**Terraform import review**

如同 [官方文件 import](https://www.terraform.io/docs/cli/import/index.html) 中描述

* Terraform 目前的 import 實作，只影響 state，不會產生 .tf resource。這是什麼意思？
* 要 import 已經存在的 infra，卻還要先寫 .tf resource 再 import，這樣不是很麻煩
* terraform import 本身設計目的就是在 state manipulation，而不是在處理 resource

然而，由於 import 既有的 remote reosurce，並產生 .tf resource 需求十分常見

* 例如中途導入既有的 infrastructure 就會一直做 terraform import
* 但實際想要的其實是 terraform import + .tf resource generation

terraformer 能符合正樣的需求

**terraformer**

[Terraformer github 專案](https://github.com/GoogleCloudPlatform/terraformer)

* 是一個 cli-tool
* reverse terraform (terraform: .tf -> infra, terraformer: infra -> .tf)
* 除了如 terraform import 產生 .state 以外
* 也會一併產生 .tf / .json

terraformer 雖然是放在 GoogleCloudPlatform organization 下，但不是一個 google 的產品，而是由社群維護的 terraform 逆向工具

terraformer

* 支援超多 providers，不管是主流公有雲，各家小雲，或是 saas 服務，都有支援

**terraformer install**

[Github Terraformer 的安裝會需要稍微注意](https://github.com/GoogleCloudPlatform/terraformer#installation)

* 由於 terraformer 也是會依賴各家的不同 api，透過中間不同的 provider，來產生實際與 remote resource API 互動的邏輯
* 例如： azure 的 vm 應該要如何透過 api 取得 vm 資料，取得資料後又要如何產生 .tf resource
* terraformer 核心是維護上述邏輯的抽象層，具體互動邏輯依賴 provider

除了有 terraformer 本身，要使用 azure 就要有 azure 的 provider plugin，以此類推

如果使用 source 來 build terraformer binary

* go build terraformer 時需要選擇需要的 provider build 即可

git clone git@github.com:GoogleCloudPlatform/terraformer.git

cd terraformer

go run build/main.go {azure}

或是使用事先 build 好的 release 版本

export PROVIDER=all

curl -LO https://github.com/GoogleCloudPlatform/terraformer/releases/download/$(curl -s https://api.github.com/repos/GoogleCloudPlatform/terraformer/releases/latest | grep tag\_name | cut -d '"' -f 4)/terraformer-${PROVIDER}-darwin-amd64

chmod +x terraformer-${PROVIDER}-darwin-amd64

sudo mv terraformer-${PROVIDER}-darwin-amd64 /usr/local/bin/terraformer

使用 terraformer

terraformer version

Terraformer v0.8.17

然後把 plugin binary 放到 ~/.terraform.d/plugins/{darwin,linux}\_amd64/ 目錄下，讓 terraformer 需要時去載入

wget https:*//releases.hashicorp.com/terraform-provider-azurerm/2.78.0/terraform-provider-azurerm\_2.78.0\_darwin\_amd64.zip*

unzip terraform-provider-azurerm\_2.78.0\_darwin\_amd64.zip

mkdir -p ~/.terraform.d/plugins/darwin\_amd64/

mv terraform-provider-azurerm\_v2.78.0\_x5 ~/.terraform.d/plugins/darwin\_amd64/

**example**

我們開一個新 root module 作為 terraformer import 的工作目錄

mkdir azure/foundation/virtual\_network\_terraformer

cd azure/foundation/virtual\_network\_terraformer

然後來 [Github Terraformer azure doc](https://github.com/GoogleCloudPlatform/terraformer/blob/master/docs/azure.md" \t "_blank) 來查詢支援 import 的 resource

* 有些新的功能可能還不支援 terraform，或是 terraformer 還不支援，就不會出現
* subnet 是支援的

依照文件所述，這邊使用 service principal 來 auth

* 也就是暫時使用 terraform 的 service principal，權限與 terraform 一樣
* 使用 terraform service principal 進行 auth

參照[前面 09-iam-for-terraform 為 terraform 配置專屬 service principal 的步驟](https://ithelp.ithome.com.tw/articles/09-iam-for-terraform" \t "_blank)，取得 auth 相關變數

