

北科 VCP-AI 運算平台

操作說明

2026-03-10 Rev.6

國立臺北科技大學

計算機與網路中心

NTUT-CNC
vComputingPlatform

目錄

壹、	平台使用規範及注意事項.....	3
貳、	平台介紹.....	5
參、	平台登入及專案申請.....	6
肆、	機器學習專案操作說明.....	10
伍、	機器學習任務操作說明.....	15
陸、	資料檔案的上傳及下載.....	17
柒、	鏡像管理及使用.....	21
捌、	使用情境範例.....	22
玖、	常見問題及聯絡資訊.....	26

壹、 平台使用規範及注意事項

一、 平台運算環境之使用規範：

1. 平台使用校園學術網路及資源，使用者需遵守本校「電腦教室及設備使用規則」與「校園網路使用規範」等規範(詳見本中心網頁)。
2. 平台運算僅提供使用者學習、研究及相關學術使用，不得作為不法、營利及妨害他人資訊使用及網路存取之用途。
3. 平台專案限定教職員方可申請，學生僅能作為專案使用者，不得申請新計畫。
4. 本平台提供之儲存區作為運算暫存用，使用者作資料處理時應自行充分備份相關數據，本中心不負資料保管責任。
5. 因空間有限，為避免影響使用者，使用服務時請將檔案總容量控制於 250GB 內，並請隨時清除未使用之檔案。
6. 運算資源使用完畢後或長期間置之服務及任務請即刻刪除，避免資料遺失及運算資源浪費。
7. 多 GPU 運算請以任務形式提交進行為主，若需佈署多 GPU 服務。
8. 長期占用機器學習服務者，系統將主動寄發信件通知，若屢次發生，同一申請人於專案結束後 30 日內不得申請及展期專案。
9. 申請將依線上使用情形進行審核，實際起始日可能於初始申請之日期不同，專案以總時長不變為原則，專案申請人應於申請審核通過後確認核發之專案起迄日期。
10. 本中心保留最終平台使用權限審核及管理之權利，若有使用者違反相關資訊使用規範，本中心得將使用者進行合理範圍內之處分(包含但不限於：警告、除權、提報資安事件等)。
11. 以上若有未盡事宜，以本中心決議為主。

二、 平台運算環境之注意事項：

1. SSH 終端機存取僅提供校內(140.124.0.0/16)連線，校外請透過本中心 VPN 連線使用。
2. 網頁終端機存取(包含 JupyterLab、Code Server 等服務)存取僅限台灣地區 IP 連線。
3. 新申請之專案，預設將僅提供**一片 GPU** 及 **7 日**使用期限，如需多 GPU，可自專案介面申請額度調升或期限延展。
4. 專案 GPU 額度調升及展期申請將依平台實際使用情形進行配發，其結果將於審核結束後以電子郵件通知申請人。
5. 為準確控管平台資源使用，專案展期申請請於專案到期再行前**十四**天申請(※到期後保留緩衝期)，提前申請之專案將先予駁回，避免因擱置過久而忽略。
6. 使用者客製容器環境；客製環境空間限制。(提供使用者以自定義環境快速部屬)
 - 單一專案自定義鏡像總容量限制為 100GB，若使用空間超過一半，系統會發告警通知。
 - 機器學習服務限使用單 GPU 進行環境測試，多 GPU 運算請以任務形式提交進行。

貳、平台介紹

一、北科 VCP 將提供使用者以線上網頁終端機的方式存取並使用 GPU 運算資源

- 平台網址 <https://vcpp.ntut.edu.tw>
- 平台進入方式：本校校園入口網站->資訊服務->北科 VCP-AI 運算平台
- 平台提供以下服務
 - ◆ 機器學習服務
 - ◆ 機器學習任務
 - ◆ 自定義鏡像管理

二、機器學習服務

- 使用環境服務可在平台建立 GPU 加速之運算環境
- 您可使用環境服務開發相關運算程式
- 服務佈署完成後可以網頁形式遠端連線進行環境的終端機操作

三、機器學習任務

- 使用者可將自行撰寫好之程式以任務的方式執行
- 如此可使用多 GPU 環境進行運算，提高運算速度

四、環境鏡像

- 目前提供主流之 GPU 運算環境包含(Cuda/Pytorch/Tensorflow)
- 若沒有符合您需求之環境，可自行客製環境，並佈署為自定義鏡像
 - ◆ 您可先佈署相近的環境
 - ◆ 使用環境服務，進入終端機環境自行安裝套件及客製環境
 - ◆ 將客製過後之環境儲存為自定義鏡像，未來使用環境服務及執行工作時皆可使用

五、儲存空間

- 本平台佈署之容器環境，於環境建立時自動掛載一空間供使用者暫存檔案，詳請閱讀環境內的說明文件
- 為避免影響使用者，使用機器學習服務時請將檔案總容量控制於 250GB 內
- 請避免將資料儲存於容器內，避免因環境刪除時一併清除
- 使用者仍應隨時備份您的資料，避免資料遺失

參、平台登入及專案申請

一、流程說明



二、登入平台

1. 北科大身分登入

- ◆ 請先登入本校校園入口網站
- ◆ 點擊 資訊服務->北科 VCP-AI 運算平台 登入系統



注意：若您作為專案使用者(及身份為學生)，至少須於專案建立前，進入平台乙次進行建檔，點擊進入即完成建檔。

三、專案申請(專案限校內教職員申請，學生只能作為專案使用者，不得申請專案)

1. 登入成功後，您尚需申請專案方能開始使用
2. 請點擊頁面左方 機器學習專案->專案列表，從「新增專案」處填寫申請



3. 填寫相關資訊

- ◆ 使用者請填入專案名稱、專案描述(選填)、專案起始時間(預設為申請當日)
- ◆ 專案時長以 30 天內為主，若需再延長可於專案到期前 14 日提出專案展延申請，每次延期以 30 日為限
- ◆ 專案成員請以校園入口網站帳號(學號/員工編號)搜尋，並指派至少一位管理員。(建議應至少一位教職員工為專案管理者)
- ◆ 參與人員需事先登入本平台，進行使用者資料建檔後方可加入專案
- ◆ 專案用途請依需求選取，且僅供教職員進行申請
- ◆ GPU 資源請斟酌申請，避免資源占用及浪費
- ◆ 您至少需於專案開始三天前申請



