0110

看起來你的研究範圍涵蓋了幾個有趣且實用的技術主題，包括 Docker、Shell（可能是 Bash 或其他）、Putty、API，以及 N8n（用於工作流程自動化的工具）。以下是這些技術的簡要介紹與建議學習方向 (**嘗試**)：

1. Docker (目前安裝到虛擬機上(還沒使用))

Docker 是一個容器化平台，可以讓你在不同的環境中執行一致的應用程式。  
學習建議：

* 基礎指令：學習如何使用 docker run、docker build、docker ps、docker stop 等指令。
* Dockerfile：了解如何撰寫 Dockerfile 來建立自定義映像。
* 應用場景：嘗試在容器中部署一個簡單的應用程式（如 Node.js 或 Python Web App）。

2. Shell (查定義) (Putty (設置金鑰))

Shell 是與作業系統溝通的工具，Putty 是一個常用的 SSH 工具，用來遠端連接伺服器。  
學習建議：

* SSH 基礎：學習如何使用 Putty 登入伺服器，設定金鑰驗證。
* Shell Script：嘗試撰寫一些基礎 Shell Script，自動化常見工作（如檔案備份、系統監控）。
* 權限與安全：了解如何管理 Linux 文件權限（如 chmod、chown）。

3. API

API（應用程式介面）是軟體系統之間的溝通橋樑。  
學習建議：

* RESTful API：理解 HTTP 方法（GET、POST、PUT、DELETE）的用途。
* 工具學習：使用工具（如 Postman 或 cURL）測試 API。
* 實作 API：嘗試用一個後端框架（如 Express.js 或 Flask）建立簡單的 API。

4. N8n (預計使用 Docker 安裝 N8n)

N8n 是一個無程式碼的自動化工作流工具，支援多種 API 整合。  
學習建議：

* 安裝與設定：學習如何使用 Docker 安裝 N8n。
* 建立工作流：嘗試設計一個簡單的工作流，例如將表單提交的資料自動發送到 Google Sheets。
* 進階整合：研究如何與其他服務（如 Slack、Webhook、REST API）進行整合。

組合應用建議

試著將以上技術整合起來，設計一個小專案：

* 使用 Docker 部署 N8n。
* 透過 API 取得外部數據（例如天氣資訊）。
* 使用 N8n 自動化流程，將數據以 Shell 腳本儲存到伺服器上。

Shell / SSH

#### **Shell 功能**：

* 本地執行命令。
* 支持**腳本和批處理**，實現自動化工作流程。
* 管理和操作本地系統的資源與設置。

#### **SSH 功能**：

* **遠程安全登錄**到其他系統（例如伺服器）。
* 遠程執行命令並管理設備。
* 支持文件傳輸和端口轉發，增強跨網絡功能。

### **1. Shell 概述**

#### **定義：**

Shell 是一種命令行介面（CLI），作為用戶與操作系統之間的橋樑。它負責解釋用戶輸入的命令並在系統中執行。

#### **主要功能：**

* **命令執行：** 直接運行用戶命令。
* **腳本支持：** 通過腳本自動化任務。
* **環境管理：** 管理環境變數和系統設置。
* **任務控制：** 處理運行、暫停或停止進程。
* **管道與重定向：** 組合多個命令並重定向輸入/輸出。

#### **常見 Shell 類型：**

1. **Bash（Bourne Again Shell）：** 廣泛使用，高兼容性。
2. **Zsh（Z Shell）：** 功能豐富，支持插件。
3. **Sh（Bourne Shell）：** 基本的 Shell，Bash 的前身。
4. **Fish（Friendly Interactive Shell）：** 使用者友好，具現代化功能。

* linux shell即bash，和windows cmd區別？

shell是一個命令直譯器(也是一種應用程式)，處於核心和使用者之間，負責把使用者的指令傳遞給核心並且把執行結果回顯給使用者，同時，shell也可以作為一門強大的程式設計語言。在linux/unix平台上，shell多半預設為Bash shell。cmd是Command shell的簡寫，微軟的定義是： CommandShell是獨立的應用程式，它為使用者提供對作業系統直接通訊的功能，它為基於字元的應用程式和工具提供了非圖形介面的運行環境，它執行命令並在螢幕上回顯MS-DOS風格的字元。所以，可以近似地認為linux shell=bash而windows=cmd，都是命令列解譯器，都是使用者與作業系統的互動介面。但是bash要比cmd強大很多，windows也有強大的shell：windows power shell。

Script:

### **2. SSH（Secure Shell）**

#### **定義：**

SSH 是一種安全協議，用於客戶端與遠程服務器之間的加密通信，通常用於遠程登錄、文件傳輸和命令執行。

#### **核心功能：**

1. **安全的遠程登錄：**
   * 語法：ssh username@hostname (-p num: 使用port num default 22)
2. **文件傳輸：**
   * 複製文件：scp file.txt username@hostname:/path/
   * 交互式傳輸：sftp username@hostname
   * Sync: rsync -avz /local/path/ username@hostname:/remote/path/
3. **端口轉發：**
   * 本地：ssh -L local\_port:remote\_host:remote\_port username@hostname
   * 遠程：ssh -R remote\_port:local\_host:local\_port username@hostname
4. **遠程命令執行：**
   * 範例：ssh username@hostname "ls -l /path/"
5. **基於密鑰的驗證：**
   * 生成密鑰：ssh-keygen -t rsa -b 4096
   * 傳輸公鑰：ssh-copy-id username@hostname
   * 設置完成後無需密碼即可登錄。

修改 ~/.ssh/config 來使用別名(alia)和簡化操作：

Host myserver (alia)

HostName example.com

User username

Port 22

IdentityFile ~/.ssh/id\_rsa

通過以下命令登錄：ssh myserver

### **3. 應用場景**

#### **Shell：**

* 自動化系統任務，如備份和數據處理。
* 管理軟體部署和測試環境。

#### **SSH：**

* 遠程服務器管理。
* 安全的文件同步（例如使用 rsync）。
* 安全隧道，用於內部網絡訪問。
* 協作式開發環境和調試。

SSH command：

**核心指令**

ssh：遠程連線 ssh-keygen：生成 SSH 密鑰 ssh-copy-id：傳輸公鑰到遠程**主機**

ssh-agent：管理私鑰 ssh-add：將私鑰添加到 SSH-Agent

**文件傳輸相關**

scp：安全複製文件 rsync：文件/目錄同步 sftp：交互文件傳輸

端口轉發與tunnel

ssh -L：local ssh -R：remote ssh -D：dynamic（SOCKS 代理）

**配置與優化**

~/.ssh/config：SSH 配置文件，用於別名和設置優化

ssh -X：啟用 X11 遠程圖形轉發 ssh -i：指定私鑰文件 ssh -p：指定端口

**診斷與測試**

ssh -v：顯示調試信息 ssh -o：臨時設置 SSH 選項（如 StrictHostKeyChecking=no） ssh-keyscan：從遠程主機獲取公鑰