

Contents

I. Remote 개발 환경 구축

- I. ssh
- II. sftp
- III. scp
- IV. VNC



I . SSH

I . SSH란?

- ssh란 무엇인가요?
 - Secure Shell 의 줄임말
 - 원격(remote) 호스트에 접속하기 위해 사용되는 보안 프로토콜
- 왜 이것을 사용하나요?
 - 기존에는 Telnet이라는 방식의 프로그램을 통해 통신 했었음
 - 당연히 암호화를 지원하지 않았고, 보안상 취약점이 항상 존재함
(서로간의 통신 데이터를 모두 알아낼 수 있음)
 - ssh 는 송신된 데이터가 네트워크에서 탈취되어도 내용을 해독할 수 없게 암호화 함
- 따라서 현재 원격 접속 보안을 위한 필수적인 요소로 사용되고 있음!

I . SSH 접속 방법

- Password 입력

- ssh를 이용해 접속을 시도하면, 사용자 ID와 Password를 입력해서 접속을 연결 시키는 방식
- 자주 접속 하는 특정 서버나 환경의 경우, 매번 입력하는 것이 불편하고, 관리 측면에서도 좋지 않음!
- 보안에 상대적으로 취약함

- Key 방식

- 접속을 하기 위한 상호간에 Key를 이용하여 서로를 인증 한 후 접속을 연결 시키는 방식
- 대칭키와 비 대칭키 2가지 방식이 존재

- Port

- 일반적으로 22번 포트를 사용 (default, 변경 가능)

I. SSH 접속하기

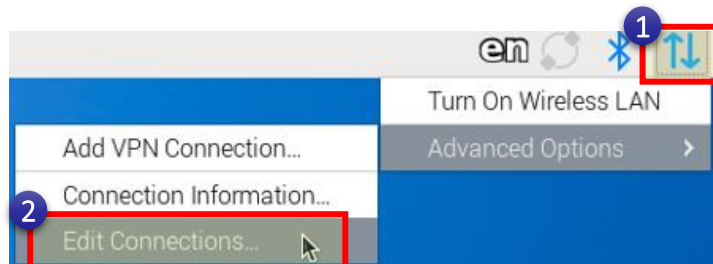
Network IP 를 manual 하게 할당하기 (Wired Network 사용)