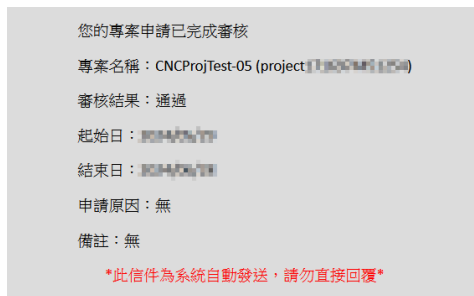
4. 申請送出後請等待本中心審核通過

- ◆ 您可於機器學習專案->專案審核列表中隨時查看審核狀態
- ◆ 審核通過時將寄發 Email 通知專案管理者

專案名稱	申請人	申請時間	類型	審核狀態
<input type="checkbox"/> CNCProjTest-05	陳明倫	2026/03/04	展延申請	已核准
<input type="checkbox"/> CNCProjTest-05	陳明倫	2026/03/04	專案申請	已核准
<input type="checkbox"/> CNCProjTest-04	陳明倫	2026/03/04	專案申請	已核准

5. 本平台管理小組可視情形修改專案核發區間，以不影響專案總時長為原則

- ◆ 請專案負責人詳閱開通郵件，確認實際專案開通區間



四、專案切換

1. 若您同時申請多個專案，可使用頁面右上角的專案選單進行專案切換



五、專案延期

1. 專案到期前 14 日，專案負責人得從平台線上申請專案展期

- ◆ 如專案狀態為「使用中」，可從機器學習專案->專案列表->專案詳細資訊頁面進行延期申請



- ◆ 如專案狀態已進入「緩衝期」，可從機器學習專案->專案列表中，選擇該專案進行延期申請



2. 專案延期注意事項

- ◆ 專案延期限原專案申請人於到期日前 14 天方可提出。
- ◆ 專案延期後，使用者於專案內建立之機器學習環境，其結束日期不會隨專案一併展期。
- ◆ 核定之專案延期區間，視平台使用情形而定，可能提前或延後，惟以延期時長不變為原則，請申請人留意。
- ◆ 單一專案在同一時間僅能送出一個展延申請。如有其他待審核的申請，請先

撤銷舊申請後，再重新提出新的展延申請

3. 專案延期申請查詢與撤銷

- ◆ 查詢：您可於機器學習專案->專案審核列表 確認專案延期審核狀態。

專案名稱	申請人	申請時間	類型	審核狀態
CNCProjTest-05	陳冠廷	2024/03/20	展延申請	待審核
CNCProjTest-05	陳冠廷	2024/03/20	展延申請	已撤回
CNCProjTest-05	陳冠廷	2024/03/20	展延申請	已駁回
CNCProjTest-05	陳冠廷	2024/03/20	展延申請	已核准

(審核完成後，系統亦將寄送電子郵件通知申請結果)

您的展延申請已完成審核
專案名稱：CNCProjTest-05 (project [11255794125])
審核結果：通過
申請天數：1
新起始日：2024/03/20
新結束日：2024/03/20
核准天數：1
申請原因：無
備註：無
此信件為系統自動發送，請勿直接回覆

您的展延申請已完成審核
專案名稱：CNCProjTest-05 (project [11255794125])
審核結果：未通過
申請天數：60
申請原因：無
備註：目前專案延期每次暫以1個月為限，如需再延長使用得於專案到期前14日向本中心申請專案展期。管理小組敬上
此信件為系統自動發送，請勿直接回覆

- ◆ 撤銷：待審核狀態的專案，亦可從機器學習專案->專案審核列表 中，撤回申請。

專案名稱	申請人	申請時間	類型	審核狀態
CNCProjTest-05	陳冠廷	2024/03/20	展延申請	待審核

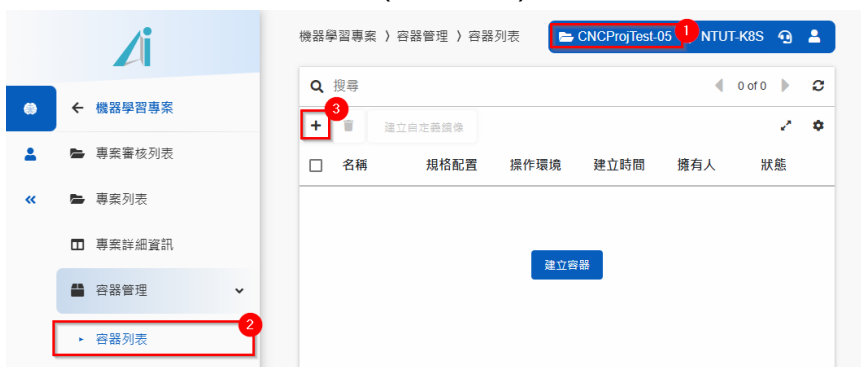
肆、機器學習專案操作說明

容器管理提供您部署公用及自定義鏡像之 GPU 加速的運算環境，您可依照說明執行下列任務：

- ◆ 建立容器
- ◆ 查詢及刪除容器
- ◆ 進入終端機環境
- ◆ 容器的使用原則

一、建立容器

1. 選擇 機器學習服務->專案列表(選擇專案)->容器管理->容器列表



2. 閱讀「終端使用者協議-容器」後，勾選同意 > 下一步
3. 輸入容器等基本資訊 > 下一步
4. 選擇硬體配置及鏡像

- ◆ 點選公共鏡像 -> 依照您的需求選擇環境鏡像
- ◆ 以您的需求選填共享記憶體

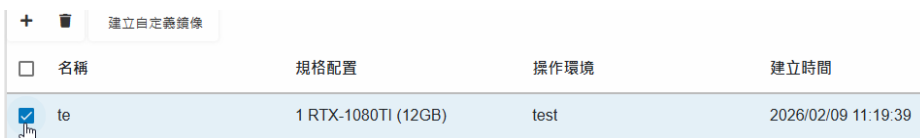
共享記憶體功能說明：允許兩個或更多程序訪問同一區塊記憶體，用於多 GPU 容器進行運算時，將運算資料置於共享記憶體，提高 GPU 平行運算效率。

注意：機器學習服務每次佈署不得超過 7 天，服務超過 7 天或超過專案結束時限，服務將自動刪除。

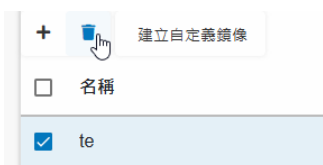
5. 返回服務列表，等待您的環境服務狀態變為「運行中」
 - ◆ 若您需要使用多 GPU 環境，建議您佈署機器學習任務
6. 佈署完成，點擊佈署的服務，即可看到容器相關資訊

