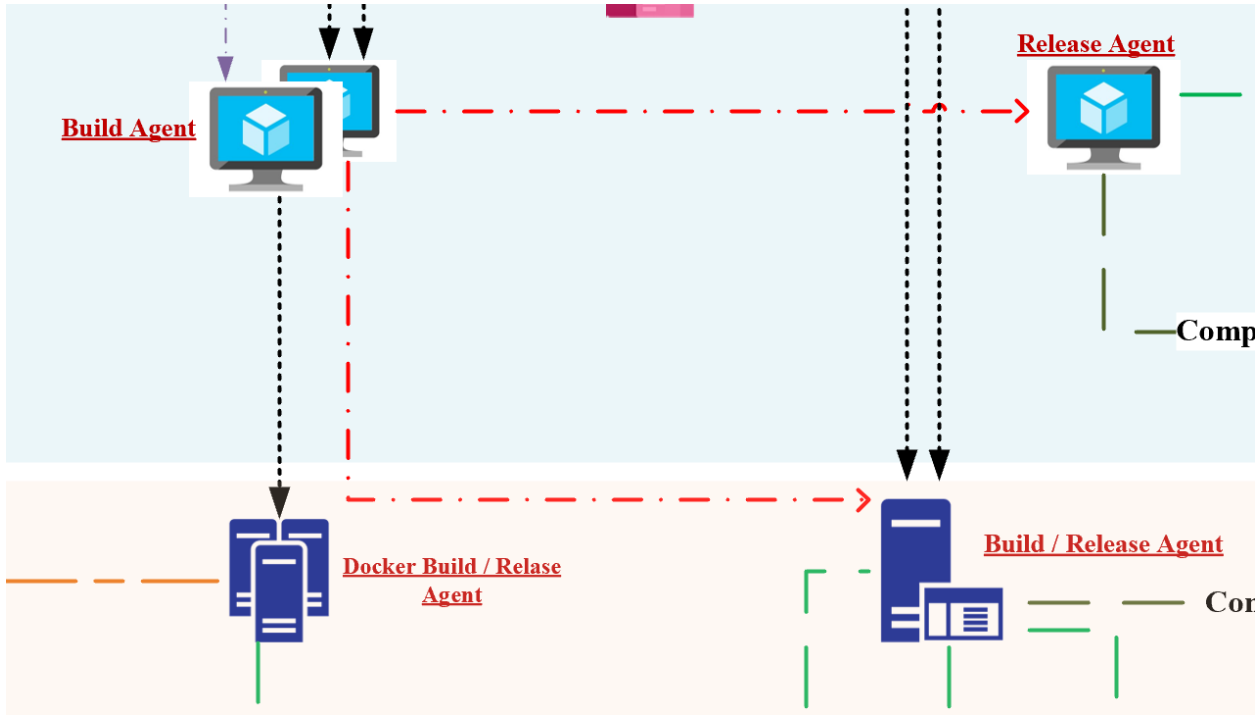
  →Agent

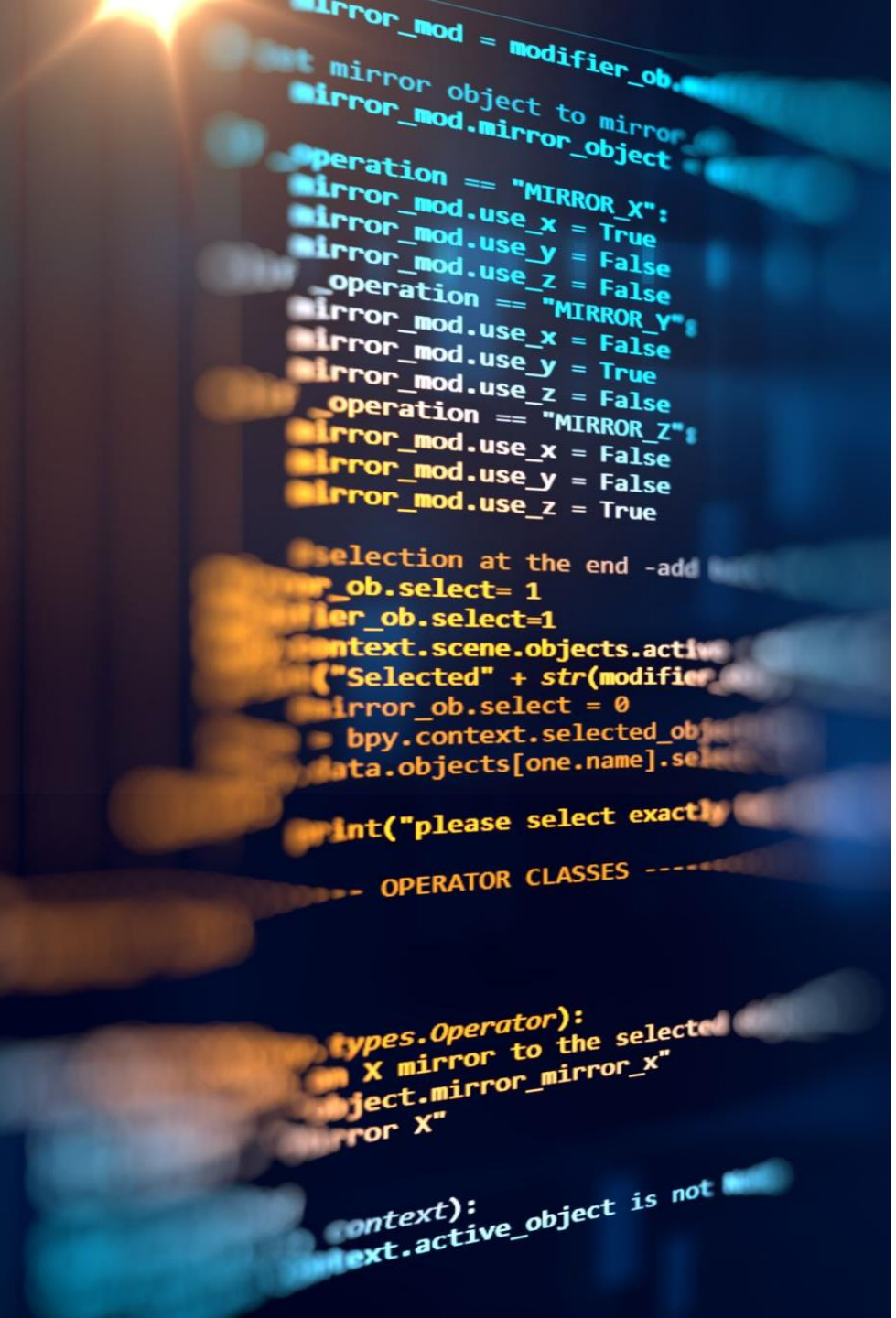
  →Host



Agent Pools

- 我們設置以下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





Microsoft STUDY4 Build School

Use Custom Agent Reason

- 加快Pull Code 速度
- 加快Build Container速度
- 避免.NET Core / .NET Framework 更新問題，導致舊系統無法Build
- Microsoft Agent 啟動速度有時太慢
- Release的環境在地端
- 省錢

Microsoft STUDY4 Build School

Agent Pool Design

- Build 和 Release Agent 要分開不同Pool
- 地端Build 和Release Agent服務帳號要不同
- 如果資源允許， Build 和 Release Agent要不同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

Microsoft STUDY4 Build School

Use Custom Agent Allocation

- 除非特殊需求才使用Private Premise的Build Agent
- 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

