

Personalized offline VC 規劃及 online VC 實作進度

Student : Sian-Yi Chen

Advisor : Tay-Jyi Lin and Chingwei Yeh

● Product descriptions

□ Introduce

- 它是一個具有離線模式的即時錄放音手持裝置
- 功能為可以在何時何地將 A(來源) 聲音轉換成 B(目標) 聲音

□ User scenario

• 初次使用

1. 登入 / 註冊

(搜尋網路，進入線上模式或離線模式)

• 線上模式

1. 透過 Google API 轉換，可直接錄放音，將 A 聲音轉換成 B 聲音
2. 長按錄音，放開則進入等待畫面進行轉換並播音

• 離線模式

1. 初次使用就跳出 320 句語料要求錄製
2. 可保存錄製進度，分次錄製
3. 錄製結束，將 320 句語料上傳至雲端轉換
4. 等待轉換完畢並下載使用模型
5. 使用離線模式時會將語音先暫存在手機端，等待連線時上傳

• DNN 模型達穩定狀態時

1. 要求每一個禮拜更新一次
2. 更新會將離線模式下錄製音檔上傳雲端保存並將本地端音檔刪除

● Key technology

□ DVC 雲端自動化

1. 開 Google 虛擬機，為了自動化考慮使用 Linux 環境
2. 架設 YMDVC 執行環境
3. 利用裝置端錄製的聲音 & 轉換後的聲音自動進行 DTW 對齊
4. 將對齊後聲音進行自動訓練 (一個禮拜一次)
5. 將取得的模型自動做資料整理 (模型轉換 pre/post norm, DNN model)
6. 讓 DNN model 與參數給予手機端

□ 線上 DVC2.0 轉換

1. 目前先透過 Google API 模擬強演算法行為
 - STT (Speech-to-text)
 - TTS (Text-to-speech)

● Online VC 實作進度

- 已開啟一台 google 虛擬機，目前使用 CentOS 7 作業系統
- 使用 SSH 登入到 google compute engine 的 VM
- 在 CentOS 7 上安裝 Python3

■ 設計考量

● 語料設定

1. 考慮到預錄句數太多而使用意願下降，希望預錄句數下修至 100 句、50 句，越少越好

● 錄製語料時機

1. 初次使用離線模式要求錄製 320 句語料
 - 優點：固定錄製語料，可先將轉換後語料準備好
 - 缺點：需進入離線模式才能錄製語料
2. 初次使用 APP 限定只能用線上模式，等藉由線上模式轉換搜集滿 320 句語料時，才能使用離線模式
 - 優點：使用線上模式的同時就在搜集語料
 - 缺點：使用者錄製的語料有可能訓練效果較提供的 320 句差

● 轉換方式 (對應錄製語料時機)

1. 將錄製的 320 句語料使用 google STT & TTS API 做轉換作為訓練音訊
2. 將線上模式輸入音檔上傳至雲端轉換 (透過 Google API) 並將輸入輸出當作訓練音訊
(錄製語料皆使用 Google API 轉換作為訓練音訊，是為了確保往後更新 DNN model 時訓練音訊一致)

雲端平台選擇

因為雲端可能要自動化，所以做 Linux 系統為選擇

	GCP	AWS	Azure
作業系統	Linux	Linux	Linux
機器類型	任意	EC2	任意
免費方案	90 天免費試用， \$300 美元的抵免額	一年， 每月 750 小時 無法更換機器類型	一年免費試用， 30 天 NT\$6,100 抵免額
計費方式	基本執行個體規格，以秒計費		
優點	最便宜	可選擇的機器類型種類較多	可選擇的機器類型種類最多
缺點	試用時間短， 目前剩餘 46 天	較貴	最貴

附錄

查看 GCP 剩餘天數

The screenshot shows the Google Cloud Platform console dashboard. The browser's address bar displays the URL: `console.cloud.google.com/home/dashboard?authuser=1&hl=zh-tw&organizationId=319425611965&project=dev-acolyte-307405`. The top navigation bar includes the Google Cloud Platform logo, a search bar, and a user profile icon. A red box labeled '1' highlights the 'Google Cloud Platform' logo. The dashboard is divided into several sections:

- 專案資訊 (Project Information):** Displays project details for 'off line converter' (Project ID: dev-acolyte-307405, Project Number: 745931310961). A red box labeled '1' highlights the 'Google Cloud Platform' logo.
- 資源 (Resources):** Shows the 'Compute Engine' resource with 1 running instance.
- 追蹤項目 (Tracked Items):** Indicates no tracking data is available for the last 7 days.
- Compute Engine:** Displays a CPU usage graph for the instance 'instance/cpu/utilization: 0.08%'. The graph shows a peak in usage around 11:15 AM.
- Google Cloud Platform 服務狀態 (Service Status):** Shows the service status as '無法提供資料' (No data available).
- 計費功能 (Billing):** Displays the estimated cost as 'TWD \$0.00' for the period 2021/4/1 to 2021/4/25. A red box labeled '2' highlights the '查看詳細扣款記錄' (View detailed billing records) link.
- Monitoring:** Provides links to '建立我的資訊主頁' (Create my dashboard), '設定快訊政策' (Set alert policy), '建立運作時間檢查' (Create uptime check), and '查看所有資訊主頁' (View all dashboards).

The bottom of the dashboard shows a warning message: 'No data is available for the selected time frame.'

查看 GCP 剩餘天數

The screenshot shows the Google Cloud Platform billing console. The left sidebar contains a menu with 'Overview' highlighted. The top navigation bar shows 'Billing account overview'. The main content area displays the trial status card, which includes the free trial credit of \$7,242 and 46 days remaining. The trial is scheduled to end on June 10, 2021. The page also shows a table of usage data and a section for the most used project.

Google Cloud Platform

帳單

帳單帳戶

我的帳單帳戶

總覽

報表

費用表

費用明細

使用承諾

承諾使用價格分析

預算與快訊

帳單匯出功能

定價

文件

交易

付款設定

付款方式

總覽

帳單帳戶總覽

付款總覽

免費試用抵免額

\$7,242

免費試用抵免額

總抵免額: \$8,352

46

剩餘天數

將於 2021年6月10日結束

在免費試用期間，您不需要支付任何費用。如要在免費試用期結束後繼續執行您的專案，請升級為付費帳戶。

升級

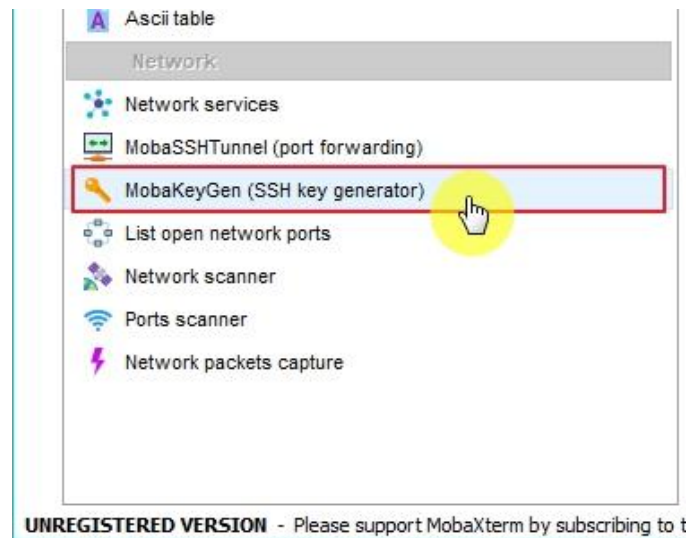
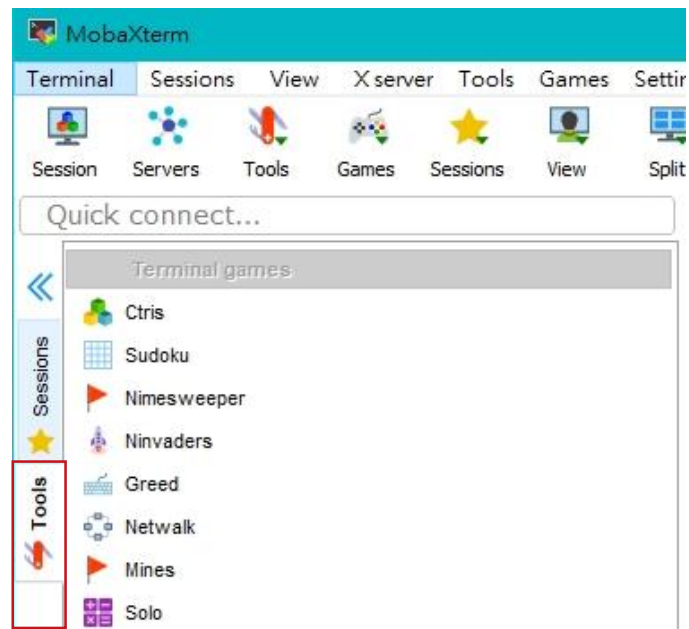
瞭解詳情