1. 라즈베리파이 무선 네트워크를 Disable

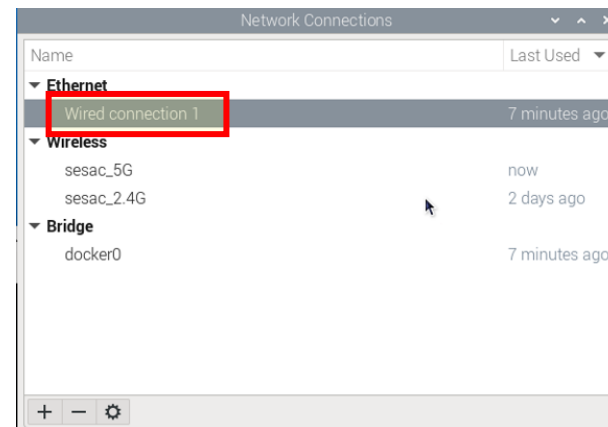


2. 라즈베리파이에 LAN cable 을 연결

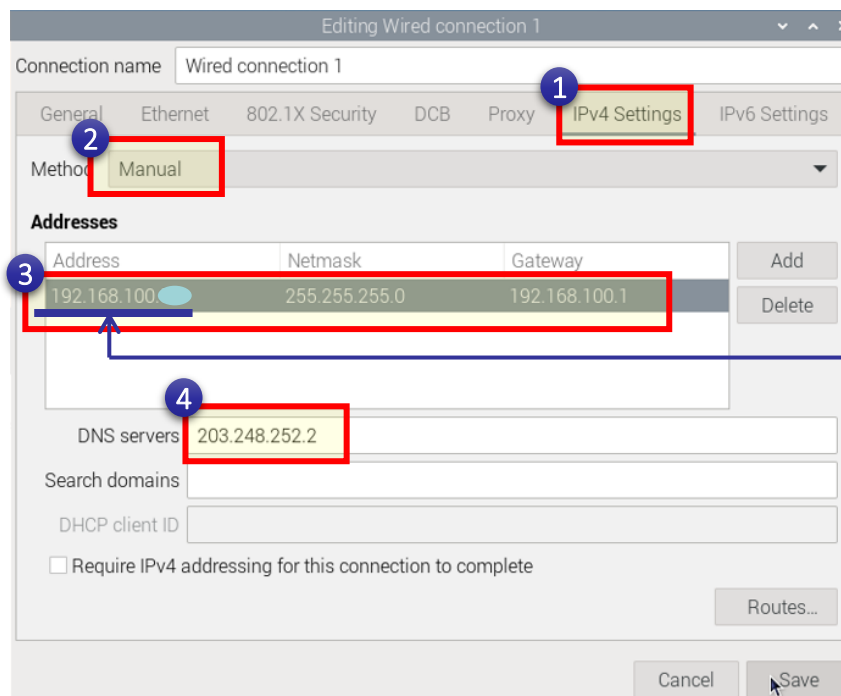
3. Network 설정 변경 메뉴 열기



4. Wired connection 1 선택해 설정메뉴 열기



5. 각자 자신에게 할당된 IP address 를 입력 후 저장



라즈베리파이용 IP address 를 각 자리마다 알려줄 예정임. 반드시 자신에게 할당된 IP address 를 입력!!

<라즈베리파이용 IP address 범위>
192.168.100.51 ~ 192.168.100.100

I . SSH 접속하기

라즈베리파이 Configuration - 원격 접속하기 (SSH: Secure Shell)

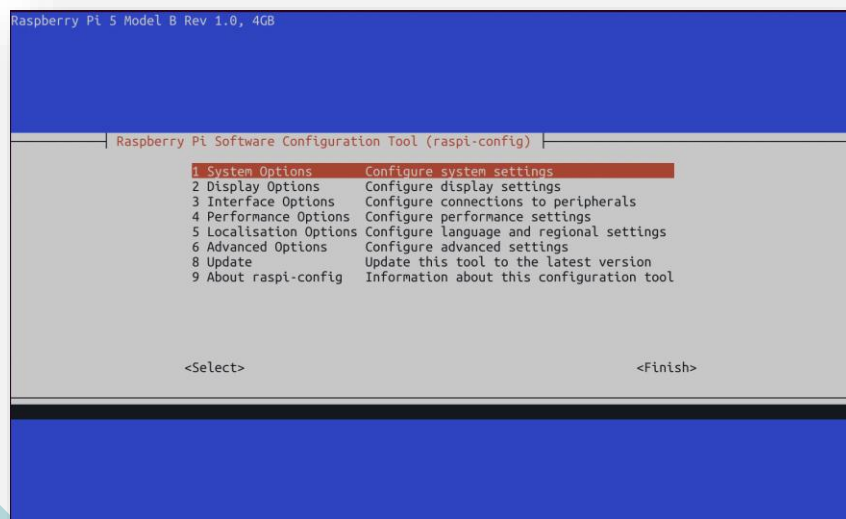
Note: SSH는 암호화된 네트워크 프로토콜로, 원격 시스템에 안전하게 로그인하고 명령어를 실행할 수 있고 이를 통해 사용자는 인터넷과 같은 보안되지 않은 네트워크를 통해서도 안전하게 서버를 관리

