內容

[1. Istio with MiniKube 研究紀錄 2](#_Toc79222864)

[1.1. 簡述 2](#_Toc79222865)

[1.2. 使用軟體清單(以Win10為例) 2](#_Toc79222866)

[1.3. Docker destop軟體安裝 3](#_Toc79222867)

[1.4. Minikube軟體安裝 4](#_Toc79222868)

[1.5. MiniKube基本操作 8](#_Toc79222869)

[1.6. 安裝Win-bash (非必要) 10](#_Toc79222870)

[1.7. MiniKube + Istio 官網Lab 17](#_Toc79222871)

[2. Istio 版本 vs K8s版本 26](#_Toc79222872)

[3. Istio JWT實作模擬Lab 28](#_Toc79222873)

[3.1. 產生公私Key步驟 28](#_Toc79222874)

# Istio with MiniKube 研究紀錄

## 簡述

※Istio (Service Mesh)官網簡介

<https://istio.io/latest/about/service-mesh/>

※Minikube

One Node的Kubernetes Cluster，可以安裝在本機上，支援Windows 與Mac Minikube 。

方便安裝，快速使用，很適合初階學習。

※目標於本機，安裝所需相關軟體，學習與熟悉minikube、istio相關技術。

## 使用軟體清單(以Win10為例)

|  |  |  |
| --- | --- | --- |
| 項目 | 版本 | 備註 |
| OS | windows 10 (Pro) |  |
| Docker Destop | 3.4.0 |  |
| Minikube | 1.21.0 |  |
| Cygwin64 或 Win bash |  | 讓windows 可以執行 linux like 之 commad  因很多介紹網頁範例都是 linux 在windows cmd下需調整，搭配方便執行 |
| Istio | 1.10.2 | istioctl.exe  istio 套件安裝於 minikube |
| Istio Sample |  | Istio sample application 測試用 |
|  |  |  |
| 備註：  開始測試時，因為機器之前已經裝過相關軟體minikube、docker、、、版本未更新，執行起來有出現一些錯誤訊息，可能為 不同軟體版本差異太遠或其他限制，建議全部重裝或更新到最新版本會比較順。 | | |

## Docker destop軟體安裝

使用 minikube 請同時先安裝docker for windows

1. 下載網址：

<https://www.docker.com/products/docker-desktop>

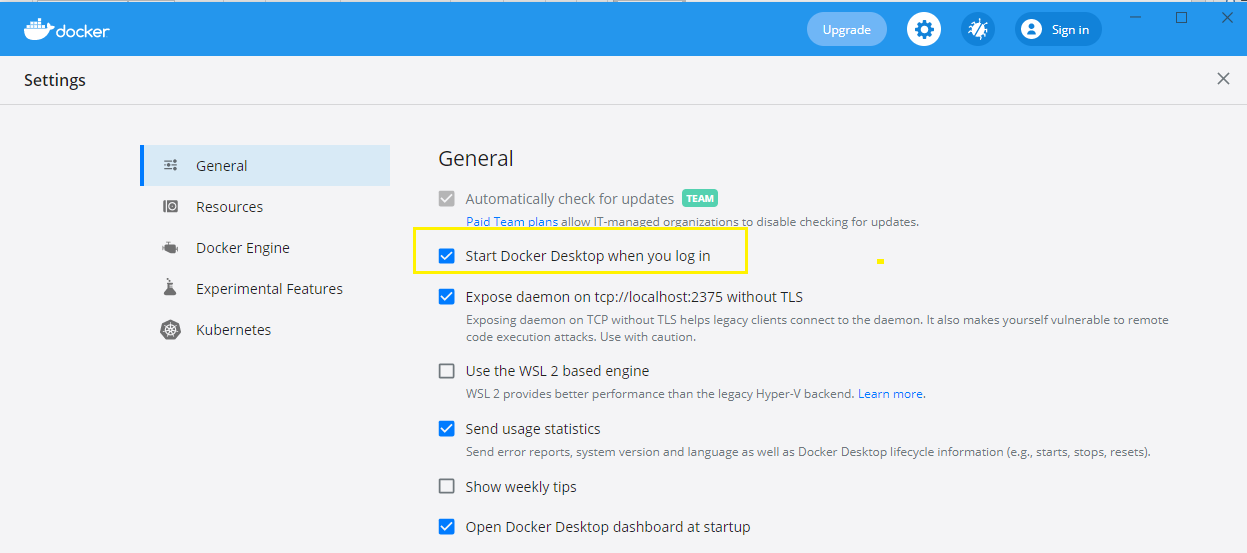
皆已預設安裝步驟進行安裝。

安裝完成後，用 cmd提示字元 執行命令，驗證顯示版本

|  |
| --- |
| docker -v |



注意重開機以後，是否設定自動啟動。如果沒有的話需要手動啟動docker destop。



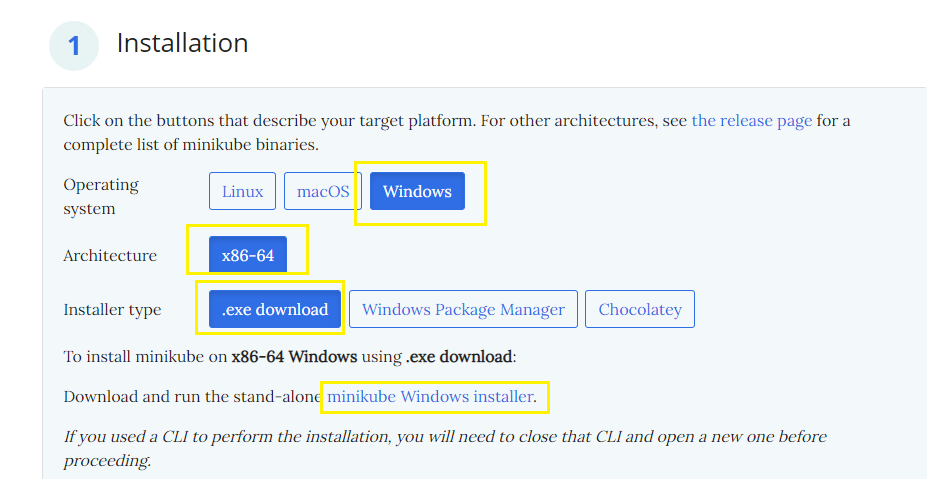
※docker 常用指令

https://joshhu.gitbooks.io/dockercommands/content/Containers/ContainersBasic.html

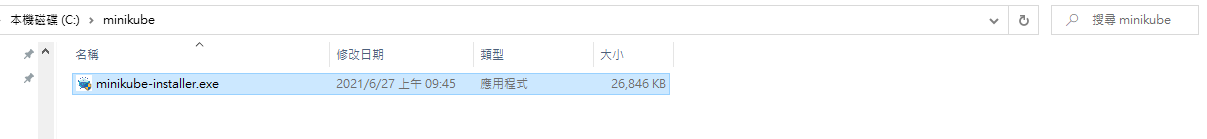
## Minikube軟體安裝

1. 下載網址：

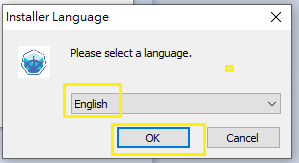
<https://minikube.sigs.k8s.io/docs/start/>