用量最多的專案

2020/4/1至2021/4/30

■ 產生 SSH 金鑰

打開 MobaXterm 之後，在 Tools 可以找到 MobaKeyGen (SSH key generator) 功能。



■ 產生 SSH 金鑰

打開之後就按下 **Generate** 按鈕就開始產生 SSH 金鑰，記得用滑鼠在空白處隨意移動進度條才會跑。

接著輸入我們的 **Google** 使用者名稱及金鑰通關密語（選用），按下 **Save private key** 把私鑰存下來。

The screenshot shows the MobaXterm SSH Key Generator window. It has a menu bar with 'File', 'Key', 'Conversions', and 'Help'. The main area is divided into sections: 'Key', 'Actions', and 'Parameters'. In the 'Key' section, the 'Public key for pasting into OpenSSH server' field is populated with a long string of characters. Below this, the 'Key fingerprint' is shown as 'ssh-rsa 2048 53:'. The 'Key comment' field contains 'fg6ts30' and is annotated with a red circle '2' and the text 'Google 使用者名稱'. The 'Key passphrase' field contains five dots and is annotated with a red circle '3' and the text '隨便的密碼'. The 'Confirm passphrase' field also contains five dots. In the 'Actions' section, there are three buttons: 'Generate' (annotated with a red circle '1'), 'Load' (annotated with a red circle '4'), and 'Save private key' (annotated with a red circle '4'). The 'Parameters' section shows 'Type of key to generate' with 'RSA' selected, and 'Number of bits in a generated key' set to '2048'.

MobaXterm SSH Key Generator

File Key Conversions Help

Key

Public key for pasting into OpenSSH server (~/.ssh/authorized_keys file):

ssh-rsa
[Blurred public key text]
IMTD/9o

Key fingerprint: ssh-rsa 2048 53: [Blurred fingerprint text] 3:ed

Key comment: 2 fg6ts30 Google 使用者名稱

Key passphrase: 3 隨便的密碼

Confirm passphrase:

Actions

Generate a public/private key pair 1 Generate

Load an existing private key file Load

Save the generated key Save public key 4 Save private key

Parameters

Type of key to generate:
☒ RSA ☐ DSA ☐ ECDSA ☐ Ed25519 ☐ SSH-1 (RSA)

Number of bits in a generated key: 2048

■ 新增 SSH 金鑰

再來我們到 GCE 操作介面，點選
中繼資料 -> SSH 金鑰，點擊編輯

The screenshot shows the Google Cloud Platform console interface. The left sidebar contains a navigation menu with the following items: 執行個體群組, 執行個體群組, 健康狀態檢查, VM 管理員, OS 修補程式管理服務, 設定, 中繼資料 (highlighted with a red box and a red circle with the number 1), 區域, 網路端點群組, 作業, 安全掃描, 設定, Marketplace, 版本資訊, and a back arrow. The main content area is titled '中繼資料' (Metadata). It shows a section for '中繼資料' (Metadata) with a red circle with the number 2 and a red box around the '安全殼層金鑰' (SSH Key) link. Below this is a red circle with the number 3 and a red box around the '編輯' (Edit) button. The text below the buttons states: '這項專案中的所有執行個體都會繼承以下 SSH 金鑰 瞭解詳情' (All instances in this project inherit the following SSH key. Learn more). A table follows with the following data:

使用者名稱 ^	金鑰
fg6ts30	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAHpi..., "expireOn": "2021-04-25T12:01:05+0000"}
fg6ts30	ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTIt..., "expireOn": "2021-04-25T12:01:01+0000"}
fg6ts30	ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEArcmG...NJm1g45IwEpW4vBwDufNZ9q1m3v9Cw== fg6ts30

At the bottom of the table, there is a link '對等 REST' (Equivalent REST).

■ 新增 SSH 金鑰

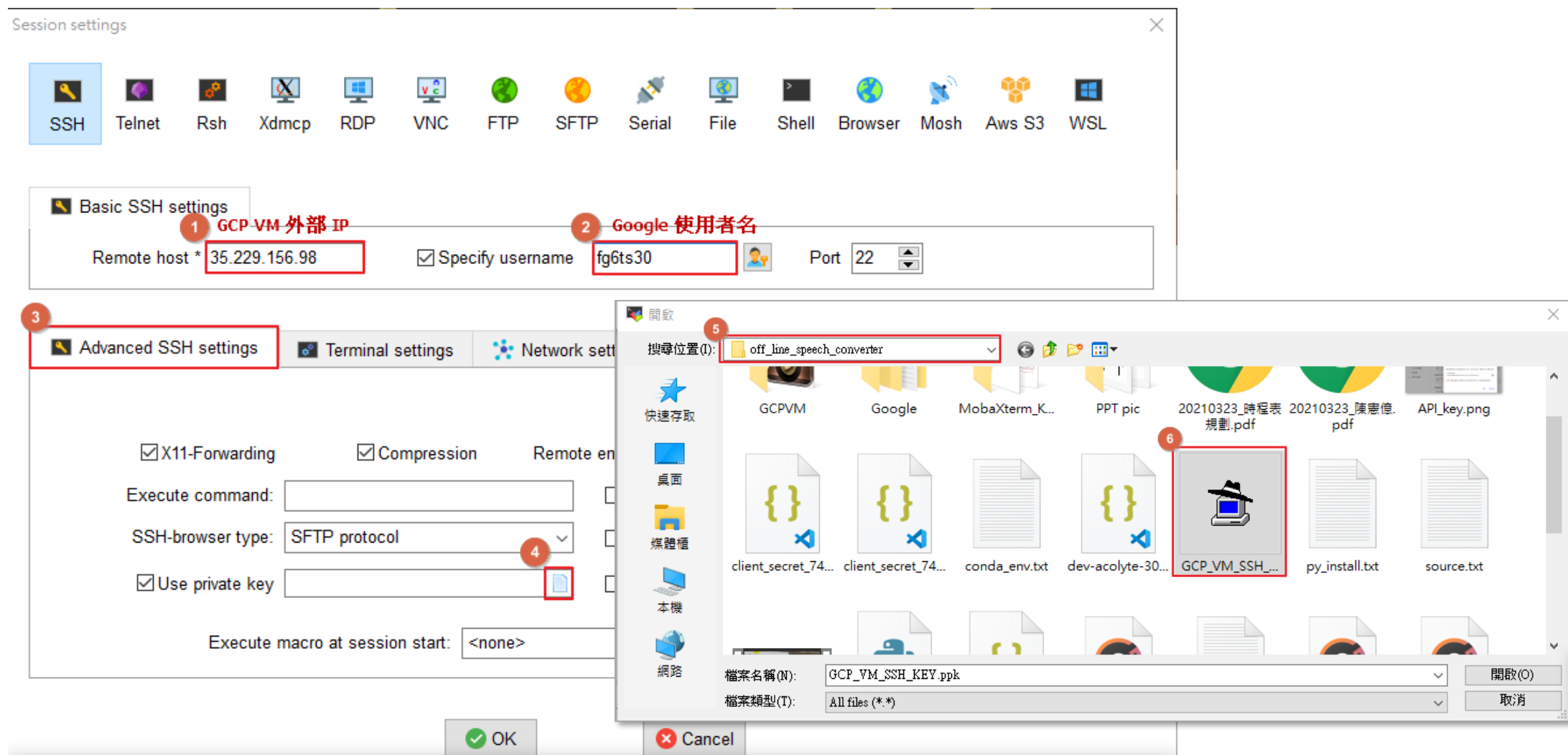
再來我們到 GCE 操作介面，點選

1. 中繼資料
2. SSH 金鑰
3. 點擊編輯
4. 新增項目
5. 剛剛產生的 SSH 金鑰貼上來
6. 儲存

The screenshot shows the Google Cloud Platform console interface. On the left, the 'Compute Engine' sidebar