1. 터미널 창을 열어서 아래 명령어를 실행하여 설정창 열기

```
$ sudo raspi-config
```

2. 아래 경로로 들어가서 SSH 를 활성화

3. Interface Options >> SSH >> Yes



3. 터미널 창에서 `ifconfig` or `ip a` 명령어를 실행하여 라즈베리 파이의 IP address 확인

```
intel@raspberrypi:~$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 2c:cf:67:6e:f6:83 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 106

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 102 bytes 8956 (8.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 102 bytes 8956 (8.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.250.74 netmask 255.255.255.0 broadcast 192.168.250.255
    inet6 2001::23b:36:f55:47c6:33fb prefixlen 64 scopeid 0x0<global>
    inet6 fe80::5131:8038:bac7:2202 prefixlen 64 scopeid 0x20<link>
    ether 2c:cf:67:6e:f6:84 txqueuelen 1000 (Ethernet)
    RX packets 11645 bytes 14175969 (13.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 14307 bytes 14337436 (13.6 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

4. 접속할 Ubuntu PC 에서 터미널 창을 열고 아래와 같이 SSH 로 접속

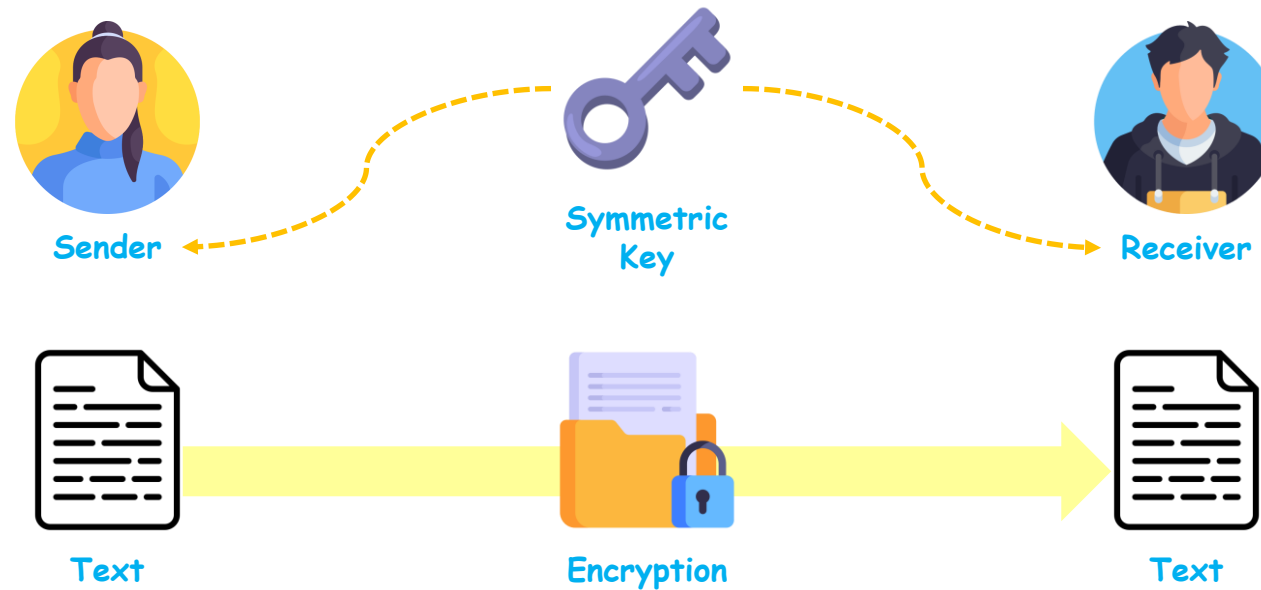
```
$ ssh <라즈베리파이의 user id>@<라즈베리파이의 IP address>
```