Microsoft STUDY4 Build School

Agent Version Challenge

- Agent 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，多半是Agent版本過舊

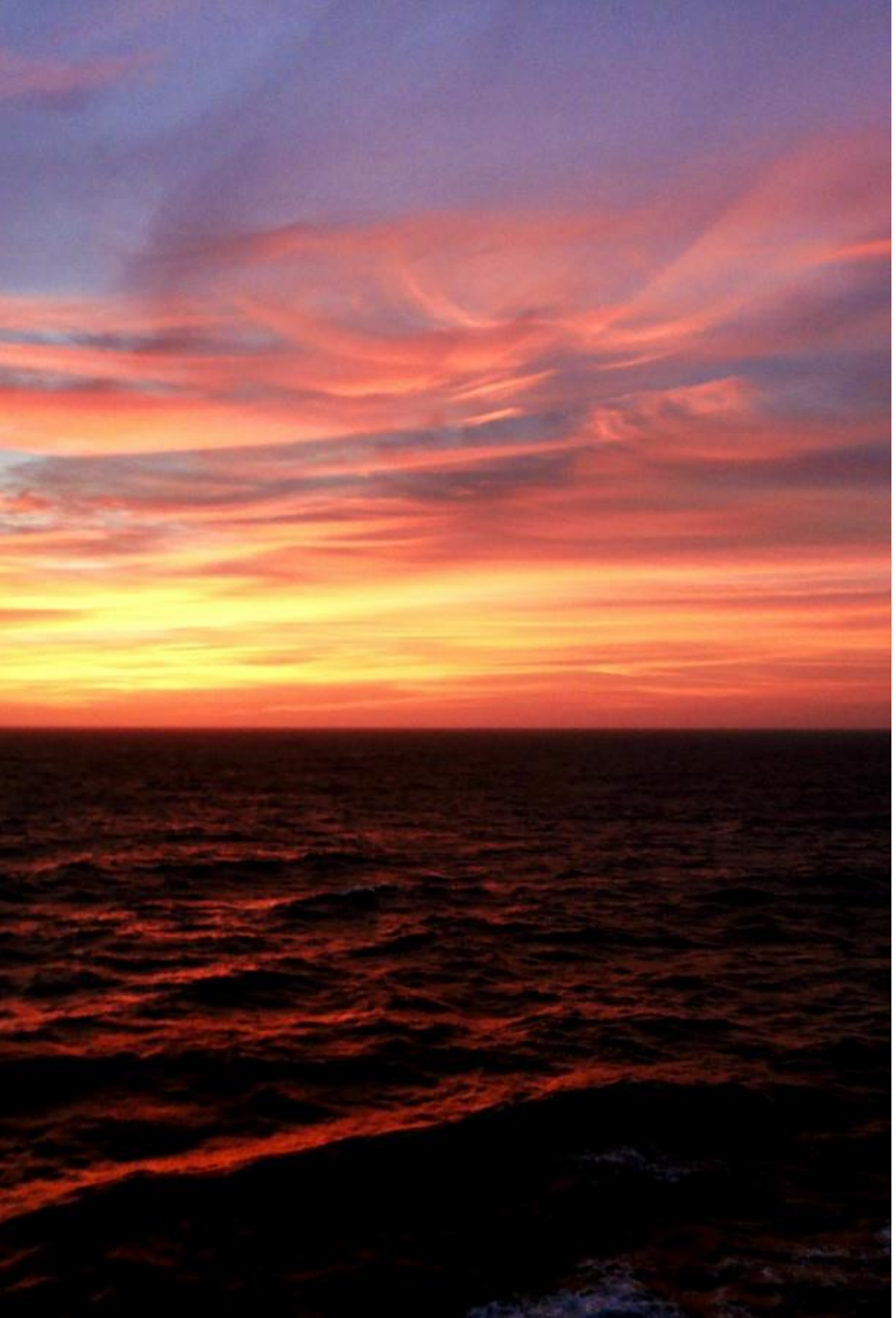
Microsoft STUDY4 Build School

Pipeline Practice

Microsoft STUDY4 Build School

Pipeline Design Patten

- 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OS相依性低
- CI 設計要能隨時可以切換Agent Pool

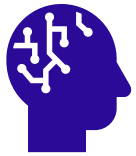


Microsoft STUDY4 Build School

Pipeline Design Patten

- Release 僅處理應用程式佈署或是進行環境佈署
- 敏感參數的安全性，用Library管理與置換
- Build Artifacts不要Keep在Local Host，一律用Upload方式Keep在Azure DevOps
- 即使一個PowerShell Script可以完成，也要區分Task處理

Pipeline Design Flow



Menu

手動執行模擬

Process

建立Task

Test

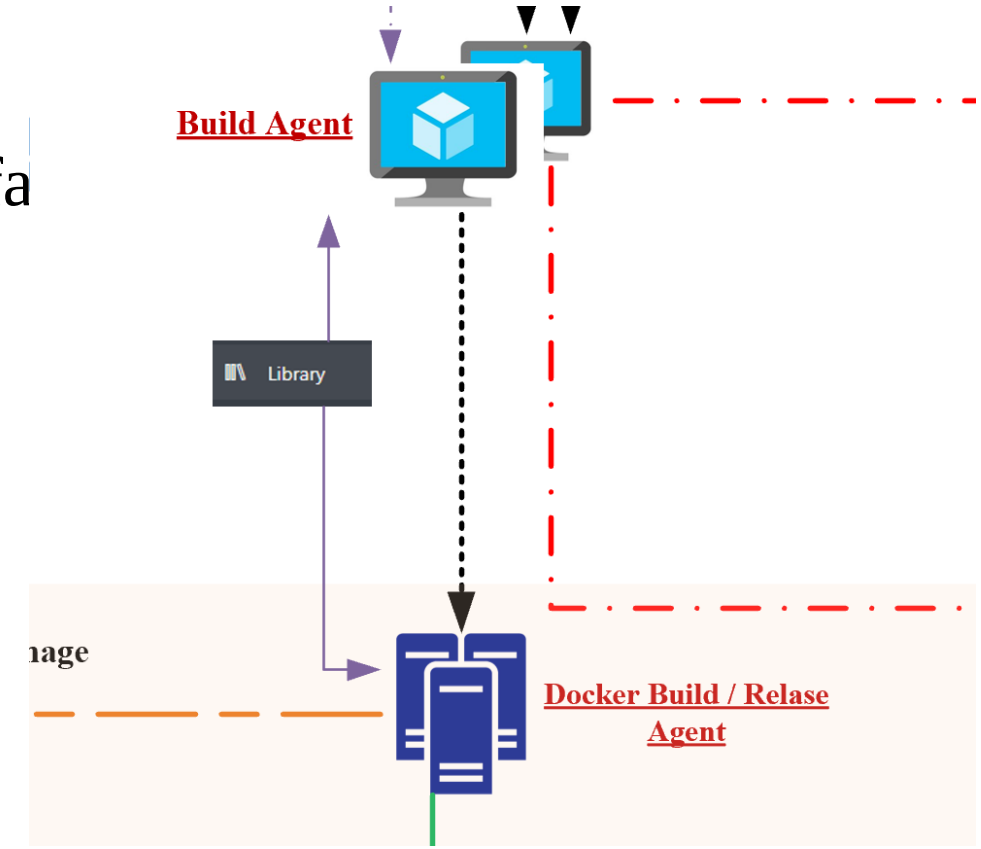
測試CI

Review

觀察執行時間

Build Common Rule

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



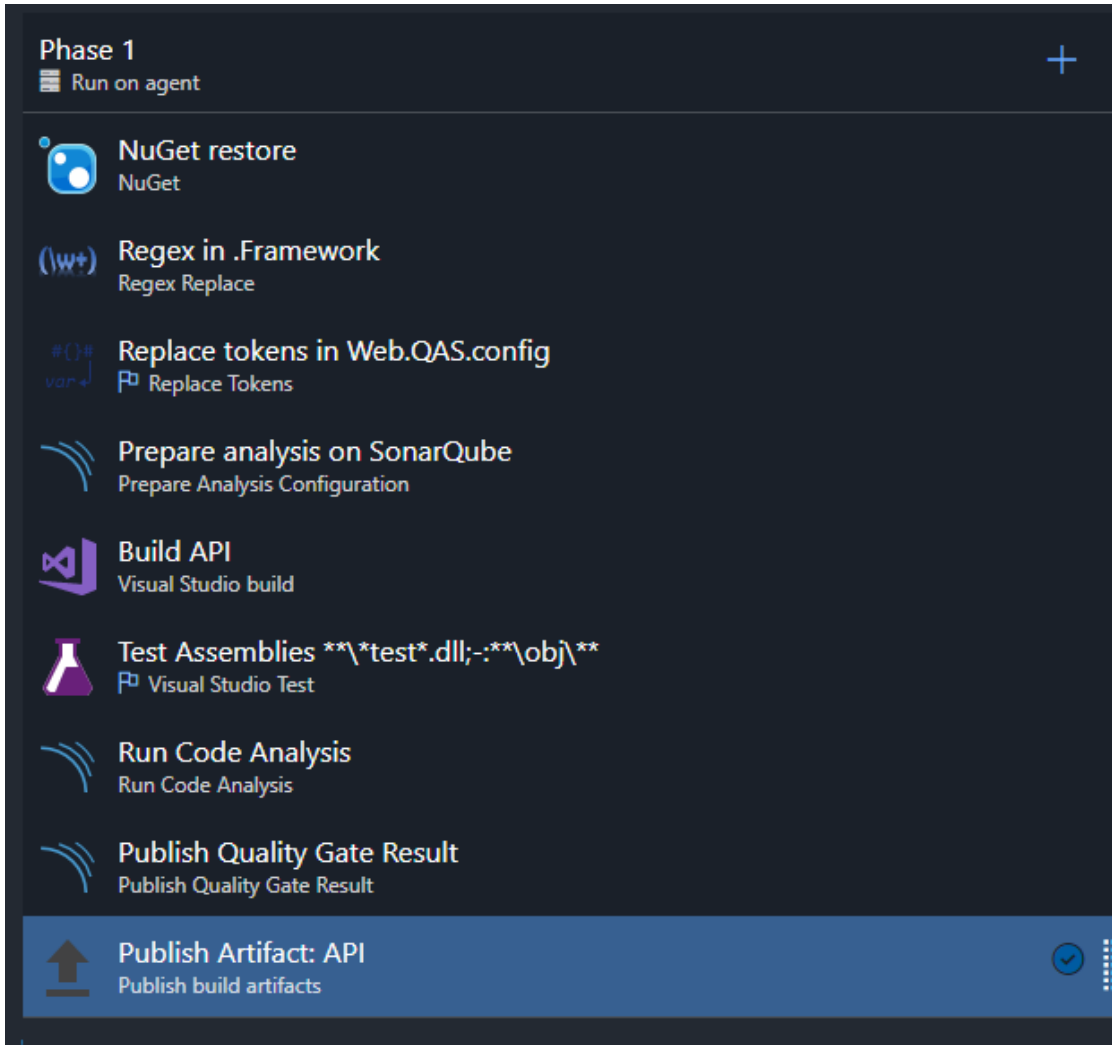
Build Type

- .NET Framework
  - ASP.NET MVC
  - API
  - Container
  - Console App
  - VSTO
- SQL Project
- Hexo Web Site

