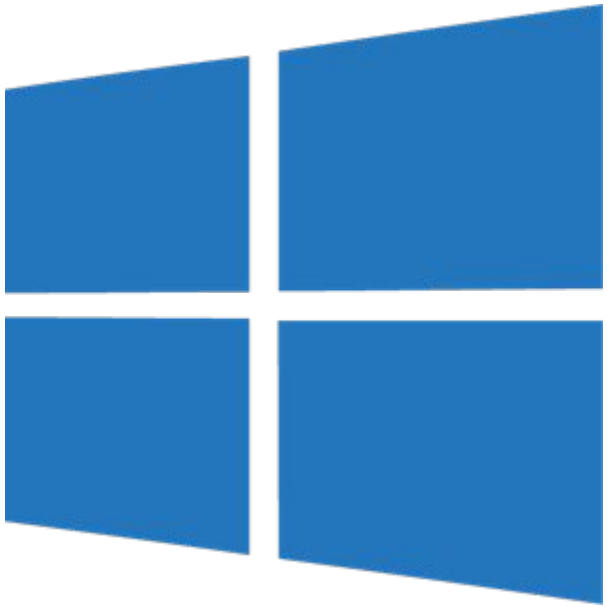
Lab1. IP info, Network Tools

TA 許仲宇 (hsuchy)
credit to 陳品劭 (cps)

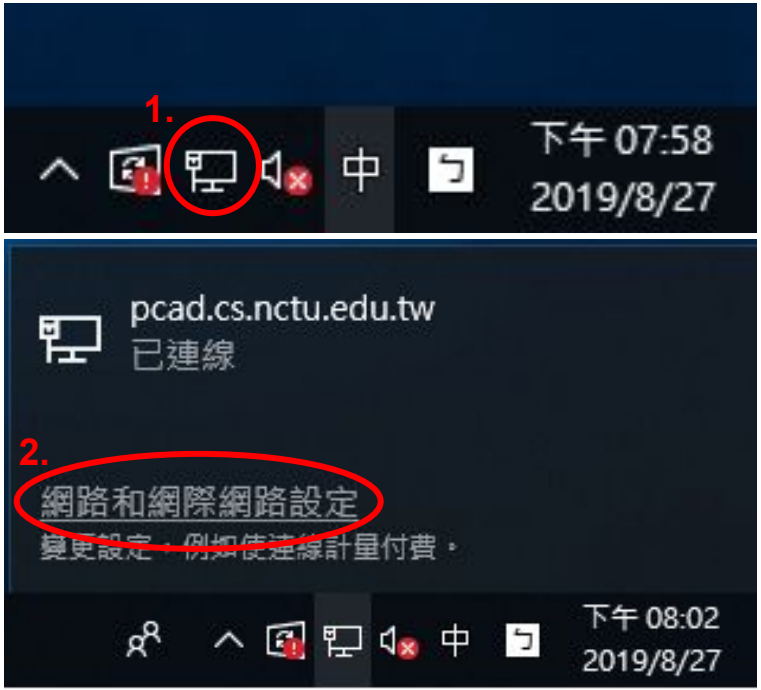
Purpose

- Windows and Linux (Ubuntu) IP Setting
- Usage of some network tools

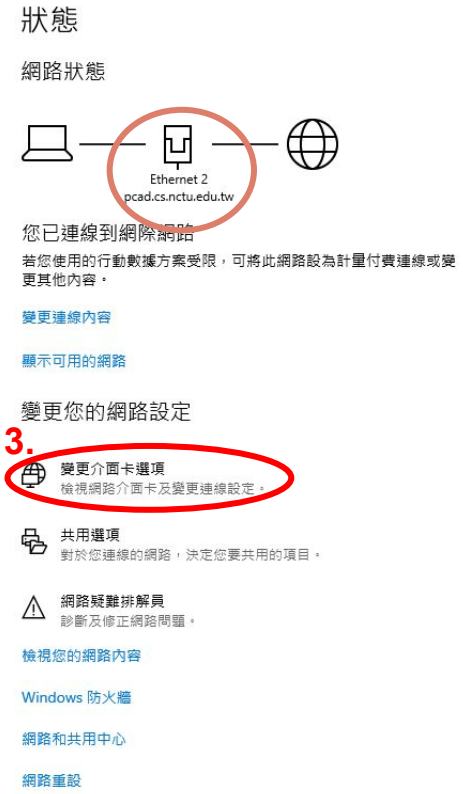
Windows



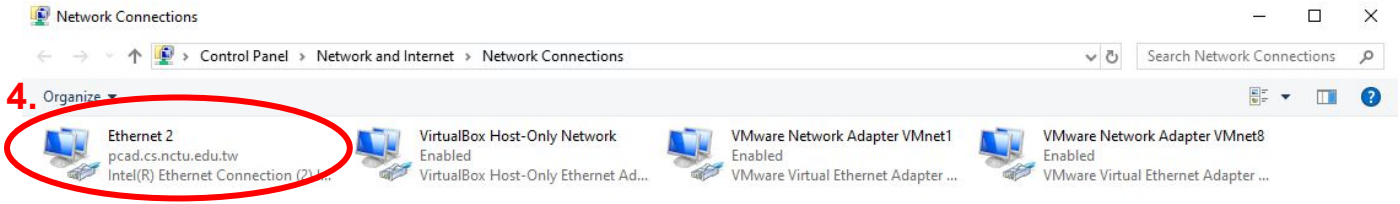
Windows 10 IP Setting(1)



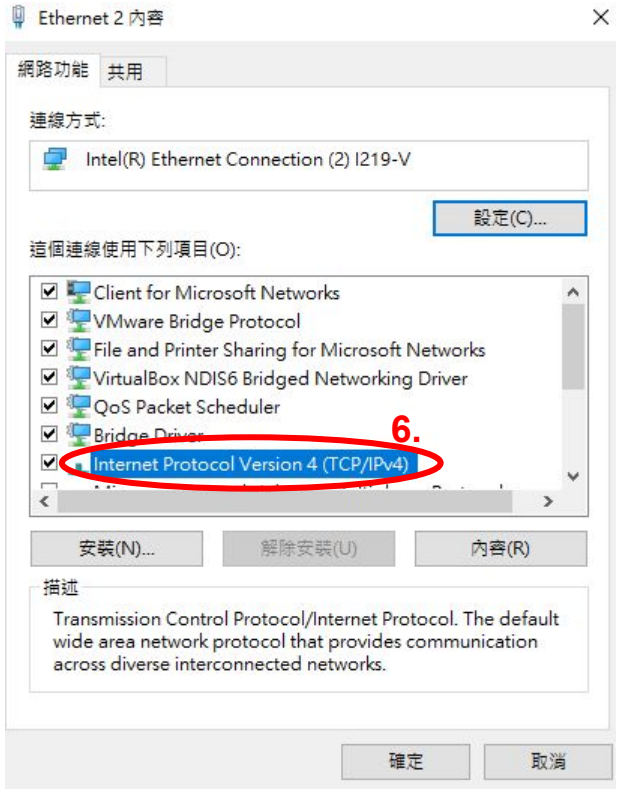
Windows 10 IP Setting (2)



