**NOTE**

* 課程範例會超出永久免費額度，會消耗 azure credit，用完請 destroy
* 範例會產出 kubeconfig 檔案，內含可以存取 cluster control panel 的 private key，請妥善保存
* 範例 AKS 是功能完整的 AKS，但不適合直接做 production 環境使用
  + 需要 補齊 security hardening

**Prerequisite & Steps**

* infrastructure 本身的知識
  + [了解 subnet](https://ithelp.ithome.com.tw/articles/10266085)
  + [了解 AKS](https://ithelp.ithome.com.tw/articles/10266085)
  + 理解 infrastructure 的參數。terraform 的文件並不會說明 resource 參數的細節，還是要對照 cloud provider 的文件。
* terraform 官方文件，查詢module 的使用方法，參數格式
  + 理解 AKS module 的 inputs, outputs
* 根據需求調整 module 參數
  + 使用 meta-argument, built-in functions 協助管理 module
* plan & apply 完整的 module

**Requirements & spec**

今天的範例內容在此 azure/dev/southeastasia/chechia\_net/kubernetes

* 假設需求是「為 [https://chechia.net](https://chechia.net/) 的後端架設 AKS 集群」
* 測試用的 dev 環境
* 網站的主要用戶在東南亞，所以 location 放在 southeastasia
* 路徑上放上 chechia\_net 路徑，把相關的 resource root module 放進來

**Content**

養成好習慣：使用任何 terraform module 時都務必檢查一下內容物

*# azure/dev/southeastasia/chechia\_net/kubernetes/terragrunt.hcl*

terraform {

source = "../../../../..//azure/modules/kubernetes\_cluster"

}

dependency "network"{

config\_path = find\_in\_parent\_folders("azure/foundation/compute\_network")

}

inputs = {

...

kubernetes\_cluster\_name = "terraform-30-days"

default\_node\_pool\_vm\_size = "Standard\_D2\_v2" # This is beyond 12 months free quota

default\_node\_pool\_count = 1

network = dependency.network.outputs.vnet\_name # acctvnet

subnet = dependency.network.outputs.vnet\_subnets[2] # dev-3

kubeconfig\_output\_path = pathexpand("~/.kube/azure-aks-terraform-30-days")

spot\_node\_pools = {

spot = {

vm\_size = "Standard\_D2\_v2"

node\_count = 1

...

}

}

}

terragrunt.hcl

* 定義 source 到本地的 module，稍後要來看 module 內容
* 定義 dependency，說明這個 aks 依賴 network
* 定義 kubeconfig output path，這個是 apply 之後的輸出產物
  + aks 建立後，會產出 kubeconfig，包含 cluster 資訊與 credential
  + 使用 kubeconfig 可以存取 aks，是敏感資料，所以要放到本地安全的地方
* inputs
  + aks 本身的 input 參數
    - 使用 vm size 是超出 12 個月免費額度的，會扣 credit
    - network, subnet, ...
  + aks 的下一層 node pool 的 input 參數
    - 使用 vm size 是超出 12 個月免費額度的，會扣 credit
    - node pool 的 vm size, node count, ...等

把不同層的參數放在最上層傳入，方便使用，但會降低可讀性（不同 resource 的參數混雜）

* 以程式語言類比，相當於上層 function 調用其他 function，把 variable 透過 function argument 送到最上層傳入
* coding style 可以團隊討論是否要這樣寫

**module content**

看一下 modules

tree -L 1 azure/modules/kubernetes\_cluster

azure/modules/kubernetes\_cluster

├── README.md

├── client\_config.tf

├── kubeconfig.tf

├── kubernetes\_cluster.tf

├── node\_pool.tf

├── output.tf

└── variables.tf

0 directories, 7 files

[透過client\_config.tf 取得 terraform 呼叫時的 provider 中的 config](https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/data-sources/client_config)

* [data block {} 與 resource block 不同，細節請見第？章](https://ithelp.ithome.com.tw/articles/10266085)