二、刪除環境服務佈署

1. 進入 機器學習服務->服務列表->選取您欲刪除的環境服務

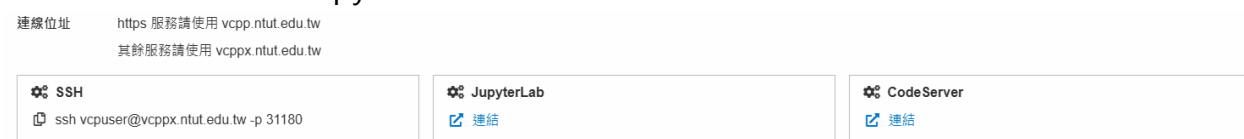


2. 點擊上方刪除按鈕

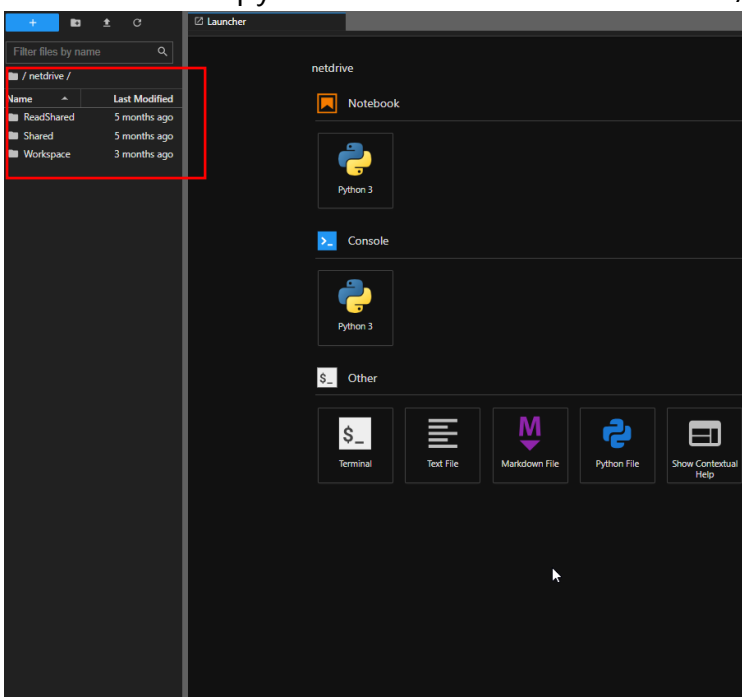


三、以 Jupyter Web 服務連線使用終端機環境

1. 點擊服務資訊欄中的 Jupyter 連線超連結



2. 進入後會看到 Jupyter Lab 環境，以下紅框處對應是/NFS 目錄



3. 點擊「Terminal」圖示，將會新開終端機，即可開始使用。
4. 其他使用

- ◆ 本平台網頁環境服務遠端存取採用開源專案 Jupyter Lab，其他進階使用請參閱 <https://jupyterlab.readthedocs.io/en/stable/>

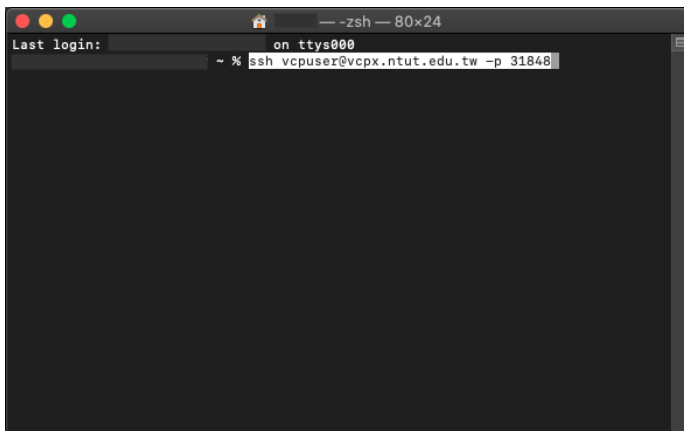
四、以 SSH 連線使用終端環境(以下為範例供參考，實際環境以現有平台為主)

1. 點擊欲進入的機器學習服務，查看下方詳細佈署資訊
2. 請注意，SSH 終端機連線限台灣境內 IP，若連線失敗，請檢查您是否使用境外 IP 或以本中心 VPN 連線使用
3. 若您使用 Linux/MAC 環境