- .NET Core
  - Vue.js & ASP.NET Core
  - API
  - Container
  - Console App
  - Component
  - Azure Function
  - WPF

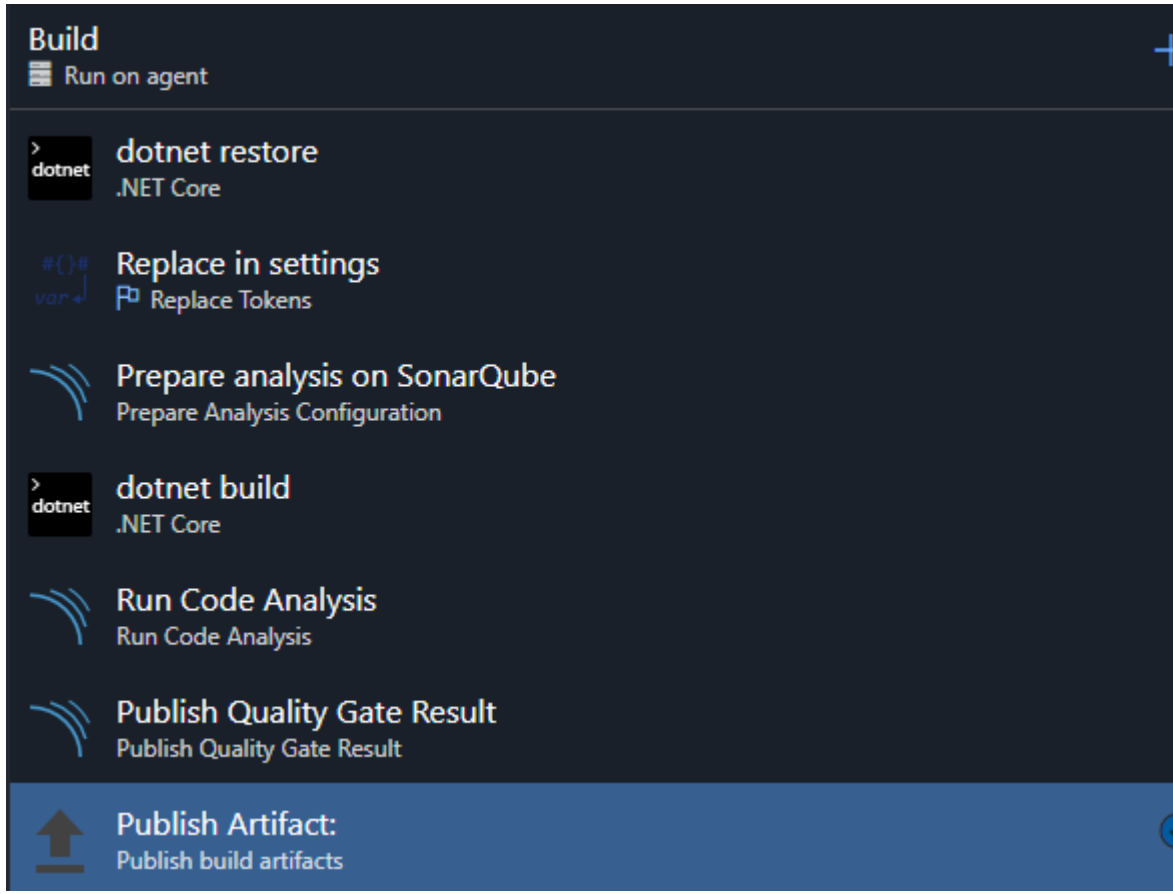
Build .NET Framework

- 順序
  - NuGet Retore : 版本過舊，可能會造成還原失敗
  - Regex : 自動更新AssemblyInfo內版本號
  - Replace Parameter : 更換Web.config的設定參數
  - SonarQube: Scan Code
  - MS Build : 編譯
  - UT : Unit Test
  - Publish Artifact : 上傳編譯好後的檔案
- 這裡並未封裝成ZIP檔案做部署



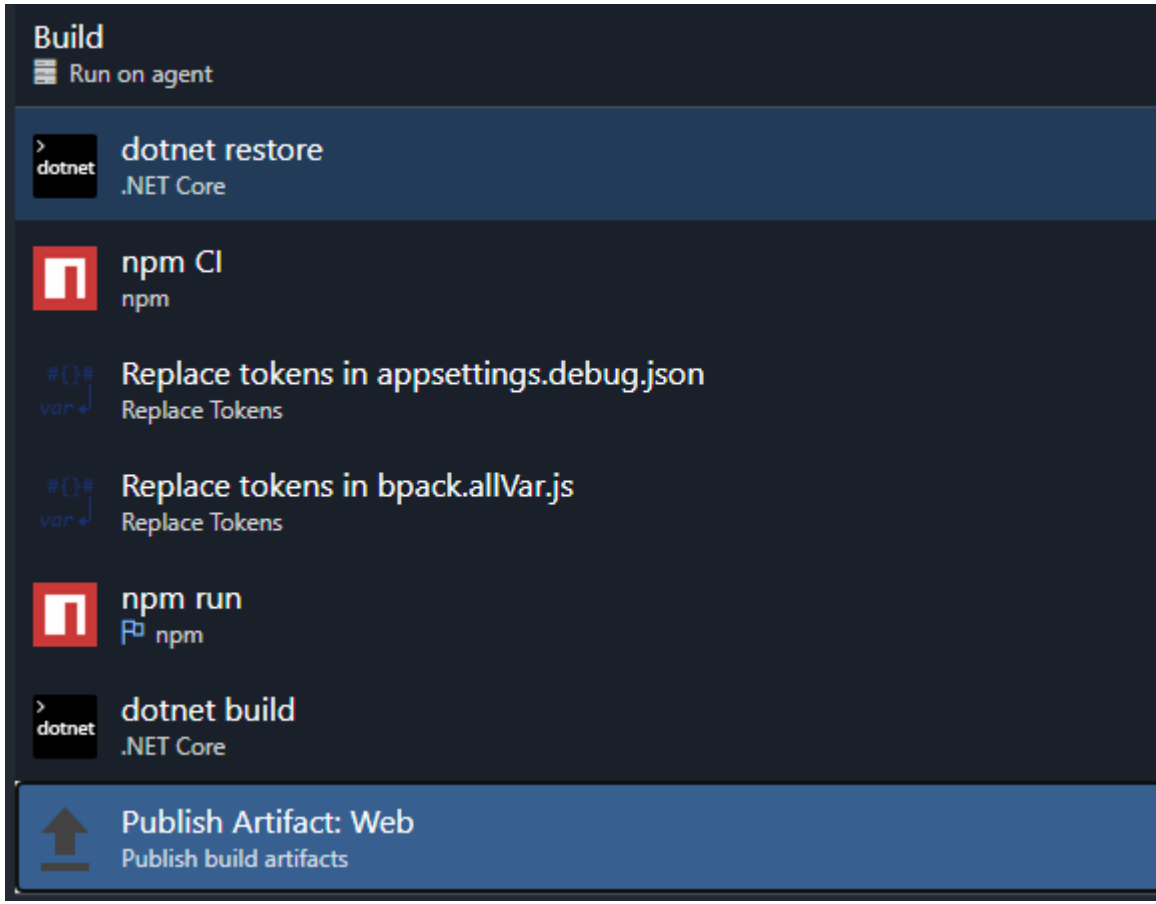
Build .NET Core

- 順序
  - .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
- 可以用ArchiveFiles@2，封裝檔案成ZIP File
- 由小到大版本，安裝.NET Core SDK