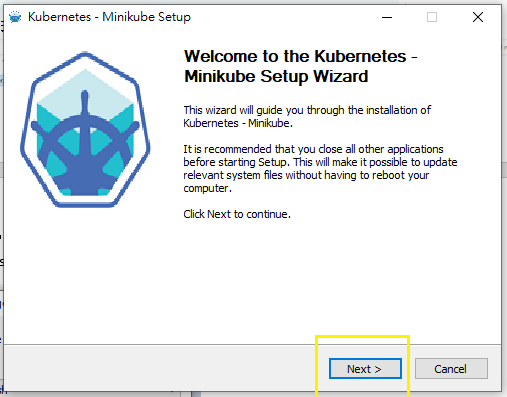
1. 執行安裝程式



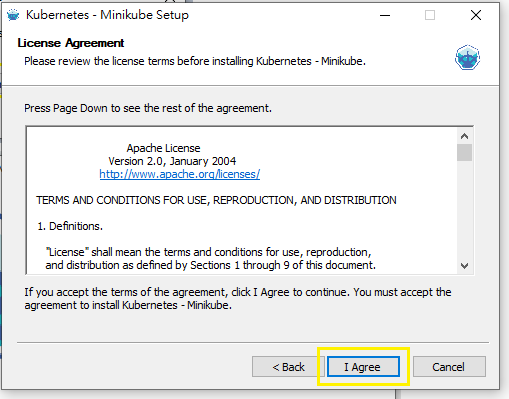
1. 允許變更電腦，點『Y』
2. 選擇語言 English，點『OK』



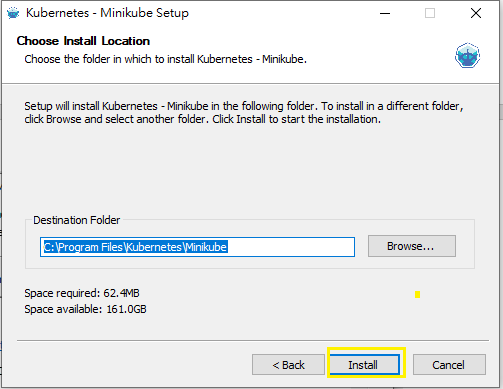
1. 點『Next』



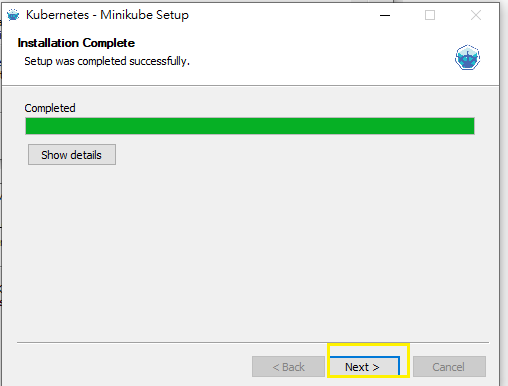
1. 點『I Agree』



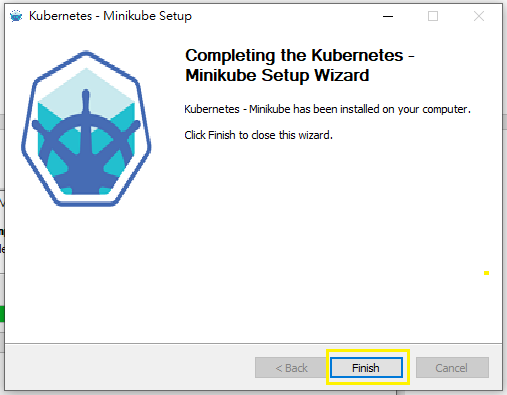
1. 點『Install』



1. 點『Next』



1. 點『Finish』



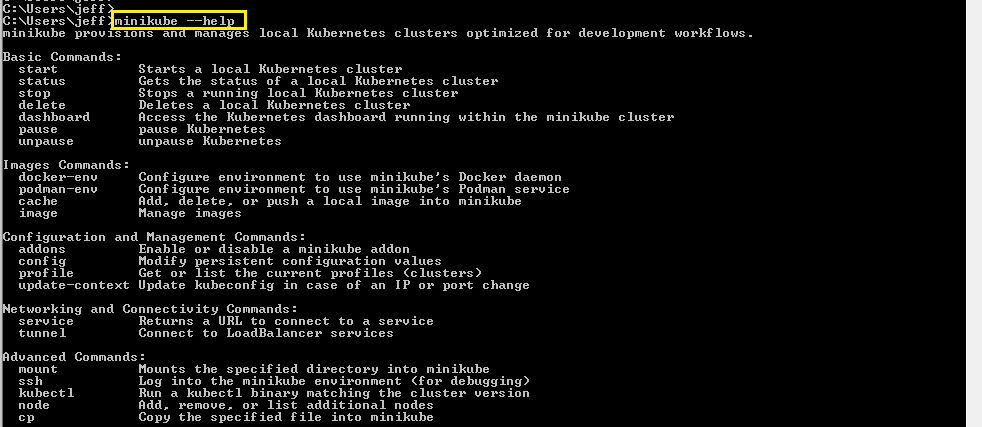
## MiniKube基本操作

### 命令執行方式

開啟『命令提示字元』，輸入 minikube ………

Ex : 看Help

|  |
| --- |
| minikube --help |

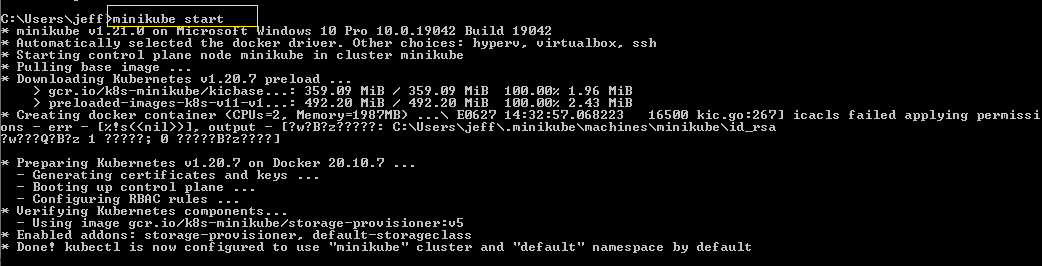


### 啟動minikube

minikube start (基本用法, 後面可以帶Option參數)

|  |
| --- |
| minikube start --memory 6144 --cpus 2 --kubernetes-version v1.16.0  // Option 指定  //--kubernetes-version 版本，可不給 用預設 (但要注意istio 跟 k8s 支援對應版本)  //--memory 8192 --cpus 2 |

過程中 minikube 可能會存取docker，docker會提示 是否要分享，點『是』



※ Start過程中有出現錯誤 icals ，沒找到解法先paas

※ 啟動完會有暫存檔放置於 C:\Users\使用者ID\.minikube

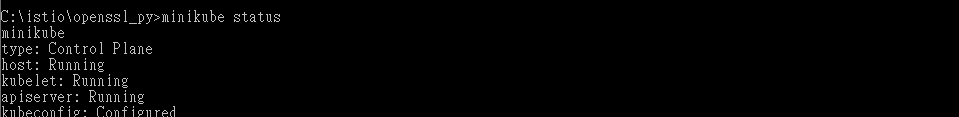
如果要專案都不要，清乾淨可以刪除該目錄下所有檔案。

* 跑不一樣的AP應用時，有時候資源需要比較大，啟動時加大資源，不然會有不明的異常或crash

### Minikube常用命令

※查看狀態

|  |
| --- |
| minikube status |



※停止minikube

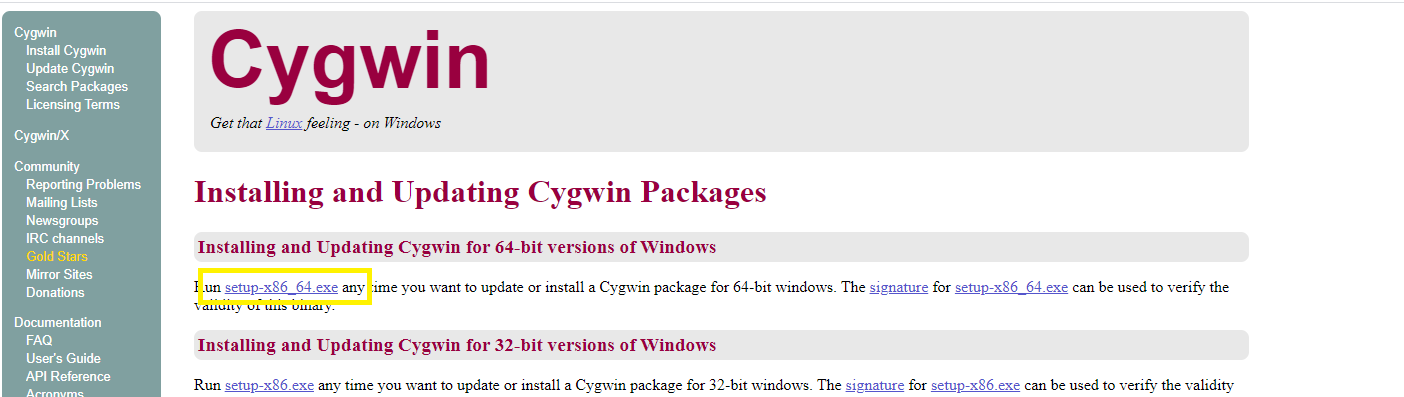
|  |
| --- |
| minikube stop |

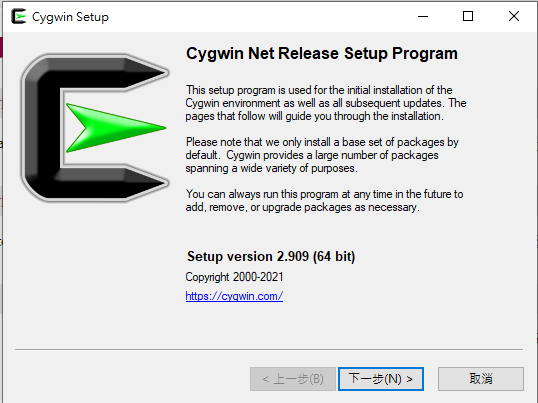
## 安裝Win-bash (非必要)

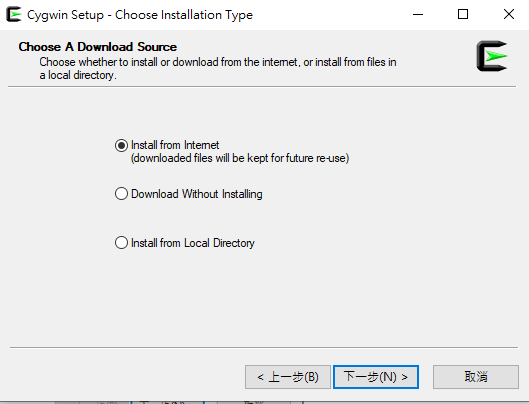
※用途為 Windows 執行linux like sh 工具 (方便 執行sample的sh script)