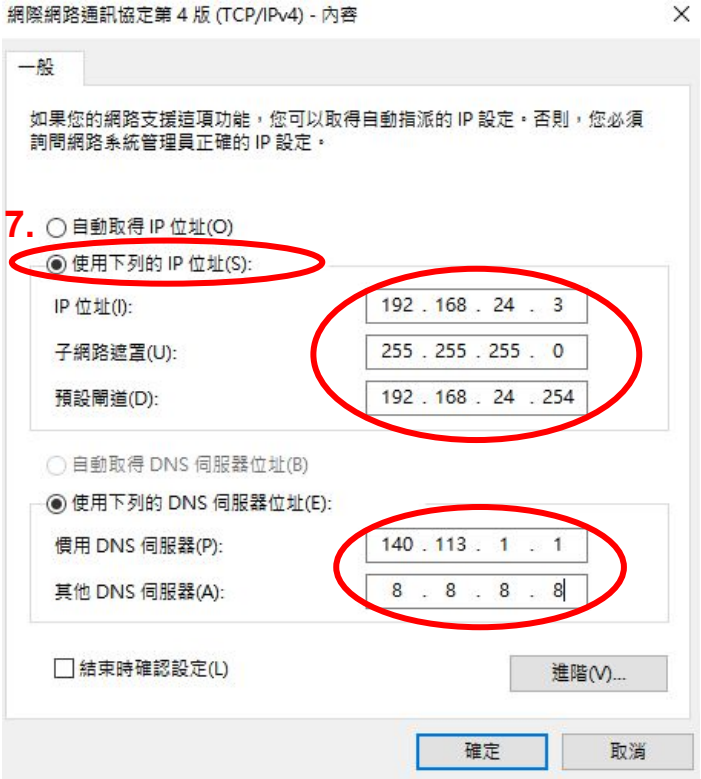
Windows 10 IP Setting (3)



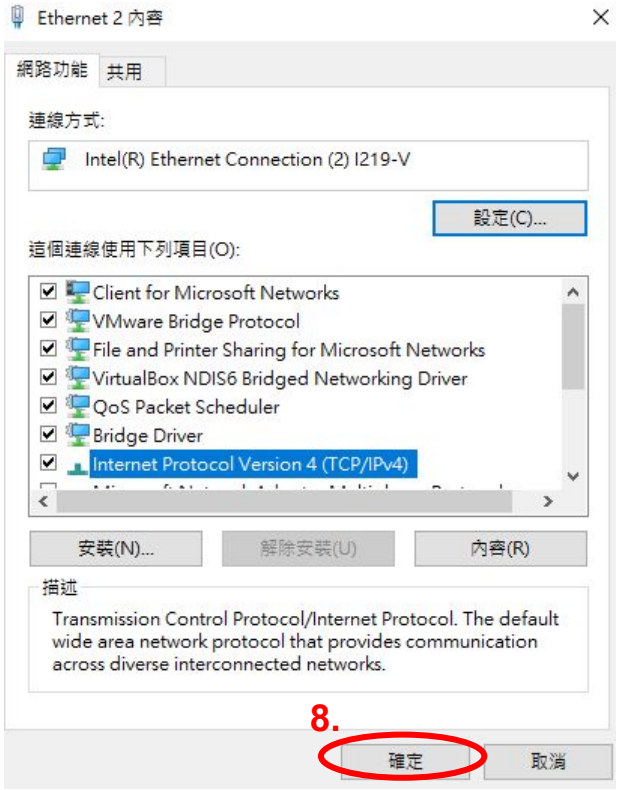
Windows 10 IP Setting (4)



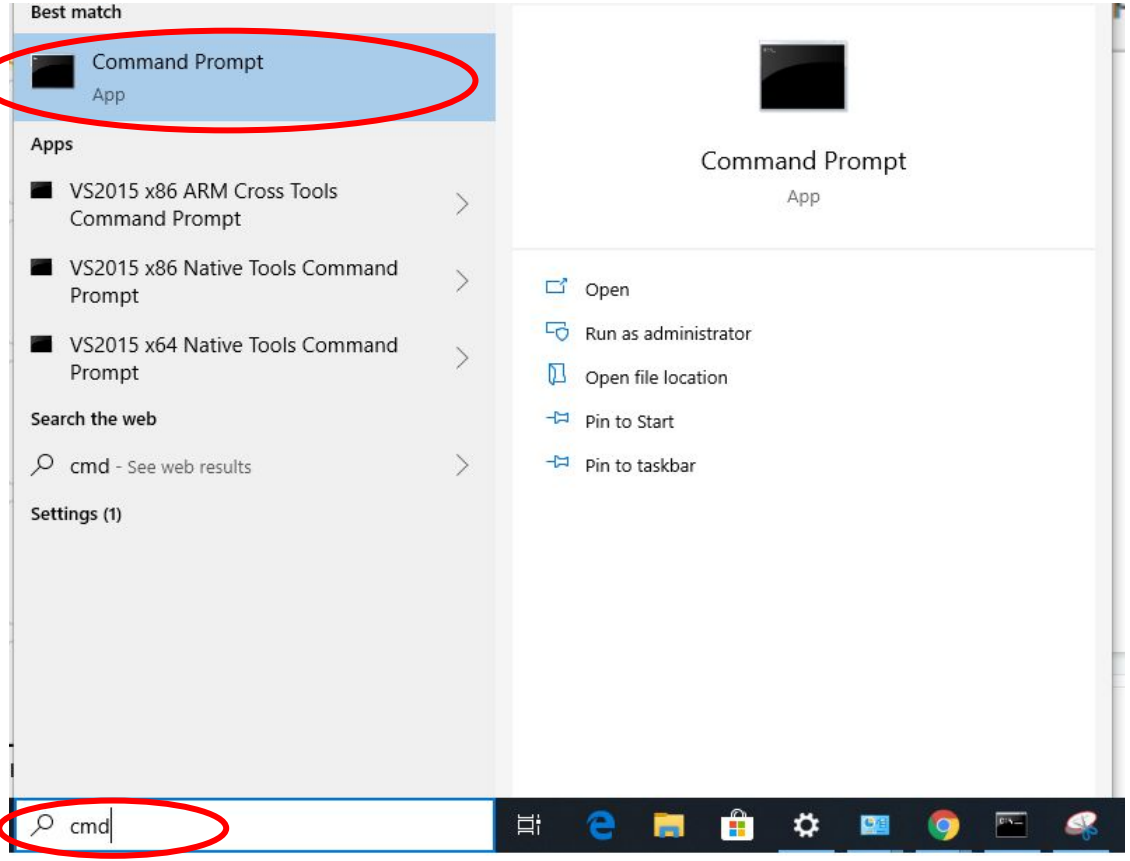
Windows 10 IP Setting (5)