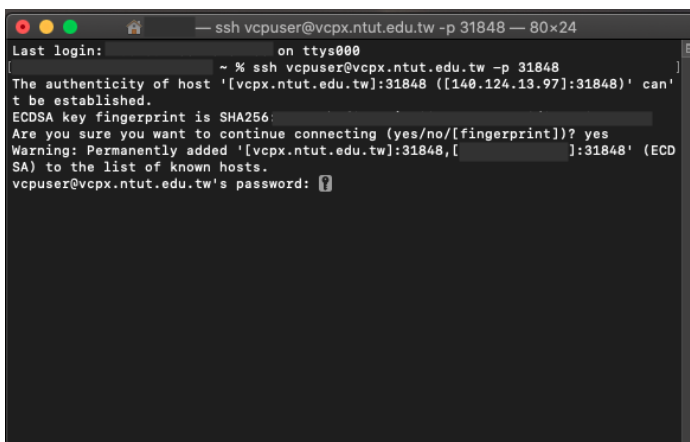
- ◆ 點擊 SSH 資訊旁複製圖示



- ◆ 開啟終端機，將複製的資訊貼至終端機執行

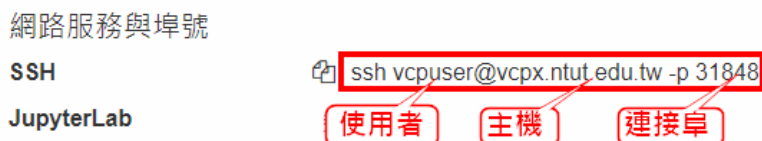


- ◆ 輸入您於佈署時設定的 SSH 密碼

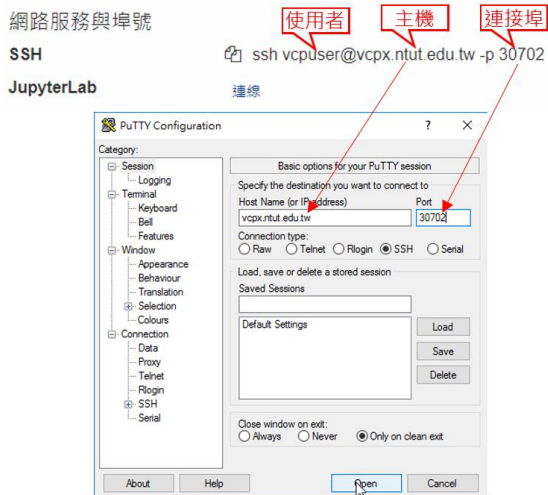


4. 若您使用 Windows 環境

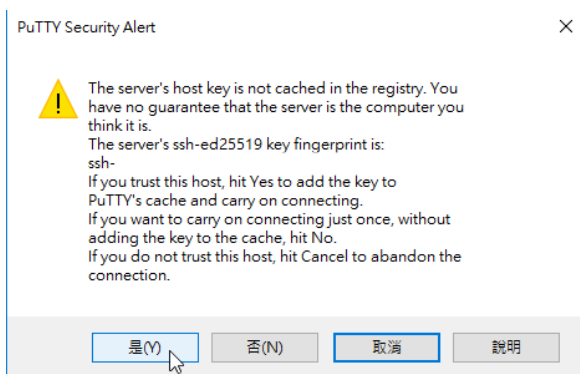
- ◆ 請紀錄 SSH 資訊



- ◆ 安裝 PUTTY(或其他 SSH 終端機軟體)
- ◆ 開啟終端機軟體，輸入下列資訊
 - 主機：vcpx.ntut.edu.tw
 - 連接埠：網頁上提供之埠號(-p 後的數字，共 5 位數字)

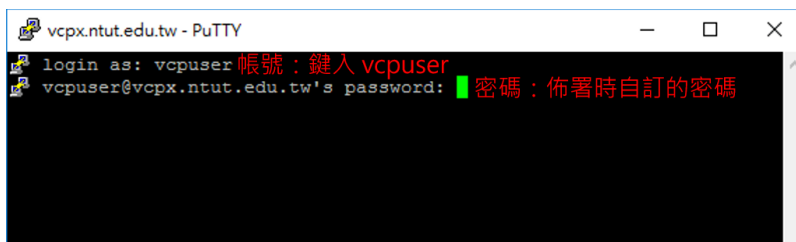


◆ 金鑰指紋碼確認，請按「是」

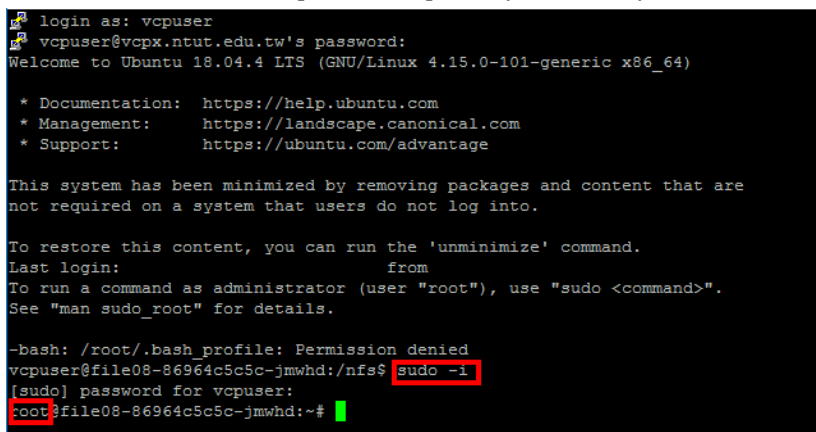


◆ 連線後輸入登入資訊

- 帳號：vcpuser
- 您於佈署時設定的 SSH 密碼



5. 連線成功後您可鍵入[sudo -i]指令(不含括號) 取得 root 權限進行操作



五、環境服務的使用原則

1. Terminal 預設進入路徑為 /nfs/

除客製環境之套件及軟體外，所有資料請務必放置於此目錄之子目錄下，勿移置

其他位置。

2. 進入終端機後便可直接執行您的工作，使用者亦可自行使用 apt-get、pip 等套件管理軟體安裝套件

```
root@fs-8885dd859-9wnzb:/nfs/Workspace# nvidia-smi
Wed Mar 25 08:58:33 2020

+-----+
| NVIDIA-SMI 418.116.00    Driver Version: 418.116.00    CUDA Version: 10.1    |
+-----+
| GPU   Name               Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|  Memory-Usage | GPU-Util  Compute M. |
+-----+-----+
|    0   Tesla V100-SXM2...    On          | 00000000:07:00:0 Off  |      0%      Default  |
| N/A   32C   P0      43W / 163W | 0MiB / 32480MiB |           |             |
+-----+-----+

+-----+
| Processes:                                                       GPU Memory |
|  GPU       PID    Type   Process name                               Usage      |
+-----+-----+
| No running processes found                                     |
+-----+

root@fs-8885dd859-9wnzb:/nfs/Workspace#
```

3. 若您連線使用時出現 SSL 憑證錯誤，可導入/nfs/底下之計網中心憑證(預設系統已導入)

```
root@20072301-ddf6cc95-2mw7n:/nfs# ls
AI_Data_Global  NTUT_Root.crt
root@20072301-ddf6cc95-2mw7n:/nfs#
```

4. 沒有使用運算資源時，請將您的環境儲存為自定義後刪除環境，以避免運算資源長期占用
5. 若您需要長時間行檔案傳輸，請依本說明書第陸章指示，佈署檔案傳輸專用環境
6. 若中心提供之主流環境不符您所需，您可選擇相近環境，亦可佈署單 CUDA 環境(僅有 NVIDIA 及 CUDA 驅動之環境)後自行客製環境
7. 切勿移除環境中的 Python、Jupyter 及 SSH 套件環境，避免您的終端機服務無法存取

伍、機器學習任務操作說明

一、建立任務

1. 選擇 任務列表 > 執行任務



2. 輸入任務詳細資料

- ◆ 輸入名稱
- ◆ 選取硬體配置及鏡像
- ◆ 輸入命令



3. 在確認視窗中確認您的任務，點選建立

規格總覽

名稱	job
GPU 型號	RTX-1080Ti (12 GB)
GPU 數量	1 片
硬體配置	1080TiX1 (2 CORE + 62 GB RAM)
鏡像類型	公共鏡像
操作環境	CUDA
Version	10.0-CUDNN7
命令	nvidia-smi

TWD 0 / 小時

取消 建立

二、確認任務運行

1. 回到任務列表，檢查任務狀態

<input type="checkbox"/>	名稱	操作環境	擁有人	建立時間	建立類型	狀態
<input type="checkbox"/>	job	CUDA:10.0-CUDNN7	管理員	---	立即	任務初始化
<input checked="" type="checkbox"/>	名稱	操作環境	擁有人	建立時間	建立類型	狀態
<input checked="" type="checkbox"/>	job	CUDA:10.0-CUDNN7	管理員	2026/03/05 15:27:40	立即	執行成功