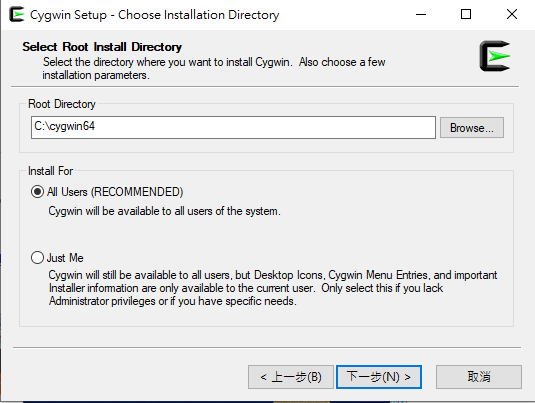
### 方案一 Cygwin64 Terminal

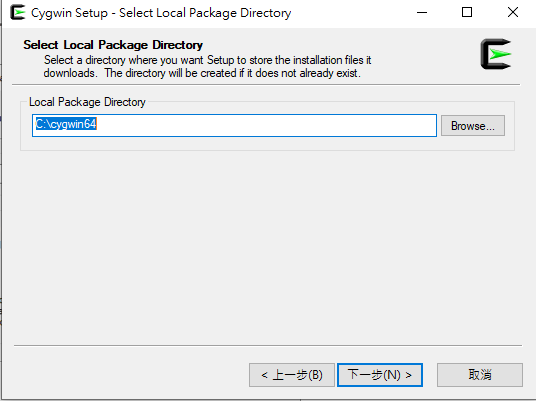
<https://cygwin.com/install.html>

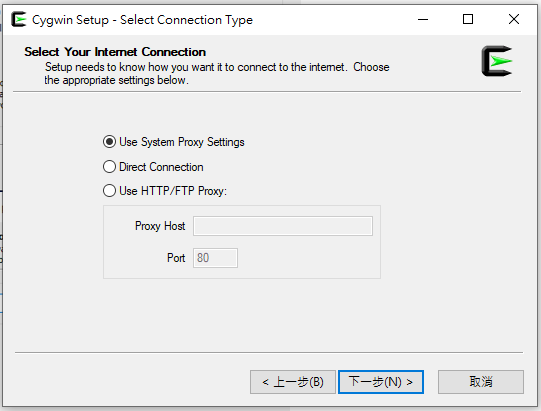


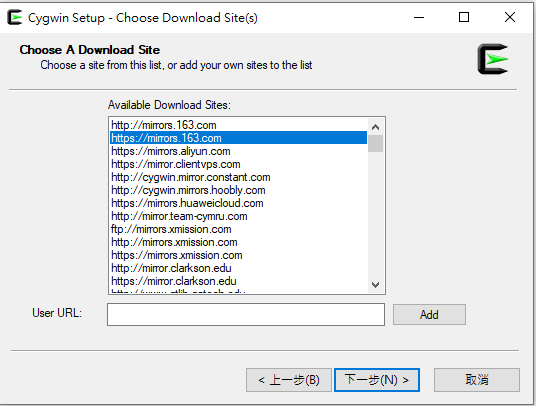


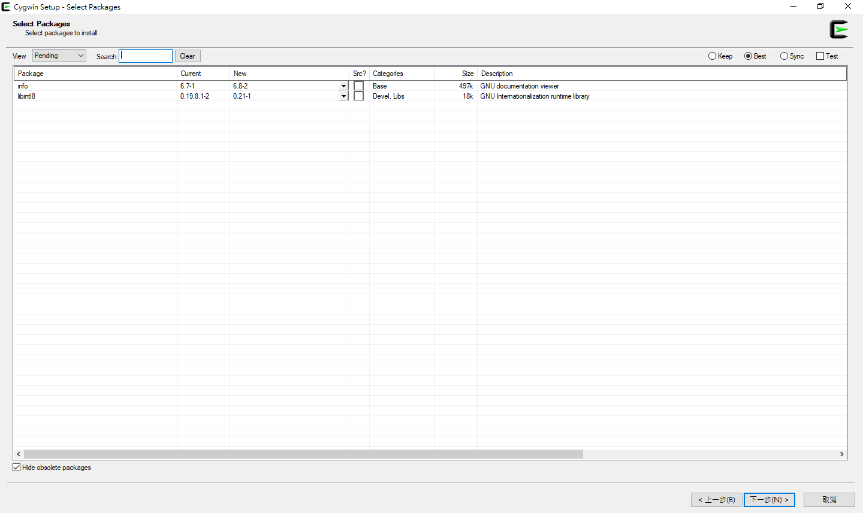


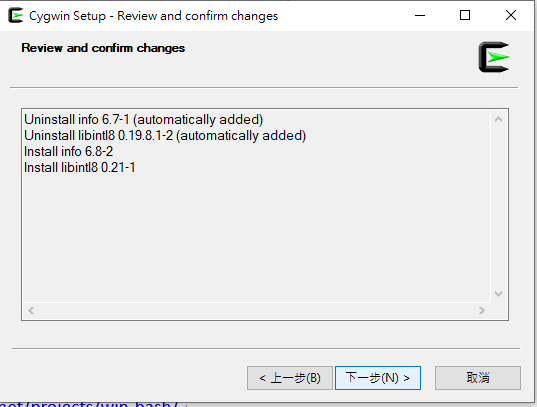


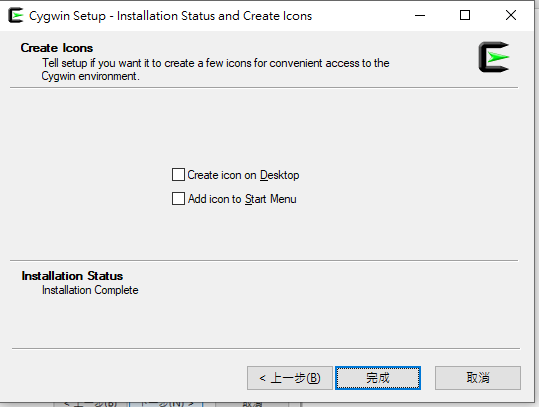






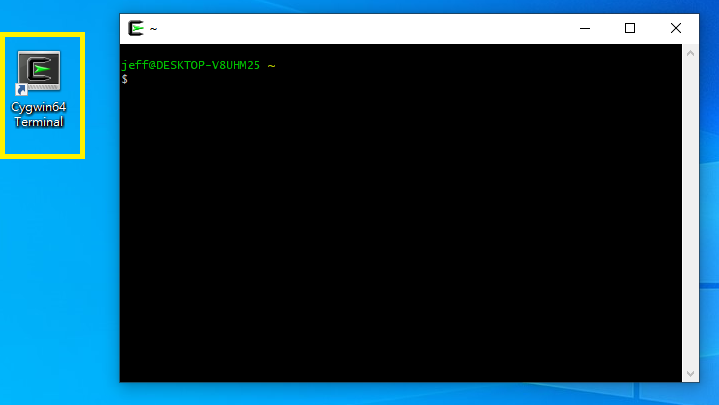






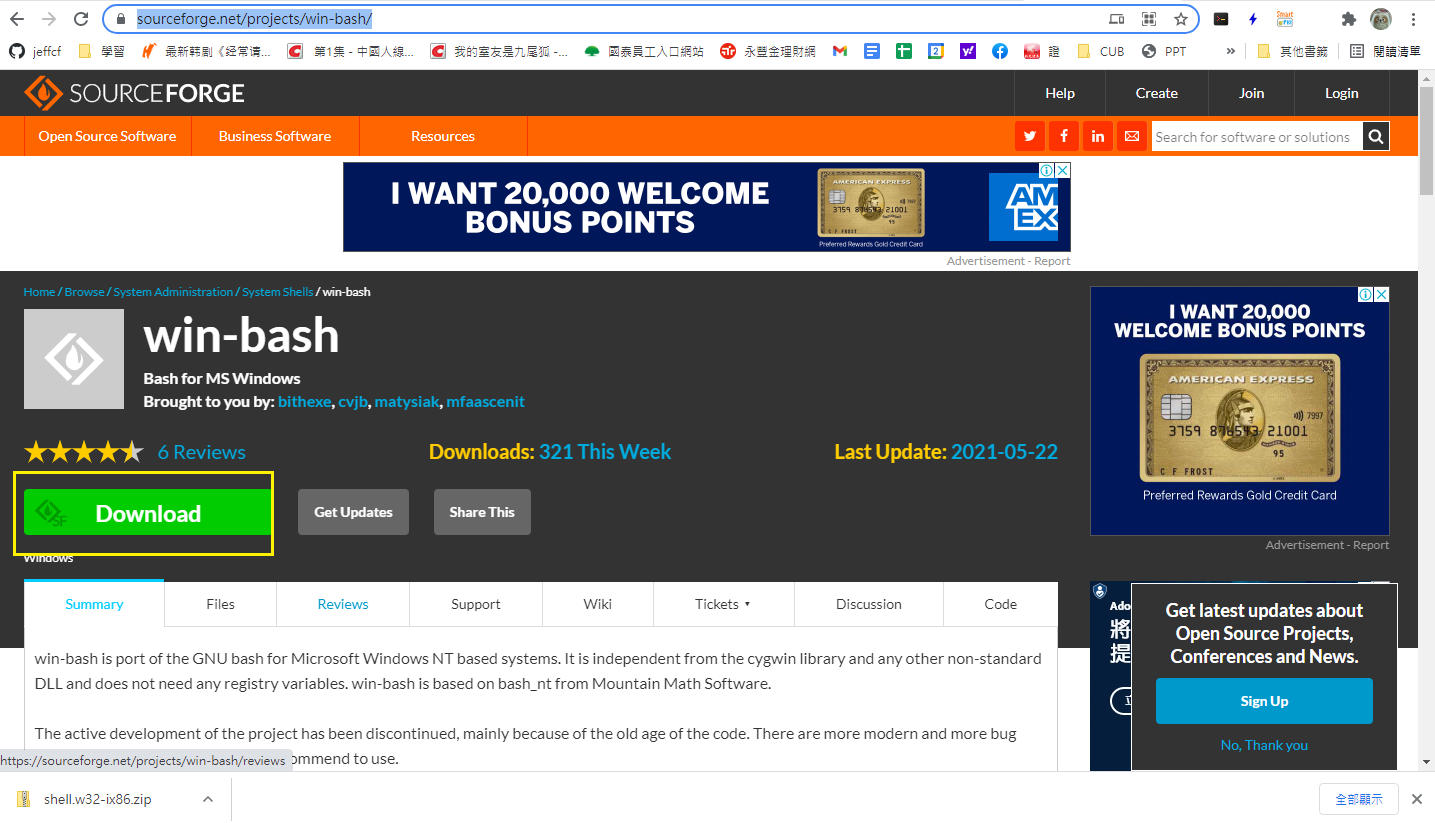
點選啟動捷徑 出現 命令視窗，即可輸入 bash linux 方式之命令

Ex: ls、grep、 | 、、、



### 方案二 win bash

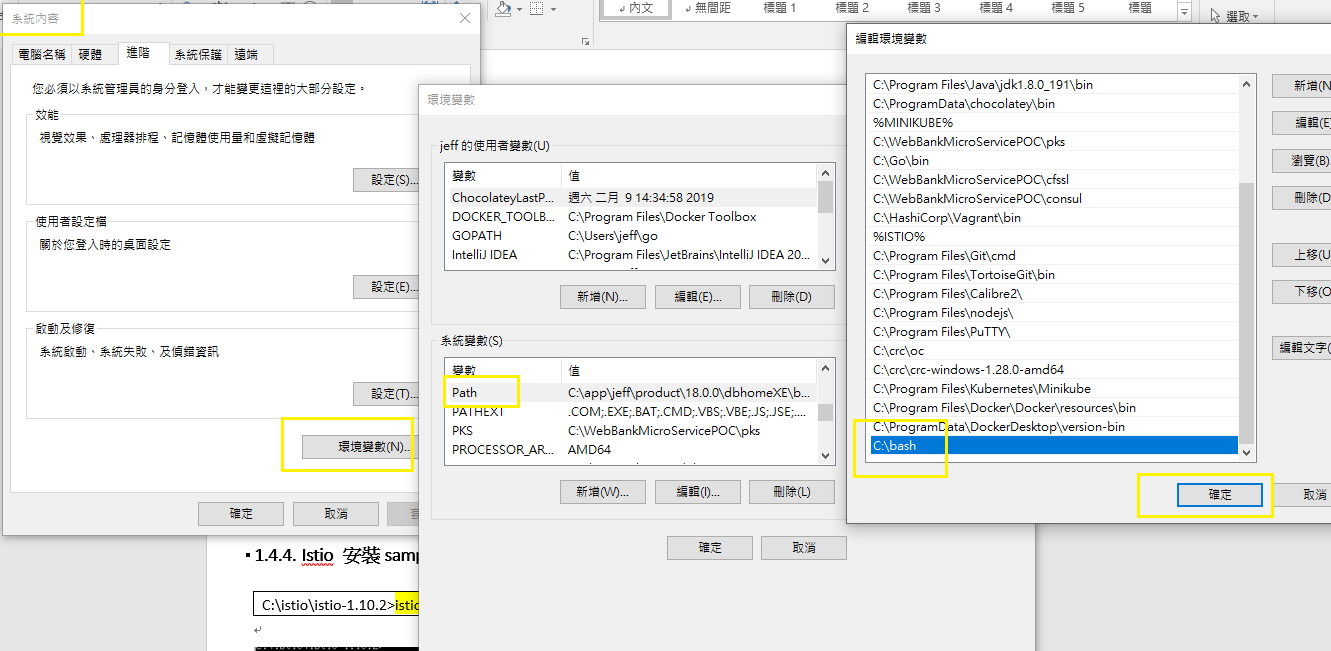
<https://sourceforge.net/projects/win-bash/>