Windows 10 IP Setting (6)



Open cmd


Best match
Command Prompt
App
Apps
VS2015 x86 ARM Cross Tools Command Prompt
VS2015 x86 Native Tools Command Prompt
VS2015 x64 Native Tools Command Prompt
Search the web
cmd - See web results
Settings (1)
Command Prompt
App
Open
Run as administrator
Open file location
Pin to Start
Pin to taskbar
cmd

Ipconfig (1)

- Used to check information of network interface cards.

```
D:\Users\kutk075>ipconfig

Windows IP 設定

...
乙太網路卡 Ethernet 2:

   連線特定 DNS 尾碼 . . . . . . . . : cccs
   連結-本機 IPv6 位址 . . . . . . . . : fe80::1521:20cd:f3e:42c5%20
   IPv4 位址 . . . . . . . . . . . . : 192.168.24.3
   子網路遮罩 . . . . . . . . . . . . : 255.255.255.0
   預設閘道 . . . . . . . . . . . . . : 192.168.24.254
...
```

Ping

- Command

```
ping ip address [-t]
```

- this command will send an packet (ICMP packet) to destination address.
- Used to check whether the route is unblocked to the destination
- -t: Repeat sending ICMP packets until pressing Ctrl+c on keyboard
    - if -t is not present, it will send ICMP packets for 3 times, by default.

Ping Example (1/2)

```
D:\Users\kutk075>ping 8.8.8.8

Ping 8.8.8.8 (使用 32 位元組的資料):
回覆自 8.8.8.8: 位元組=32 時間=3ms TTL=54
回覆自 8.8.8.8: 位元組=32 時間=3ms TTL=54
回覆自 8.8.8.8: 位元組=32 時間=3ms TTL=54
回覆自 8.8.8.8: 位元組=32 時間=3ms TTL=54

8.8.8.8 的 Ping 統計資料:
    封包: 已傳送 = 4, 已收到 = 4, 已遺失 = 0 (0% 遺失),
大約的來回時間 (毫秒):
    最小值 = 3ms, 最大值 = 3ms, 平均 = 3ms
```

- 位元組
  - 電腦發送的封包大小
- 時間
  - 封包從主機到目的地，再從目的地返回主機所花的時間（ = round-trip time, RTT）
- TTL (Time to Live)
  - 一般從 64 開始扣減
  - TTL = 54 代表中間經過 10 台主機

Route Print

- Command

```
route print
```

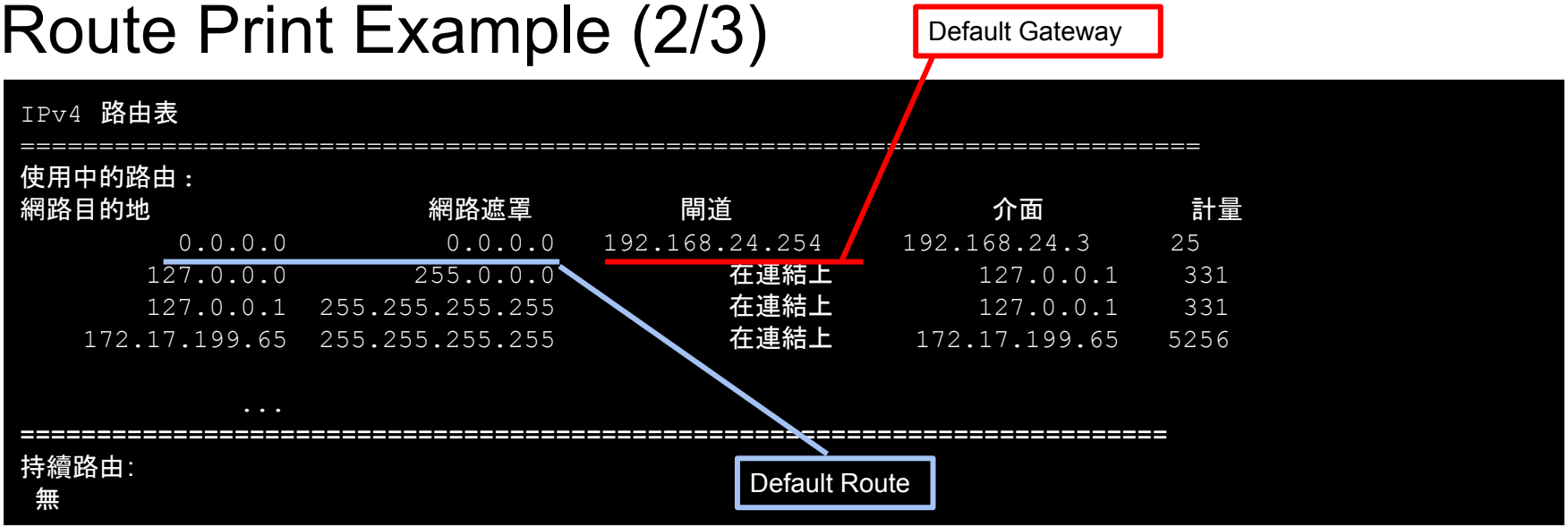
- Used to check routing table
- By routing table, we can know information of routes. Such as,
  - corresponding dest ip and mask of a route.
  - which route will be used when sending a packet.
  - which NICs (Network Interface Card) will a route use to send out packet.
  - will a route go through gateway or not.
  - etc…
- Useful for hosts with multiple NICs

Route Print Example (1/3)

```
D:\Users\kutk075>route print
```

```
===========================================================================
介面清單
 11...0a 00 27 00 00 0b ......VirtualBox Host-Only Ethernet Adapter
 20...10 7b 44 18 cb a2 ......Intel(R) Ethernet Connection (2) I219-V
  8...00 50 56 c0 00 01 ......VMware Virtual Ethernet Adapter for VMnet1
  1...........................Software Loopback Interface 1
  3...00 50 56 c0 00 08 ......VMware Virtual Ethernet Adapter for VMnet8
===========================================================================
```