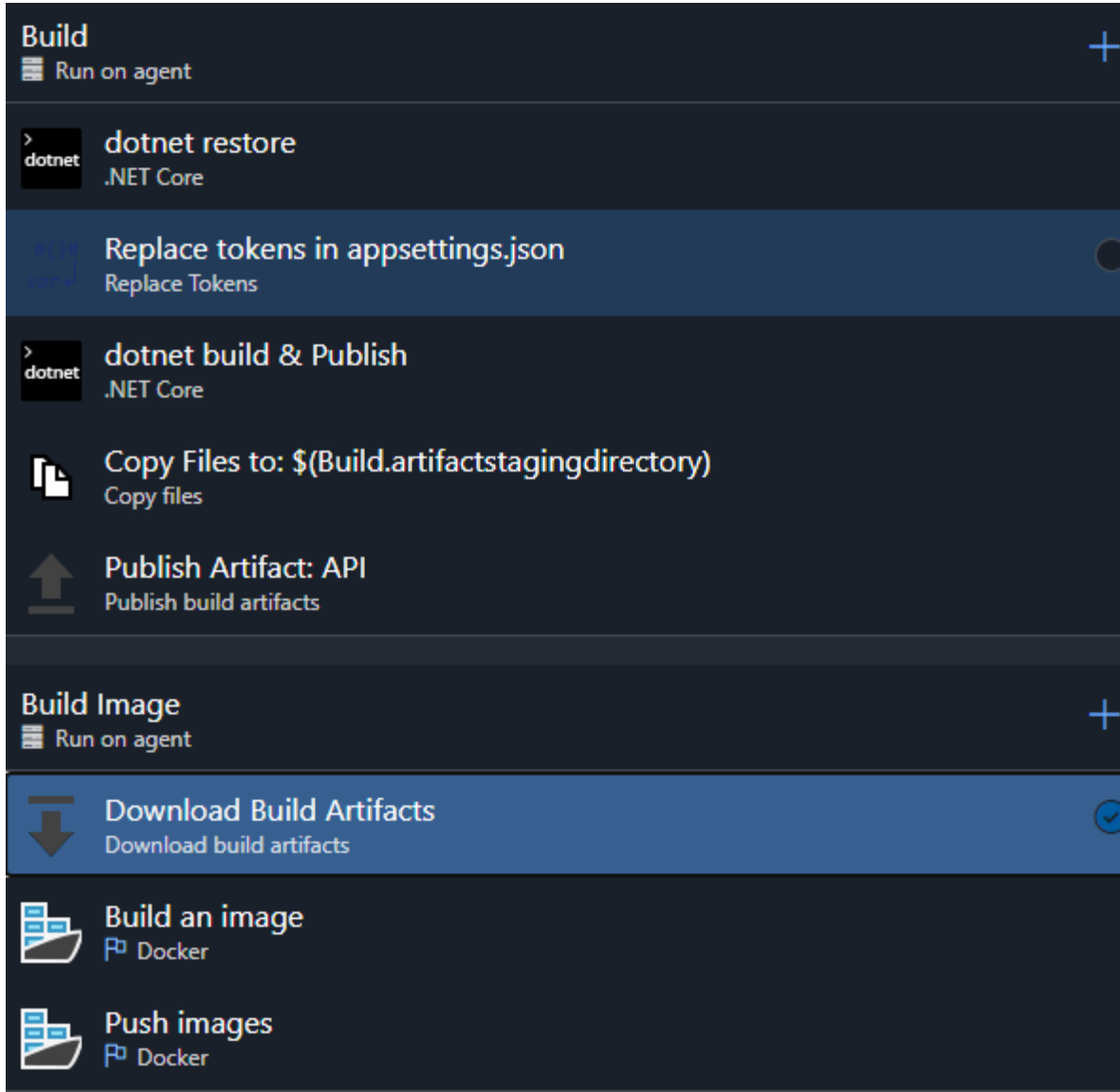
Build Vue.js & .NET Core

- 順序
  - .NET Core : 還原套件
  - Npm : 用CI Command還原npm套件
  - Replace Tokens : 置換Appsetting參數
  - Replace Tokens : 置換Webpack參數
  - Npm : 用Run Command建置前端程式
  - .NET Core : 編譯ASP .NET Core
  - Publish Artifact :上傳編譯後的檔案
- npm run 的參數要在package.json設定好



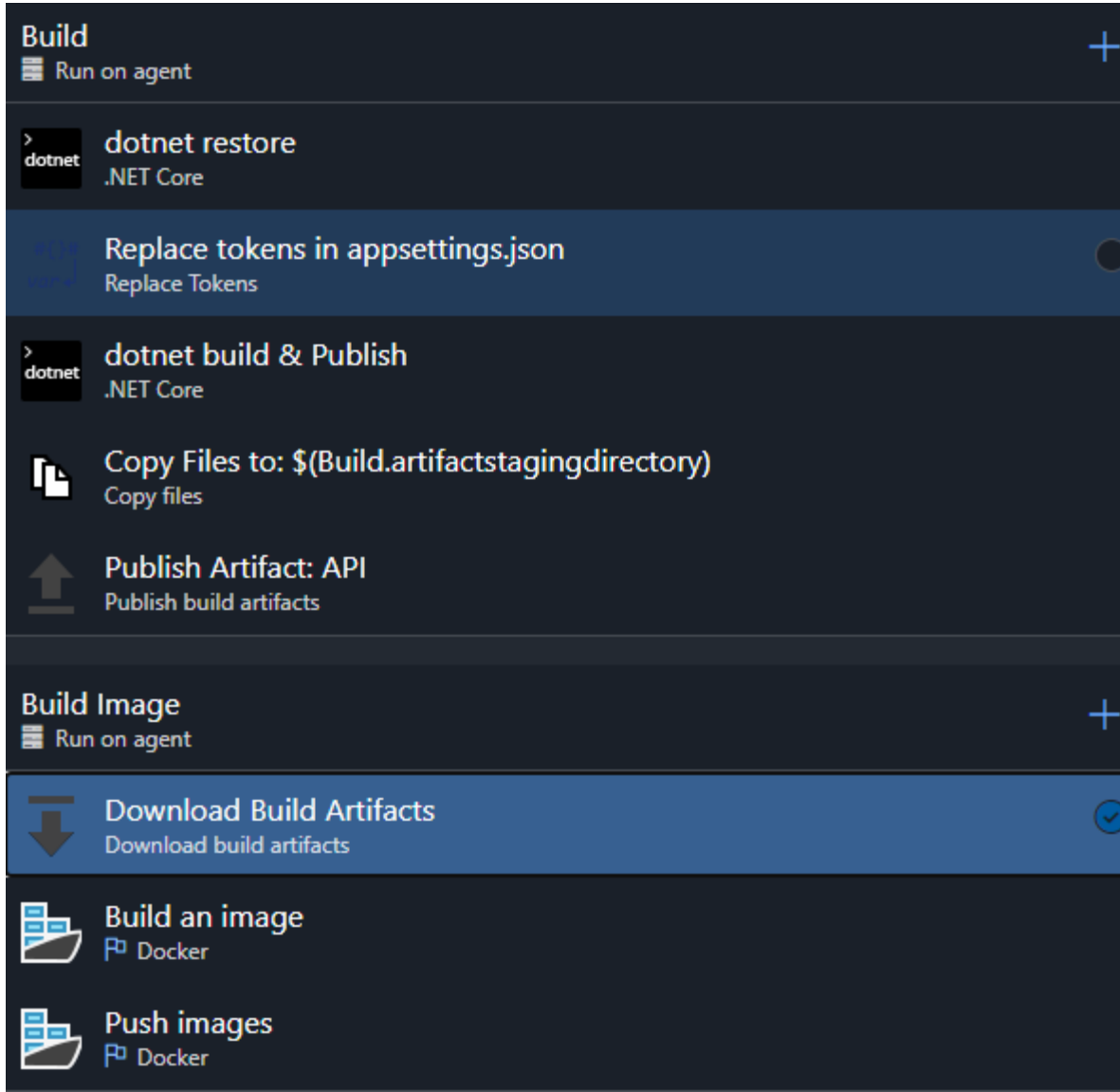
Build .NET Core & Container

- 順序
  - 與Build .NET Core 順序相同
  - Copy File : Copy Dockerfile 到Artifact資料夾
  - Publish Artifact :上傳編譯後的檔案和Dockerfile
  - Download build artifacts : 下載編譯後檔案和Dockerfile
  - Build Docker : 將檔案Copy到Container並編譯Docker
  - Push Docker : Push Container到ACR



