從網路基礎到 SRE
整合 DevOps 實踐提高自動化效率

May 30, 2024 at National Yang Ming Chiao Tung University

Tsung-Yi Yu (tsungyi@comp.nus.edu.sg)

whoami

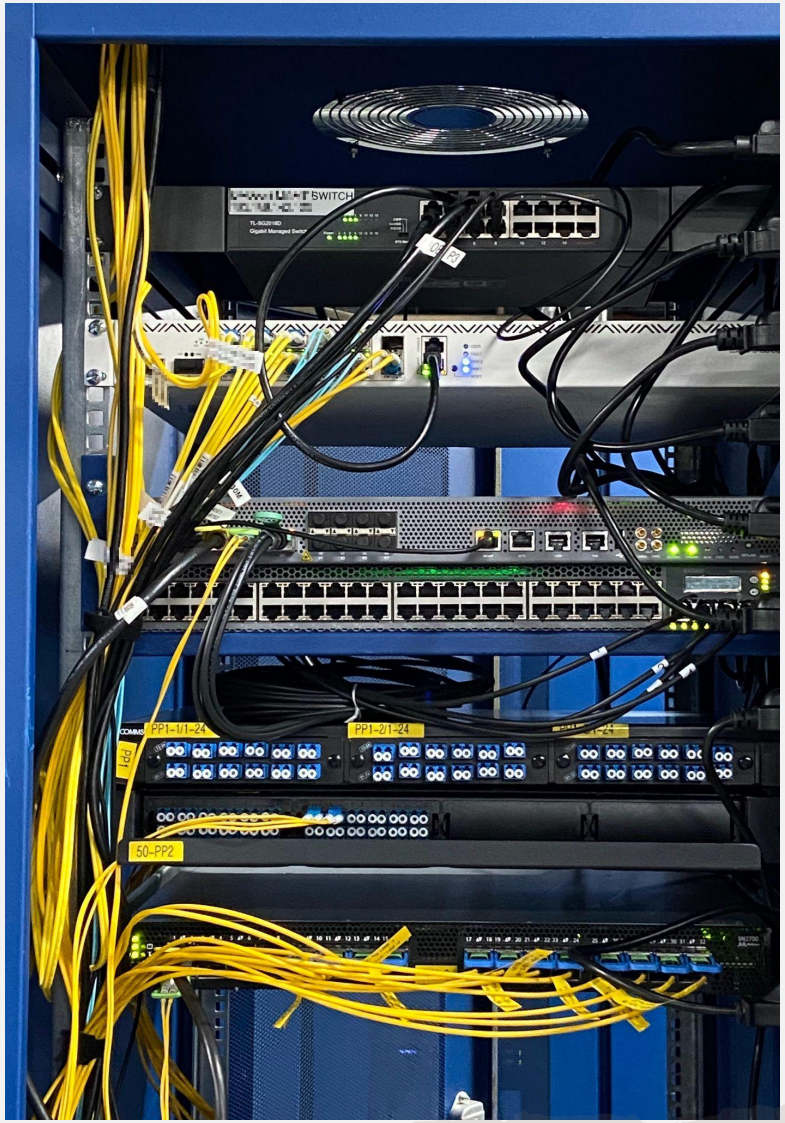
- AS7480 owner & operator
- Internet geek
- Research Assistant at National University of Singapore (NUS)
- Ex SRE Intern at LINE Taiwan
- APNIC/APSIG Fellow Alumni
- Focus on IDC, Border Network or other Layer-3 Protocol

為什麼我想架設自己的 BGP 網路？

- 炫砲
- 讓每台機器的 IP 都能從網際網路連到
- 用自己的 IP 上網
- 選擇不同的網路出口

成果

- 機架式伺服器
- 網管式交換機
- 超多核心和記憶體的伺服器
- 利用開源軟體建置的路由器
- 超過 100G 的對外頻寬



```
steveyiyo@steveyiyo-testvm:~$ curl ipinfo.io
{
  "ip": "44.31.73.1",
  "city": "Taipei",
  "region": "Taiwan",
  "country": "TW",
  "loc": "25.0478,121.5319",
  "org": "AS7480 STEVEYI NETWORK",
  "timezone": "Asia/Taipei",
  "readme": "https://ipinfo.io/missingauth"
```

My traceroute [v0.94]

cr1-tpe.steveyi.net (103.69.92.110) -> nasa.cs.nycu.edu.tw 2024-05-24T19:11:04+0000

Keys: Help Display mode Restart statistics Order of fields quit

		Packets		Pings				
Host		Loss%	Snt	Last	Avg	Best	Wrst	StDev
1. AS945	agg.tp.ip.wakuwaku.co (74.122.39.0)	0.0%	5	0.1	0.1	0.1	0.2	0.0
2. AS???	11-222-163-203-static.tpix.net.tw (203.163.222.11)	0.0%	5	0.4	0.5	0.3	0.6	0.1
3. AS9505	175-41-58-145.twgate-ip.twgate.net (175.41.58.145)	0.0%	5	0.8	1.0	0.8	1.9	0.5
4. AS9505	203-78-181-246.twgate-ip.twgate.net (203.78.181.246)	0.0%	5	0.7	0.7	0.6	0.7	0.0
5. AS9505	203-160-226-14.twgate-ip.twgate.net (203.160.226.14)	0.0%	4	2.3	2.5	2.3	3.1	0.4
6. AS18185	211.76.255.193 (211.76.255.193)	0.0%	4	2.3	2.4	2.3	2.8	0.3
7. AS9916	140.113.0.77 (140.113.0.77)	0.0%	4	2.3	2.3	2.3	2.4	0.0
8. AS9916	140.113.3.177 (140.113.3.177)	0.0%	4	3.0	2.9	2.8	3.0	0.1
9. AS9916	140.113.3.241 (140.113.3.241)	0.0%	4	3.2	3.3	3.2	3.4	0.1
10. AS9916	nasa.cs.nctu.edu.tw (140.113.17.32)	0.0%	4	2.8	2.9	2.8	2.9	0.0

靜態路由

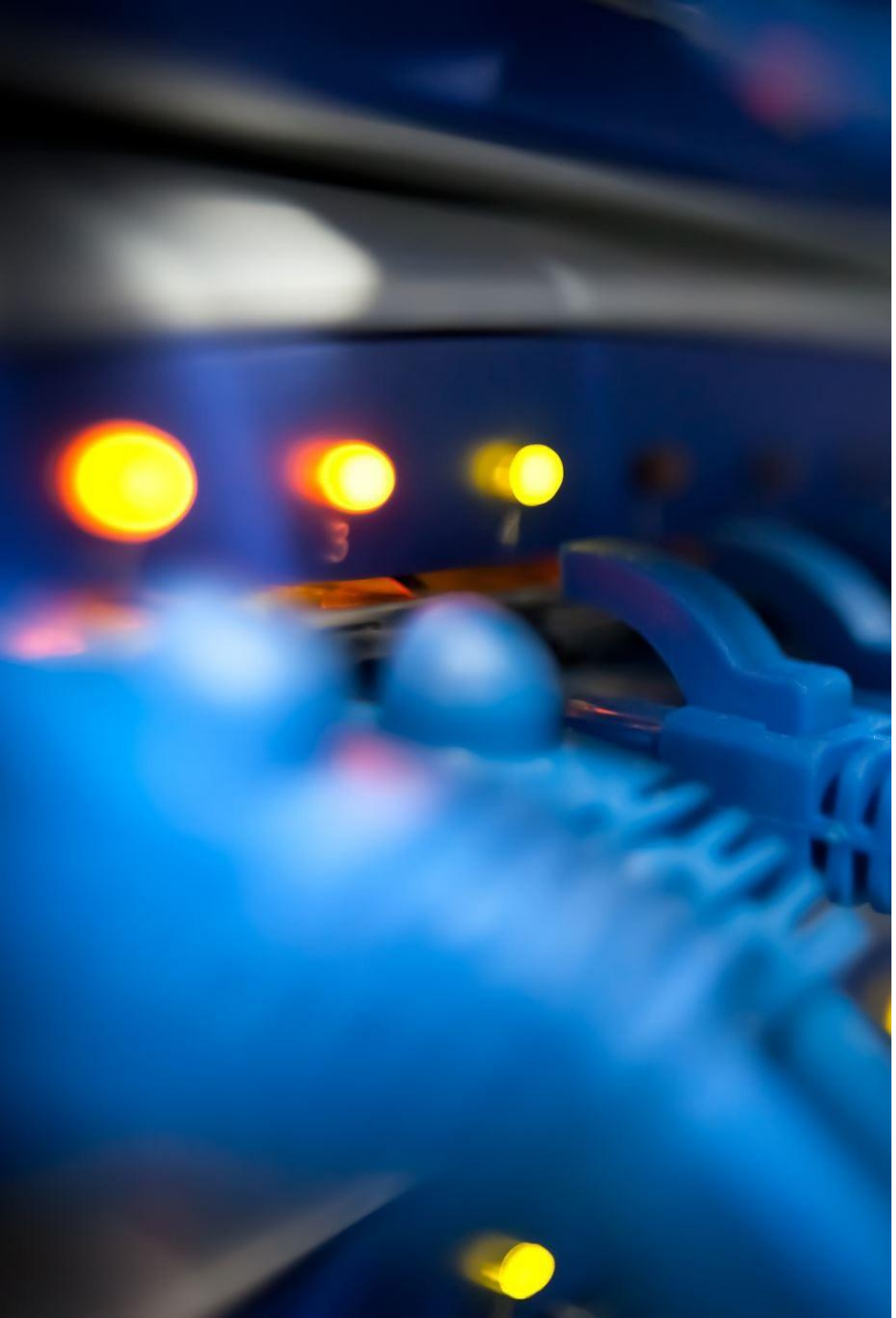
優點

- 預設路由
- 適合小型網路
- 完全控制路由走向

缺點

- 不適合大型網路
- 無高可用性

在 Linux 下使用 iproute2 新增靜態路由
sudo ip route add 8.8.8.0/24 via 10.1.1.1 dev eth0



路由器種類

- Hardware Router
  - 中華電信的小烏龜
  - 市面上的硬體路由器
- Software Router (x86 Router)
  - OpenWrt
  - Router OS
  - VyOS
  - pfSense
  - Customized…?