Route Print Example (2/3)



- 閘道
  - 在連結上表示不需經過 gateway，因為該目的地在同一子網域
- 介面
  - 電腦送出封包的 IP 位址
- 計量
  - 傳送成本的**其中一個**參考數字 (數字越低優先權越高)
- 持續路由
  - 重開機還會保存

Tracert

- Command

```
tracert [-d] [-h maximum_hops] target_name
```

- Used to know the route from your host to the destination host.
- -d: Not to resolve addresses to host names of hosts on the route
- -h: Limits the maximum hops to the target host

Tracert Example (1/3)

```
D:\Users\kutk075>tracert 8.8.8.8

在上限 30 個躍點上
追蹤 dns.google [8.8.8.8] 的路由:

  1     5 ms     2 ms     3 ms  192.168.24.254
  2     2 ms     2 ms     2 ms  IP-168-126.cs.nctu.edu.tw [140.113.168.126]
  3     1 ms     1 ms     2 ms  IP-23-150.cs.nctu.edu.tw [140.113.23.150]
  4     1 ms     3 ms     1 ms  140.113.3.242
  5     1 ms     5 ms     3 ms  140.113.3.178
  6     2 ms     2 ms     1 ms  140.113.0.78
  7     4 ms     5 ms     4 ms  142.250.175.52
  8     5 ms     5 ms     7 ms  142.251.55.135
  9     4 ms     4 ms     4 ms  142.251.226.171
 10     4 ms     4 ms     4 ms  dns.google [8.8.8.8]
```

Default gateway

Domain name

Send three times for each hop

Tracert Example (2/3)

- tracert -d *target_name*

```
D:\Users\kutk075>tracert  -d 8.8.8.8

在上限 30 個躍點上追蹤 8.8.8.8 的路由

  1     2 ms     1 ms     2 ms  192.168.24.254
  2     1 ms     3 ms     2 ms  140.113.168.126
  3     1 ms     1 ms     1 ms  140.113.23.150
  4     2 ms    <1 ms     2 ms  140.113.3.242
  5     2 ms     1 ms     2 ms  140.113.3.178
  6     1 ms     3 ms    10 ms  140.113.0.78
  7     5 ms     4 ms     6 ms  142.250.175.52
  8     3 ms     3 ms     4 ms  142.251.55.135
  9     5 ms     5 ms     7 ms  142.251.226.171
 10     6 ms     5 ms     6 ms  8.8.8.8
```

Tracert Example (3/3)

- tracert -h *maximum_hops target_name*

```
D:\Users\kutk075>tracert  -h 5 8.8.8.8

在上限 5 個躍點上
追蹤 dns.google [8.8.8.8] 的路由:

  1     2 ms     4 ms     3 ms  192.168.24.254
  2     2 ms     2 ms     1 ms  IP-168-126.cs.nctu.edu.tw [140.113.168.126]
  3     1 ms     1 ms    12 ms  IP-23-150.cs.nctu.edu.tw [140.113.23.150]
  4     4 ms     2 ms     3 ms  140.113.3.242
  5     3 ms    23 ms     7 ms  140.113.3.178

追蹤完成。
```

Linux (Ubuntu)

Enter terminal of your ubuntu

- Enter terminal

About Your Computer

- Show IP address

```
$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether 00:50:56:93:88:3b brd ff:ff:ff:ff:ff:ff
    inet 10.1.0.13/24 brd 10.1.0.255 scope global noprefixroute ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe93:883b/64 scope link
       valid_lft forever preferred_lft forever
```

About Your Computer

- Show IP route

```
$ ip route
default via 192.168.24.254 dev ens224 proto static metric 25
10.1.0.0/24 dev ens192 proto kernel scope link src 10.1.0.13 metric 100
10.1.0.0/16 via 10.1.0.254 dev ens192 proto static metric 100
140.113.235.0/24 dev ens224 proto kernel scope link src 140.113.235.153
metric 101
```

Windows

```
IPv4 路由表
===========================================================================
使用中的路由：
網路目的地                網路遮罩          閘道          介面          計量
          0.0.0.0          0.0.0.0  192.168.24.254   192.168.24.3    25
        127.0.0.0        255.0.0.0         在連結上         127.0.0.1   331
        127.0.0.1  255.255.255.255         在連結上         127.0.0.1   331
    140.113.235.0    255.255.255.0         在連結上  140.113.235.153   101
              ...
===========================================================================
```

Linux Network Tools

- Common network tools
  - ping
  - traceroute
  - mtr
  - nslookup

Ping

- Ping example
  - mostly same as windows
  - one major difference is Linux ping by default keep sending packet until stopped (Ctrl + C)

```
$ ping nctu.edu.tw
PING nctu.edu.tw (140.113.2.73) 56(84) bytes of data.
64 bytes from gw1-mail.nctu.edu.tw (140.113.2.73): icmp_seq=1 ttl=60 time=1.09 ms
64 bytes from gw1-mail.nctu.edu.tw (140.113.2.73): icmp_seq=2 ttl=60 time=1.01 ms
64 bytes from gw1-mail.nctu.edu.tw (140.113.2.73): icmp_seq=3 ttl=60 time=1.02 ms
```

- You can use **man ping** to check the manual

Traceroute

- Traceroute example

```
$ traceroute nctu.edu.tw
traceroute to nctu.edu.tw (140.113.2.73), 30 hops max, 60 byte packets
 1  Cisco-2960-Pri-EC347.cs.nctu.edu.tw (140.113.235.252)  0.663 ms  0.760 ms  0.866 ms
 2  140.113.3.174 (140.113.3.174)  0.337 ms  0.332 ms  0.305 ms
 3  140.113.3.178 (140.113.3.178)  1.128 ms  1.165 ms  1.163 ms
 4  not-a-legal-address (140.113.0.82)  1.119 ms  1.113 ms  1.103 ms
 5  * * *
 6  * * *
 7  * * *
```

Attempt sending 3 times for each hop. 1 round-trip time for each attempt.