2. 任務事件可查詢詳細執行明細

任務詳細資訊 日誌 事件

類型	原因	訊息	最早出現於	最近出現於	發生次數
Normal	Started	Started container tf	2026/03/05 15:27:19	2026/03/05 15:27:19	1
Normal	Created	Created container tf	2026/03/05 15:27:19	2026/03/05 15:27:19	1
Normal	Pulled	Container image "140.124.2...	2026/03/05 15:27:19	2026/03/05 15:27:19	1
Normal	Scheduled	Successfully assigned proje...	---	---	---

3. 任務的終端機輸出(STDOUT)可於日誌欄位查詢

任務詳細資訊 日誌 事件

匯出文字檔


```

Thu Mar 5 07:27:19 2026
+-----+
| NVIDIA-SMI 550.127.05           Driver Version: 550.127.05   CUDA Version: 12.4   |
+-----+-----+-----+-----+
| GPU  Name          Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
|                                           | MIG M.         |                      |
+-----+-----+-----+-----+
|  0   NVIDIA GeForce GTX 1080 Ti      Off   | 00000000:0F:00:0 Off |          N/A   |
| 23%   23C    P8              9W / 250W |  0MiB / 11264MiB |      0%   Default  |
|                                           |                      | N/A   |
+-----+-----+-----+-----+

+-----+
| Processes:                                |
| GPU  GI  CI           PID  Type  Process name                        GPU Memory |
|                                           | Usage   |
+-----+-----+-----+
| No running processes found                |
+-----+

```

4. 工作完成，請刪除工作

+ 

名稱

job

陸、 資料檔案的上傳及下載

使用平台時，若您欲使用資料集及個人檔案，可使用前述終端機環境進行檔案上傳集下載，本章節將提供以下說明：

- ◆ 檔案目錄說明及使用原則
- ◆ 佈署檔案傳輸服務環境
- ◆ 使用 Jupyter 服務進行檔案傳輸
- ◆ 使用 SFTP 進行檔案傳輸

一、目錄說明及使用原則

1. 平台有自動掛載一空間供使用者儲存檔案

- ◆ netdrive/ReadShare 目錄及其內容 (專案唯讀目錄)
- ◆ netdrive/Shared 目錄及其內容 (專案共享目錄)
- ◆ netdrive/Workspace 目錄及其內容 (使用者個人目錄)

2. 目錄使用原則

- ◆ 請使用 /nfs 目錄下的資料夾做為您存取資料及工作的目錄
- ◆ 預設終端機即資料傳輸的工作目錄即為/nfs/，您可直接切換至目錄下之資料夾，進行檔案傳輸及下達指令，請盡量避免切換至/nfs 以外之資料夾
- ◆ 非雲端硬碟空間(上述 2.之目錄)內資料，會隨您的服務刪除後同步清空，且無法救回，請注意備份及儲存您的檔案
- ◆ 為避免儲存空間不足影響其他使用者，請盡量將您的資料上傳容量控制於 **250GB** 以內
- ◆ 若有大量碎檔，建議您先壓縮成 tar 或 zip 格式，上傳後再於終端機解壓縮
- ◆ 若您需要長時間行檔案傳輸，請依本章指示，佈署檔案傳輸專用環境，避免長期占用運算資源

二、佈署檔案傳輸服務環境

1. 依前述「機器服務」，以 1 GPU 之硬體配置佈署 FileUpload 映像

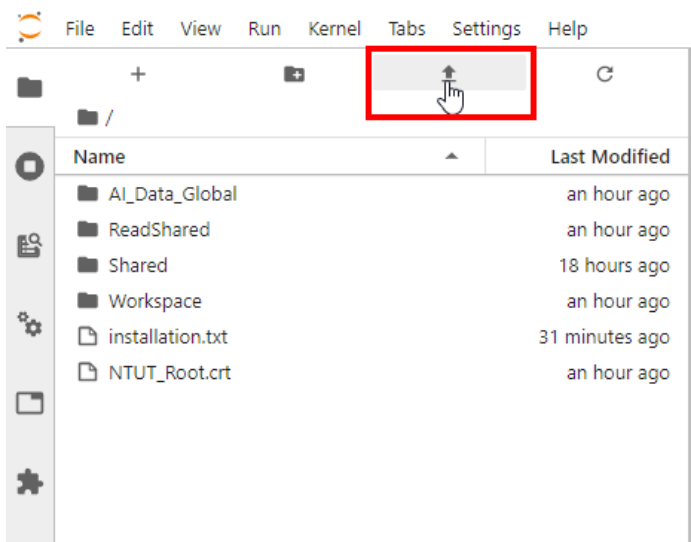
- ◆ 若您並非同時間進行 GPU 運算時，請部署檔案傳輸專用環境進行上傳及下載，避免佔用 GPU 資源



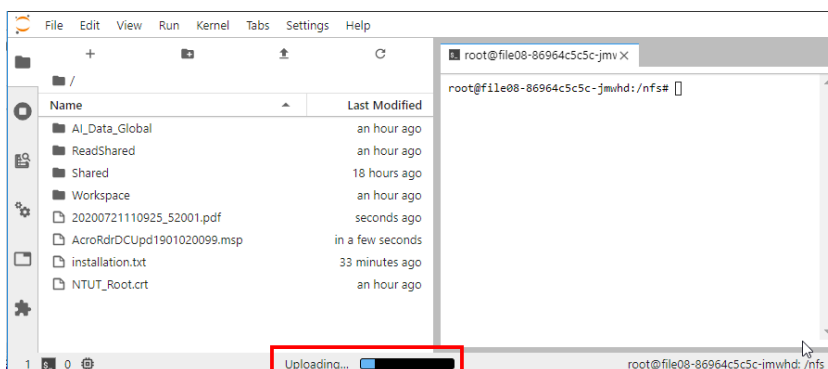
2. 依前述指示進入 Jupyter Web 或 SSH 服務
 3. 佈署後即可進入您專屬的儲存區，進行資料傳輸
- 三、使用 Jupyter 服務進行檔案傳輸(以下為範例供參考，實際環境以現有平台為主)