is visible. The '中繼資料' (Metadata) tab is selected. The main content area shows the '中繼資料' (Metadata) page. A modal window is open for editing the 'SSH keys' (SSH 金鑰). The modal shows a list of existing keys and a '新增項目' (Add item) button. The '儲存' (Save) button is highlighted.

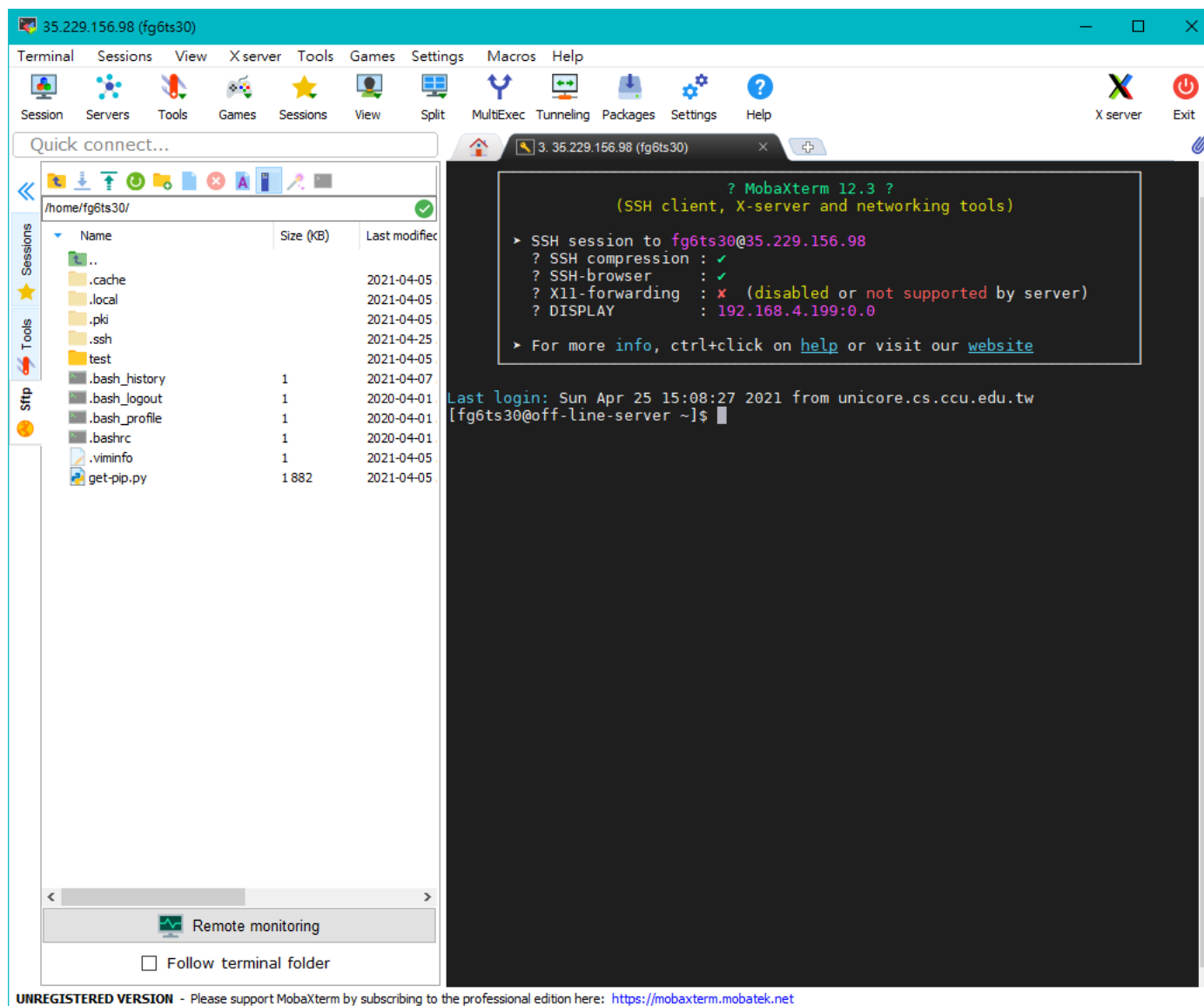
Google Cloud Platform console interface showing the 'Compute Engine' page. The '中繼資料' (Metadata) tab is selected. The 'SSH 金鑰' (SSH Keys) section is highlighted. A modal window is open for editing the 'SSH 金鑰' (SSH Keys). The modal shows a list of existing keys and a '新增項目' (Add item) button. The '儲存' (Save) button is highlighted.