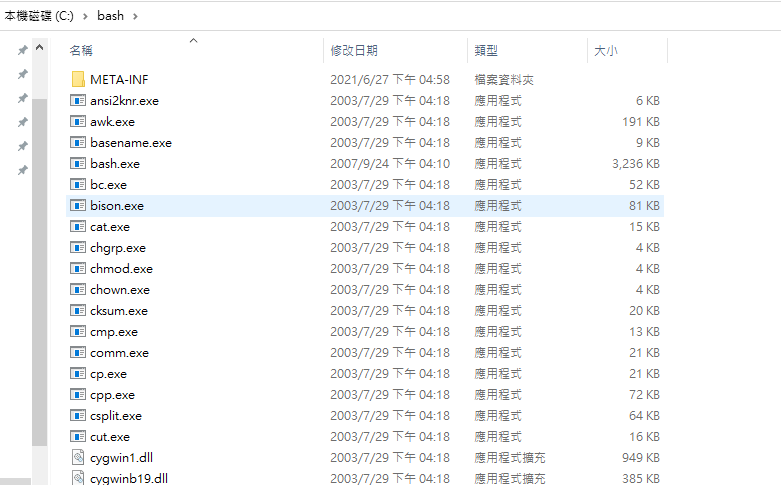


解壓縮到 c:/bash

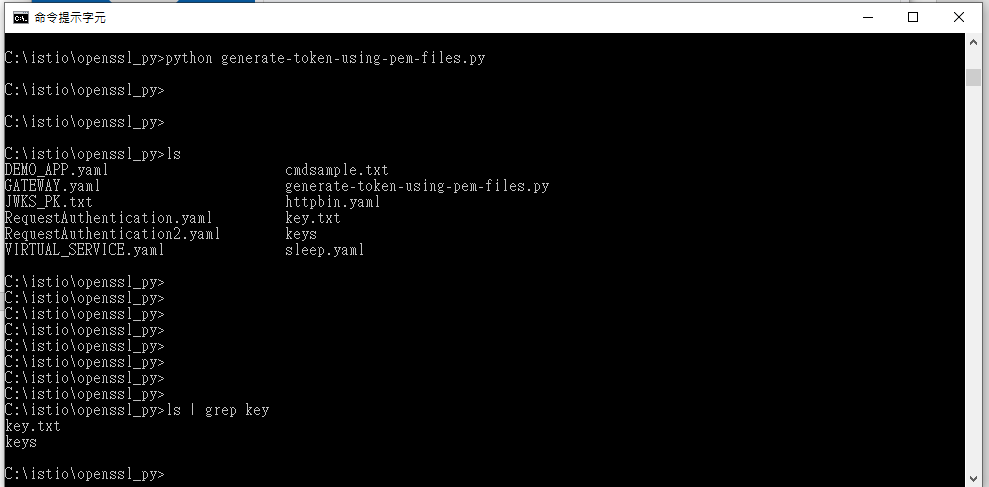


將 c:/bash 新增到環境變數 Path中





※直接用 命令提示字元視窗 即可 使用命令



## MiniKube + Istio 官網Lab

### 參考說明

<https://istio.io/latest/docs/setup/getting-started/>

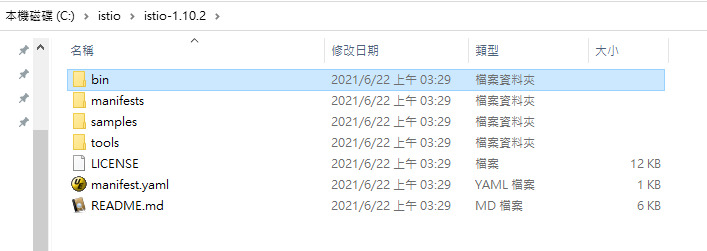
<https://ithelp.ithome.com.tw/articles/10249647>

### Istio相關軟體安裝

※下載網址

<https://github.com/istio/istio/releases/tag/1.10.2>

※解壓縮



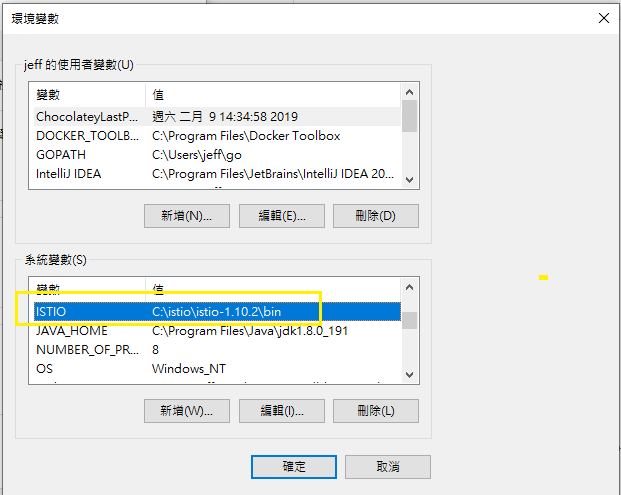
C:\istio\istio-1.10.2\bin 下有 istio 控制命令工具

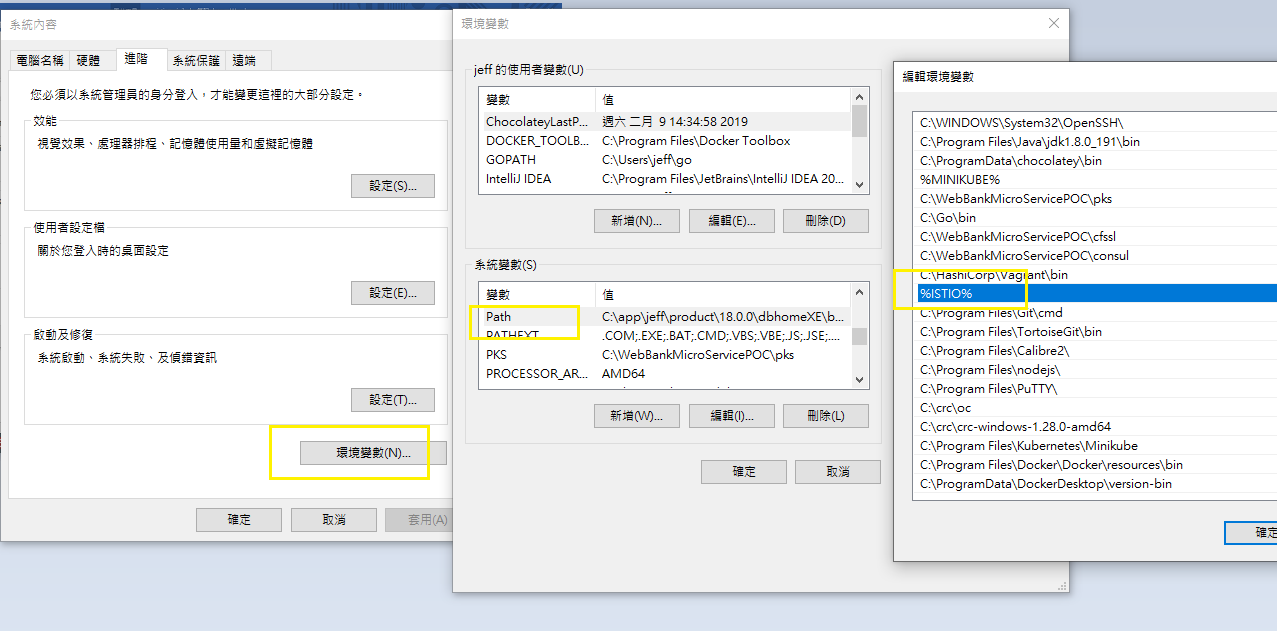


※設定Windows環境變數Path

ISTIO = istioctl.exe下載存放路徑 ex: C:\istio\istio-1.10.2\bin

Path 中，新增 %ISTIO

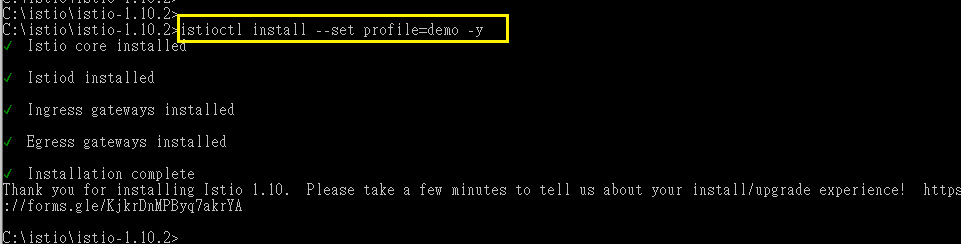




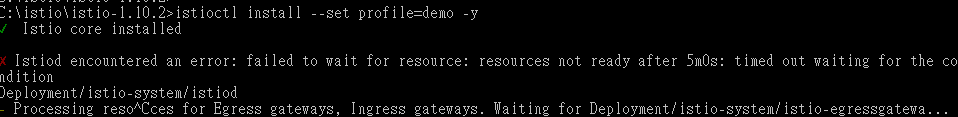
### Istio 安裝sample

**※安裝istio**

|  |
| --- |
| C:\istio\istio-1.10.2> istioctl install --set profile=demo -y |



※後來重作變成有錯誤



※全部清除以後重啟加大資源好像又正常

minikube start --memory 12000 --cpus 3 起來做一次又可以 (?? Mem加大)

**※設定namespace default inject sidecar**

Add a namespace label to instruct Istio to automatically inject Envoy sidecar proxies when you deploy your application later:

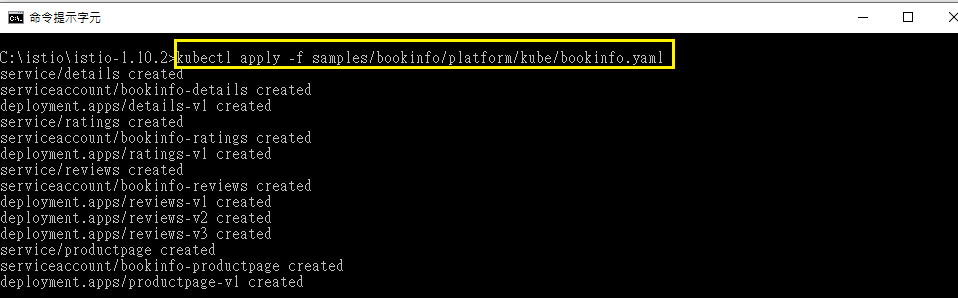
|  |
| --- |
| C:\istio\istio-1.10.2> kubectl label namespace default istio-injection=enabled |

**※apply佈署範例**

|  |
| --- |
| C:\istio\istio-1.10.2>kubectl apply -f samples/bookinfo/platform/kube/bookinfo.yaml |