路由器的運作原理

路由選擇

- 路由表
- 最佳路由

封包轉發

- TCP
- UDP
- ICMP
- Protocols

IPv4 不夠用怎麼辦

- RFC 1918 (Private Network IP Address)
- SNAT (or NAT Forwarding)
- DNAT (or Port Forwarding)

如果我是網路提供商，
有一條鏈路斷掉了怎麼辦？

動態路由

優點

- 適用於大型網路（校園網路、網路提供商等）
- 路由協議
- 最佳化路由

缺點

- 入門門檻高
- 一般路由器不支援

路由協議

- IGP (Interior Gateway Protocols)
  - OSPF
  - RIP
  - IS-IS
  - EIGRP
- EGP (Exterior Gateway Protocols)
  - BGP

IGP carries infrastructure links and loopbacks, while BGP is used for carrying Internet and customer prefixes.

IGP: OSPF (Open Shortest Path First)

- 為什麼選擇 OSPF？
- 基於 LSA
- 使用 Dijkstra 算法計算最短路徑
- 區域劃分
- 適用於大型網絡

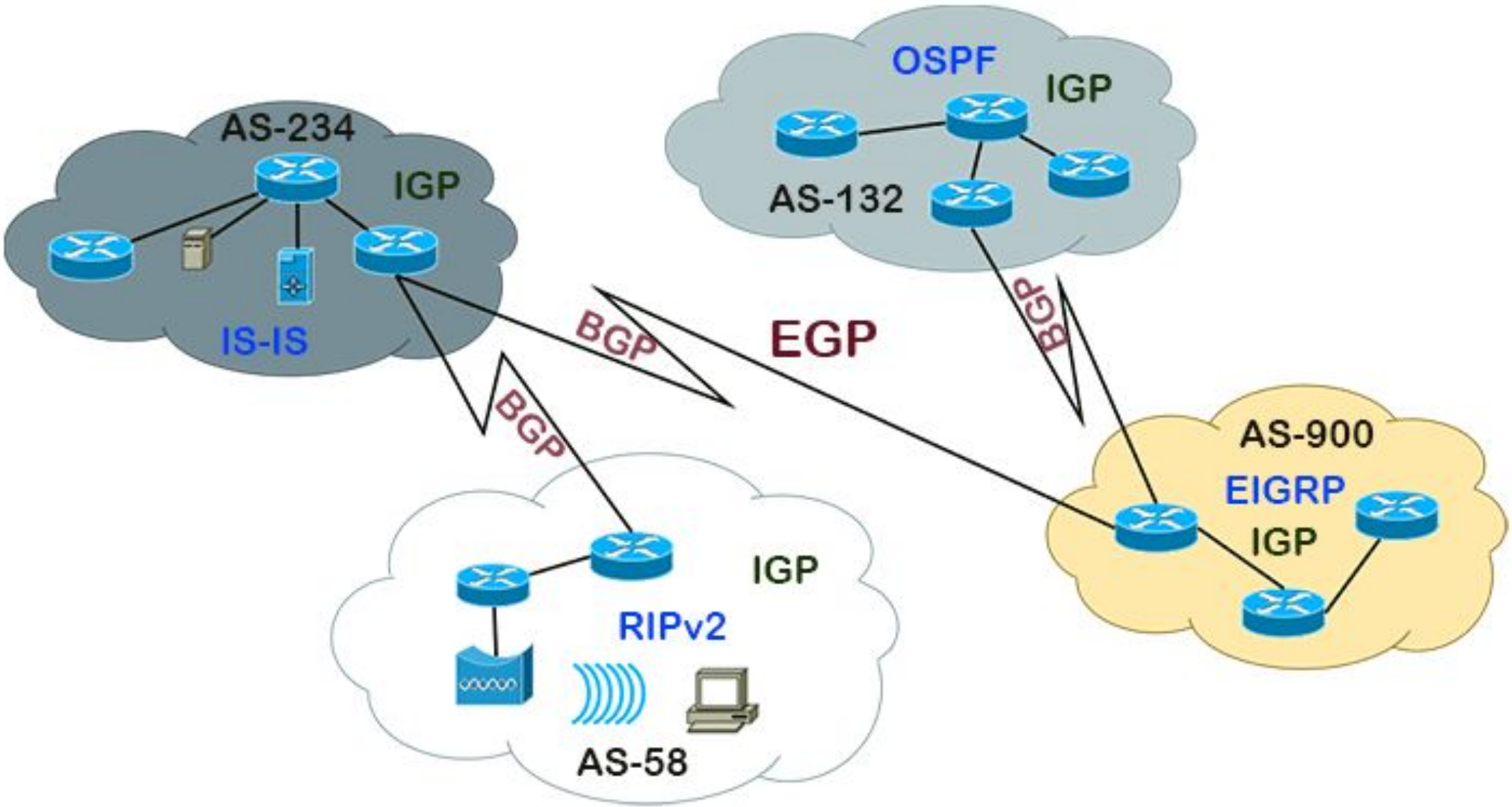
```
Hello, this is FRRouting (version 9.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

cr1-tpe.steveyi.net# show ip ospf vrf vrf_INTERNAL database
VRF Name: vrf_INTERNAL

       OSPF Router with ID (10.37.2.254)

                Router Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
10.37.1.254    10.37.1.254      594 0x800001b3 0x9d3b 1
10.37.2.254    10.37.2.254      473 0x800001b4 0x02cd 1

                Net Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum
10.37.2.2      10.37.1.254     1664 0x800001af 0xd810

                AS External Link States

Link ID         ADV Router      Age  Seq#       CkSum  Route
10.37.0.0      10.37.2.254      593 0x800001b0 0x7d31 E2 10.37.0.0/24 [0x0]
10.37.1.0      10.37.1.254      674 0x800001af 0x7b34 E2 10.37.1.0/24 [0x0]
10.37.1.254    10.37.1.254     1124 0x800001af 0x852b E2 10.37.1.254/32 [0x0]
10.37.4.0      10.37.2.254      433 0x800001b0 0x5159 E2 10.37.4.0/24 [0x0]
103.69.92.119  10.37.2.254      213 0x800001b0 0x9bc1 E2 103.69.92.119/32 [0x0]


cr1-tpe.steveyi.net#
```

OSPF 的基本運作原理

- Hello packets
  - 路由器之間互相偵測。
- LS-Update
  - 交換路由器之間的鏈路狀態。

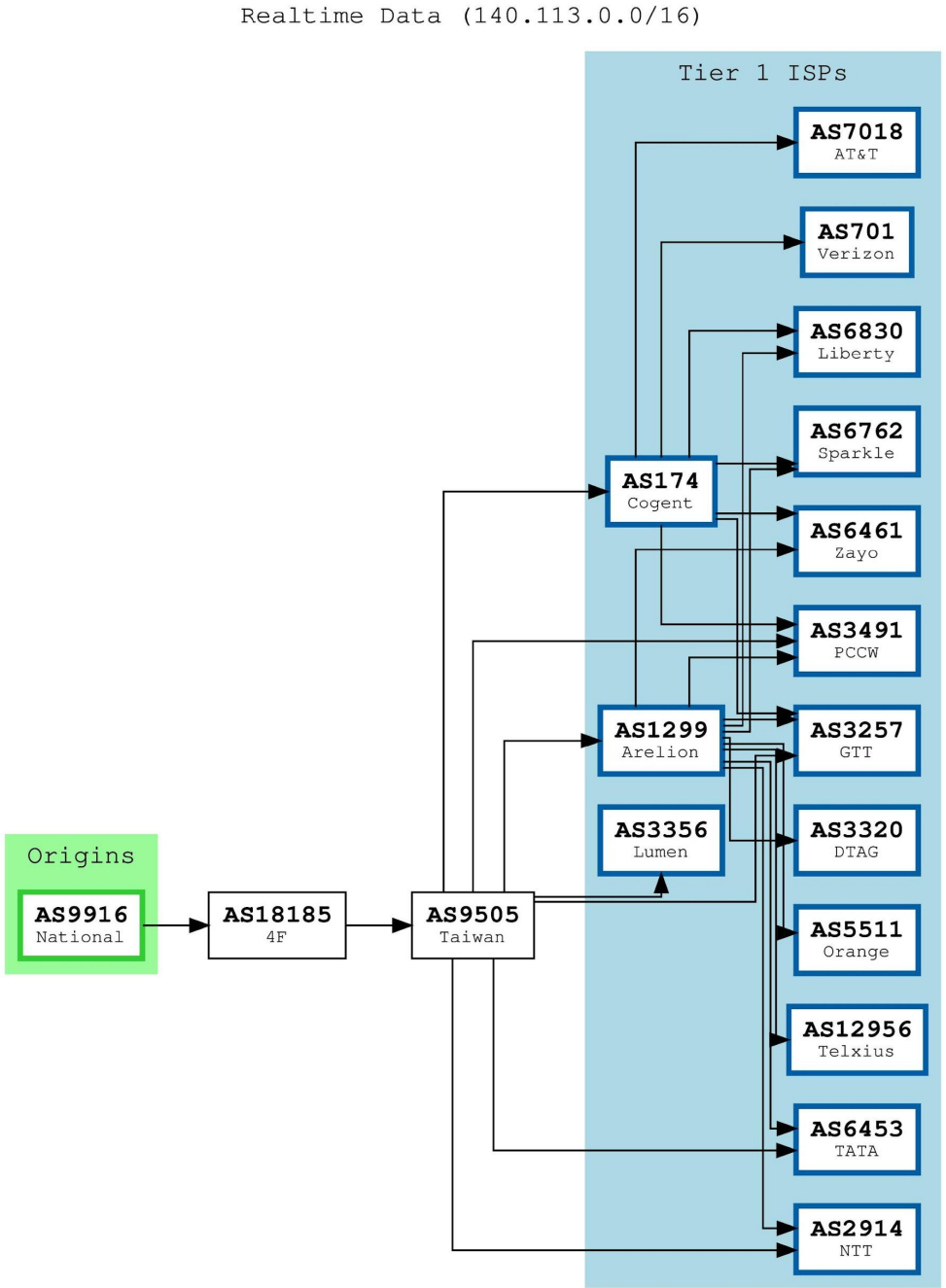
```
steveyiyo@cr1-tpe:~$ sudo tcpdump -i vlan1383 ip proto ospf -v -n
tcpdump: listening on vlan1383, link-type EN10MB (Ethernet), snapshot length 262144 bytes
21:33:22.699532 IP (tos 0xc0, ttl 1, id 41036, offset 0, flags [none], proto OSPF (89), length 72)
    10.37.2.2 > 224.0.0.5: OSPFv2, Hello, length 52
        Router-ID 10.37.1.254, Area 0.0.0.1, Authentication Type: none (0)
        Options [External]
          Hello Timer 10s, Dead Timer 40s, Mask 255.255.255.0, Priority 1
          Designated Router 10.37.2.2, Backup Designated Router 10.37.2.254
          Neighbor List:
            10.37.3.254
            10.37.2.254
21:33:22.699553 IP (tos 0xc0, ttl 1, id 41037, offset 0, flags [none], proto OSPF (89), length 84)
    10.37.2.2 > 224.0.0.5: OSPFv2, LS-Update, length 64
        Router-ID 10.37.1.254, Area 0.0.0.1, Authentication Type: none (0), 1 LSA
          LSA #1
          Advertising Router 10.37.1.254, seq 0x800001b3, age 1s, length 16
            Router LSA (1), LSA-ID: 10.37.1.254
            Options: [External]
            Router LSA Options: [ASBR]
              Neighbor Network-ID: 10.37.2.2, Interface Address: 10.37.2.2
                topology default (0), metric 1
21:33:23.270099 IP (tos 0xc0, ttl 1, id 14621, offset 0, flags [none], proto OSPF (89), length 72)
    10.37.2.254 > 224.0.0.5: OSPFv2, Hello, length 52
        Router-ID 10.37.2.254, Area 0.0.0.1, Authentication Type: none (0)
        Options [External]
          Hello Timer 10s, Dead Timer 40s, Mask 255.255.255.0, Priority 1
          Designated Router 10.37.2.2, Backup Designated Router 10.37.2.254
          Neighbor List:
            10.37.1.254
            10.37.3.254
```