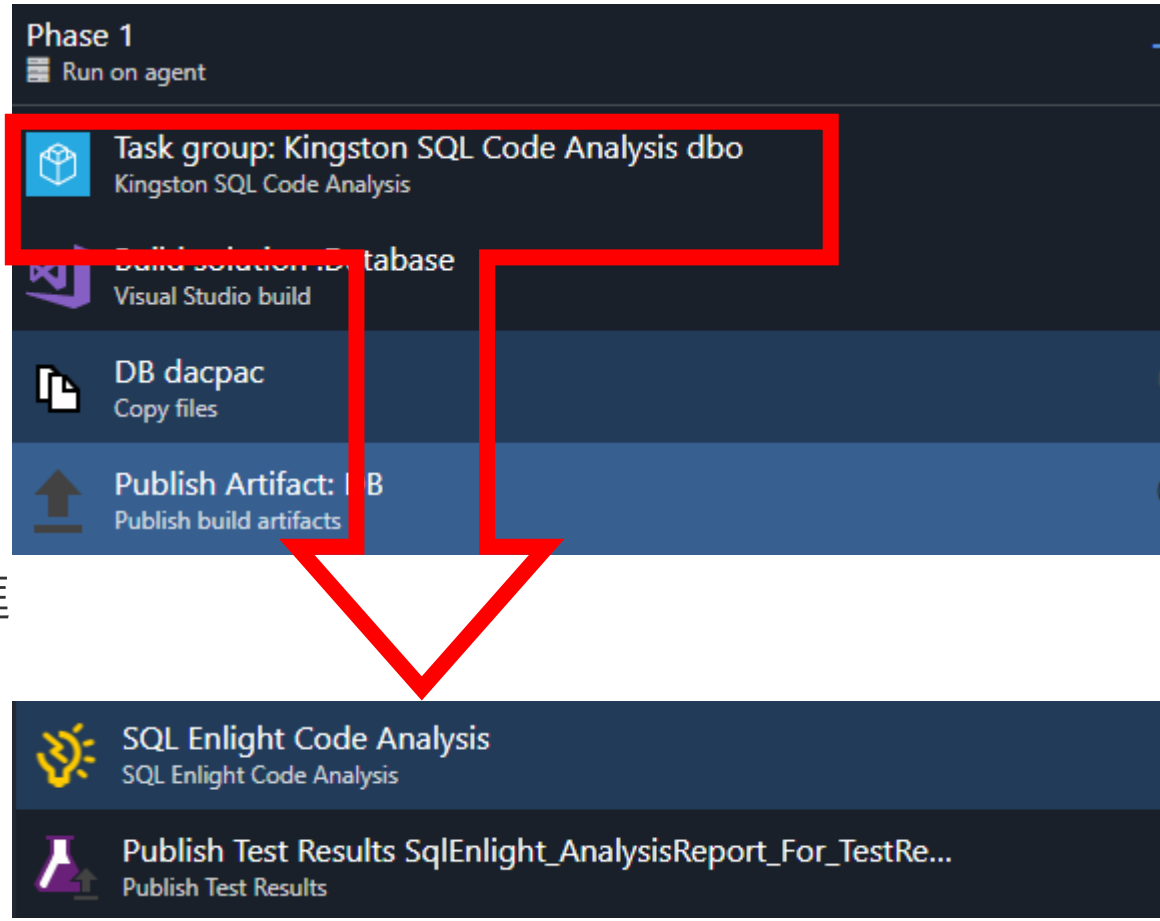
Build .NET Core & Container

- 分批建置Container Application
  - Application Agent
  - Docker Agent
- 不需要Container時候，可以單獨發布
- 能單獨針對編譯後的Application除錯
- Container編譯時候，帶入版本號



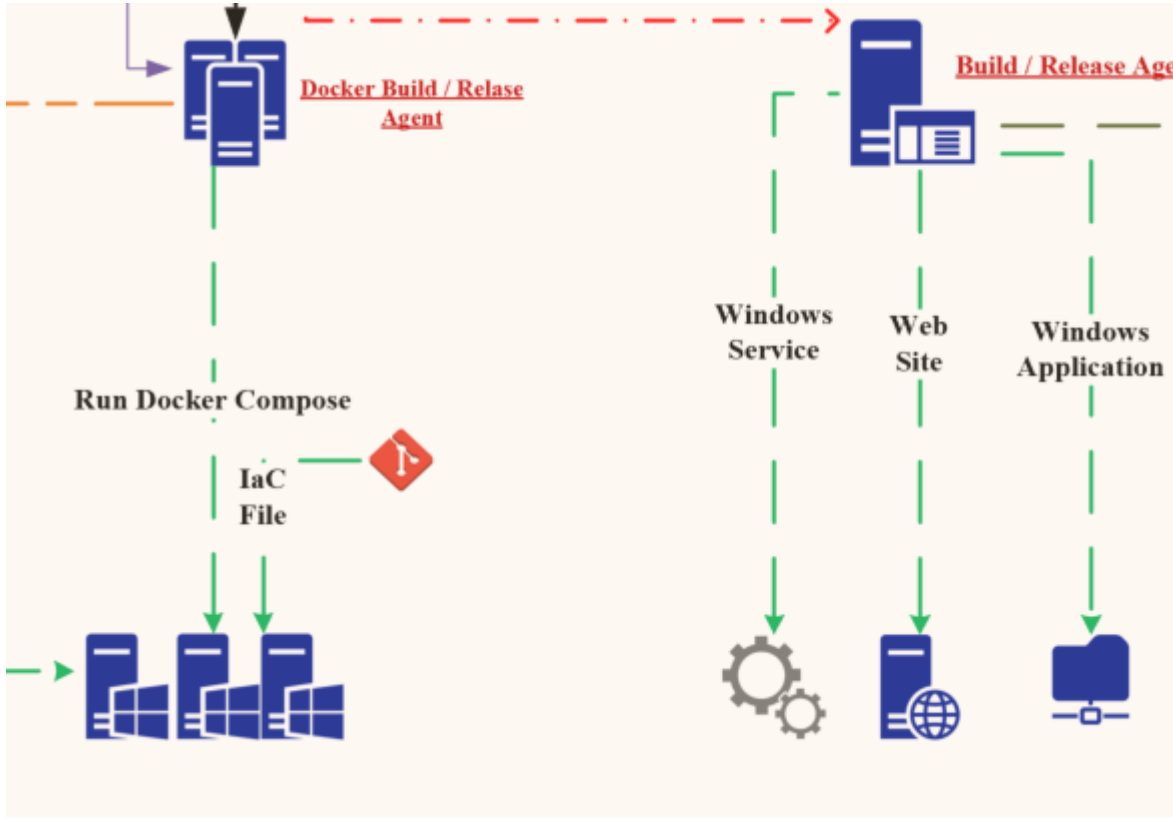
Build SQL Project

- 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資料夾
  - Publish Artifact :上傳編譯後的檔案
- 首先必須先用SQL Project，針對資料庫程式進行版控



Release Common Rule

- 所有編譯後的檔案，一律從Build Artifact取得
- 正式環境部署皆需要人員Approve
- 非Container的佈署，大多用Copy File方式處理
- 佈署用的參數也存放在Library
- Release Agent不放在被佈署的Host
- Compose File額外Repos管理

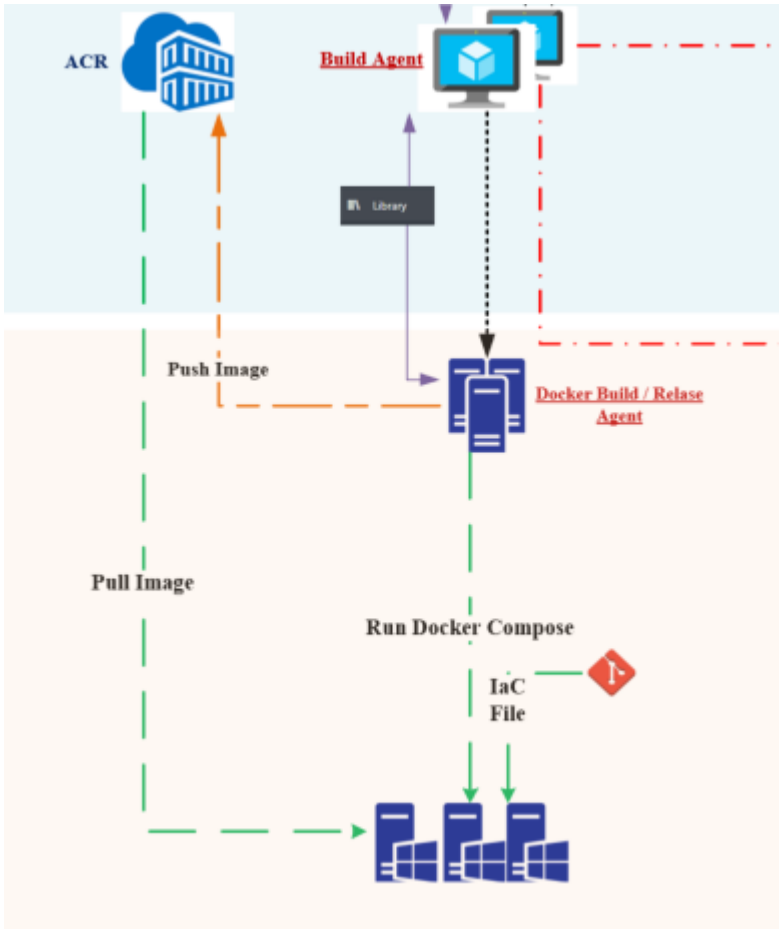


Release Type

- NuGet Package
- On-Premise
  - Container by Docker Host
  - Windows Service by Server & Single PC
  - Trigger FTP
  - Web Site
  - API
  - Single DLL File
  - MS SQL Database
  - VSTO Application
  - WPF / Windows Form Application
- Azure
  - Azure Web App Service
  - Azure Function
  - Azure SQL Database