清除

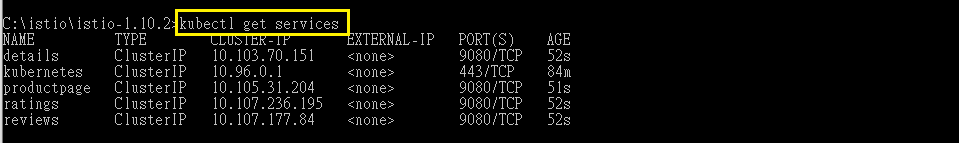
kubectl delete -f samples/bookinfo/platform/kube/bookinfo.yaml



**※查看 服務**

|  |
| --- |
| C:\istio\istio-1.10.2>kubectl get services |

應該會出現下列 服務



**※查看 pods**

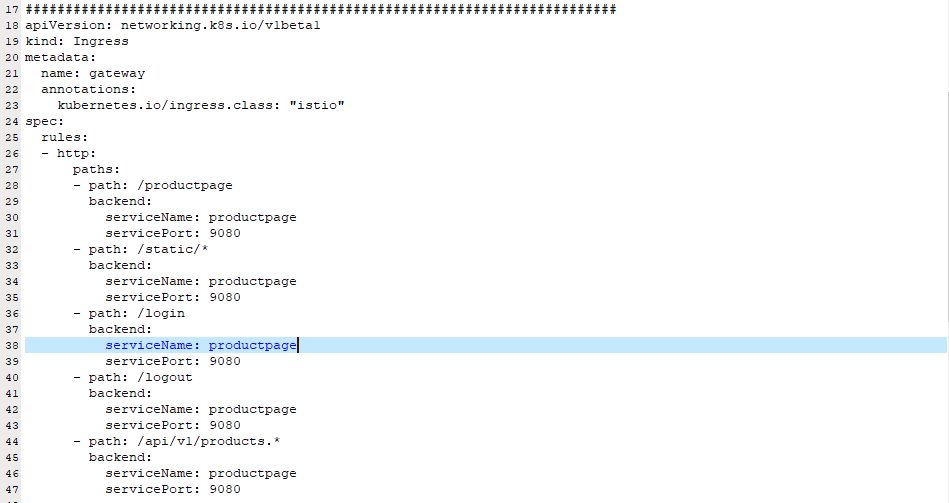
|  |
| --- |
| C:\istio\istio-1.10.2>kubectl get pods |

應該會出現下列 pods



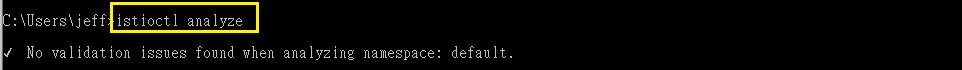
※Associate this application with the Istio gateway 設定gateway進入的mapping

|  |
| --- |
| kubectl apply -f samples/bookinfo/networking/bookinfo-gateway.yaml |



※Associate this application with the Istio gateway:

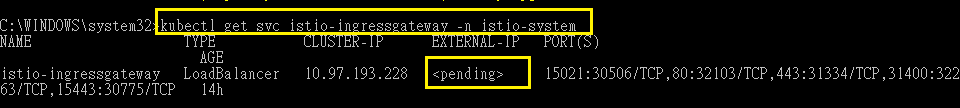
|  |
| --- |
| istioctl analyze |



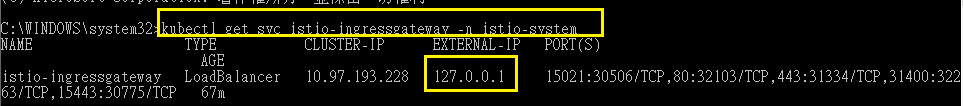
※查看 istio-ingressgateway 服務

|  |
| --- |
| kubectl get svc istio-ingressgateway -n istio-system |

此範例環境沒經過騎他設定時 EXTERNAL-IP是 **<pending>**，外部無法直接存取到內部。



透過 後面minikube tunnel 命令 ，EXTERNAL-IP LB暴露 出127.0.0.1



※讓minikube 提供一個LoadBalance給 Istio (啟動 minikune的 tunnel)

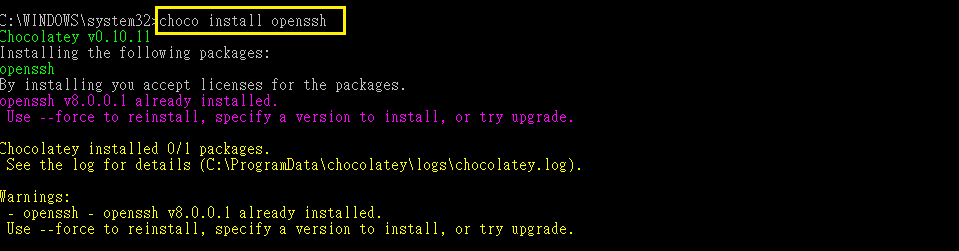
|  |
| --- |
| minikube tunnel  minikube tunnel –cleanup //清除 |

執行後會出現提示OpenSSH clients版本問題提示 (先透過後面安裝，在執行，一樣有提示)



※安裝OpenSSH

<https://minikube.sigs.k8s.io/docs/handbook/accessing/#access-to-ports-1024-on-windows-requires-root-permission>



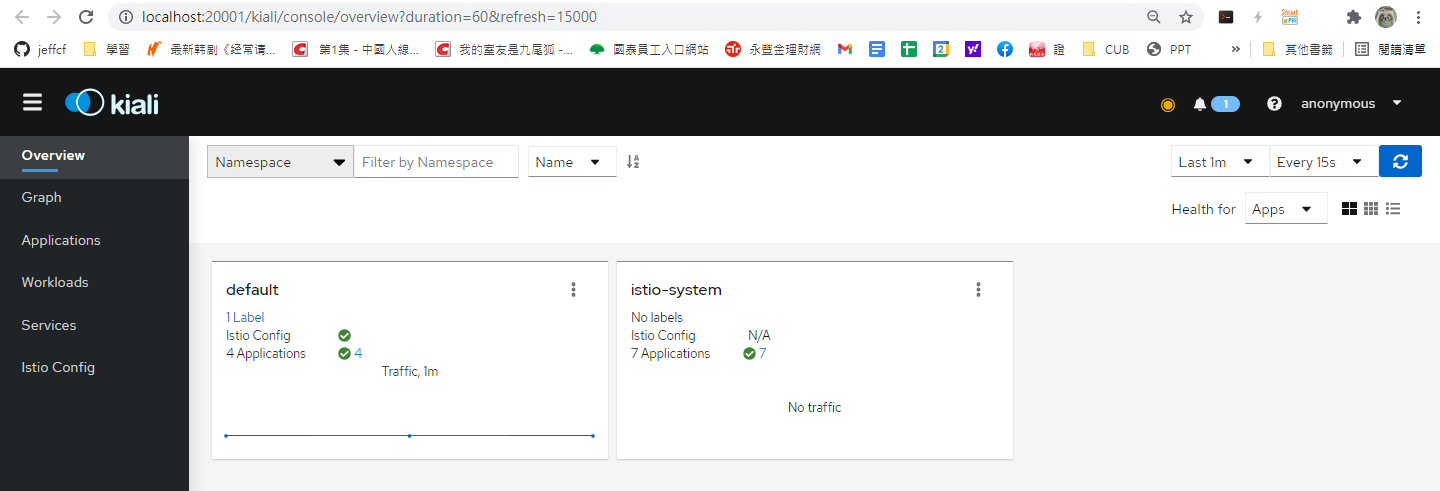
※用 kiali 觀察 istio 服務

|  |
| --- |
| kubectl apply -f samples/addons  kubectl rollout status deployment/kiali -n istio-system |

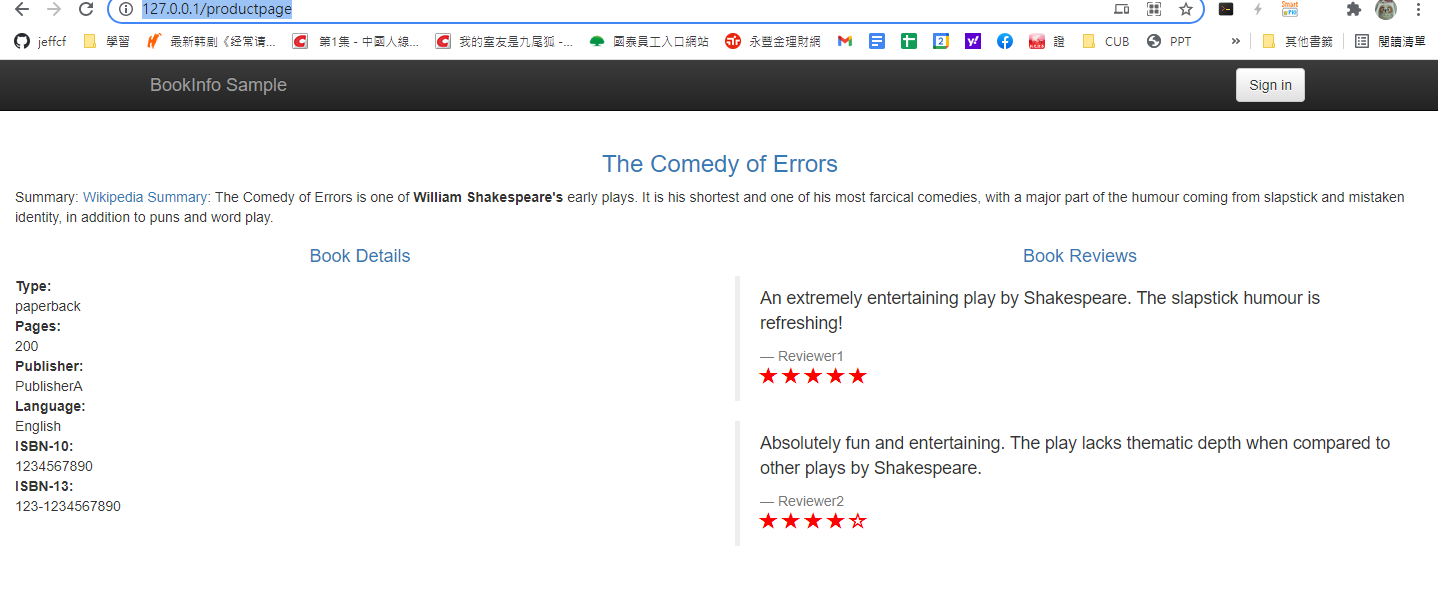
|  |
| --- |
| istioctl dashboard kiali |