No response within exceeded time. Default 5 seconds

MTR

- Combines the functionality of the **traceroute** and **ping** programs
- MTR example

```
$ mtr nctu.edu.tw
```

```
                             My traceroute  [v0.92]
bsd1.cs.nctu.edu.tw (140.113.235.131)                2019-08-28T22:43:59+0800
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                   Packets               Pings
 Host                            Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. Cisco-2960-Pri-EC347.cs.nctu.ed  0.0%     3    0.8   0.8   0.7   0.9   0.1
 2. 140.113.3.174                    0.0%     3    0.5   0.5   0.5   0.5   0.0
 3. 140.113.3.178                    0.0%     3    1.1   1.1   1.0   1.2   0.1
 4. not-a-legal-address              0.0%     3    1.6   1.4   1.3   1.6   0.2
 5. gw1-mail.nctu.edu.tw             0.0%     3    1.5   1.3   1.2   1.5   0.2
```

Nslookup

- Name server lookup
- Finds information about a named domain.
- nslookup example

```
$ nslookup nctu.edu.tw
Server:         10.1.1.1
Address:        10.1.1.1#53

Non-authoritative answer:
Name:   nctu.edu.tw
Address: 140.113.2.73
```

name server’s name and ip

Nslookup

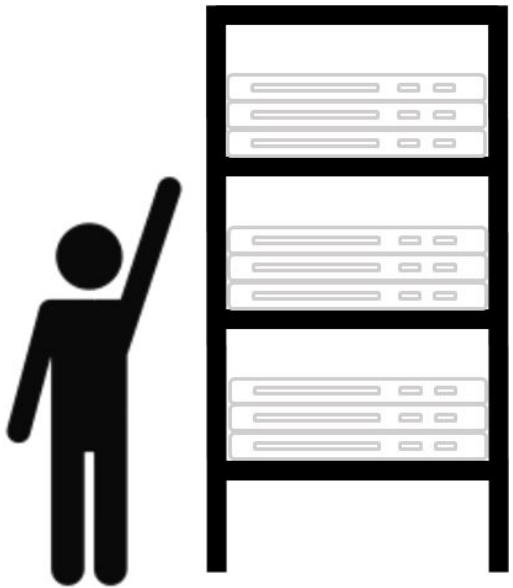
- Can also specify which name server to look up
- nslookup example

```
$ nslookup apache.com 8.8.8.8
Server:   dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:     apache.com
Address:  67.227.199.17
```

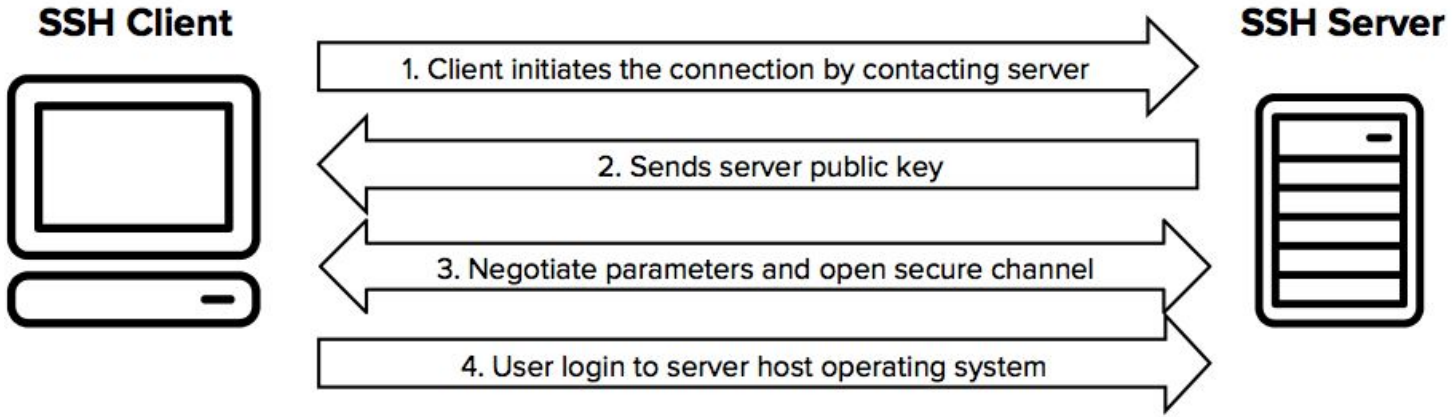
Specify name server's name/ip

How Can I Reach My Machine?



Secure Shell (SSH)

- A method for secure remote login from one computer to another.
  - Provides several alternative options for strong authentication.
  - Protects the communications security and integrity with strong encryption

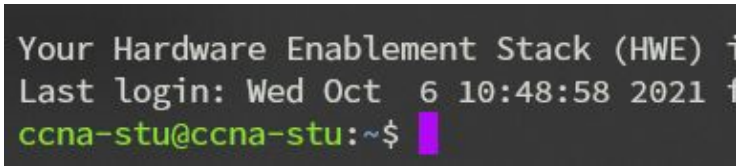


https://www.ssh.com/

SSH Server (Ubuntu)

- Enter terminal

`ctrl` + `alt` + `t`



- Update package info

```
$ sudo apt update
```

- Install SSH server

```
$ sudo apt install openssh-server
```

※ What is sudo ?

Wireshark

- A free and open-source packet analyzer
- Environment
  - Windows
  - Linux
  - MacOS
  - …
- Live data can be read from
  - Ethernet
  - Bluetooth
  - USB
  - ...

Wireshark (Ubuntu)