```
max@maxim-mobl:~$ ssh intel@192.168.250.74
intel@192.168.250.74's password:
Warning: No xauth data; using fake authentication data for X11 forwarding.
Linux raspberrypi 6.12.20+rpt-rpi-2712 #1 SMP PREEMPT Debian 1:6.12.20-1+rpt1-bpo12+1 (2025-03-19) aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Apr 21 16:45:49 2025 from 192.168.250.247
intel@raspberrypi:~$ |
```

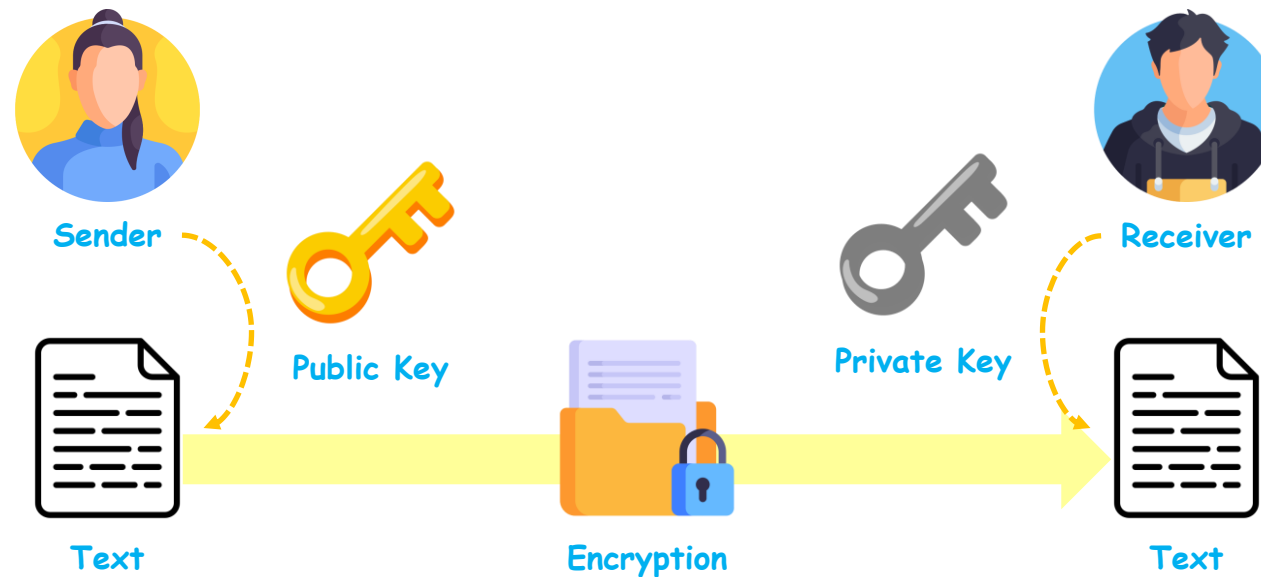
I . SSH Key



■ 대칭키 (Symmetric Key)

- 정보를 보내는 곳(Sender)와 정보를 받는 곳(Receiver)가 **같은 Key**를 가지고 있다고 가정한다
- Sender는 Key를 이용해 정보를 암호화 한 후 정보를 전송한다
- Receiver는 Key를 이용해 전송 받은 정보를 복호화 한다

I . SSH Key (cont,)



■ 비 대칭키 (Key Pair - Asymmetric Key)

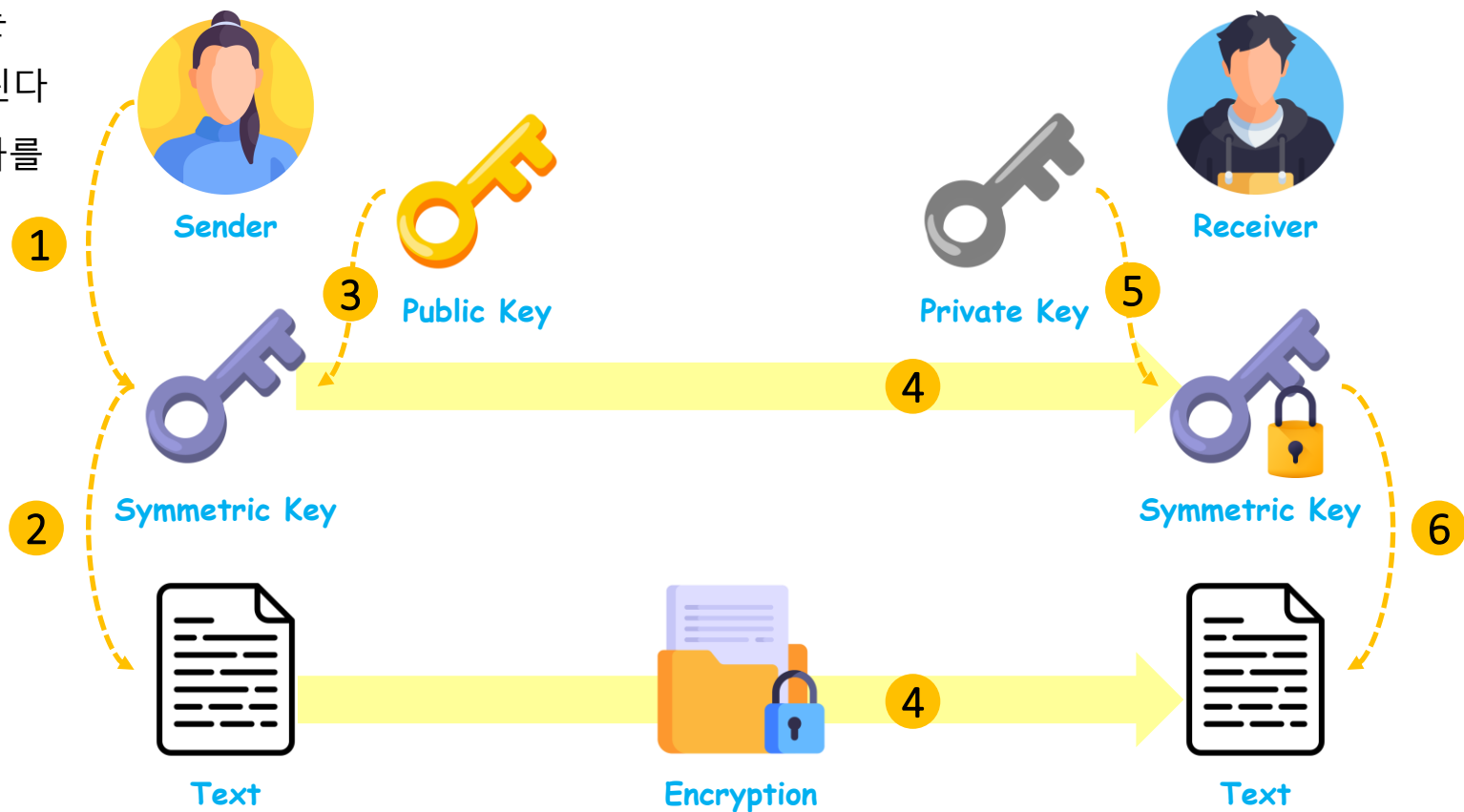
- 2개의 Key를 활용하여 문서를 암호/복호 하는 기법
- Receiver는 Public Key와 그에 맞는 Private Key를 제작한 후, Public Key를 공개 한다
- Sender는 공개된 Public Key를 이용해 정보를 암호화 한 후 Receiver에게 전송 한다
- Receiver는 Private Key를 이용해 정보를 복호화 하기 때문에 Public Key가 잘못 되어 있다면 복호에 실패 한다

I . SSH Key (cont,)

■ ssh Key 사용 방식

- 대칭키는 키 보관이 어렵고, 비 대칭키는 매번 키 확인 작업으로 시간이 오래 걸린다
- 따라서, 두 방식을 혼합하여 처음 사용자를 증명할 때만 비 대칭키를 쓰고, 서로 확인이 되면 대칭키를 이용해 통신

1. 대칭키 생성
2. 대칭키를 사용하여 정보 암호화
3. 대칭키를 공개키로 암호화
4. 두 암호 정보를 전송
5. 개인키로 대칭키 복호화
6. 대칭키로 정보 복호화



I . SSH Key Hands-on



- Hands-on
 - 라즈베리파이에서 ssh 패키지 관련 설치 및 ssh 통신을 위한 환경 설정 (openssh-server)