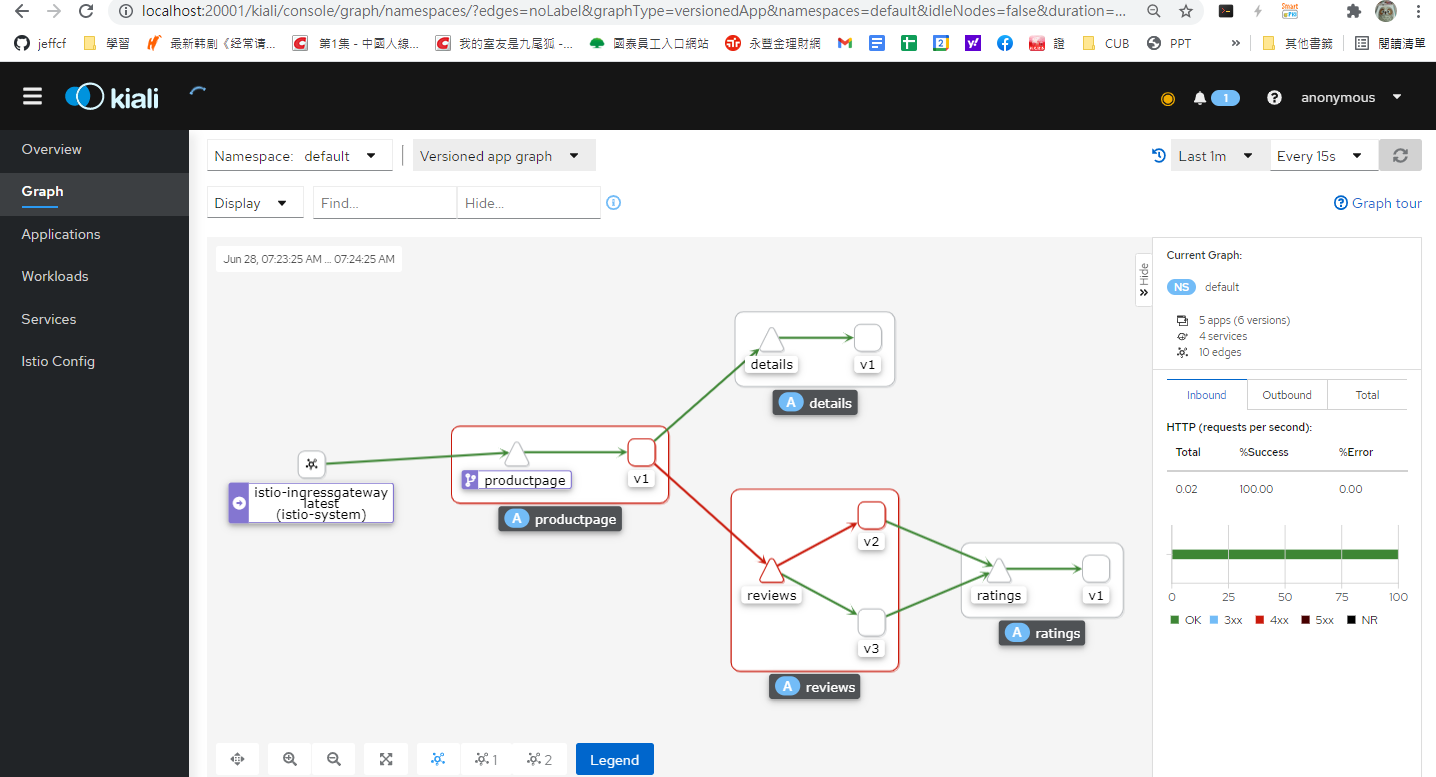
會自動跳出 下面Kiali網頁

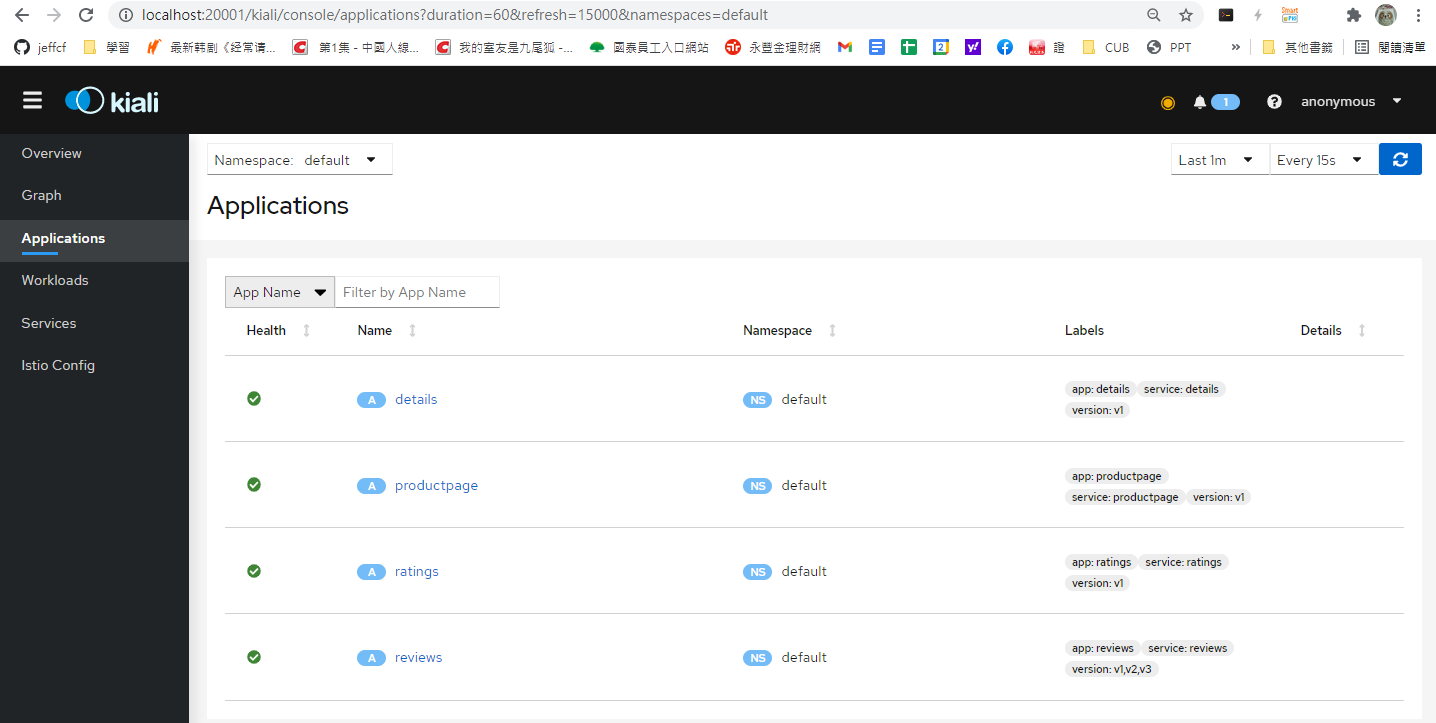


先試打 範例網頁 <http://127.0.0.1/productpage>



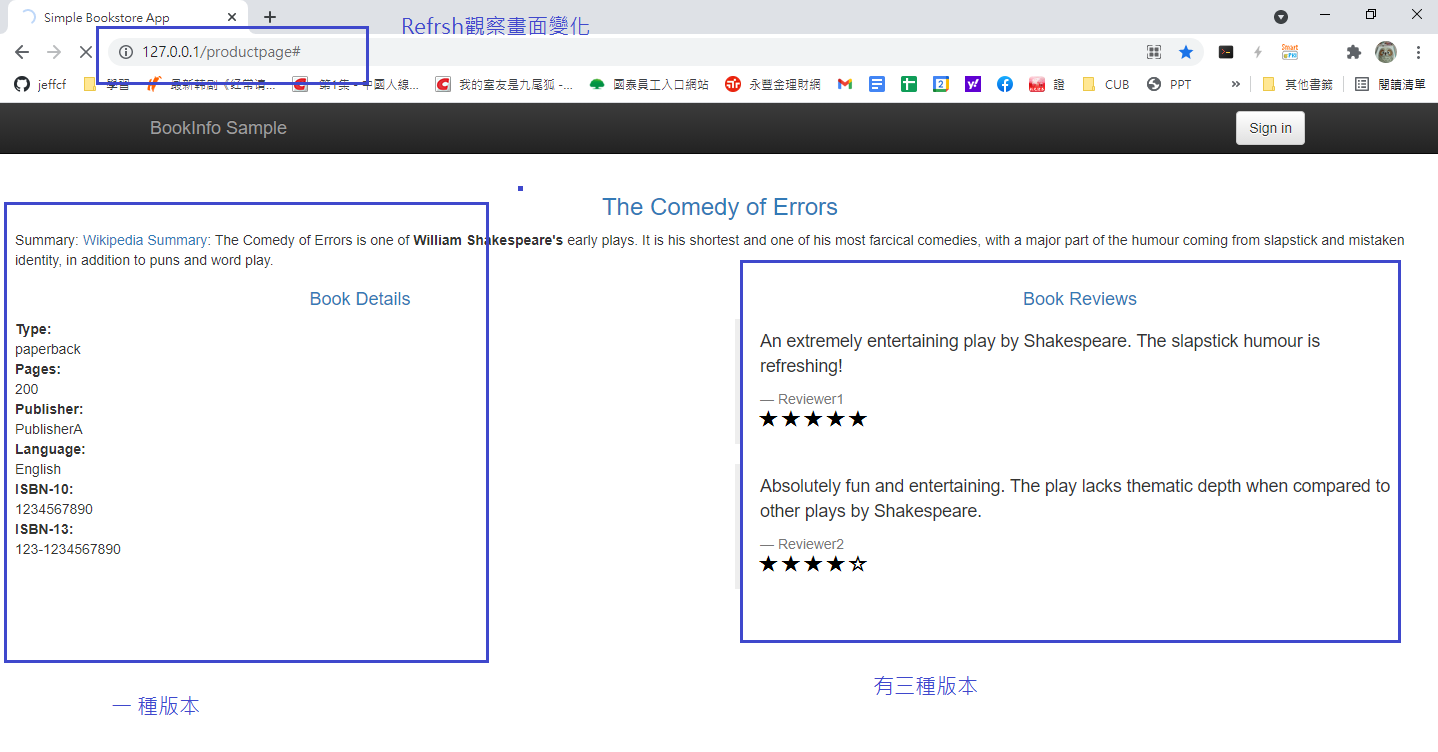
再回去用kiali 就可以看到Graph





### bookinfo範例內容筆記

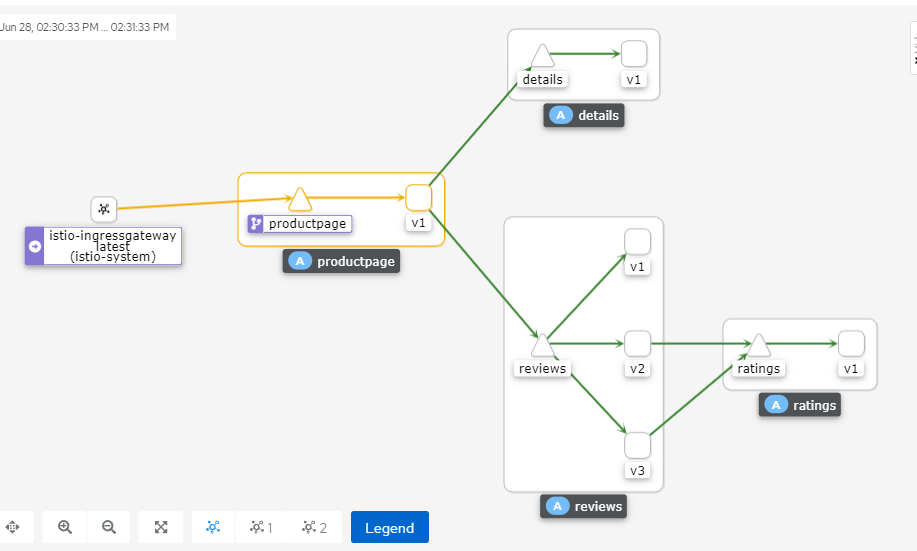
1. 開啟 samples\bookinfo\platform\kube\bookinfo.yaml
2. Refresh網頁觀察畫面變化