- Enter terminal

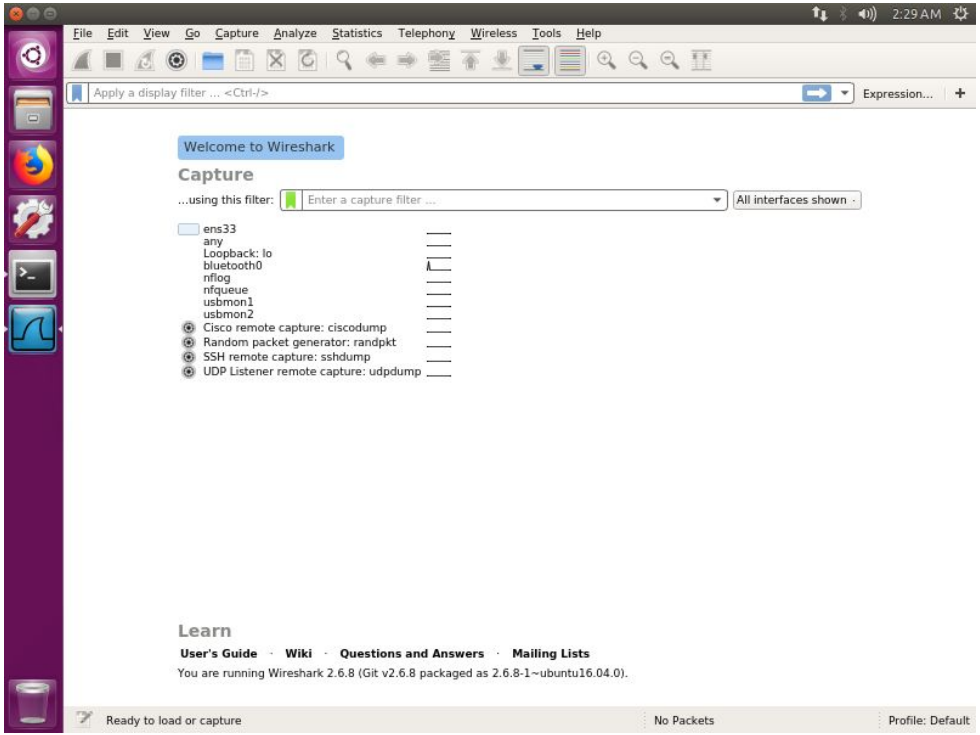
`ctrl` + `alt` + `t`

- Install Wireshark

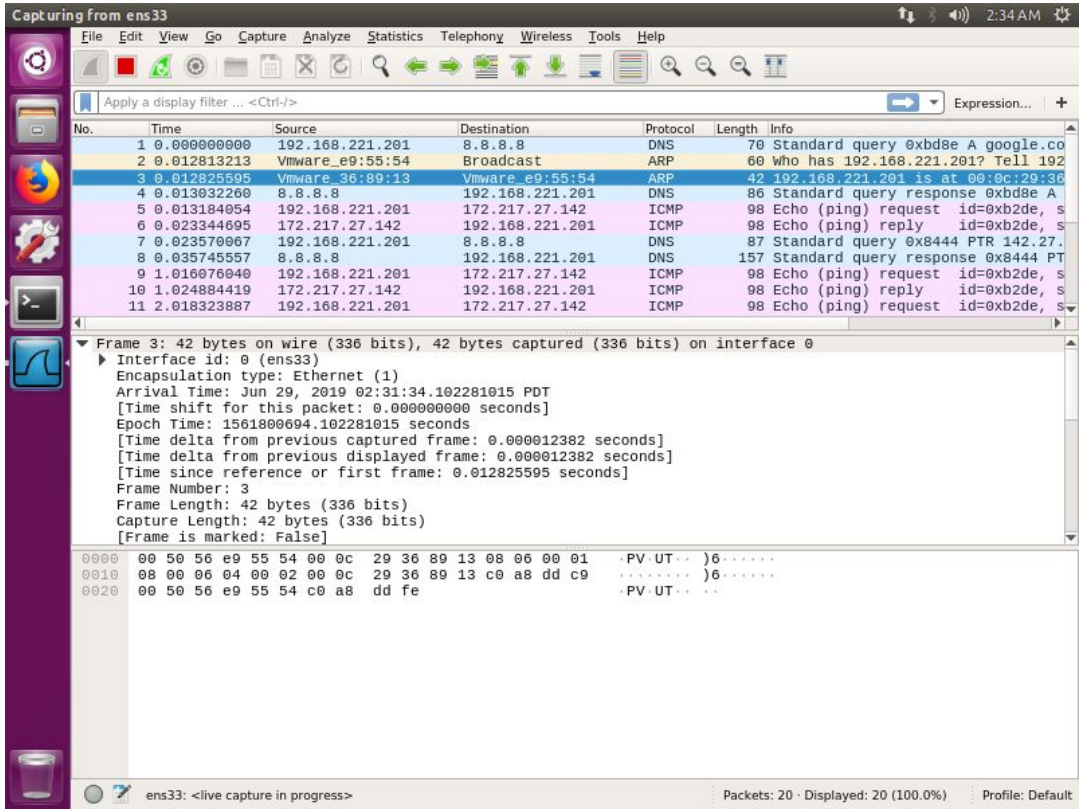
```
$ sudo apt install wireshark
```

- Execute Wireshark

```
$ sudo wireshark
```



Wireshark



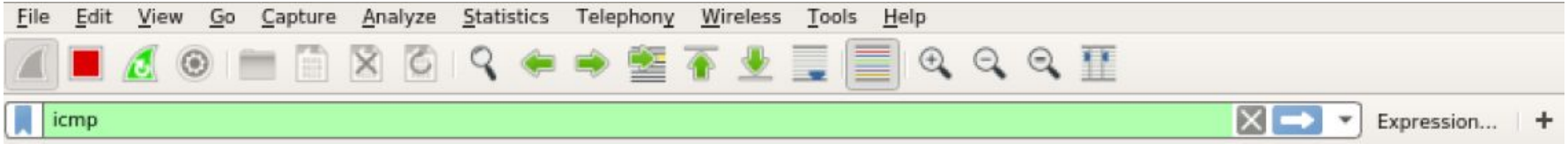
Packet List

Packet Details

Packet Bytes

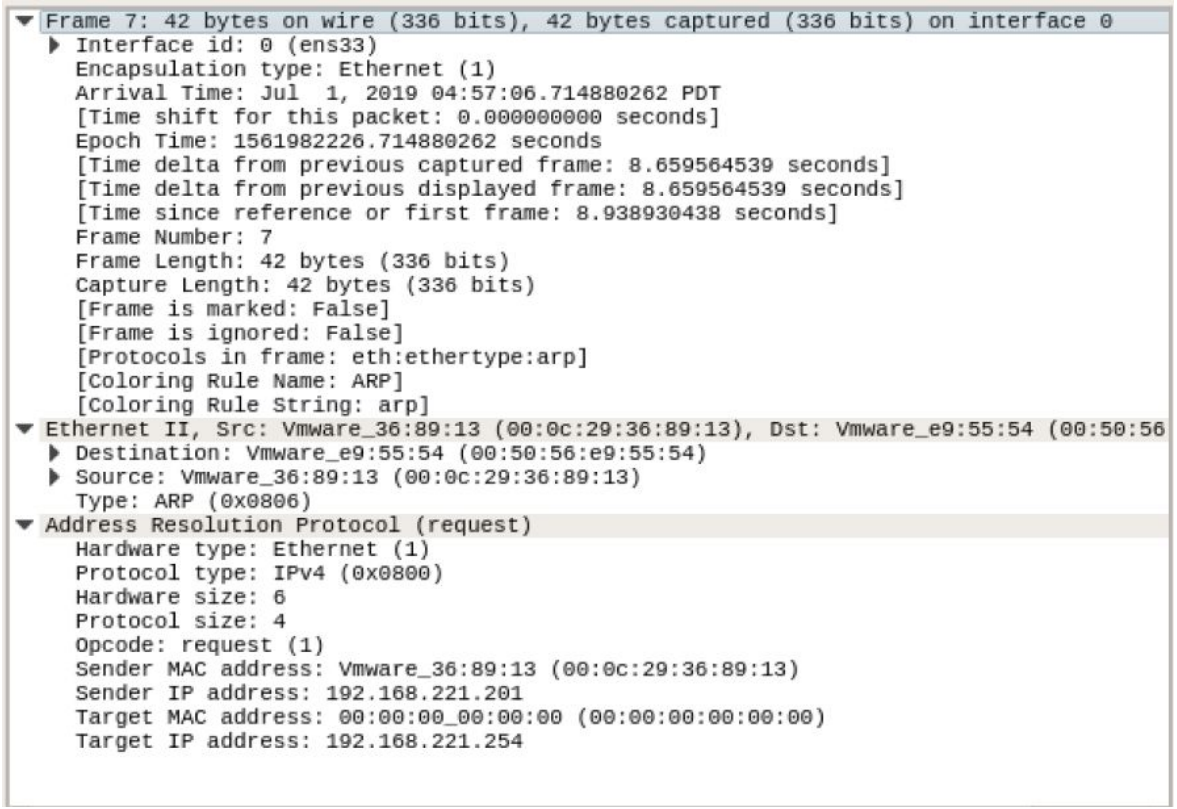
Wireshark

- Press Enter after filling the blank



- Turn red if the condition is wrong

Wireshark Example (ARP)



Wireshark Example (ARP)

- When an Ethernet frame is sent on LAN from one host to another
  - It is the 48-bit Ethernet address that determines for which interface the frame is destined

	Bits 0–7	Bits 8–15	Bits 16–23	Bits 24–31
0	Hardware address type (HTYPE)		Network protocol type (PTYPE)	
32	Length of hardware address (HLEN)	Length of protocol address (PLEN)	Operation	
64	Senders' MAC address			
96				
112	Senders' IP address			
144	Recipients' MAC address			
176				
192	Recipients' IP address			

```
▼ Frame 7: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
  ▶ Interface id: 0 (ens33)
    Encapsulation type: Ethernet (1)
    Arrival Time: Jul  1, 2019 04:57:06.714880262 PDT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1561982226.714880262 seconds
    [Time delta from previous captured frame: 8.659564539 seconds]
    [Time delta from previous displayed frame: 8.659564539 seconds]