sudo tcpdump -i vlan1383 ip proto ospf



Source: https://www.computernetworkingnotes.com/ccna-study-guide/igp-egp-and-autonomous-system-explained.html

BGP

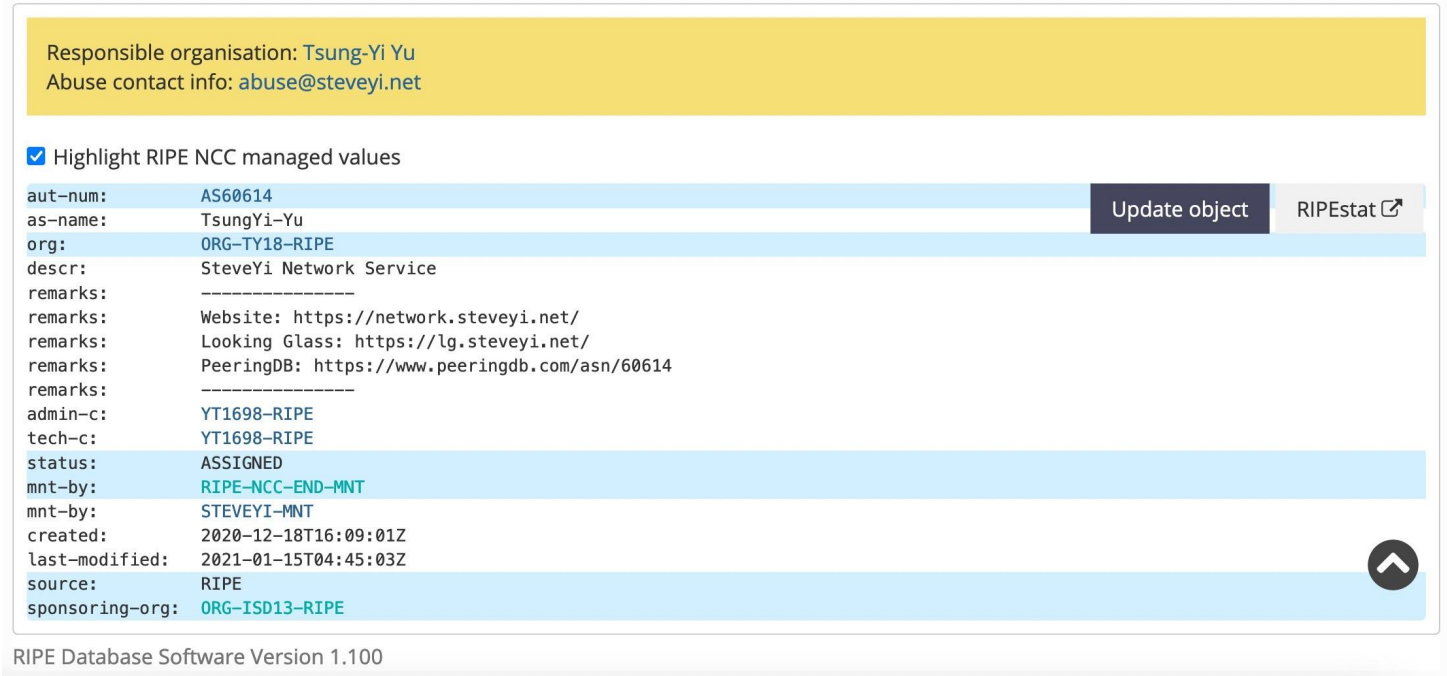


- Border Gateway Protocol
- RFC 4271
- 與其他 ASN 交換路由
- iBGP & eBGP
- 多種路由路徑可選

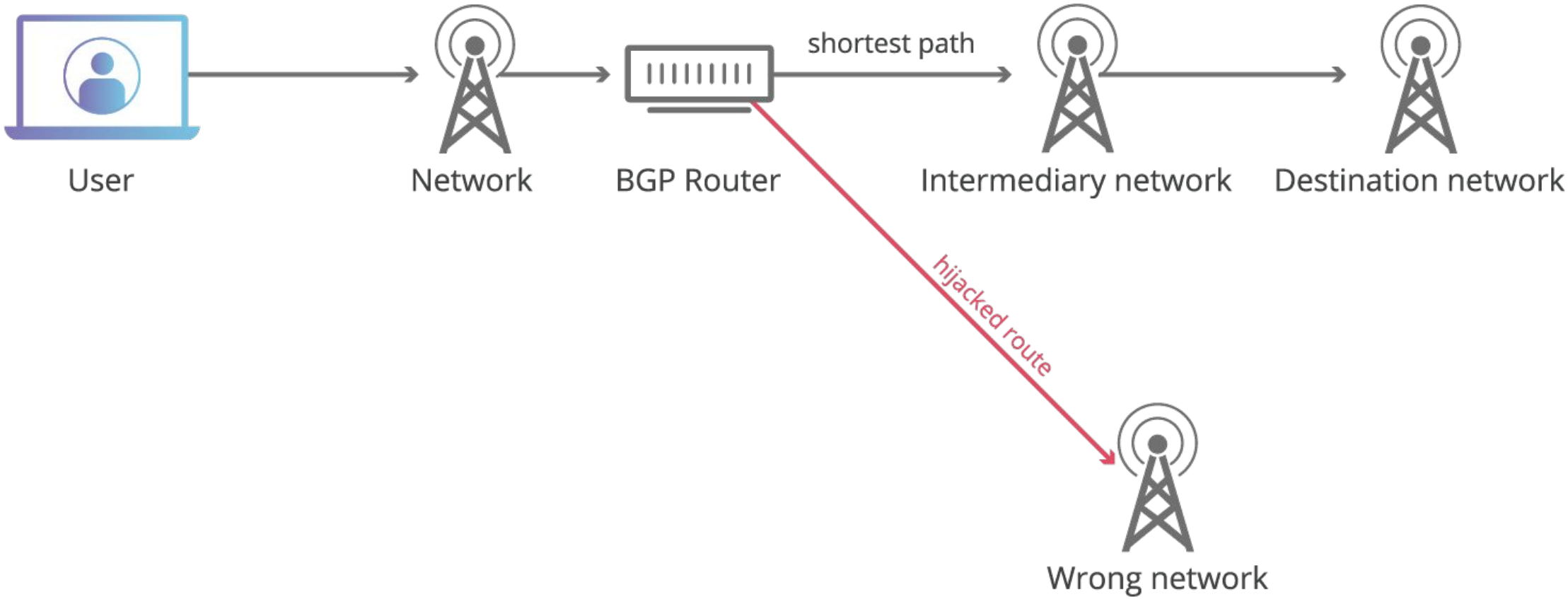
https://bgp.tools/prefix/140.113.0.0/16#connectivity