比對跟頁面執行的flow

Yaml 中有

五個Deployment 對應到下圖結果



|  |
| --- |
| kubectl get destinationrules  kubectl get virtualservices  kubectl apply -f samples/bookinfo/networking/destination-rule-all.yaml  kubectl delete -f samples/bookinfo/networking/destination-rule-all.yaml |

# Istio

## Istio版本

Isito V1.5 V1.6看起來是目前較多支援的版本 (20210806)

|  |  |  |
| --- | --- | --- |
| GKE | https://cloud.google.com/istio/docs/istio-on-gke/versions | 1.4.x |
| OKD  OCP | Istio v1.5 was tested for a later 4.x version of OpenShift, but was concerned with trying it since OKD has a much older Kubernetes base at v1.11.  <https://access.redhat.com/documentation/zh-cn/openshift_container_platform/4.6/html/service_mesh/deploying-applications-ossm-v1x>  OCP4.6 – Istio 1.6  <https://access.redhat.com/documentation/zh-cn/openshift_container_platform/4.6/html/service_mesh/service-mesh-release-notes>  OKD, Istio v1.5, Maistra = confussion |  |

※參考網址

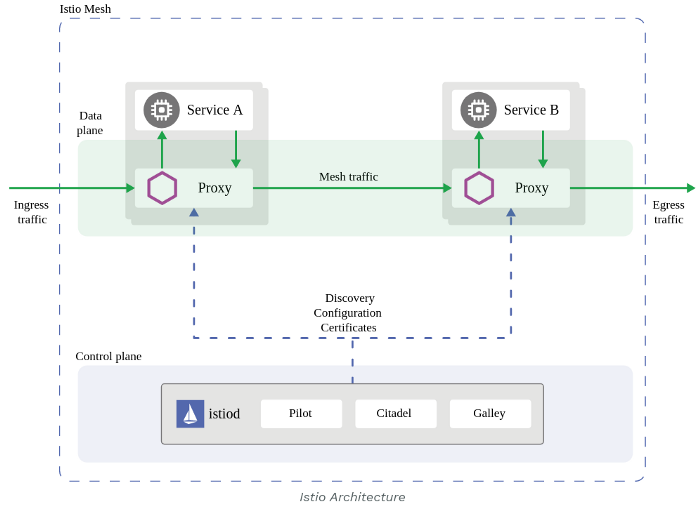
Restricting User Access Via Service Mesh 2.0 and Red Hat Single Sign On

<https://cloud.redhat.com/blog/restricting-user-access-via-service-mesh-2.0-and-red-hat-single-sign-on>

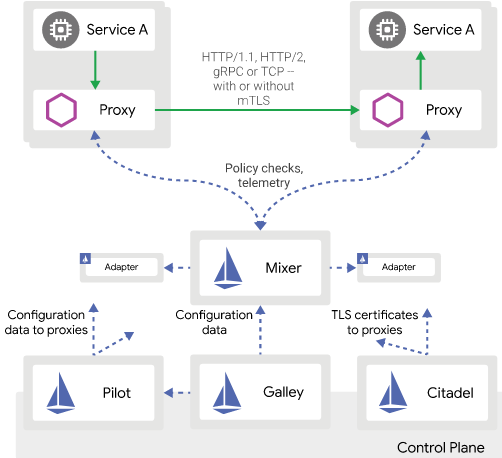
OpenShift Service Mesh Essentials — Part I — The Why and What of it

<https://medium.com/swlh/openshift-service-mesh-essentials-part-i-the-why-and-what-of-it-a3ef09bf8aa8>

※Istio v1.5 之後 (簡化合併元件)



Istio V1.5 之前的架構



## 安裝Istio Configuration Profile

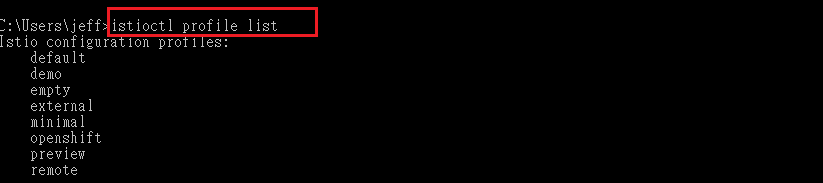
customization of the

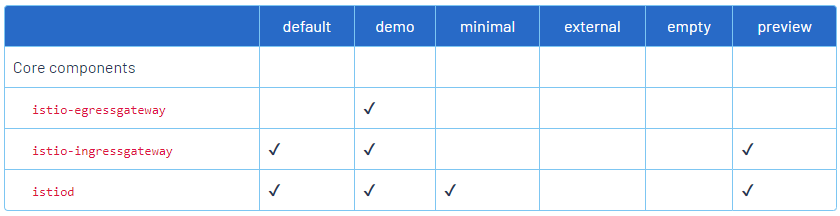
Istio control plane and

the sidecars for the Istio data plane

內建profiles如下表

|  |
| --- |
| istioctl profile list  istioctl profile dump |





## 安裝Istio Gateway

Mesh 的出入口

## 安裝SideCar

<https://istio.io/latest/docs/setup/additional-setup/sidecar-injection/>

自動by namespace

istio-injection=enabled

手動

# Istio JWT實作模擬Lab

參考：

<https://medium.com/intelligentmachines/istio-jwt-step-by-step-guide-for-micro-services-authentication-690b170348fc>

簡述：

步驟：

1. 製作 公鑰、私鑰 private-key.pem、public-key.pem 檔。(使用OpenSSL Win版)
2. 拿私鑰產生JWT (使用範例提供的 python 程式)

Ex:

jwt.generate\_jwt(payload, jwk.JWK.from\_json(private\_key), 'RS256', datetime.timedelta(minutes=500000))

1. 將公鑰 製作 jwks(JSON Web Key Set) 格式
2. 公鑰 解開，驗證是否正確 (公鑰無保存安全問題，各服務可以自行持有做驗證)

於Istio 交由RequestAuthentication 處理，驗證JWT是否正確

## 產生公私Key步驟

### OpenSSL 安裝 (Widnows可用Win64包裝版)

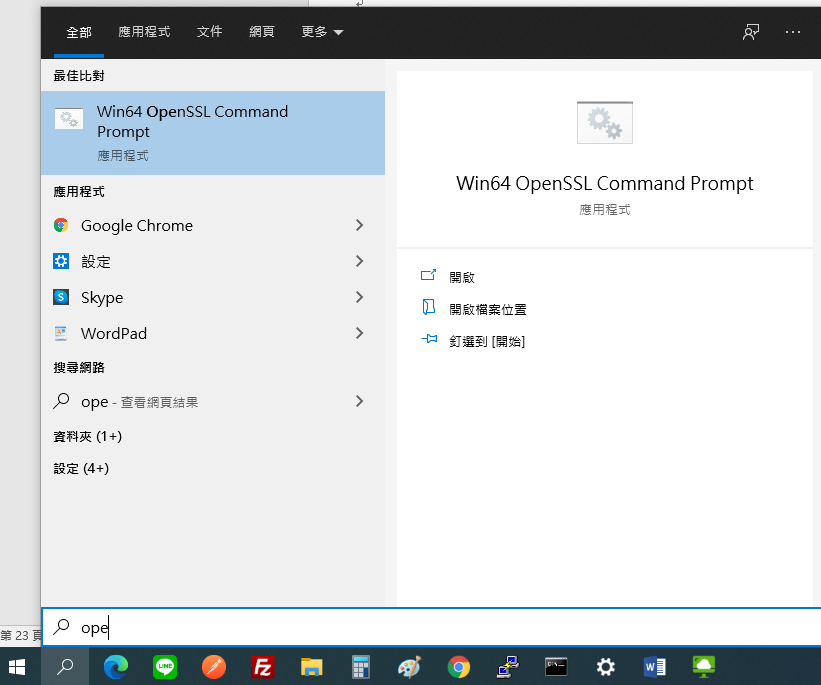
<https://slproweb.com/products/Win32OpenSSL.html>



已預設安裝即可。(最後一頁，不要按donate)

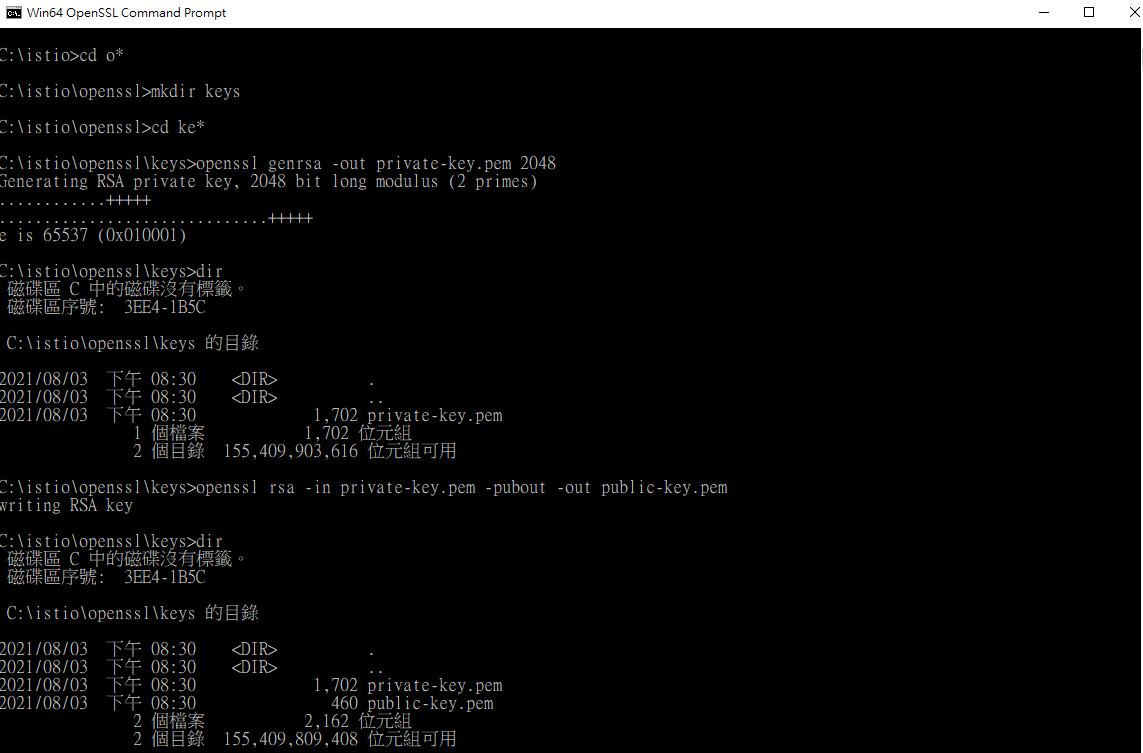
### 產生公私Key

※啟動 Win64 OpenSSL Command 視窗



※輸入下列命令 (自建一個Lab目錄存放)

|  |
| --- |
| mkdir keys  cd keys  *openssl genrsa -out private-key.pem 2048*  *openssl rsa -in private-key.pem -pubout -out public-key.pem* |