# client\_config.tf

data "azurerm\_client\_config" "current" {}

kubeconfig.tf 是透過 local\_file resource，將建立 cluster 後的 cluster config 與 credential 存入本地檔案

* [kubeconfig 的細節可以參見 k8s 官方文件](https://kubernetes.io/docs/concepts/configuration/organize-cluster-access-kubeconfig/)

kubernetes\_cluster.tf 描述 azurerm\_kubernetes\_cluster resource，以及 aks 本身依賴的其他 resource

* 例如：要不要開 log analytics，這個是另外一個 resource（可能在 Azure 上也是另外一隻獨立的 API）
* kubernetes\_cluster.tf 使用了 [terraform resource meta-argument: count](https://ithelp.ithome.com.tw/articles/10266085) 與 [conditional expression:](https://ithelp.ithome.com.tw/articles/10266085)，有空我們後面細講

**how to create module**

azure/modules/kubernetes\_cluster 是如何寫出來的？其實就是圍繞需求，慢慢補齊 module 的功能

* 查找 terraform aks resource 的文件，參考範例先出 kubernetes\_cluster.tf
* 在 azure/dev/southeastasia/chechia\_net/kubernetes 嘗試 plan 與 apply
  + 檢查結果，做功能性測試，這個 module 是否能用，有無缺什麼設定與參數？...
* 繼續修改，例如
  + 增加 node\_pool.tf
  + 使用 [spot instance](https://ithelp.ithome.com.tw/articles/10266085) 作為 node
  + 發現會有多個 node\_pool 所以使用了 [for\_each 語法(細節請見第？章)](https://ithelp.ithome.com.tw/articles/10266085)
* 過程中有需要傳入的參數，就寫在 variables.tf 讓最上層呼叫的時候傳入

由於我們的設計是會有 dev-aks, stag-aks, prod-aks 使用相同的 module 產生，所以要把不同環境下的不同參數，傳到最上層的 terragunt.hcl

* 實際上線，可能會有底下幾個環境，透過不同路徑的 terragrunt.hcl 產生
  + azure/dev/southeastasia/chechia\_net/kubernetes/terragrunt.hcl
  + azure/stag/southeastasia/chechia\_net/kubernetes/terragrunt.hcl
  + azure/prod/southeastasia/chechia\_net/kubernetes/terragrunt.hcl
* 不同環境使用相同 module 產生，可以摻生多個環境的 infrastructure
  + 有測試的 dev, stag 之後才會上到 prod
  + 降低多環境維運的常見問題：dev 會動但是 prod 不會動
  + 將低不同環境的維護成本

**自幹的 AKS module**

* 一組 vpn subnets，將 node group 放在 vpn subnet 上
  + foundation/compute\_network 已經把 vpn subnets 產生出來了
    - 使用 terragrunt dependency 來產生兩個 root module 的依賴
    - 將 network 中的 output 作為 kubernetes cluster 的 input 傳入
* 一個 aks cluster
  + AKS cluster 是代管的 control panel，費用為 $0.1 / hr
    - [Azure Kubernetes Service Pricing](https://azure.microsoft.com/en-us/pricing/details/kubernetes-service?WT.mc_id=AZ-MVP-5003985)
  + 一個 default on-demand node group，創建時必須產生
  + (Optional) 一個 on-demand node group
  + (Optional) 一個 spot node group

**Init, Plan & Apply**

看過內容，變來實際跑看看

* 好習慣不怕提醒：apply 前仔細看，apply 下去就都要收費了XD

cd azure/dev/southeastasia/chechia\_net/kubernetes

terragrunt init

terragrunt plan

terragrunt apply

apply 時間較長

* aks kubernetes control panel 的創建時間約為 2-3 mins
* aks node group 的創建時間約為 2-3 mins
* 由於 node group depends on control panel，所以無法平行創造，整體花費時間會拉長

**access ake with kubectl**

不熟悉 AKS / Kubernetes 的朋友可以搭配azure official doc: [Create a Kubernetes cluster with Azure Kubernetes Service using Terraform](https://docs.microsoft.com/zh-tw/azure/developer/terraform/create-k8s-cluster-with-tf-and-aks?WT.mc_id=AZ-MVP-5003985)