    [Time since reference or first frame: 8.938930438 seconds]
    Frame Number: 7
    Frame Length: 42 bytes (336 bits)
    Capture Length: 42 bytes (336 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:arp]
    [Coloring Rule Name: ARP]
    [Coloring Rule String: arp]
▼ Ethernet II, Src: Vmware_36:89:13 (00:0c:29:36:89:13), Dst: Vmware_e9:55:54 (00:50:56
  ▶ Destination: Vmware_e9:55:54 (00:50:56:e9:55:54)
  ▶ Source: Vmware_36:89:13 (00:0c:29:36:89:13)
    Type: ARP (0x0806)
▼ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Vmware_36:89:13 (00:0c:29:36:89:13)
    Sender IP address: 192.168.221.201
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.221.254
```

Appendix. Add virtual machine

Add virtual machine(1/8)

— Preparation

- First install VirtualBox For your computer
- You may also need an OS image for your virtual machine
  - You can get Ubuntu desktop image here
- You may need enable virtualization for your CPU in BIOS first
  - Instruction for Win11

Add virtual machine (2/8)

— Create a new VM

- press New to start adding a VM

Add virtual machine (3/8)

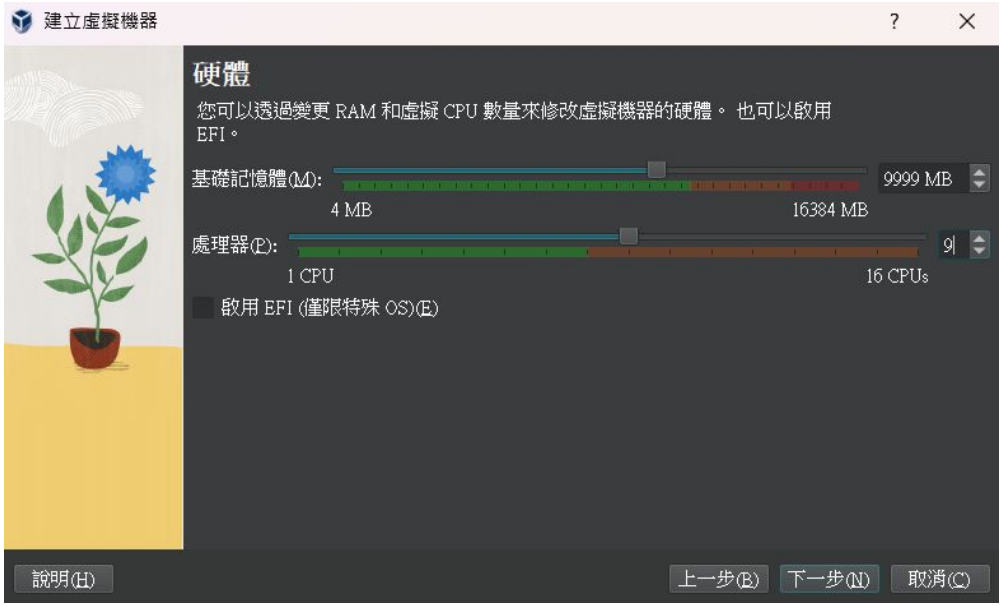
— Configure your VM setting

- Choose your VM name, location and OS image.
  - suggest creating a folder to collect VMs
- For first time installation, let’s skip unattended installation

Add virtual machine (4/8)