APP\_NAME=terraform-30-days

az ad sp list --display-name ${APP\_NAME}

TENANT\_ID=$(az ad sp list --display-name ${APP\_NAME} | jq -r '.[0].appOwnerTenantId')

SERVICE\_NAME=$(az ad sp list --display-name ${APP\_NAME} | jq -r '.[0].servicePrincipalNames[0]')

az login --service-principal \

--username ${SERVICE\_NAME} \

--tenant ${TENANT\_ID} \

--password ~/.ssh/terraform-30-days.keycrt > /tmp/azure-login-profile

export ARM\_CLIENT\_ID=$(cat /tmp/azure-login-profile | jq -r '.[0].user.name')

export ARM\_CLIENT\_CERTIFICATE\_PATH="/Users/che-chia/.ssh/terraform-30-days.pfx"

export ARM\_CLIENT\_CERTIFICATE\_PASSWORD=

export ARM\_SUBSCRIPTION\_ID=$(az account subscription list | jq -r '.[0].subscriptionId')

export ARM\_TENANT\_ID=$(cat /tmp/azure-login-profile | jq -r '.[0].tenantId')

將 auth export 到環境變數後，就可以使用 terraformer

terraformer import azure -h

terraformer import azure list

analysis

app\_service

container

cosmosdb

data\_factory

database

databricks

disk

dns

eventhub

keyvault

load\_balancer

network\_interface

network\_security\_group

private\_dns

private\_endpoint

public\_ip

purview

redis

resource\_group

scaleset

security\_center\_contact

security\_center\_subscription\_pricing

storage\_account

storage\_blob

storage\_container

subnet

synapse

virtual\_machine

virtual\_network

terraformer import azure -r subnet

2021/09/25 23:00:52 Testing if Service Principal / Client Certificate is applicable for Authentication..

2021/09/25 23:00:52 Using Service Principal / Client Certificate for Authentication

2021/09/25 23:00:52 Getting OAuth config for endpoint https:*//login.microsoftonline.com/ with tenant 5dc1c3ed-d350-4c3b-ba3d-db5ac4bfe072*

2021/09/25 23:01:36 Testing if Service Principal / Client Certificate is applicable for Authentication..

2021/09/25 23:01:36 Using Service Principal / Client Certificate for Authentication

2021/09/25 23:01:36 Getting OAuth config for endpoint https:*//login.microsoftonline.com/ with tenant 5dc1c3ed-d350-4c3b-ba3d-db5ac4bfe072*

2021/09/25 23:01:41 azurerm importing... subnet

2021/09/25 23:01:48 azurerm done importing subnet

2021/09/25 23:01:48 Number of resources for service subnet: 7

2021/09/25 23:01:48 Refreshing state... azurerm\_subnet.tfer--dev-002D-3

2021/09/25 23:01:48 Refreshing state... azurerm\_subnet.tfer--dev-002D-2

2021/09/25 23:01:48 Refreshing state... azurerm\_subnet\_network\_security\_group\_association.tfer--aks-002D-subnet\_network\_security\_group\_association

2021/09/25 23:01:48 Refreshing state... azurerm\_subnet\_network\_security\_group\_association.tfer--base-002D-external\_network\_security\_group\_association

2021/09/25 23:01:48 Refreshing state... azurerm\_subnet.tfer--dev-002D-1

2021/09/25 23:01:48 Refreshing state... azurerm\_subnet.tfer--base-002D-external

2021/09/25 23:01:48 Refreshing state... azurerm\_subnet.tfer--aks-002D-subnet

2021/09/25 23:01:53 Filtered number of resources for service subnet: 7

2021/09/25 23:01:53 azurerm Connecting....

2021/09/25 23:01:53 azurerm save subnet

2021/09/25 23:01:53 azurerm save tfstate for subnet

**import results**

cd azure/foundation/compute\_network\_terraformer

tree

.

└── generated

└── azurerm

└── subnet

├── outputs.tf

├── provider.tf

├── subnet.tf

├── subnet\_network\_security\_group\_association.tf

├── terraform.tfstate

└── variables.tf

內容是在 terraform service principal 中可見的 subnets，已經變成 .tf，然而內容還需要整理

**don't commit generated files**

不要 commit 產生的 generated 資料夾到 git 上

* state 檔案不 commit，內有 sensitive data
* .tf 檔案都還蓄要整理內容

將 generated 加到 .gitignore 中