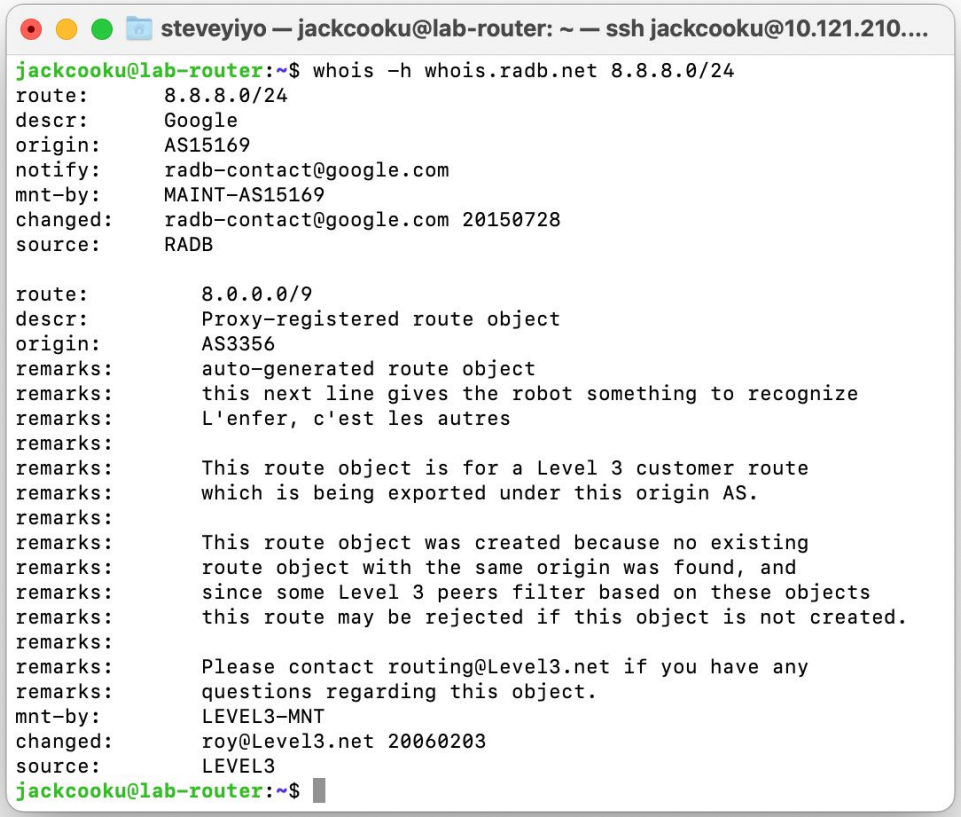
ASN

- 自治系統編號
- 由五大 RIR 分配
- 用於 BGP 宣告路由



那… 如果有人亂宣告路由，害我被路由劫持了，怎麼辦

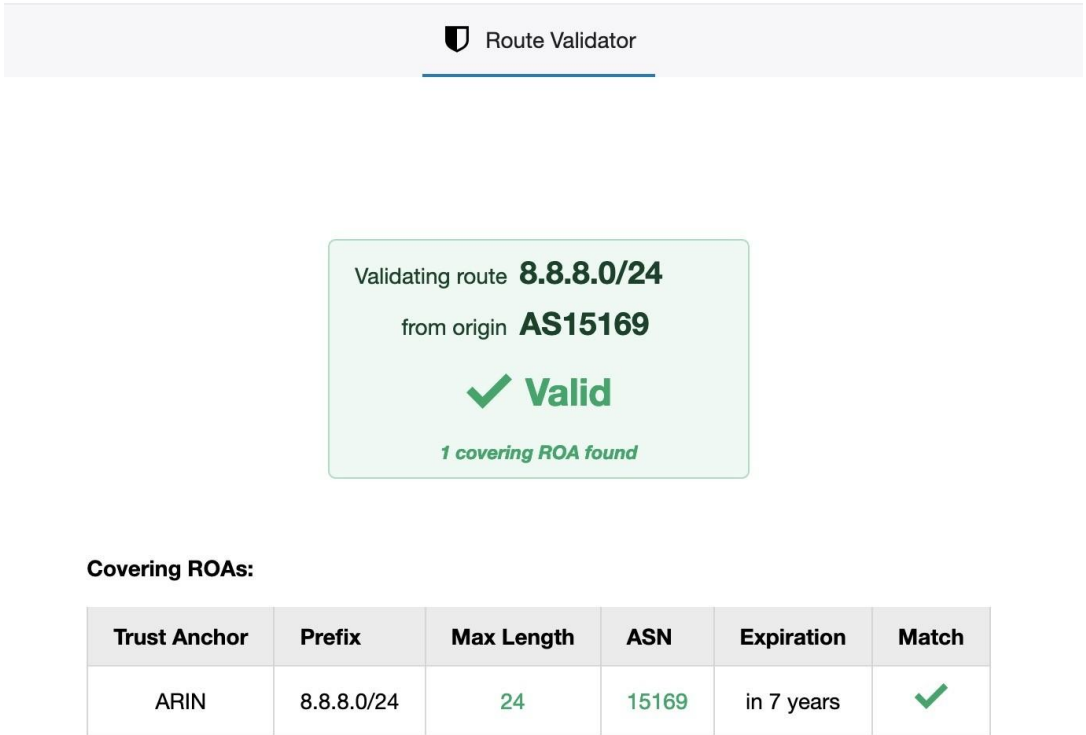




BGP Filter

- Route Object
- AS-SET
- RIR / RADB / ALTDB

BGP Filter



- RPKI
- RFC 3779
- 基於公共密鑰基礎建設框架
- 路由來源授權

BGP Community

- 由各 AS 自行定義，通常會在 WHOIS、PeeringDB 或網站上公告。部分 AS 則為內部使用。
- 用於標記路由，控制 BGP 路由選擇。
- 有兩種類型
  - Standard Community：由兩部分組成，格式為 ASN:代碼 (e.g. 7480:100)。
  - Extended Community：通常用於 MPLS VPN
- 常見的使用案例
  - 影響路由的宣告：如限制路由到特定的國家或地區，或是禁止宣告給其他 AS 網路。
  - 調整路由優先級：高優先級的路由可以設定更高的 LOCAL_PREF。

BGP 與 OSPF 選路方面的不同

- OSPF 為 LSA 最短路徑
- BGP 為最「佳」路徑
    - 最佳有時候不是最短的，有可能是最便宜的線路。
    - 因為某些政策目的，特地繞遠路。
    - e.g. 假設 HE 是 TWGate 的 Provider，而 TWGate 則是其他 ISP 的 provider，即使其他 ISP 跟 HE 在 IX 有 peering，HE 仍然會走 TWGate 去其他 ISP。
        - Customer 的 LOCAL_PREF 一般都會比 Peer 高。
        - 目的是要盡可能把流量丟給客戶（不然線路都用不滿就不可能升級啦）。

我也想玩 BGP，我可以用 VM 組網嗎？

當然可以！

選擇開源路由套件

- FRRouting
- BIRD
- GoBGP
- 或是基於 RFC 4271 自己開發一個（x

此外你還要…

修改 sysctl 允許 Linux 轉發封包

- 預設不會將封包轉發給第三者

開放相關防火牆

- 設定 VLAN or VRF（視情況而定）

如果我想異地組網，沒有實體互聯？

- 利用 VPN Tunnel
  - WireGuard
  - GRE
  - SIT
  - … 任意 VPN 協議皆可
- 防火牆不能擋住相關 Port 及路由
- 切記不要覆蓋 VPN Endpoint

覆蓋 VPN Endpoint？那是什麼？

宣告的 Prefix 覆蓋了 VPN 連接的 Endpoint

e.g. 假設 VPN Server IP 為 10.0.0.1，透過該 Tunnel 進行互聯，並宣告 10.0.0.0/24。路由會直接導向到 VPN Interface，進一步將 Internet 路由覆蓋。

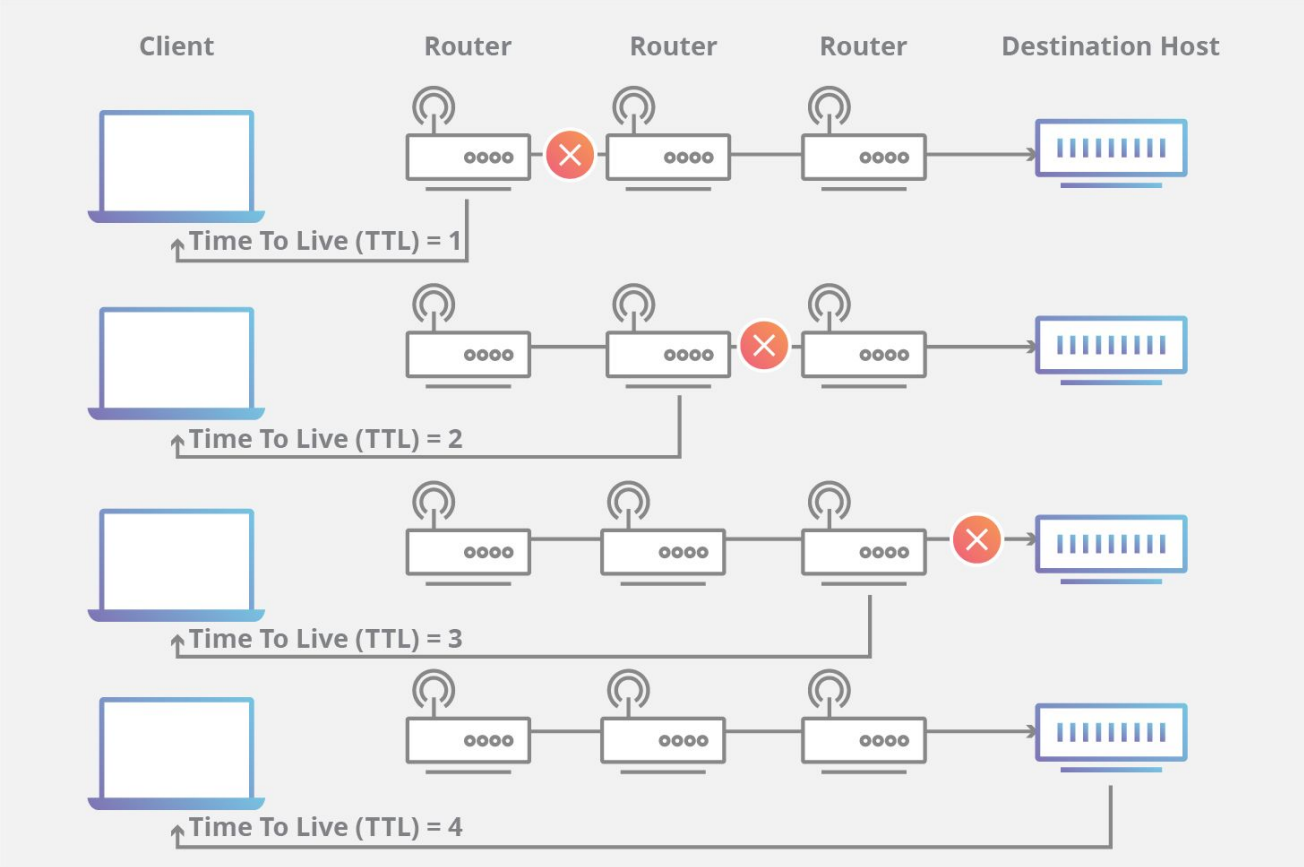
如何解決路由覆蓋問題？

- 利用 VRF (Virtual routing and forwarding)
- 使用靜態路由 (e.g. sudo ip route add 10.0.0.1/32 via x.x.x.x)