— Configure RAM and CPU

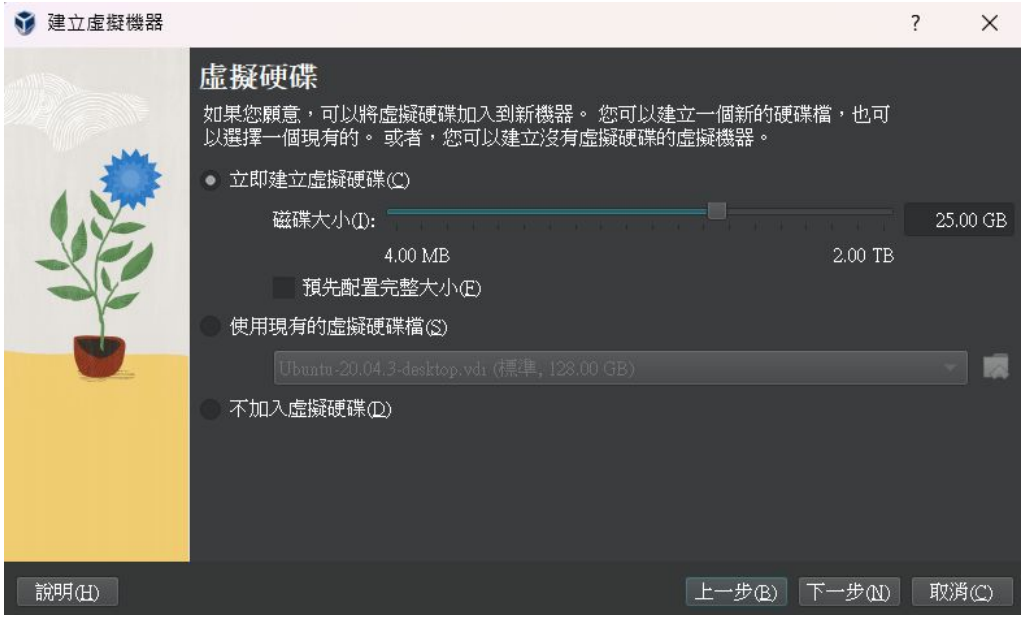
- Recommend at least 8GB RAM and 4 CPUs



Add virtual machine (5/8)

— Configure disk

- Create or import your virtual disk.
- By default, the disk will scale dynamically.
- Recommend: at least 25GB.
  - you can add other disk later.



Add virtual machine (6/8)

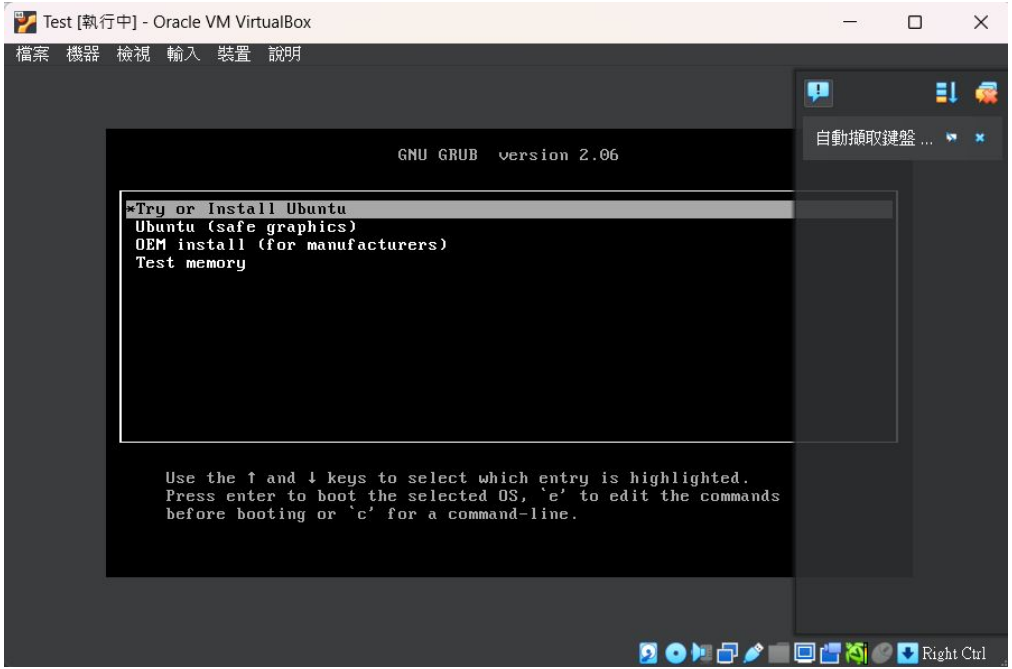
— Launch VM

- Press start to launch

Add virtual machine (7/8)

— Install Ubuntu

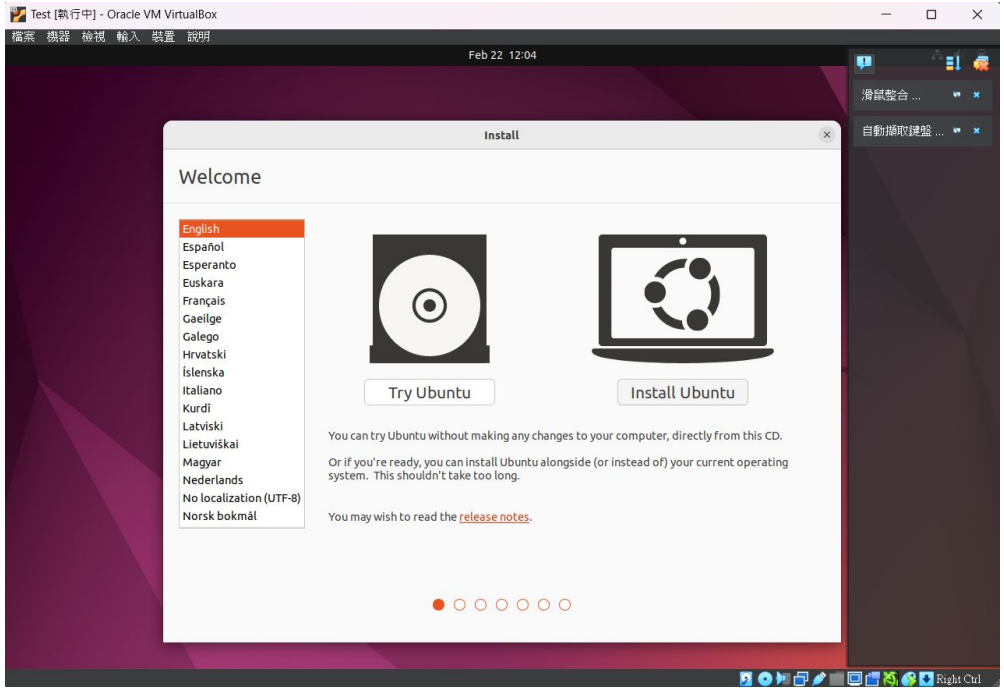
- Press Enter to start install Ubuntu
  - if your mouse stucked in your virtual box window, press Right Contorl.



Add virtual machine (8/8)

— Install Ubuntu

- Click Install Ubuntu
- simply follow the instructions to complete the installation.



Reference

- https://ubuntu.com/tutorials/how-to-run-ubuntu-desktop-on-a-virtual-machine-using-virtualbox#1-overview