Release Container

- 順序
  - 先從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啟動Container



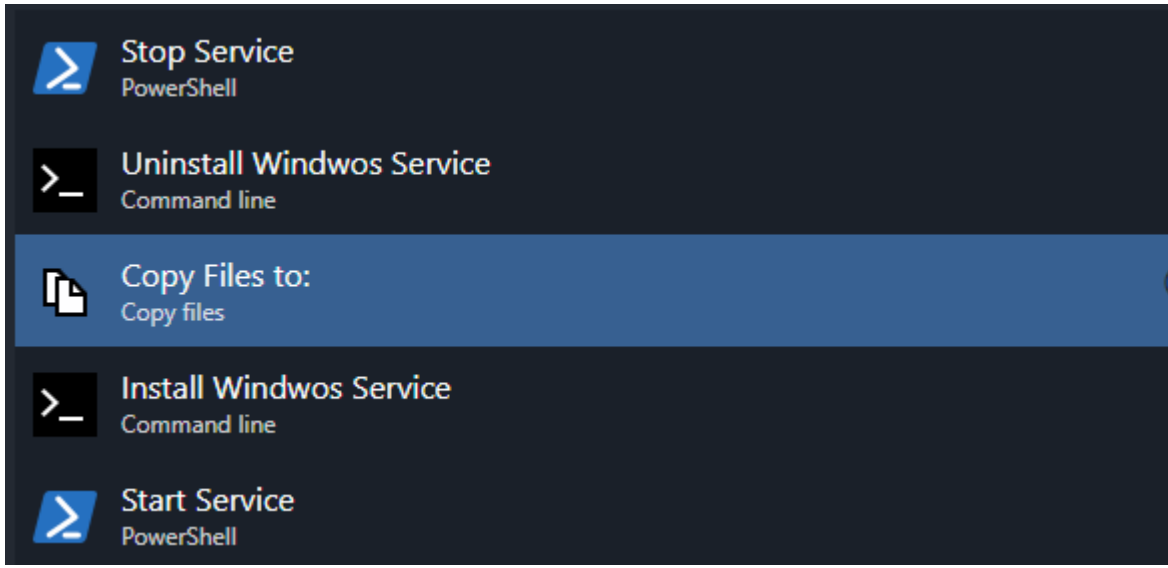
Release Container

- 每次Release的Container必須要有版號，才可以進行管理與快速復原
- 用$(Release.Artifacts.Pipeline Name.BuildNumber) 取得CI 版本
- 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"
```

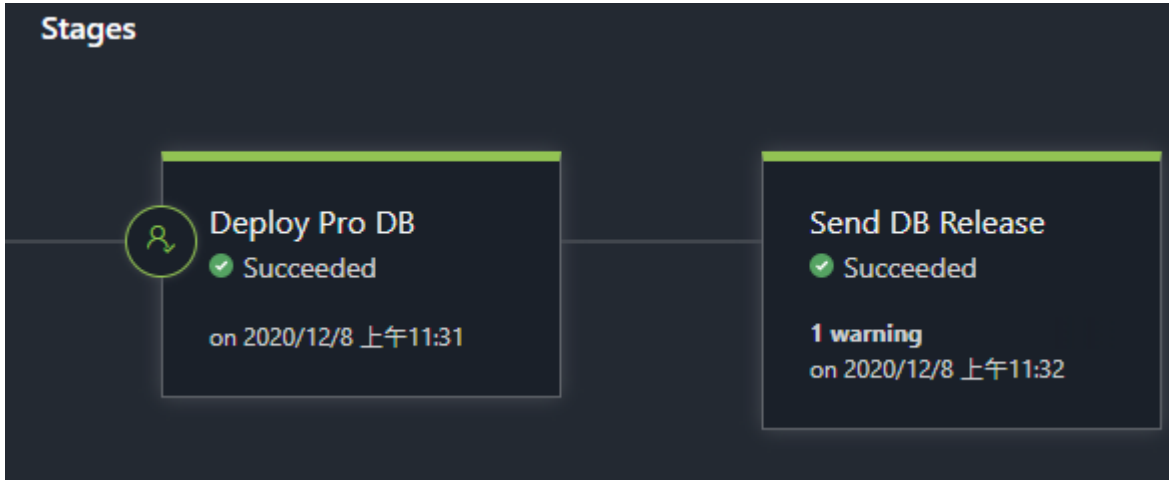
Release Windows Service

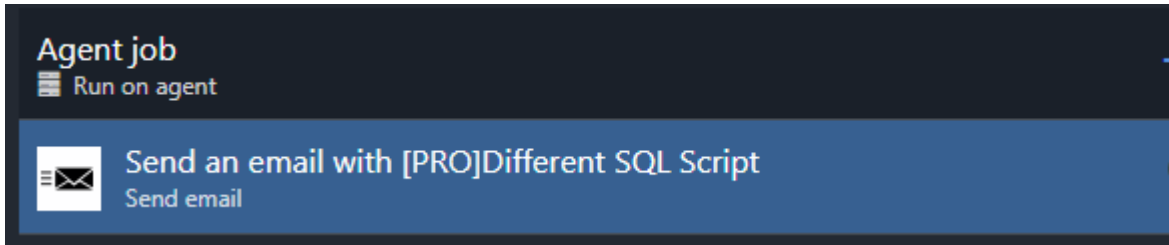
- 順序
  - PowerShell : 停止WS在遠距Host
  - Command Line : 卸載WS
  - Copy File : 複製新的Application
  - PowerShell : 安裝WS到遠距Host
  - Command Line : 啟動WS
- 目前並無專門Task，必須透過Command處理



Compare PRD Database

- 順序
  - Download Build Artifacts : 下載需要佈署DB相關的Script
  - 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
- 前提DB程式碼要用SQL Project做管理
- Agent Host要安裝sqlpackage.exe





Compare PRD Database

- 差異化比較屬性，用DB.Publish.xml管理
- 執行差異化比較的Script，用PowerShell執行
- 在PowerShell內執行DB.Publish.xml

```
#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+'
$SQLPackagePath=$SQLPackagePath+' --% '+$SQLPackageCmd+' '+$CompareC

"Start Generate Release SQL Script file"