什麼是 VRF？

- Virtual routing and forwarding
- 擁有獨立路由表及路由策略，互不干擾
- 常用於 MPLS 網路中
- 實質上還在同一台物理設備上
- E.g. 可以將透過 main table 運作的 VPN Interface 變成 VRF 內專用

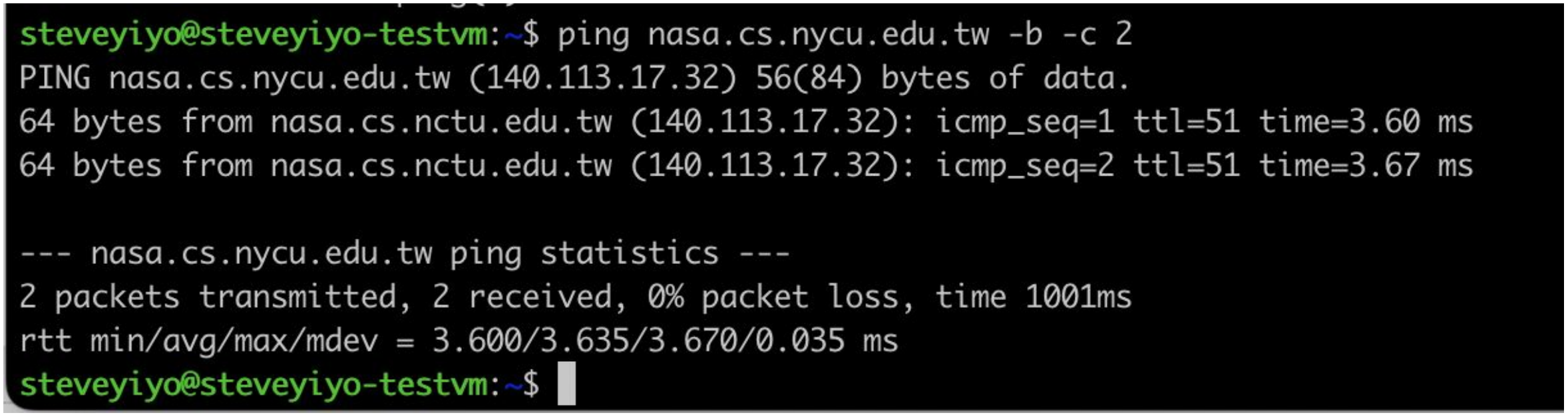
如果路由互相指向，
造成無限循環…？



TTL (Time to Live)

- 當封包轉發一次，TTL 減一。
- 防止路由設定錯誤，導致封包無限輪迴造成網路設備故障。

- 此圖原 TTL 為 64，經過多節點後變成 55。



- 封包 Out 為 TTL 64
- 封包 In 為 TTL 51

```
2 steveyiyo@steveyiyo-testvm:~/sonic-buildimage$ sudo tcpdump -i any host 140.113.17.32 -v
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
19:16:50.843181 eth0  Out IP (tos 0x0, ttl 64, id 8037, offset 0, flags [DF], proto ICMP (1), length 84)
    44.31.73.1 > nasa.cs.nctu.edu.tw: ICMP echo request, id 49229, seq 1, length 64
19:16:50.846771 eth0  In  IP (tos 0x0, ttl 51, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
    nasa.cs.nctu.edu.tw > 44.31.73.1: ICMP echo reply, id 49229, seq 1, length 64
19:16:51.844903 eth0  Out IP (tos 0x0, ttl 64, id 8268, offset 0, flags [DF], proto ICMP (1), length 84)
    44.31.73.1 > nasa.cs.nctu.edu.tw: ICMP echo request, id 49229, seq 2, length 64
19:16:51.848496 eth0  In  IP (tos 0x0, ttl 51, id 0, offset 0, flags [DF], proto ICMP (1), length 84)
    nasa.cs.nctu.edu.tw > 44.31.73.1: ICMP echo reply, id 49229, seq 2, length 64
```

FRRouting 路由器

- 建置 eBGP 網路，與其他 AS 交換路由。
- 控制每個 AS Neighbor 能走的路。
- 自行控制優先級。
- 開源，不用花費很多錢。

```
Hello, this is FRRouting (version 9.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

cr1-tpe.steveyi.net# show bgp vrf vrf_PUBLIC summary

IPv4 Unicast Summary (VRF vrf_PUBLIC):
BGP router identifier 103.69.92.110, local AS number 7480 vrf-id 26
BGP table version 13245659
RIB entries 1747837, using 160 MiB of memory
Peers 9, using 181 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt Desc
23.142.145.2    4       7480         0         0        0    0    0    never      Connect        0 N/A
103.69.92.33    4        945   8034180      9275 13245659    0    0 6d10h17m       796606        6 AUGUST
103.69.92.37    4      48024         0         0        0    0    0    never      Active         0 NEROCLOUD
103.69.92.39    4      38254      6805   4574420 13245659    0    0 14:00:34            1   951582 ITLAB
103.69.92.43    4     212358         0         0        0    0    0    never      Active         0 DA21510
103.69.92.51    4       7480   5023863   5036831 13245659    0    0 6d10h17m            0   951582 RR-TPE
103.69.92.53    4     208137      9263   5074739 13245659    0    0 6d10h17m            1   951582 LINLEE
103.172.40.158  4        983         0         0        0    0    0    never      Connect        0 AKARI
154.18.24.177   4        174   1712399      9420 13245659    0    0 6d10h17m       916093        6 COGENT

Total number of neighbors 9

IPv6 Unicast Summary (VRF vrf_PUBLIC):
BGP router identifier 103.69.92.110, local AS number 7480 vrf-id 26
BGP table version 2818238
RIB entries 388785, using 36 MiB of memory
Peers 5, using 101 KiB of memory

Neighbor             V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt Desc
2001:df2:5741:2::2   4        945   4449693      9267  2818238    0    0 6d10h17m       204909        4 AUGUST
2001:df2:5741:4::2   4       7480   1319284   1335982  2818238    0    0 6d10h17m            0   205421 RR-TPE
2402:4480:2:6::16:1  4        174    953874      9265  2818238    0    0 6d10h17m       180150        4 COGENT
2407:cdc0:eeee::158:1 4       983         0         0        0    0    0    never      Active         0 AKARI
2602:feda:d90:2::1   4      38254      6790   1084636  2818238    0    0 14:00:34            1   205421 ITLAB

Total number of neighbors 5
```

Lab01: Install FRRouting

```
# Install dependencies
sudo apt update && apt upgrade -y
sudo apt install -y curl gnupg2 traceroute

# Import Source and add the GPG Key
curl -s https://deb.frrouting.org/frr/keys.asc | sudo apt-key add -
FRRVER="frr-stable”
echo deb https://deb.frrouting.org/frr $(lsb_release -s -c) $FRRVER | sudo tee -a /etc/apt/sources.list.d/frr.list

# Install FRRouting and enable all feature
sudo apt update -y && sudo apt install -y frr frr-pythontools
sudo sed -i "s/=no/=yes/g" /etc/frr/daemons
sudo systemctl start frr
```

Lab01: Install FRRouting

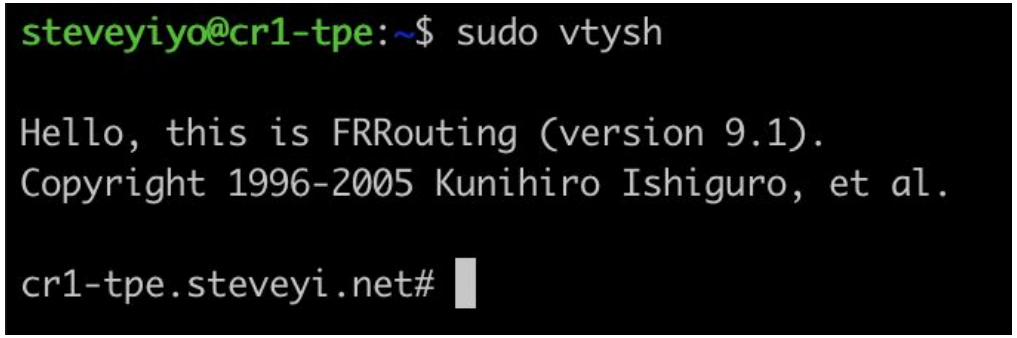
```
# Enable IP Forwarding
echo "
net.ipv4.conf.all.forwarding = 1
net.ipv6.conf.all.disable_ipv6 = 0
net.ipv6.conf.default.disable_ipv6 = 0
net.ipv6.conf.lo.disable_ipv6 = 0
net.ipv6.conf.default.forwarding = 1
net.ipv6.conf.all.forwarding = 1
net.ipv6.conf.all.proxy_ndp = 1
net.ipv6.conf.all.accept_ra = 2
" | sudo tee -a /etc/sysctl.conf

sudo sysctl -p
```

```
sudo sysctl -p

net.ipv4.conf.all.forwarding = 1
net.ipv6.conf.all.disable_ipv6 = 0
net.ipv6.conf.default.disable_ipv6 = 0
net.ipv6.conf.lo.disable_ipv6 = 0
net.ipv6.conf.default.forwarding = 1
net.ipv6.conf.all.forwarding = 1
net.ipv6.conf.all.proxy_ndp = 1
net.ipv6.conf.all.accept_ra = 2

net.ipv4.conf.all.forwarding = 1
net.ipv6.conf.all.disable_ipv6 = 0
net.ipv6.conf.default.disable_ipv6 = 0
net.ipv6.conf.lo.disable_ipv6 = 0
net.ipv6.conf.default.forwarding = 1
net.ipv6.conf.all.forwarding = 1
net.ipv6.conf.all.proxy_ndp = 1
net.ipv6.conf.all.accept_ra = 2
root@cr1-tpe:~#
```