1. 佈署檔案傳輸服務環境後，依照本說明書第肆章進入 Jupyter Lab 環境
2. 上傳資料

◆ 上傳資料方式：點擊工具列上傳圖示，選擇欲上傳檔案後「開啟」，檔案即開始上傳。

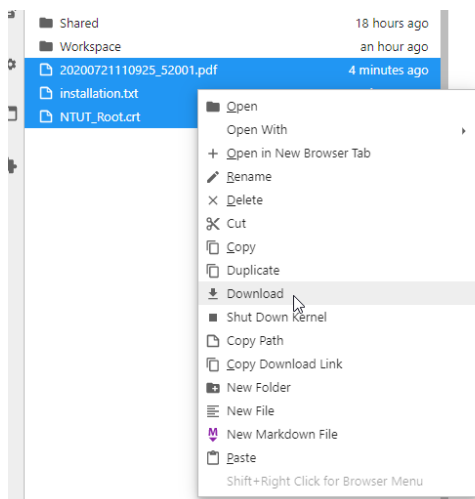


◆ 您可於頁面下方查看上傳進度



3. 下載資料

- ◆ 下載資料方式：選取欲下載檔案，按右鍵，選擇「Download」即可



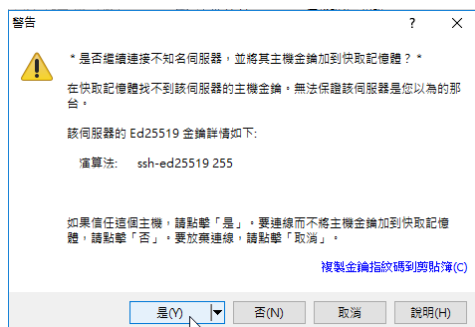
四、使用 SFTP 服務進行檔案傳輸

1. 佈署檔案傳輸服務環境後，紀錄 SSH 相關資訊
2. 請注意，**SFTP 檔案傳輸限台灣境內 IP**，若連線失敗，請檢查您是否使用境外 IP 或以本中心 VPN 連線使用
3. 您可以 SFTP 軟體進行檔案傳輸，此處以 WinSCP 為範例

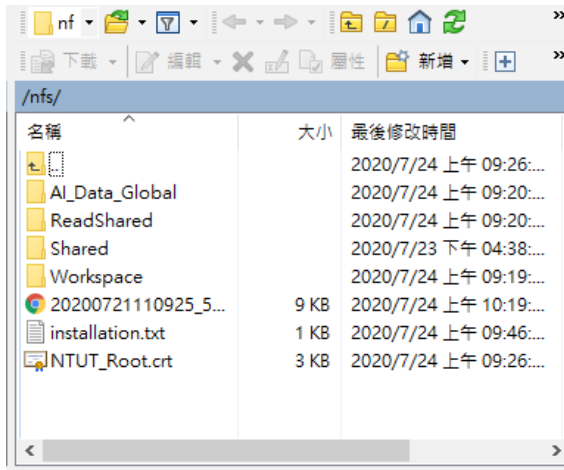
- ◆ 開啟 WSCP 軟體，並鍵入相關資訊



- ◆ 金鑰確認視窗，請按「是」



- ◆ 連線成功後，即可進行檔案上傳，預設目錄為/nfs



柒、鏡像管理及使用

除了本中心提供之主流環境外，您亦可自行客製環境，並佈署為自定義鏡像，提供下次環境服務及執行任務時使用，本章節涵蓋以下說明

- ◆ 鏡像類別
- ◆ 建立自定義鏡像

一、鏡像類別

1. 在您佈署環境服務或執行任務時，可選擇使用「公共鏡像」或「自定義鏡像」
2. 公共鏡像為本中心預先佈署之主流深度學習環境
3. 您可依公共鏡像為基礎佈署環境服務，再進入終端機中客製環境，客製過後您即可將自定義鏡像佈署至環境服務或執行任務而無須重新客製

二、建立自定義鏡像

1. 先建立環境服務後依操作說明進入終端機客製環境

◆ 請注意，在建立自定義鏡像時，**請勿上傳資料集與檔案**

2. 點選機器學習服務->服務列表->選取您建立的環境

名稱	規格配置	操作環境	建立時間	擁有人	狀態
te	1 RTX-1080TI (12GB)	test	2026/02/09 11:19:39		運行中

3. 點選建立自定義鏡像

搜尋

+ 建立自定義鏡像

名稱

te

4. 輸入名稱、描述及鏡像相關資訊(鏡像名稱通常為主要軟體名稱;標籤通常為版本)

部署詳細資訊 服務資訊 監控 事件 建立自定義鏡像 x

tensorflow
長度為2-128個字元，只能包含小寫英文字母、數字、底線_與減號(-)

鏡像標籤*

1.1
長度為2-128個字元，開頭不得為減號(-)或點(.)，其餘內容可使用大小寫英文字母、數字、底線_、減號(-)與點(.)

鏡像類型*

私有

專案：僅有與您相同專案的使用者可以使用該鏡像，包含您自己

5. 回到自訂定義像列表，建立完成
6. 建立完成後即可以自定義的鏡像部署環境服務及執行任務，如此便不必重複客製您的環境

名稱	規格配置	操作環境	建立時間	擁有人	狀態
test1	mycustomimage	tensorflow.tf	私有	JupyterLab:8889 ...	建立中

注意：單一專案之自定義鏡像容量加總不得超過 250GB，超過將無法儲存額外自定義鏡像。

捌、使用情境範例

一、運行重大專案-多 GPU 需求

1. 使用流程說明



2. 佈署環境服務

- ◆ 依本說明書前述指示佈署與您使用環境接近之環境服務

GPU 型號*

RTX-1080Ti (12 GB)

GPU 數量*

1片 2片 3片 4片 8片

目前最大可用量為 8 片，若使用高於該數量的容器需要等候建立

硬體配置*

規格名稱	CPU (Core)	RAM (GB)
<input checked="" type="checkbox"/> 1080TiX1	2	62

根據不同 GPU 片數，硬體配置將有所調整

鏡像類型*

公共鏡像 自定義鏡像

操作環境*

CUDA 10.0-CUDNN7

3. 進入終端客製環境(以下為範例供參考，實際環境以現有平台為主)

