**Terraform state rm**

<https://www.terraform.io/docs/cli/commands/state/rm.html>

如同[Terraform 官方文件](https://www.terraform.io/docs/cli/commands/state/rm.html" \t "_blank) 所述，我們再比較三個部分的差異

* .tf 維持原狀
* state rm 將已經存在的 terraform state 移除
* remote resource 仍然存在

state 移除消失，對 terraform 會有什麼影響？回想一下 state 的作用

* 內含有 remote resource 的 metadata (例如：id)
* 內含有 terraform 的 intermediate data

將上面兩個都移除，也就是 terraform 會"忘記" 這個 resource 曾經存在過

* 如果 remote resource 仍然存在，terraform 也沒有這個 remote reosurce 的連結
* remote resource 會變成孤兒 orphan resource，無法再透過 terraform 管理
* 如果 .tf 內仍然有 resource，plan 與 apply 也只會 create 新的 resource

由於上述原因，我們只有在非常少數的狀況下，才會使用 state rm，這是最後手段，通常是 state 與 provider 出現嚴重錯誤，導致 state 無法被正確移除時，我們才手動操作

* 而且會先確認 remote reosurce 有正確的被移除
* 不然 orphan resources 仍然是要收錢的，而且會在看不到的地方影響其他正常 resource 的運作

**Let's do state rm**

一樣回到 azure/foundation/compute\_network 的範例

cd azure/foundation/compute\_network

terragrunt state list

module.network.data.azurerm\_resource\_group.network

module.network.azurerm\_subnet.subnet[0]

module.network.azurerm\_subnet.subnet[1]

module.network.azurerm\_subnet.subnet[2]

module.network.azurerm\_virtual\_network.vnet

我們可以針對其中一個 resource addressing 做 state rm  
或是針對整個 module 做 rm

針對整個 module 做 rm 的話，之後要復原的 address 數量就會比較多  
我們這邊為了示範 rm 而做 rm，並沒有實際的需求，所以只示範其中一個 subnet rm

一樣能 dry-run 先 dry-run

terragrunt state rm --dry-run "module.network.azurerm\_subnet.subnet[0]"

Would remove "module.network.azurerm\_subnet.subnet[0]"

然後 state rm

terragrunt state rm --dry-run "module.network.azurerm\_subnet.subnet[0]"

Removed module.network.azurerm\_subnet.subnet[0]

Successfully removed 1 resource instance(s).

state list 可以看到 state address 已經消失了

terragrunt state list

module.network.data.azurerm\_resource\_group.network

module.network.azurerm\_subnet.subnet[1]

module.network.azurerm\_subnet.subnet[2]

module.network.azurerm\_virtual\_network.vnet

嘗試 plan

terragrunt plan

Terraform used the selected providers to generate the following execution

plan. Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

*# module.network.azurerm\_subnet.subnet[0] will be created*

+ resource "azurerm\_subnet" "subnet" {

+ address\_prefix = (known after apply)

+ address\_prefixes = [

+ "10.2.1.0/24",

]

+ enforce\_private\_link\_endpoint\_network\_policies = false

+ enforce\_private\_link\_service\_network\_policies = false

+ id = (known after apply)

+ name = "dev-1"

+ resource\_group\_name = "terraform-30-days"

+ service\_endpoints = []

+ virtual\_network\_name = "acctvnet"

}

Plan: 1 to add, 0 to change, 0 to destroy.

terraform 覺得應該要 create，增加一個

* 由於 .tf 中仍然有 module.network.azurerm\_subnet.subnet[0]
* 而 state 中沒有
* 然而透過 azure web console 仍然可見 subnet 存在，我們只 rm state，remote resource 仍然存在
* 此時 module.network.azurerm\_subnet.subnet[0] 雖然是 terraform 產生的，但由於 terraform 移除 state 後，已經失去所有與 remote resource 的關聯，也就是忘記有這個 subnet 存在，這個 subnet 已經是 orphan resource，無法透過 terraform 管理了