Lab01: Install FRRouting

Enter the FRRouting interactive CLI

sudo vtysh

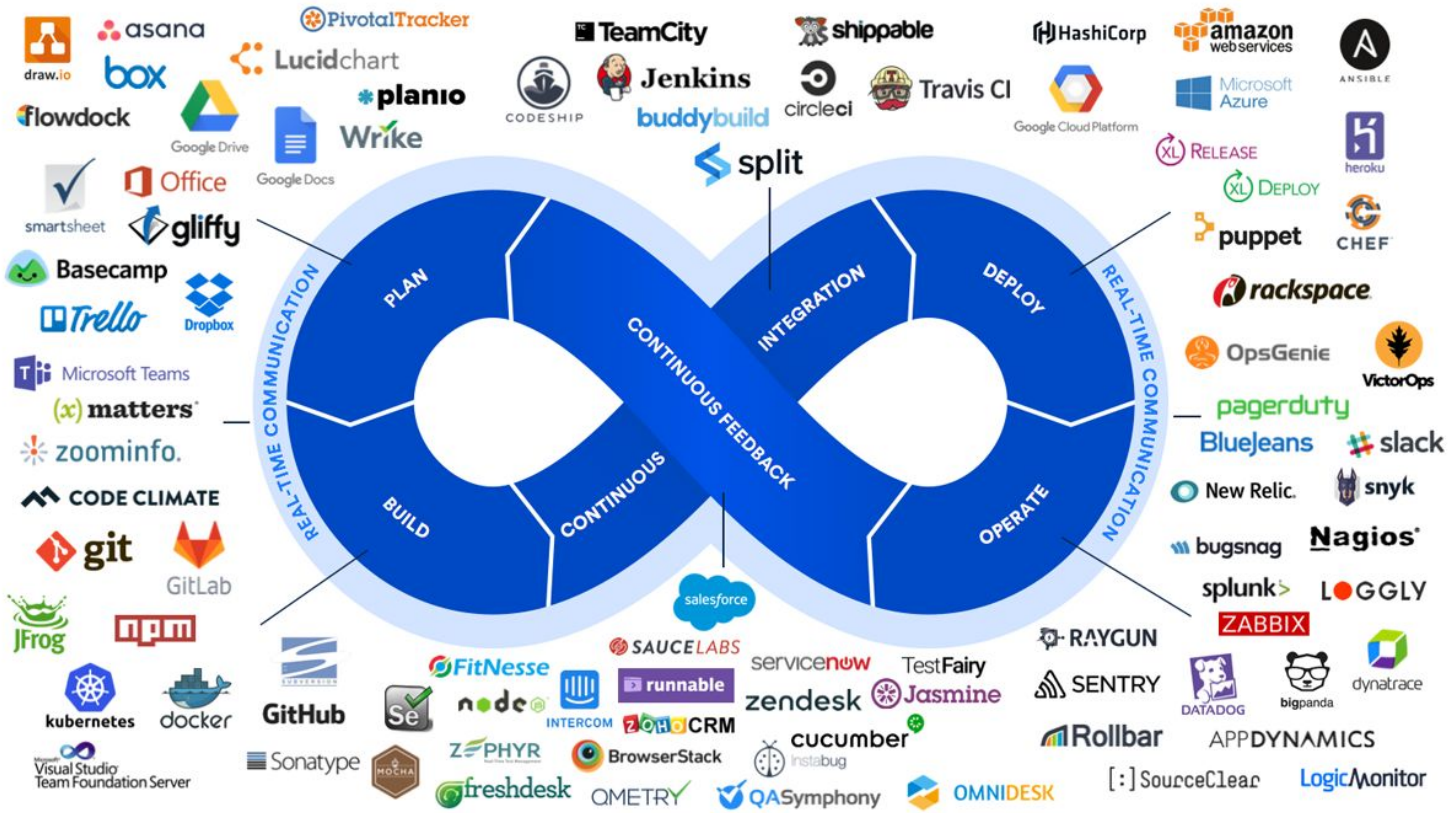


可以透過軟體來管理 BGP 路由宣告嗎？

- 當然可以！
- CI/CD 是一個好工具！

DevOps

通過自動化「軟體交付」和「架構變更」



CI/CD？

- 將程式的流程以自動化的方式呈現
  - 開發
  - 單元測試 (Unit testing)
  - 整合測試 (Integration testing)
  - 部署
  - 維運
  - 版本更新
  - … 任何你想得到的功能，運用 CI/CD 工具幾乎都可以做到！

常見的 CI/CD 工具

GitHub Action, GitLab CI, Jenkins, Drone CI …

Q：我們要如何選擇工具的使用呢？

A：使用取決於你的工具搭配，以及想不想自己維護 CI 伺服器。
部分開發者會使用 GitHub Action，並搭配 VPN 來部署服務到自己的伺服器。

某些企業也會因為安全因素，所以自己 host 唷！

那 那到底要怎麼利用 CI 呀？

- CI 有多種觸發機制 (Trigger)
  - 定時觸發 e.g. 自動撈資料
  - Git 的版本改變 e.g. push 新的 commit
  - Git 打版，進行版本更新 e.g. 將新的程式部署到伺服器
  - Docker Image 打包、單元邏輯測試，整合測試等等...

將 DevOps 整合進路由器的優點？

- 利用 Unit Test 及 Integration testing 確保符合預期。
- 在 Docker 中提前測試成果，一切正常後 Merged 到生產環境上。
- 以程式化執行，降低錯誤機率。
- 利用 Git 來達到版本控制。
- ... SDN 網路

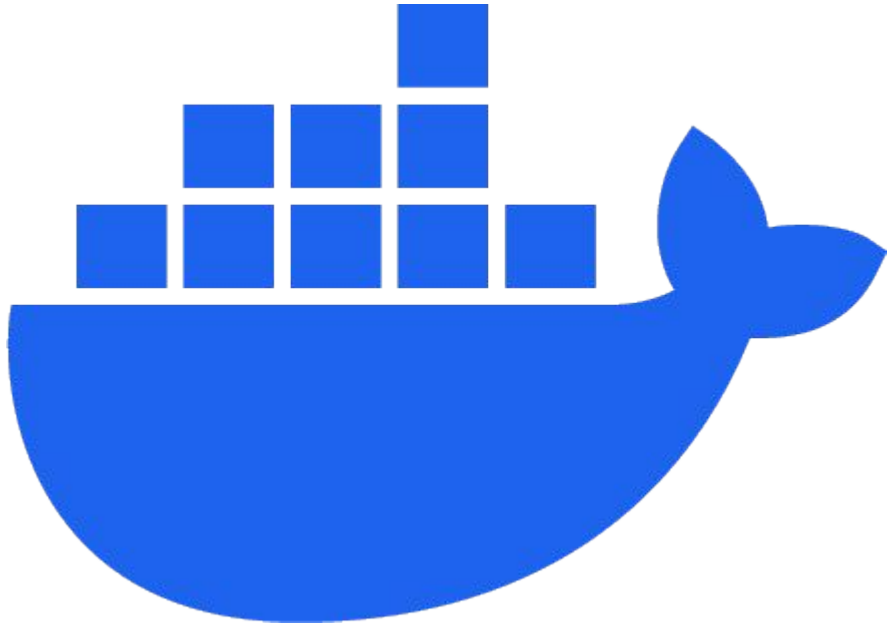
為簡化 CI/CD 解說，簡報中會以網站伺服器為例

Docker 簡介

- 容器化。
- 可以在任何系統上執行，無需擔心環境問題。
- 乾淨，適合 Application 部署。
- 容易與 kubernetes 整合，利用水平擴充及自動規模化 (Autoscaling) 達到冗余功能。
- 當然，也可以用於 CI 測試環境。

Container 及虛擬化的差異

- **容器化 Containerization**
  - 環境乾淨
  - 輕量
  - 速度快
  - 一致性與可攜性
  - 適合微服務
- **虛擬化 Virtualization**
  - 隔離性
  - 硬體抽象化
  - 容錯和災難恢復
  - 資源分配和擴展性