- ◆ 依本說明書第四章連入終端機，客製您的環境

```
root@sample-d9d55f7b5-9xhzh:/nfs/global/LibraryAndInstaller/openCV# ls
installer.sh
root@sample-d9d55f7b5-9xhzh:/nfs/global/LibraryAndInstaller/openCV# bash installer.sh
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [835 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [37.0 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [7904 B]
Get:10 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [870 kB]
34% [7 Packages 1970 kB/11.3 MB 17s] 408 kB/s 33s

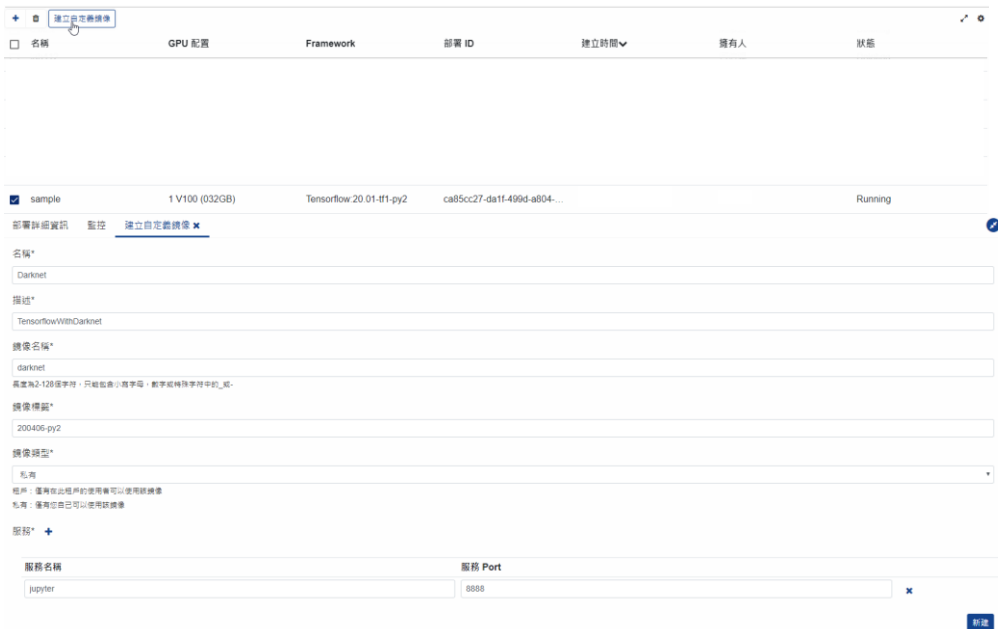
root@sample-d9d55f7b5-9xhzh:/nfs/Workspace/darknet# var="GPU=1"
root@sample-d9d55f7b5-9xhzh:/nfs/Workspace/darknet# sed -i "1s/.*/$var/" Makefile
root@sample-d9d55f7b5-9xhzh:/nfs/Workspace/darknet# make
gcc -Iinclude/ -Isrc/ -DGPU -I/usr/local/cuda/include/ -Wall -Wno-unused-result -Wno-unknown-pragmas -Wfatal-errors -fPIC -Ofast -DGPU -c ./src/gemm.c -o obj/gemm.o
./src/gemm.o: In function 'time_gpu':
./src/gemm.c:232:9: warning: 'cudaThreadSynchronize' is deprecated [-Wdeprecated-declarations]
     cudaThreadSynchronize();
     ^
In file included from /usr/local/cuda/include/cuda_runtime.h:96:0,
                 from include/darknet.h:11,
                 from ./src/utils.h:5,
                 from ./src/gemm.c:2:
/usr/local/cuda/include/cuda_runtime_api.h:957:57: note: declared here
extern_CUDA_DEPRECATED_host_cudaError_t_CUDARTAPI_cudaThreadSynchronize(void);
gcc -Iinclude/ -Isrc/ -DGPU -I/usr/local/cuda/include/ -Wall -Wno-unused-result -Wno-unknown-pragmas -Wfatal-errors -fPIC -Ofast -DGPU -c ./src/convolutional_layer.c -o obj/convolutional_layer.o
./src/convolutional_layer.c: In function 'time_gpu':
./src/convolutional_layer.c:232:9: warning: 'cudaThreadSynchronize' is deprecated [-Wdeprecated-declarations]
     cudaThreadSynchronize();
     ^
In file included from /usr/local/cuda/include/cuda_runtime.h:96:0,
                 from include/darknet.h:11,
                 from ./src/utils.h:5,
                 from ./src/convolutional_layer.c:2:
/usr/local/cuda/include/cuda_runtime_api.h:957:57: note: declared here
extern_CUDA_DEPRECATED_host_cudaError_t_CUDARTAPI_cudaThreadSynchronize(void);
gcc -Iinclude/ -Isrc/ -DGPU -I/usr/local/cuda/include/ -Wall -Wno-unused-result -Wno-unknown-pragmas -Wfatal-errors -fPIC -Ofast -DGPU -c ./src/deconvolutional_layer.c -o obj/deconvolutional_layer.o
./src/deconvolutional_layer.c: In function 'time_gpu':
./src/deconvolutional_layer.c:232:9: warning: 'cudaThreadSynchronize' is deprecated [-Wdeprecated-declarations]
     cudaThreadSynchronize();
     ^
In file included from /usr/local/cuda/include/cuda_runtime.h:96:0,
                 from include/darknet.h:11,
                 from ./src/utils.h:5,
                 from ./src/deconvolutional_layer.c:2:
/usr/local/cuda/include/cuda_runtime_api.h:957:57: note: declared here
extern_CUDA_DEPRECATED_host_cudaError_t_CUDARTAPI_cudaThreadSynchronize(void);
gcc -Iinclude/ -Isrc/ -DGPU -I/usr/local/cuda/include/ -Wall -Wno-unused-result -Wno-unknown-pragmas -Wfatal-errors -fPIC -Ofast -DGPU -c ./src/convolutional_layer.c -o obj/convolutional_layer.o
```

- ◆ 同時也可將您撰寫執行任務時的腳本並存放於使用者資料夾內

```
root@sample-d9d55f7b5-9xhhz:/nfs/Workspace# vi sample.py
root@sample-d9d55f7b5-9xhhz:/nfs/Workspace# tail sample.py
import cv2
print(cv2)
root@sample-d9d55f7b5-9xhhz:/nfs/Workspace#
```

◆ 請注意，進行環境客製時，請勿上傳資料集與檔案

4. 將環境儲存為自定義鏡像



5. 以自定義鏡像執行任務

- ◆ 依前述指示以自定義鏡像建立任務
- ◆ 注意：命令與參數間請以空行分隔，若參數為檔案，請以絕對路徑表示(EX: 若檔案位於使用者資料夾，請在檔名前加入「/nfs/Workspace」前綴

* 名稱
可使用小寫英文字母與數字，不可包含其他特殊符號

* 硬體配置
✓ 個人 GPU 使用額度：1 + 1 / 4 片
✓ 個人 CPU 使用額度：6 + 4 / ∞ Core
✓ 個人 RAM 使用額度：72 + 64 / ∞ GB