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

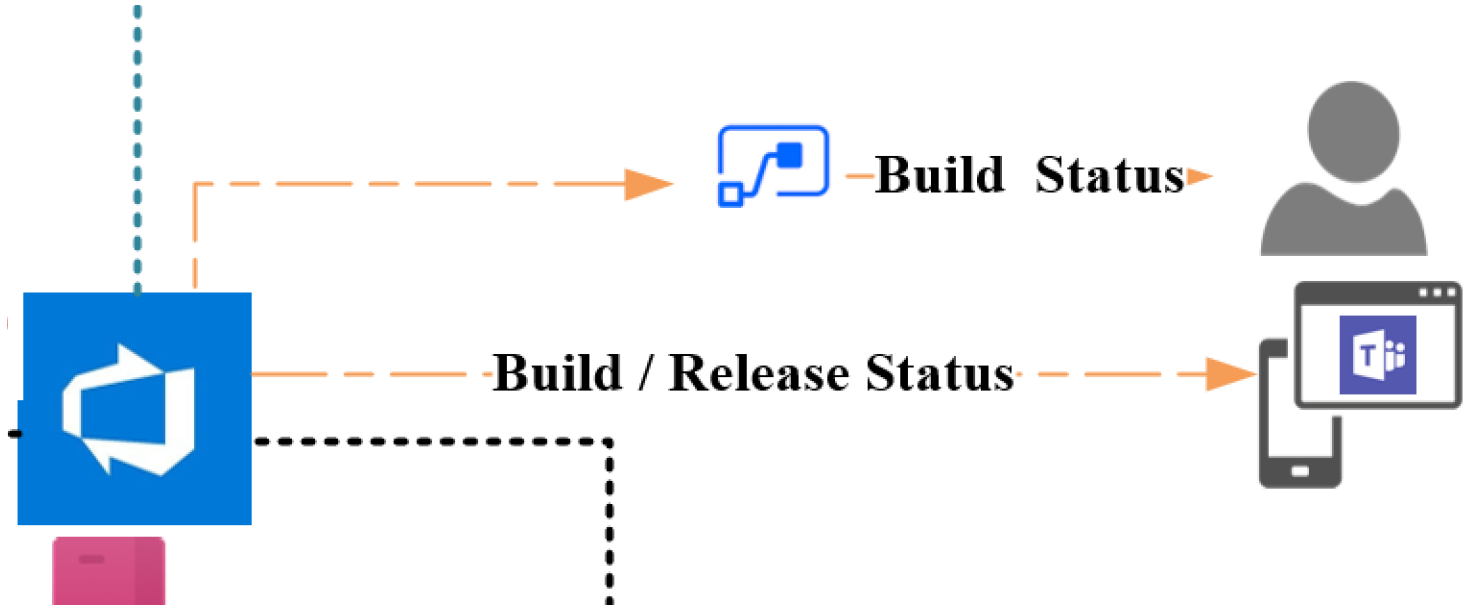
如果可以，就讓Pipeline做自動化的平台

Microsoft STUDY4 Build School

Azure DevOps Notifications

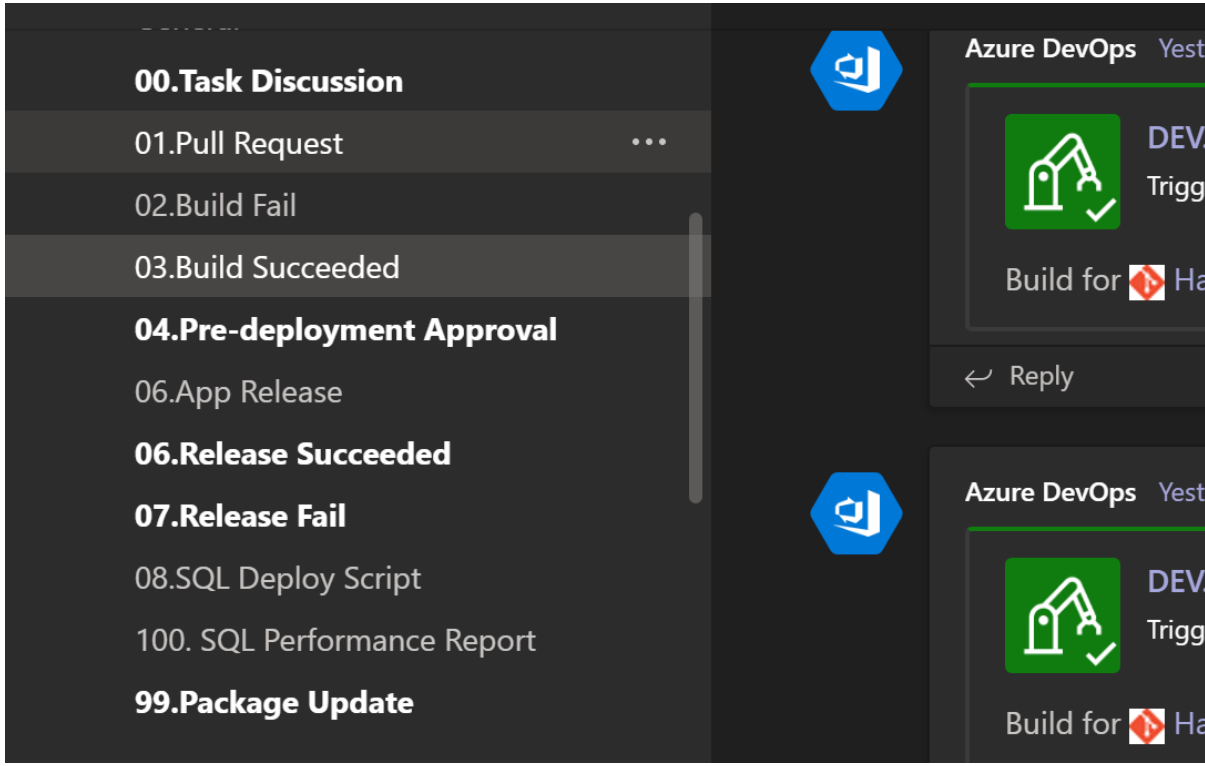
Build & Release Notifications

- 所有Build / Release 的資訊都要通知到團隊頻道
- 訊息發送到Microsoft Teams
  - Azure DevOps Connector
    - Webhook方式傳送
  - Azure Pipelines
    - Azure DevOps bot



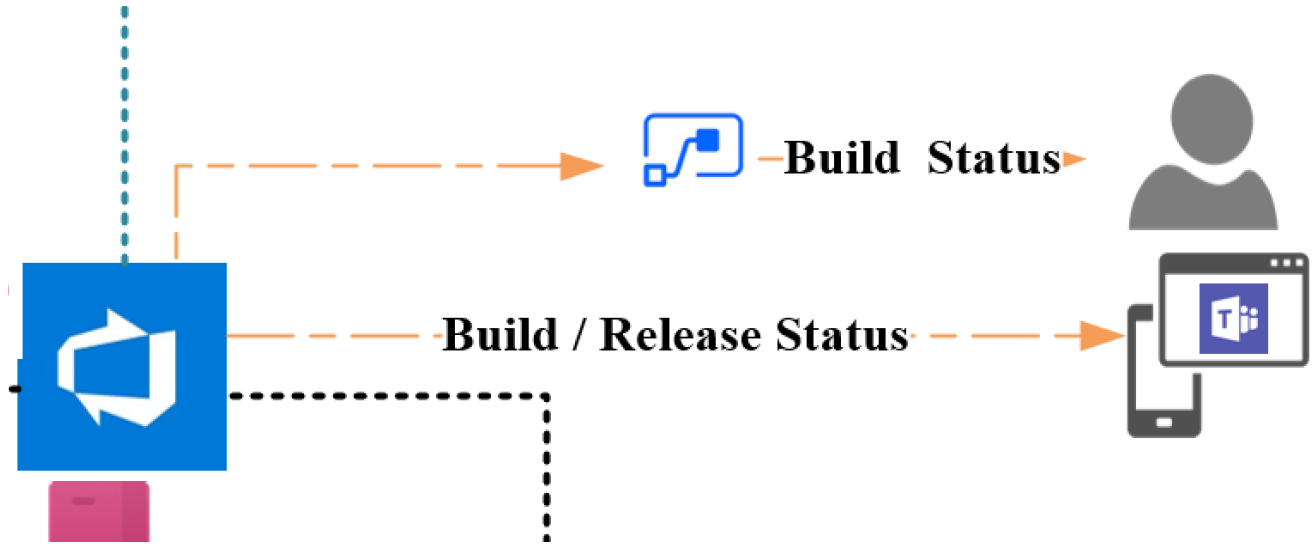
Build & Release Notifications

- 所有Build / Release 的資訊都要通知到團隊頻道
- 訊息發送到Microsoft Teams
  - Azure DevOps Connector
    - Webhook方式傳送
  - Azure Pipelines
    - Azure DevOps bot



Build & Release Notifications

- Advance
  - Build Fail 通知到個人 Teams Chat
  - Service Hooks + Power Automate



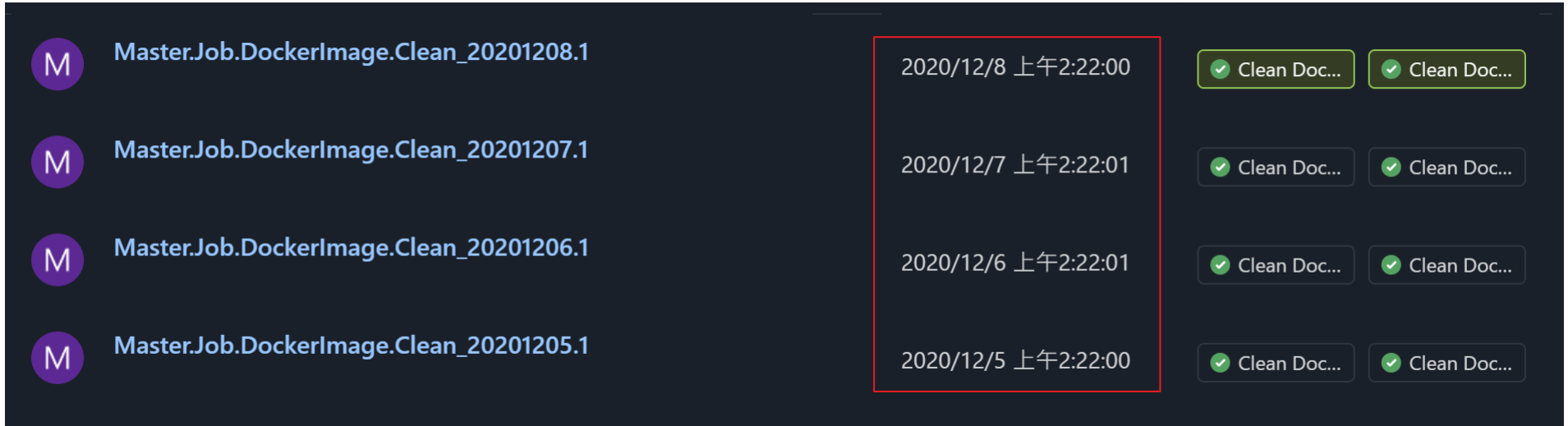
Microsoft STUDY4 Build School

另類使用方式

非關DevOps，使用模式僅參考，因此，使用前要深思

Schedule Controller

- 透過Release定時驅動功能，作為系統 / 維運 的排程器
  - 不需要使用Windows Schedule
  - 有完整歷程記錄
  - 可以設定執行排程的流程
  - 如果用Azure DevOps Service，還可以雲端手動啟動地端排程