Docker 使用範例

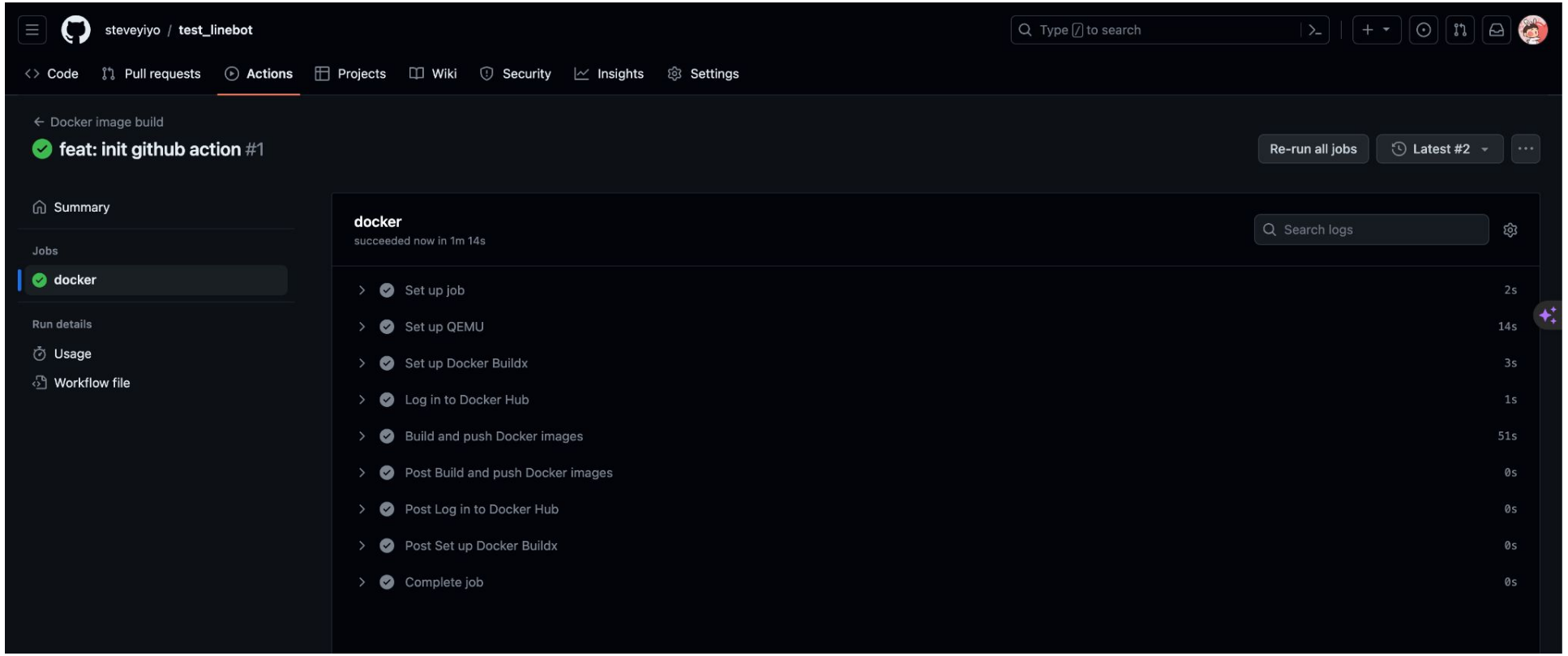
- 程式開發完成後，打包為 Docker Image。
- 透過 Linux Shell 測試網路架構的成果。
- e.g. OpenVswitch, Mininet, BGP Routing, FRRouting, etc …

那麼 CI 到底要怎麼做呢？

- GitHub Action 為例。
- 透過 YAML 檔來定義 CI 的工作內容，包括：
  - 觸發條件（當有新的 commit push 到特定 branch）
  - 執行環境（Ubuntu 20.04）
  - 工作順序（登入容器託管平台、打包、上傳）
  - 環境變數（包括密鑰、API Key 等等）
  - 上傳以及 tag docker image

為方便解說，右圖以 Docker Image Build 為例，打包完成後上傳到容器管理平台。

成效


steveyiyo / test_linebot
Type / to search
Code
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Docker image build
feat: init github action #1
Re-run all jobs
Latest #2
Summary
Jobs
docker
Run details
Usage
Workflow file
docker
succeeded now in 1m 14s
Search logs
Set up job 2s
Set up QEMU 14s
Set up Docker Buildx 3s
Log in to Docker Hub 1s
Build and push Docker images 51s
Post Build and push Docker images 0s
Post Log in to Docker Hub 0s
Post Set up Docker Buildx 0s
Complete job 0s

如果要讓 CI 直接將服務部署上去呢？

- 平常怎麼部署，就讓程式執行一樣的事情。
- e.g. 原本將程式開發完成後，打包成 Docker Image 後，手動到 SSH 內執行相關部署指令。

產生 SSH Key

- ssh-keygen
- 加密算法自行選擇

```
steveyiyo@MacBook-Air-3 CI_Test % ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/steveyiyo/.ssh/id_rsa): ./id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ./id_rsa
Your public key has been saved in ./id_rsa.pub
The key fingerprint is:
SHA256:e6bfeciM7n+t6MfeST62/z//RGb4892ogbJ7VViN9xk steveyiyo@MacBook-Air-3.local
The key's randomart image is:
+---[RSA 4096]----+
|               o |
|              oEo|
|             o .+|
|            . .o.|
|        S    .. +|
|         . ..  = |
|        o +=.o +o|
|         *o.++BBB|
|        +*+oB*==^|
+----[SHA256]-----+
```

General

Access

Collaborators

Code and automation

Rules

Actions

Webhooks

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Actions

Codespaces

Dependabot

Integrations

GitHub Apps

Email notifications

Actions secrets / New secret

Name *

SSH_PRIVATE_KEY

Secret *

ovUpXXuoHeLXkNkMkf4aB2nSuD6thoGNm7gnGp+qHbXoj/Lpu+YIpENr4K+HEeoqNTUqEl

-----END OPENSSH PRIVATE KEY-----

Add secret

如果要讓 CI 直接將服務部署上去呢？

- 定義 CI 工作內容
  - 新增 SSH Private Key
  - 執行 SSH 連接及指令

worker.yaml

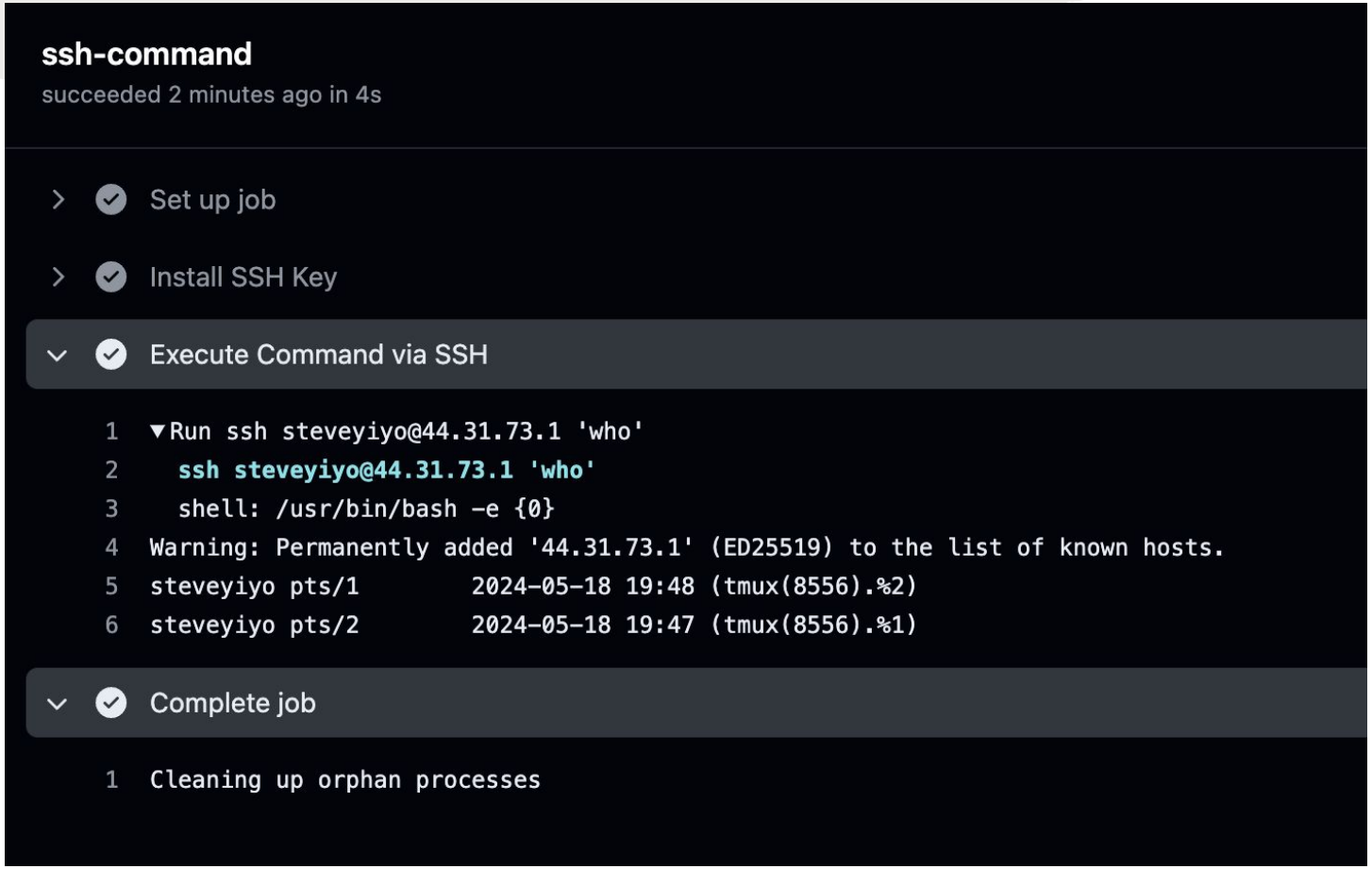
.github > workflows > worker.yaml

```yaml
name: Run SSH Commands on Remote Server

on: [push]

jobs:
  ssh-command:
    runs-on: ubuntu-latest
    steps:
    - name: Install SSH Key
      run: |
        mkdir -p ~/.ssh
        echo "${{ secrets.SSH_PRIVATE_KEY }}" > ~/.ssh/id_rsa
        chmod 600 ~/.ssh/id_rsa
        echo "Host *" >> ~/.ssh/config
        echo "  StrictHostKeyChecking no" >> ~/.ssh/config
        echo "  UserKnownHostsFile=/dev/null" >> ~/.ssh/config

    - name: Execute Command via SSH
      run: ssh steveyiyo@44.31.73.1 'who'