Use private key



Use private key

按下 OK 並輸入密碼就登入成功，可以利用 MobaXterm 登入 GCP 的 VM 了

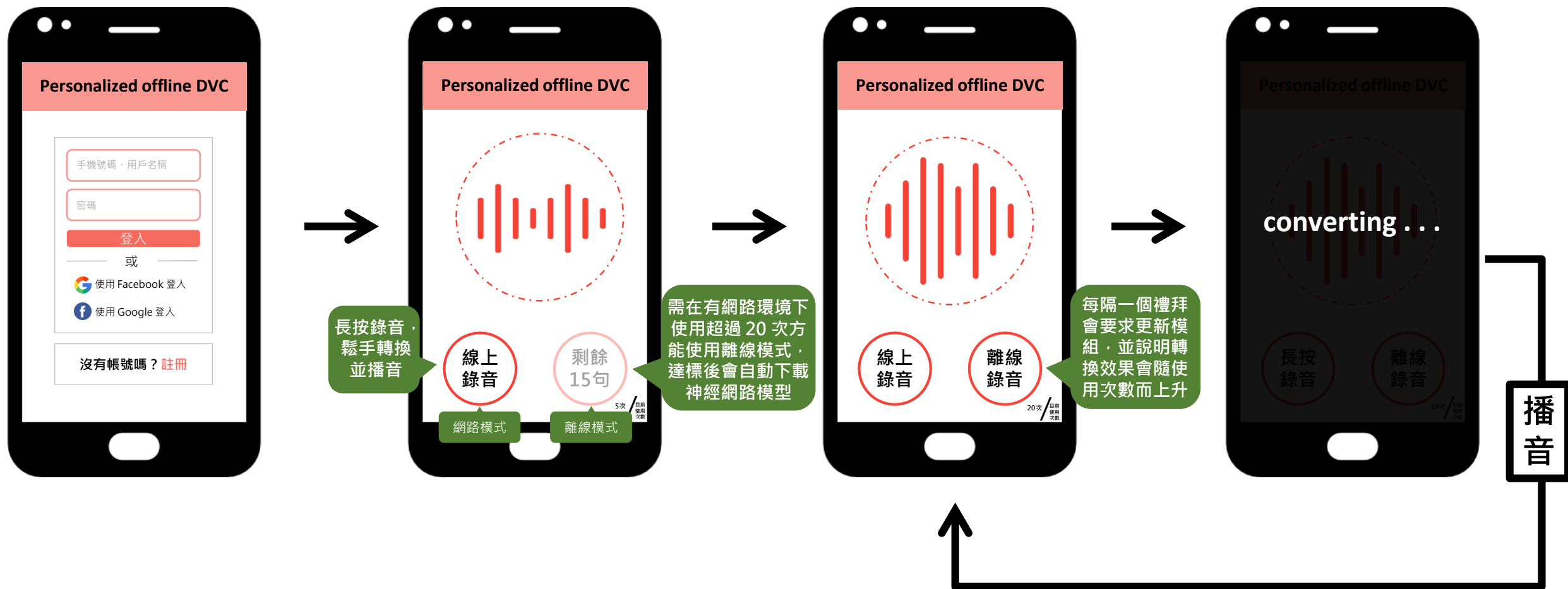


GCP VM

```
fg6ts30@off-line-server:~ - Google Chrome
ssh.cloud.google.com/projects/dev-acolyte-307405/zones/asia-east1-c/instances/off-line-server?authuser=1&hl=zh_T...
Downloading grpcio-1.36.1-cp36-cp36m-manylinux2014_x86_64.whl (4.1 MB)
| 4.1 MB 66.7 MB/s
Collecting keras-applications>=1.0.8
  Downloading Keras Applications-1.0.8-py3-none-any.whl (50 kB)
  | 50 kB 8.9 MB/s
Collecting gast==0.2.2
  Downloading gast-0.2.2.tar.gz (10 kB)
Collecting h5py
  Downloading h5py-3.1.0-cp36-cp36m-manylinux1_x86_64.whl (4.0 MB)
  | 4.0 MB 52.3 MB/s
Collecting setuptools>=41.0.0
  Downloading setuptools-54.2.0-py3-none-any.whl (785 kB)
  | 785 kB 53.0 MB/s
Collecting markdown>=2.6.8
  Downloading Markdown-3.3.4-py3-none-any.whl (97 kB)
  | 97 kB 9.6 MB/s
Collecting werkzeug>=0.11.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
  | 298 kB 66.6 MB/s
Collecting importlib-metadata
  Downloading importlib_metadata-3.10.0-py3-none-any.whl (14 kB)
Collecting cached-property
  Downloading cached_property-1.5.2-py2.py3-none-any.whl (7.6 kB)
Collecting zipp>=0.5
  Downloading zipp-3.4.1-py3-none-any.whl (5.2 kB)
Collecting typing-extensions>=3.6.4