### Lab中驗證公私Key的 python 範例

|  |  |
| --- | --- |
| 要先安裝python，  將執行結果 導入 key.txt (內容會顯示相關的key, token   * ) * 安裝Python (下載，已預設下一步完成安裝)   <https://www.python.org/downloads/>   * 安裝Python jwt 套件   <https://pyjwt.readthedocs.io/en/latest/>  ※開啟命令提示列   |  | | --- | | C:\istio\openssl\_py> python generate-token-using-pem-files.py > key.txt |   結果範例如下    Python jwt API Doc  <https://pyjwt.readthedocs.io/en/latest/usage.html> |

### Install Istio

要先安裝Istio 參考，前面章節『1.7.3. Istio 安裝sample』的前兩個步驟

### Install the demo application

|  |
| --- |
| 原始sample  因與 windows cmd 不同 需變換一下  kubectl create ns foo  kubectl apply -f <(istioctl kube-inject -f samples/httpbin/httpbin.yaml) -n foo  kubectl apply -f <(istioctl kube-inject -f samples/sleep/sleep.yaml) -n foo  kubectl create ns bar  kubectl apply -f <(istioctl kube-inject -f samples/httpbin/httpbin.yaml) -n bar  kubectl apply -f <(istioctl kube-inject -f samples/sleep/sleep.yaml) -n bar  kubectl create ns legacy  kubectl apply -f samples/httpbin/httpbin.yaml -n legacy  kubectl apply -f samples/sleep/sleep.yaml -n legacy |

|  |
| --- |
| 變更後 cmd  前面步驟拆分兩步驟，於 C:\istio\istio-1.10.2 執行  istioctl kube-inject -f samples/httpbin/httpbin.yaml 將結果存檔，httpbin.yaml檔  istioctl kube-inject -f samples/sleep /sleep.yaml 將結果存檔，sleep.yaml檔  C:\istio\openssl\_py    kubectl create ns foo  kubectl apply -f httpbin.yaml -n foo  kubectl apply -f sleep.yaml -n foo  kubectl create ns bar  kubectl apply -f httpbin.yaml -n bar  kubectl apply -f sleep.yaml -n bar  kubectl create ns legacy  kubectl apply -f httpbin.yaml -n legacy  kubectl apply -f sleep.yaml -n legacy |

### Install Gateway

|  |
| --- |
| C:\istio\openssl\_py  kubectl apply -f GATEWAY.yaml |

### Install Virtual Service

|  |
| --- |
| C:\istio\openssl\_py  kubectl apply -f VIRTUAL\_SERVICE.yaml |

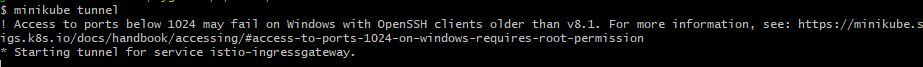
### Test驗證

1. 取得 IP AND export

|  |
| --- |
| Lab原始命令  export INGRESS\_HOST=$(kubectl -n istio-system get service istio-ingressgateway -o jsonpath='{.status.loadBalancer.ingress[0].ip}')  curl $INGRESS\_HOST/headers -s -o /dev/null -w "%{http\_code}\n"  windows沒辦法直接執行  分步拆解 kubectl -n istio-system get service istio-ingressgateway    因為EXTERNAL-IP是空的所以無法往下GO 改用 minikube tunnel |

minikube tunnel

暴露IP 127.0.0.1

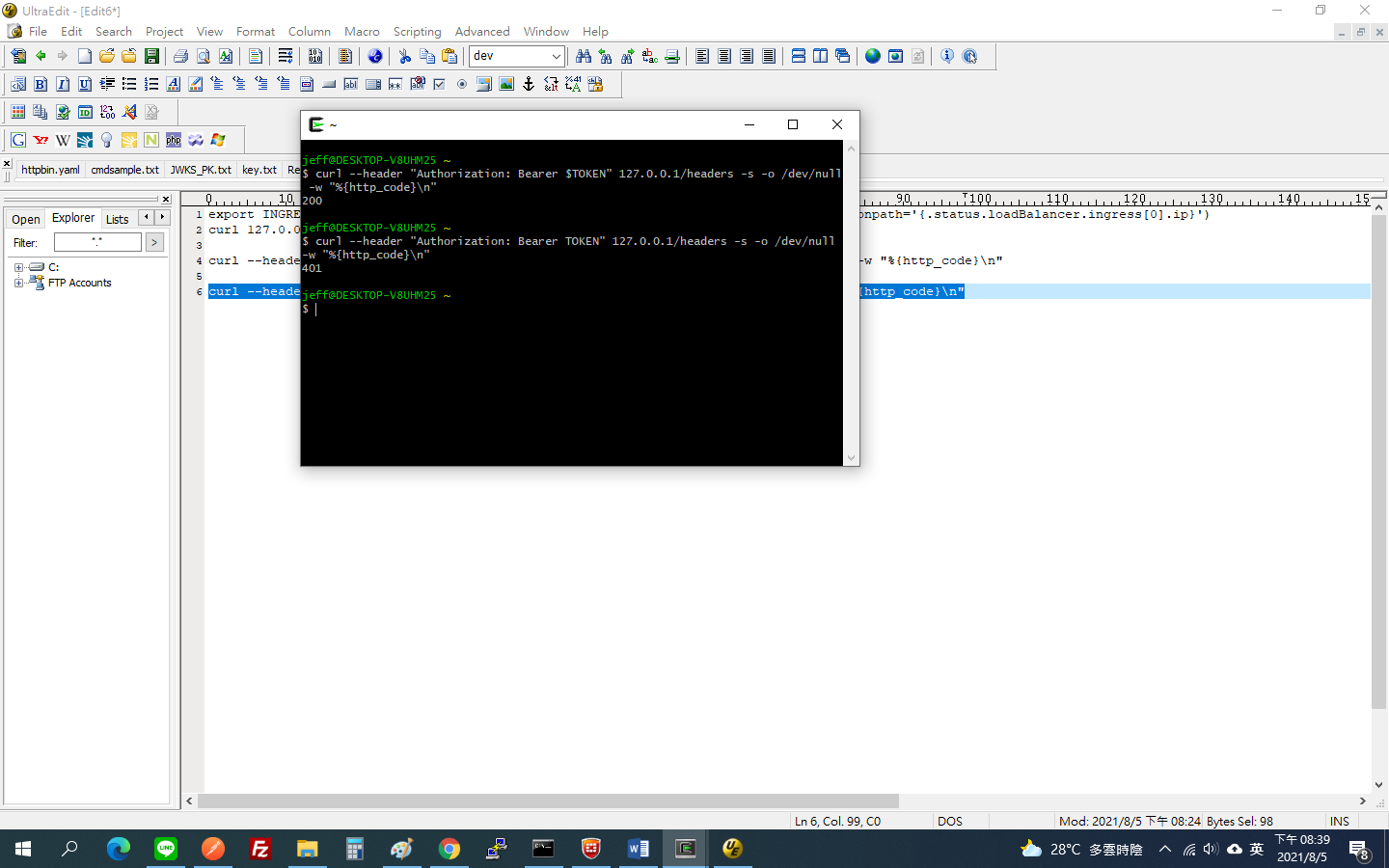


export Token="eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJBVURJRU5DRSIsImV4cCI6MTY1ODE2NDYyMywiaWF0IjoxNjI4MTY0NjIzLCJpc3MiOiJJU1NVRVIiLCJqdGkiOiJ2S0pFUE9zZ1k2MWpjeGNsYy1mZUhRIiwibmJmIjoxNjI4MTY0NjIzLCJwZXJtaXNzaW9uIjoicmVhZCIsInJvbGUiOiJ1c2VyIiwic3ViIjoiU1VCSkVDVCJ9.szcvYWyzm3ChRbSIuGFowJuvhYtBrShv5PFz22ti\_iivA\_6\_A1DwFKHreOX-tQW40p7qltCetTDp4Y8G0zI7008TidzcZHg-iJ9jzVSy4fjAGi19jCKPS4uGZ0\_VZ2DYXDb8se4gO\_y07t5mq8t9FXtsYNRGRY68e0sdnJqsQM3JSuDwPGgN2ZECwnL-XplcPlF6Qmg5\_QFDg6Gd3RumguLWOWgMPMKktDiTipxn2S4G36Ia3asDQPVDcOw-JaxpMpfKnkVKTRCG8jIKSTLCzhFP\_uPWNOHCMAHdamW9Xb8Y04FeW03FZAl14DuzBD-2GUjgbsIX\_wH42r-8k\_IQgQ"

export INGRESS\_HOST="127.0.0.1"

curl --header "Authorization: Bearer bla-bla-bla" 127.0.0.1/headers -s -o /dev/null -w "%{http\_code}\n"

curl --header "Authorization: Bearer $TOKEN" 127.0.0.1/headers -s -o /dev/null -w "%{http\_code}\n"



其他

※**Flagger 漸進式佈署**

https://cloudnative.to/blog/automated-canary-deployments-with-flagger-and-istio/

<https://www.gushiciku.cn/pl/puVo/zh-tw>

<https://blog.51cto.com/zero01/2572820>

※NameSpace 隔離Pod/Service

https://www.sdnlab.com/23944.html

Gateway

https://cloudnative.to/blog/how-to-pick-gateway-for-service-mesh/