

- 一般如果開發人員或是網管人員在操作電腦時大部份是會直接 ssh 進去，或是 windows 的話就是透過遠端桌面，但：
 - Windows 同一個帳號似乎一次只能一個人登入
 - 能做什麼事，怎麼做，通常**沒有文件**
 - 如果同時要對多台機器作同樣的操作，就要瘋狂的**登入登出**，並下 bash 指令或是 cmd
 - 有沒有勾起誰的回憶呢？
- 如果電腦沒幾台，而且文件齊全的話通常這樣是沒問題的，但：
 - 大部份的實際情況都是**極為複雜的**
 - 如果網管人員要建立一台**一模一樣環境**的電腦，他必須花很長的時間才能知道：
 - **什麼套件**被安裝了，裝了什麼版本，各有什麼設定
 - 記錄並更新這些相關的文件
 - **手動**或寫腳本移除、安裝、更新各軟體
 - **寫腳本**似乎是一個很常見的解，但：
 - 必須考慮到所有的情境，而且通常漏考慮了就掛了
 - 此外，同時要在上百上千台電腦跑腳本，除非是排程類型的，不然光是執行就夠累人的
- 恭禧你活在 2021 年，因為這樣的工具 2000 年左右就出現了，比較有名的像是 CFEngine、Puppet、Chef, TerraForm ... 等
 - 上面的工具主要是在管理、維護組態的
 - 什麼是組態？英文就是 Configuration
 - 其實就是整個電腦環境目前應該有的狀態，或是做什麼樣的動作時會吃什麼參數、先後順序... 等等
 - 但為什麼要放棄腳本？
 - 腳本簡單快速方便
 - 用了這麼多年，已經累積一定的量
 - 究竟有什麼吸引人的點要使用 CM(Configuration Management)？
- 向各位介紹 Ansible
 - 它是由懂腳本指令的網管及工程師開發出來的一套 CM
 - 以往透過腳本可以做到的事，Ansible 都能達到，且更趨向中央化，可重覆性
 - 你還是手癢想寫腳本？ Ansible 還是可以讓你寫，並且有條理的寫成**劇本**(Playbook)，讓每一次的執行結果都會維持在定義好的**狀態**
 - 其他像是 Puppet、Chef...等都需要會一點它們特有的 DSL，或是一些程式語言，像是 Ruby
 - Ansible 可以一次性向所有在你管控之下的機器統一同步發出指令操作之，而且不需要特邊安裝什麼 Client 端的介面軟體，只要能透過 ssh 溝通即可
 - 它執行的結果是等幕(Idempotent)的
 - 通常某些腳本執行一次以上就可能發生一些問題
 - 事實上絕大多數的 Ansible 指令或模組都是等幕的，就算不是，它也能讓你定義：
 - 何時該執行什麼
 - 怎樣算是發生錯誤或是變更
- 以上我知道，聽起來很玄 ... 沒錯，這就是 DevOps(X)
 - 我也是第一次碰 Ansible，但一用就愛上它了
 - 我再也不用管 Linux shell 怎麼寫，Windows Powershell 或是 batch command 要怎麼寫出對應的
 - 我不用再 ssh 進去每一台機器，一開始就會有一個地方可以針對每台機器(IP/Domain)定義、分群、額外設定參數...等
 - 我本人目前也不熟 Ansible，但我什麼不會就會看書，並整理一些重點：
 - 讓大家有空能跟著一起學 Ansible
 - 可以思考怎樣讓 Ansible 能搭配既有的 CI/CD 流程
- 下回預告: 安裝
 - 啊不是說不用安裝？
 - 沒錯...要操控的電腦不用(除非是 Windows)

- 安裝也不會太複雜，只要有 Linux 或 Mac，會相對容易一點，Windows 我也沒試過，但就做看唄