  Downloading typing_extensions-3.7.4.3-py3-none-any.whl (22 kB)
Using legacy 'setup.py install' for gast, since package 'wheel' is not installed.
Using legacy 'setup.py install' for termcolor, since package 'wheel' is not installed.
Using legacy 'setup.py install' for wrapt, since package 'wheel' is not installed.
Installing collected packages: zipp, typing-extensions, six, numpy, importlib-metadata, cached-property, wheel, wer
kzeug, setuptools, protobuf, markdown, h5py, grpcio, absl-py, wrapt, termcolor, tensorflow-estimator, tensorboard,
opt-einsum, keras-preprocessing, keras-applications, google-pasta, gast, astor, tensorflow
  Running setup.py install for wrapt ... done
  Running setup.py install for termcolor ... done
  Running setup.py install for gast ... done
Successfully installed absl-py-0.12.0 astor-0.8.1 cached-property-1.5.2 gast-0.2.2 google-pasta-0.2.0 grpcio-1.36.1
h5py-3.1.0 importlib-metadata-3.10.0 keras-applications-1.0.8 keras-preprocessing-1.1.2 markdown-3.3.4 numpy-1.19.
5 opt-einsum-3.3.0 protobuf-3.15.7 setuptools-54.2.0 six-1.15.0 tensorboard-1.15.0 tensorflow-1.15.0 tensorflow-est
imator-1.15.1 termcolor-1.1.0 typing-extensions-3.7.4.3 werkzeug-1.0.1 wheel-0.36.2 wrapt-1.12.1 zipp-3.4.1
[fg6ts30@off-line-server ~]$
[fg6ts30@off-line-server ~]$
[fg6ts30@off-line-server ~]$
```