安裝 [kubectl](https://kubernetes.io/docs/tasks/tools/install-kubectl-macos/)

* terraform module apply 同時已經將 aks kubeconfig 寫在 ~/.kube/azure-aks 路徑
* 設定 kubeconfig 路徑~/.kube/azure-aks
* 檢查一下 kubeconfig 內容
* 使用 kubectl access k8s

cat ~/.kube/azure-aks

KUBECONFIG\_OUTPUT\_PATH="/Users/che-chia/.kube/azure-aks"

kubectl *--kubeconfig ${KUBECONFIG\_OUTPUT\_PATH} cluster-info*

kubectl *--kubeconfig ${KUBECONFIG\_OUTPUT\_PATH} get node*

NAME STATUS ROLES AGE VERSION

aks-default-44401806-vmss000000 Ready agent 9m4s v1.20.7

可以使用 kubectl 控制 cluster 就成功了

**Module Known Issues**

這個自幹的 AKS module 有著以下問題

* variable type(any) 這個蠻糟的XD
* tfsec security issues

自幹的 module 通常會有比較多問題，所以安全性的掃描工具（ex. tfsec）是十分必要的

或是就不要自幹，使用社群維護的 module 版本

**Alternative: AKS Example**

實務上，除了自己寫 module 外使用，也可以直接使用開源的 module

以上面的 AKS 範例，可以使用 [azurerm AKS module](https://registry.terraform.io/modules/Azure/aks/azurerm/latest)

使用開源的 module，有幾個條件

* 作者/團隊是有名的，或是在 terraform registry 官方認證的
  + module 更安全，更新更穩定，bug 少
* 務必挑選經常更新的 module 使用
  + 如果是沒在維護的 module ，使用後才會發現沒有更新，最後還是要自己刻一版
* 仍然需要看完完整 module 內容
  + 沒看懂內容就 apply 到雲端上蠻危險的。除了惡意的 module 內容，也有可能搞錯原本設計的用途，導致誤用

建議常見的 resource 與泛用性高的基本架構可以使用社群

專門為公司服務打造的上層 infrastructure，可以自幹，放在私有 repository，並自己維護

* 例如我的 [https://chechia.net](https://chechia.net/)
  + 可能有 network Vnet，後端 AKS，資料庫 DB，前端 VM scaleSets (Scaling Group)，firewall rules,....
  + 將這些東西打包成上層 terraform module，方便管理，傳遞 module 間的參數，也建立彼此的依賴關係
  + 底下則是調用社群維護的 module，只要固定維護升級 module 就好

**Open Source modules**

合適的開源 module，除了 google terraform module aks 以外，可以到以下地方尋找

* [terraform registry}()
* [github: search terraform module xxx](https://ithelp.ithome.com.tw/articles/10266085)

public cloud provider 都有出自家的 module，方便用戶使用

* [terraform official module](https://ithelp.ithome.com.tw/articles/10266085)
* [aws official module](https://ithelp.ithome.com.tw/articles/10266085)
* [azure official module](https://ithelp.ithome.com.tw/articles/10266085)
* [gcp official module](https://ithelp.ithome.com.tw/articles/10266085)

**Cleanup**

destroy 整座 cluster

**Homework**

* 繼續嘗試 azure/modules/kubernetes\_cluster
  + 新增 spot node group
* 使用 [azurerm AKS module](https://registry.terraform.io/modules/Azure/aks/azurerm/latest) 部署 AKS
  + 閱讀 README.md 以及範例
  + 調整 inputs
  + plan, apply 創建 AKS
  + 比較使用 azurerm 維護的 module，與講者隨手做的 module
* 增加 firewall rule
  + 限制可以存取 master 的 CIDR

**Summary**

本章節主要是代大家走過一次實務的開發流程

* 查 cloud provider 文件
* 查 terraform resource 文件
* init, plan, apply, test, fix 然後不斷迭代改進
* 可以善用社群維護的 module
* 務必使用安全性掃描工具