```

* 需要先將對應的 SSH Public Key 加到遠端主機

如果要讓 CI 直接將服務部署上去呢？

- CI 連到遠端 SSH 主機
- 執行 who 指令

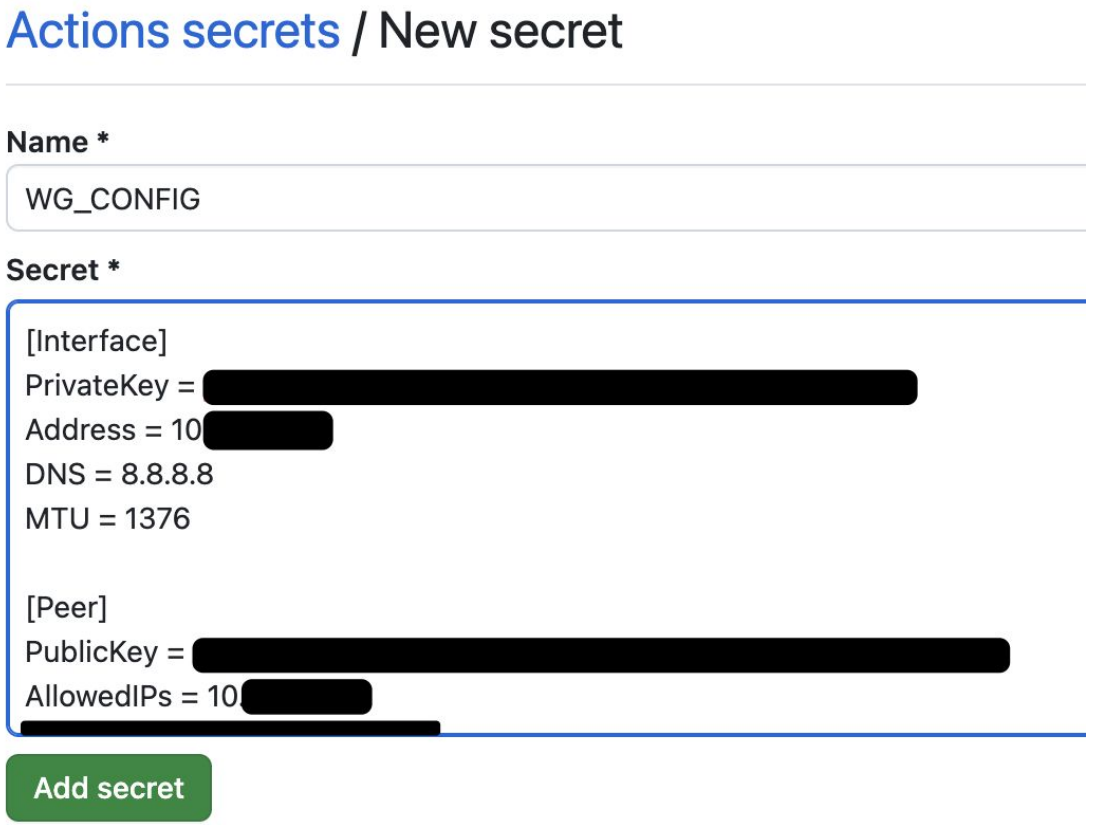


如果我的遠端 SSH 主機要透過 VPN 連接？

- 當然也可以讓你的 CI 在工作時連接 VPN！
- 自行在內網部署 CI Worker，解決連接問題。* 部分企業也是 Self Hosted

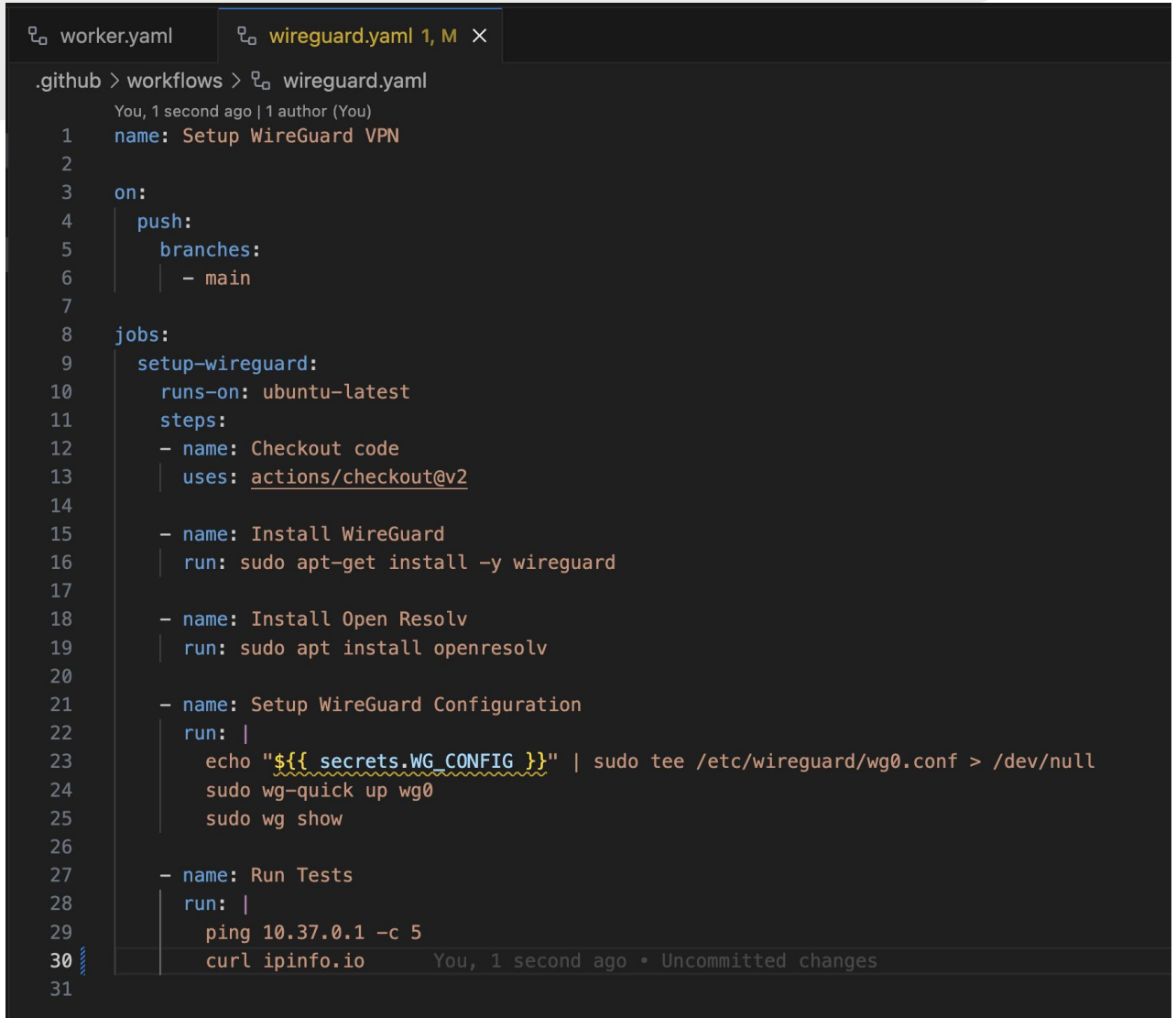
如果我的遠端 SSH 主機要透過 VPN 連接？

- 將 WireGuard Config 新增進 Secret
- DNS 可設可不設，取決於網路架構及程式
- MTU 取決於網路架構
- AllowedIPs 不得為 0.0.0.0/0，否則會故障



如果我的遠端 SSH 主機要透過 VPN 連接？

- 以 Ubuntu 最新版作為 Worker
- 安裝 WireGuard 及 Open Resolv
- 新增 WireGuard Config
- 執行成果



Run Tests

```
▶ Run ping 10.37.0.1 -c 5

PING 10.37.0.1 (10.37.0.1) 56(84) bytes of data.
64 bytes from 10.37.0.1: icmp_seq=1 ttl=62 time=395 ms
64 bytes from 10.37.0.1: icmp_seq=2 ttl=62 time=247 ms
64 bytes from 10.37.0.1: icmp_seq=3 ttl=62 time=248 ms
64 bytes from 10.37.0.1: icmp_seq=4 ttl=62 time=247 ms
64 bytes from 10.37.0.1: icmp_seq=5 ttl=62 time=247 ms
--- 10.37.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4001ms
rtt min/avg/max/mdev = 247.415/276.928/394.772/58.921 ms
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100   263  100   263    0     0   7317      0 --:--:-- --:--:-- --:--:--  7514
{
  "ip": "20.55.15.212",
  "city": "Washington",
  "region": "Virginia",
  "country": "US",
  "loc": "38.7135,-78.1594",
  "org": "AS8075 Microsoft Corporation",
  "postal": "22747",
  "timezone": "America/New_York",
  "readme": "https://ipinfo.io/missingauth"
}
```

SRE 是什麼？

- 最早由 Google 提出的概念。
- 以軟體工程方法來解決維運問題。
- 自動化系統的可擴展性、可用性和性能。
- 這個概念提出時，是想解決「開發者快速迭代」以及「維運者想保持穩定」的矛盾。

各種 SRE 團隊？

- 在大型企業中，除了將 SRE 團隊分離出來以外，還會在其進行二次分離。每個團隊負責不同的業務。通常有...
- Infrastructure SRE
  - 維運基礎設施，包括網路交換機、路由器、伺服器以及機房等等。通常會有一批專門的團隊來處理。e.g. Cloudflare, Google, Meta, etc.
- Platform SRE
  - 有點類似開發 / 維運內部私有雲環境，在基礎設施上建制 SaaS 服務，提供其他團隊使用如 VM, S3, CDN, k8s cluster, GitLab, Harbor, etc.
- Application SRE
  - 這個團隊更偏向在生產環境的產品，比如我們常見的網站伺服器，遊戲的後端等等。

如何成為 SRE 工程師？

- 取決於職位跟經驗，通常會要求擁有部署及維護程式，部分職位則希望有基礎設施或網路的經驗。
- 找到 JD (Job Description)，實作專案！

CI/CD, DevOps 與 SRE 的關聯？

- DevOps 和 SRE 都旨在改善軟體開發和維運的效率和效果
- CI/CD 流程則實現快速、穩定的軟體發布。同時也為 DevOps 和 SRE 提供了工具和流程，確保軟體可以持續、自動地構建、測試及部署。

嘗試使用這些技術，提高你的開發效率及能力！

問答時間

Thank You!