共享記憶體 啟用

* 鏡像類型

* Framework

* 命令
每個指令及參數之間請以換行做為區隔

配置費用 TWD 40 / 小時

6. 佈署監視

- ◆ 以執行工作之「事件」監控佈署情形

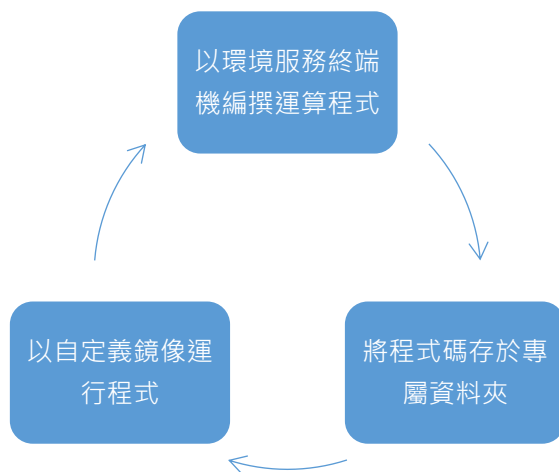
類型	原因	訊息	最早出現於	最近出現於	發生次數
Normal	Started	Started container ff	2020/04/06 08:25:17	2020/04/06 08:25:17	1
Normal	Created	Created container ff	2020/04/06 08:25:16	2020/04/06 08:25:16	1
Normal	Pulled	Successfully pulled image "10.10.10.130000/..."	2020/04/06 08:25:14	2020/04/06 08:25:14	1
Normal	Pulling	Pulling image "10.10.10.130000/..."	2020/04/06 08:23:33	2020/04/06 08:23:33	1

◆ 日誌欄位可察看執行時的輸出



二、重大專案-持續修改程式內容

1. 流程說明



2. 以環境服務終端機編撰運算程式

- ◆ 在您的環境已客製完成並存為自訂義鏡像的情況下
- ◆ 以 FileUpload 鏡像，並搭配無 GPU 環境做為修改程式之終端

* 硬體配置

全域 GPU 使用額度: 3 + 0 / 8 片
 個人 CPU 使用額度: 4 + 2 / ∞ Core
 個人 RAM 使用額度: 62 + 8 / ∞ GB

* 鏡像類型 公共鏡像 自定義鏡像

* Framework

3. 進入終端機，將程式碼存於專屬資料夾

```
root@d-67b76b9cf8-dfncr:/nfs/Workspace# vi sample.py
```

4. 以自定義鏡像運行程式

- ◆ 注意若使用檔案作為參數，請使用檔案的絕對路徑

* 名稱
可使用小寫英文字母與數字，不可包含其他特殊符號

* 硬體配置 ▾
● 僅入 GPU 使用額度：1 + 1 / 4 片
● 僅入 CPU 使用額度：6 + 4 / ∞ Core
● 僅入 RAM 使用額度：72 + 64 / ∞ GB

共享記憶體 啟用

* 鏡像類型

* Framework ▾

* 命令
每個指令及參數之間請以換行做為區隔

配置費用 TWD 40 / 小時

5. 於任務執行欄位查看程式輸出

任務詳細資訊 日誌 事件

[匯出文字檔](#) 📄

```
<module 'cv2.cv2' from '/usr/local/lib/python2.7/dist-packages/cv2/cv2.so'>
```

6. 您可重複上述 2-4 步驟，持續修改您的程式碼

玖、常見問題及聯絡資訊

一、常見問題

1. 我的帳號無法成功登入

- ◆ 請確認您的登入資訊可正常登入本校校園入口網站

<https://nportal.ntut.edu.tw>

- ◆ 若您忘記密碼或需要修改密碼，請參閱入口網站下方「忘記密碼說明」



2. 登入後顯示「無可用專案」



- ◆ 若您為學生，請詢問您的指導老師協助申請專案
- ◆ 若您為教職員，請依照本手冊第參章進行專案申請

3. 建立機器學習服務/任務時，系統顯示「目前非計畫使用期間，無法建立新服務」

- ◆ 請與專案負責人聯繫，確定專案核發區間 (可能於申請時不同，將於審核通知信內告知負責人)。
- ◆ 請確認專案是否已過期，若需展期，可請專案負責人來信申請展延。

4. SSH 終端機無法連線

- ◆ SSH 連線限台灣境內 IP
- ◆ 請檢查您是否使用境外 IP 或使用非本中心 VPN 連線
- ◆ 必要時可使用本中心 VPN 進行 SSH 連線，詳見本中心網頁
<https://cnc.ntut.edu.tw/p/405-1004-110297.php?Lang=zh-tw>

5. SSH 連線於一段時間後會自行斷線

- ◆ 為應本中心資訊安全政策，SSH 連線設有 4 小時連線逾時設定
- ◆ 建議您可將運算相關指令透過 JupyterLab 下達，避免斷線造成指令終止
- ◆ 您亦可使用 tmux、screen 等終端機工作階段控管軟體，使指令繼續於終端機斷線後持續運行

6. 於 JupyterLab 上及建立的檔案及目錄，無法透過 SFTP 及 SSH 等方式存取
 - ◆ 由於 JupyterLab 及 SSH 連線時所使用之身分不同所導致
 - ◆ 您可透過 (chmod o+w [目錄名稱]) 指令進行權限變更
 - ◆ 變更權限時請注意檔案存取安全，並於傳輸完成後以 (chmod o-w [目錄名稱]) 指令恢復原權限
7. 安裝套件時，系統提示 SSL 錯誤或憑證錯誤
 - ◆ 請導入位於您專屬資料夾內的 NTUT_Root.crt 憑證即可
 - ◆ 系統預設已導入，導入操作會因套件管理程式不同而異，詳細使用請參見各套件管理軟體使用說明
 - ◆ 詳見本中心網頁 <https://cnc.ntut.edu.tw/p/406-1004-93484,r1522.php?Lang=zh-tw>
 - ◆ 您亦可以「Import Private CA Certificate」加上您使用的軟體、套件作為關鍵字進行搜尋

```
root@20072301-ddf6cc95-2mw7n:/nfs# ls
AI_Data_Global NTUT_Root.crt
root@20072301-ddf6cc95-2mw7n:/nfs#
```

8. 一直收到北科 VCP-AI 運算平台硬碟用量告警提醒
這時可以進家目錄底下，執行以下這兩個指令，查看是否為.cache 或/opt 的使用空間過大，造成頻繁收到容器空間使用量告警提醒，記得暫時沒有用到就暫時清掉。
du -sh /* 2>/dev/null | sort -hr | head -n 10
du -sh .[!]* *

二、聯絡及連結資訊

1. 若於您有任何問題

- ◆ 來電分機 3200/3295 (上班時間隨時可提供服務)
- ◆ 洽共同科館 111 室計網中心值班櫃台
- ◆ 來信 cnc@mail.ntut.edu.tw

2. 專案負責人資訊

- ◆ 行政諮詢組 林小姐
- ◆ 校內分機 3217
- ◆ Email : yeanning319@ntut.edu.tw