目前建置了 CentOS 7，並使其可以使用 pip install 套件

產品預想圖





■ 程式架構 (裝置端)

1. 麥克風輸入音檔
2. 是否連上網際網路

if (能連網) {

3. 將輸入的音檔上傳至雲端虛擬機
4. Speech-to-Text
5. Text-to-Speech
6. 將轉換後的音檔上傳至雲端虛擬機
- if(觸發條件)
7. 下載 DNN 模型參數
8. 寫 Python 或是 Java 版 DVC

} else if (不能連網) {

9. 使用儲存的 DNN 模型執行儲存的參數

}

10. 播放音檔

- 初次使用，請在有網路的狀況下至少說 20 句話再使用離線功能，原本想設定 320 句，但次數太多會降低使用意願
- 觸發條件 (按下 button 或是更新下載)
 - 模型參數手動下載or 自動下載
 - 轉換/下載期間跳出預估等待時間

■ 程式架構 (雲端)

1. 開 Google 虛擬機
2. 架設 YMDVC 執行環境
3. 利用裝置端錄製的聲音 & 轉換後的聲音自動進行 DTW 對齊
4. 將對齊後聲音進行自動訓練 (一個禮拜或是一個月一次)
5. 將取得的模型自動做資料整理 (模型轉換 pre/post norm, DNN model)
6. 將轉換完模型放進資料夾內方便裝置端存取

■ 參考文獻

Google Cloud 建立服務帳戶：

<https://support.google.com/cloudidentity/answer/7378726?hl=zh-Hant>

第一次開Google VM就上手 – Compute Engine操作簡介：

<https://blog.cloud-ace.tw/compute-engine/start-a-google-vm-compute-engine/>

使用其他 SSH 用戶端登入到 Google Compute Engine 的 Linux VM：

<https://dotblogs.com.tw/supershowwei/2017/08/23/120313>

在 CentOS 7 上安裝 Python3

<https://kirin.idv.tw/python-install-python3-in-centos7/>

■ 程式架構(1/3) (使用 **Python** 製作雛形)

```
麥克風輸入音檔           // speech_recognition.microphone() , 函式可使麥克風輸入
是否連上網際網路         // 使用系統指令 “os.system()” 函數 ping 網域 , 來測試是否
if ( 能連網 ) {           // 有網路 , 回傳 0 or 1
    upload                // SpeechRecognition 套件
    Speech-to-Text         // &
    Text-to-Speech         // gTTS 套件
    儲存音檔              // 使用 paramiko 套件使用 ssh 連線遠端 VM 複製檔案
    下載 DNN 模型參數
    寫 python 版 DVC1.0
} else if (不能連網){
    使用儲存的 DNN 1.0 模型執行儲存的參數
    // 寫 DVC 1.0 python 版本 , 需處理傳回來的參數
}
播放音檔                 // 使用 pygame 套件 mixer() 函式播放音檔
```

■ 程式架構(2/3) (使用 **Java** 實作手機 APP)

麥克風輸入音檔

// Android 提供 MediaRecorder 達成手機收音、儲存、播音

是否連上網際網路

// 加入權限，再使用內建提供函數判斷是否連網

if (能連網) {

Speech-to-Text

// google 有提供 libraries

Text-to-Speech

儲存音檔

下載 DNN 模型

// 利用 ssh 遠端連線到 Google VM 複製檔案

} else if (不能連網){

使用儲存的 DNN 1.0 模型執行儲存的參數

}

// 寫 DVC 1.0 java 版本，需處理傳回來的參數

播放音檔