**az-cli list**

透過 az-cli 查詢實際的 vnet / subnet 狀態

* 發現 subnet/dev-1 確實存在

az network vnet list

{}

az network vnet list | jq '.[].id'

"/subscriptions/.../resourceGroups/MC\_base\_general\_southeastasia/providers/Microsoft.Network/virtualNetworks/aks-vnet-39532258"

"/subscriptions/.../resourceGroups/terraform-30-days/providers/Microsoft.Network/virtualNetworks/acctvnet"

az network vnet subnet list -g terraform-30-days --vnet-name acctvnet

{}

az network vnet subnet list -g terraform-30-days --vnet-name acctvnet | jq '.[].id'

"/subscriptions/.../resourceGroups/terraform-30-days/providers/Microsoft.Network/virtualNetworks/acctvnet/subnets/dev-1"

"/subscriptions/.../resourceGroups/terraform-30-days/providers/Microsoft.Network/virtualNetworks/acctvnet/subnets/dev-3"

"/subscriptions/.../resourceGroups/terraform-30-days/providers/Microsoft.Network/virtualNetworks/acctvnet/subnets/dev-2"

**how to recover? how to un-do state rm**

那如何把 state rm 掉的 subnet 加回來？我們可以嘗試 terraform apply

terragrunt apply

Terraform will perform the following actions:

*# module.network.azurerm\_subnet.subnet[0] will be created*

+ resource "azurerm\_subnet" "subnet" {

+ address\_prefix = (known after apply)

+ address\_prefixes = [

+ "10.2.1.0/24",

]

+ enforce\_private\_link\_endpoint\_network\_policies = false

+ enforce\_private\_link\_service\_network\_policies = false

+ id = (known after apply)

+ name = "dev-1"

+ resource\_group\_name = "terraform-30-days"

+ service\_endpoints = []

+ virtual\_network\_name = "acctvnet"

}

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?

Terraform will perform the actions described above.

Only 'yes' will be accepted to approve.

Enter a value: yes

module.network.azurerm\_subnet.subnet[0]: Creating...

╷

│ Error: A resource with the ID "/subscriptions/6fce7237-7e8e-4053-8e7d-ecf8a7c392ce/resourceGroups/terraform-30-days/providers/Microsoft.Network/virtualNetworks/acctvnet/subnets/dev-1" already exists - to be managed via Terraform this resource needs to be imported into the State. Please see the resource documentation for "azurerm\_subnet" for more information.

│

│ with module.network.azurerm\_subnet.subnet[0],

│ on .terraform/modules/network/main.tf line 15, in resource "azurerm\_subnet" "subnet":

│ 15: resource "azurerm\_subnet" "subnet" {

│

╵

ERRO[0129] 1 error occurred:

\* exit status 1

terragrunt apply 時，azure API 回傳 error

* virtualNetworks/acctvnet/subnets/dev-1 依然存在，terraform 想 apply 一個相同 name (id) 的 resource，自然會失敗

雖然我們的需求是希望把移除的 state 加回來，但是terrafrom 並沒有"把移除的 state 加回來“的概念

* 記得 state 基本概念中提到的，state 就是 .tf resource 與 remote resource 的連結
* state rm 強迫 terraform 遺忘 state 存在
* 換作是其他的 module，就算可以 apply 成功，也是 create 一個新的，原本的 resource 仍然留在原地，變成有兩個 remote resource，一個可以透過 terraform 管理，另外一個是 orphan

**So really, how to fix this?**

那現在整個 azure/foundation/compute\_network 不能用了，同事要氣噗噗了ＸＤ，該要怎麼修復？

error 後面有一句訊息

to be managed via Terraform this resource needs to be imported into the State. Please see the resource documentation for "azurerm\_subnet" for more information.

意思是

* terraform 嘗試 apply，卻找到相同 name 的 resource
* terraform 建議可以把這個 terraform 不認得的 resource，import 到 terraform 中管理
* 從既有的 remote resource，建立 remote resource 到 .tf resource 的連結

具體該如何操作？我們看下一節課 terraform state import