Backup All Repos File

- 基於ISO需求，需要將雲端 Repos File 備份到地端
  - 用PowerShell & Git 指令
  - 呼叫Azure DevOps API
  - 搭配Release排程功能

```
$PATToken="$(PAT)"

$base64AuthInfo=
[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.
GetBytes(":$($PATToken)"))

$AzureDevopsOrg=“Org"

$AzureDevOpsProject=“Name"

$LocalfilePath="$(BackupFilePath)"

$ProjectUrlAPI =
"https://dev.azure.com/$($AzureDevopsOrg)/$($AzureDevOpsPr
oject)/_apis/git/repositories?api-version=6.1-preview.1"

$Repo = (Invoke-RestMethod -Uri $ProjectUrlAPI -Method Get -
UseDefaultCredential -Headers @{Authorization=("Basic {0}" -f
$base64AuthInfo)})

$RepoName= $Repo.value.name
```

Backup All Repos File

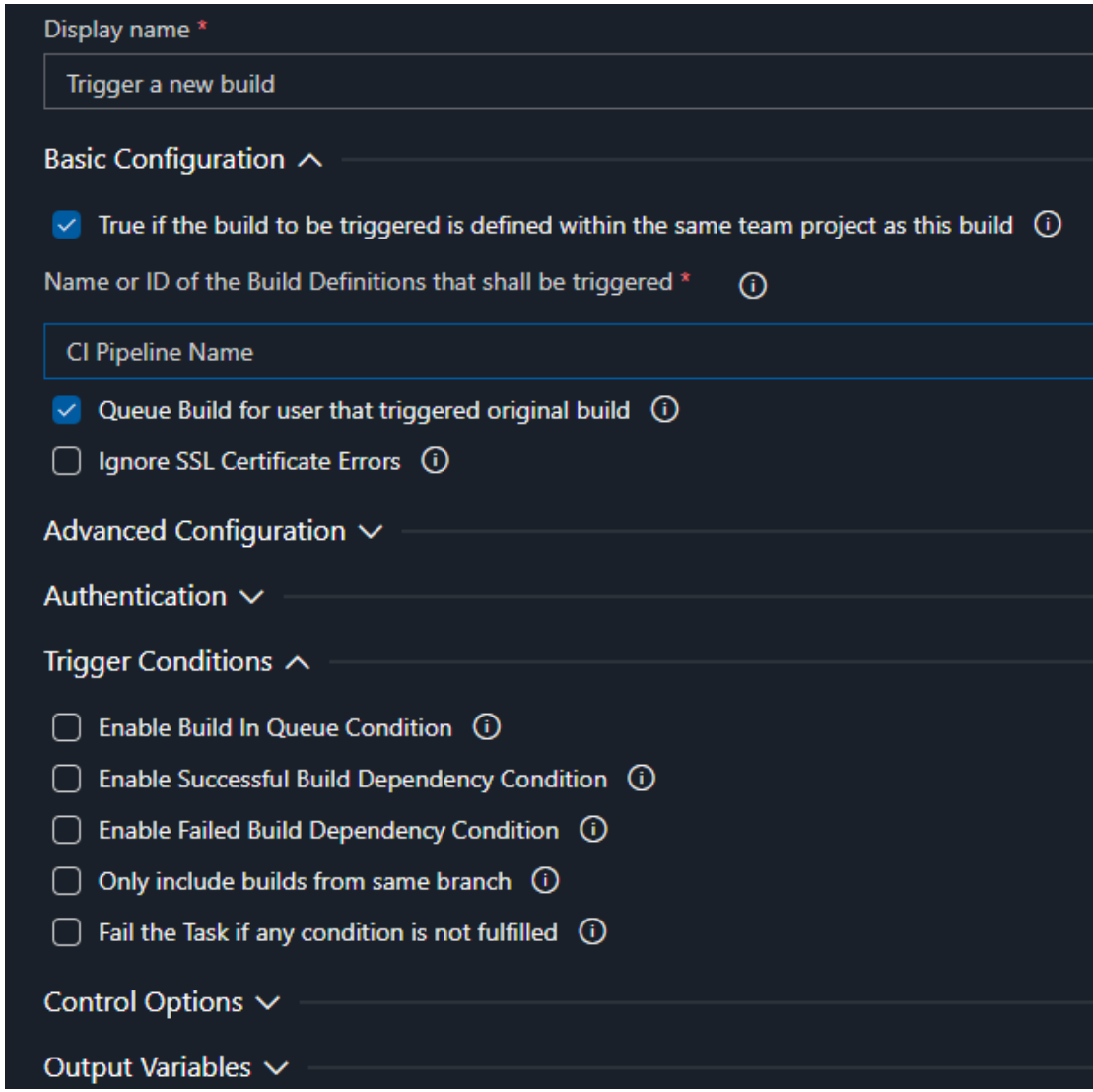
- 基於ISO需求，需要將雲端 Repos File 備份到地端
  - 用PowerShell & Git 指令
  - 呼叫Azure DevOps API
  - 搭配Release排程功能

```
ForEach ($name in $RepoName)
{

$ReposUrl="https://anything:$($PATToken)@dev.azure.com/$($AzureDevopsOrg)
/$($AzureDevOpsProject.Replace(" ","%20"))/_git/$($name.replace(" ","%20"))"
    $FilePath="$($LocalfilePath)\$($name)"
     if(Test-Path $FilePath)
    {
        cd  $FilePath
        write $ReposUrl
        git pull origin UAT
        git pull origin master
        }
    else {
        write $ReposUrl
        git clone $ReposUrl $FilePath -b master
    }
}
```

Trigger CI on Release Stage

- 當某個重要Package被Release後，必須自動驅動某系統的CI，讓系統重新編譯取得最新的Package 版本



Trigger CI on Release Stage

- 當某個重要Package被Release後，必須自動驅動某系統的CI，讓系統重新編譯取得最新的Package 版本
- CI 必須要在NuGet Package添加取得最新版本Package

Use NuGet Task

Command : Custom

Command and arguments:

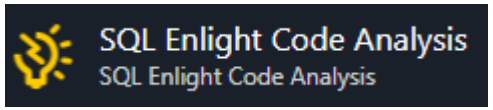
update “Project.sln" -Id “NuGet Package Name" -ConfigFile "nuget.config"

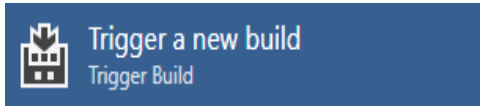
Microsoft STUDY4 Build School

常用的Marketplace套件

Often Extension

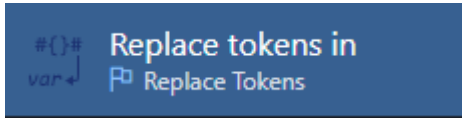
- SQL Enlight Code Quality Task
  - 掃描SQL語法是否有符合SQL撰寫規範
- SonarQube Task
  - 整合SonarQube服務，進行Code掃描
- Trigger Build Task
  - 在現有Pipeline 流程，驅動其他CI Pipeline流程

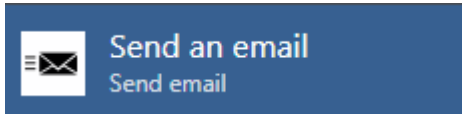




Often Extension

- Regex Replace Task
  - 用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Library使用
- Send Mail Task





Summary

Our Challenge

- 不容易明白的錯誤訊息
- 介面與功能常常改變
- 網路穩定性
- 不一定能Work的Preview功能
- 有部分功能可能沒有，但是在.NET世界，基本上都具備
- Pipeline Task版本更新或是Deprecated
- Pipeline的管理
- Pipeline 失敗不一定是Application問題

Use YAML

- 使用YAML
  - CI 使用YAML，可以讓有相同CI流程系統，可以快速建立Pipeline
  - YAML讓Pipeline具有版控
  - YAML支援CD，但功能尚不完善，不建議使用
  - 可以離線編輯Pipeline
- 挑戰
  - 語法需要熟悉，遇到第三方Task的YAML，不容易撰寫
  - 沒有可視化流程
  - 除非很熟悉，不然需要比對文件才知道Task屬性

Azure DevOps CLI & API

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

Microsoft STUDY4 Build School

Azure DevOps 可以是DevOps工具
也可以是一個應用平台

Reference

- https://docs.microsoft.com/en-us/azure/devops/cli/?view=azure-devops
- https://docs.microsoft.com/en-us/rest/api/azure/devops/?view=azure-devops-rest-6.1
- https://github.com/edwardkuo/AzureDevOpsPipelineSample

Microsoft STUDY4 Build School

Thanks for joining!

Ask questions on Twitter using #dotNETConf

.NET Conf 2020

特別感謝

91APP Technical Network

Red Hat

KKTIX

JET BRAINS

HackMD

MVP Microsoft® Most Valuable Professional

Microsoft

Build School

STUDY4 為學習而生

以及各位參與活動的你們