 - Host에서 비대칭키 생성 해보기(ssh-keygen)
 - Public Key를 복사해서 라즈베리파이에 파일로 생성한 후 Key를 시스템에 등록(ssh-add)
 - ssh명령어로 host(Ubuntu Desktop)에서 라즈베리파이로 접속해보기 (ssh <user id>@<raspberry pi IP Address>)

I . SSH Key Hands-on



I . SSH Hands-on



■ Hands-on

- Ubuntu Host 에서 ssh 를 사용해서 raspberrypi 의 ~/Bookshelf 경로에 어떤 파일 리스트 들이 있는지 확인하기. 명령어 한줄로만 실행해야 하며 ssh 로 raspberrypi 에 접속 후 ls -la ~/Bookshelf 명령어를 입력하는 건 안됨

→

- 현재 raspberrypi 의 ssh 의 default port 번호는 22 이다. 이것 2222 로 변경해 보자.

→

→

쉬어가기 – 데몬이 뭘까? (daemon)

■ 개념

- 어원: 데몬(daemon)이라는 단어는 고대 그리스어에서 유래된 것으로, 신화, 종교, 철학에 등장하는 인간과 신들 사이에 위치하거나 죽은 영웅의 영혼 등을 나타내는 "다이몬(δαίμων)"에서 비롯됨. 보이지 않는 수호신 정도...
- 컴퓨터 세계에서 데몬(daemon)은 백그라운드에서 지속적으로 실행되는 프로그램(프로세스)을 의미함. 이러한 프로그램은 유령처럼 눈에 보이지 않게, 하지만 항상 작동하며 시스템을 돕는 역할

■ systemd 는 뭘까?

- system daemon의 줄임말로 시스템 부팅 후 처음 올라오는 최상위 관리자 프로세스 (일종의 슈퍼 데몬)

```
pi@raspberrypi:/etc/systemd/system/multi-user.target.wants $ ls -la
total 8
drwxr-xr-x  2 root root 4096 May 23 07:29 .
drwxr-xr-x 18 root root 4096 May 23 06:37 ..
lrwxrwxrwx  1 root root  40 May 13 09:07 avahi-daemon.service -> /lib/systemd/system/avahi-daemon.service
lrwxrwxrwx  1 root root  41 May 13 09:07 console-setup.service -> /lib/systemd/system/console-setup.service
lrwxrwxrwx  1 root root  32 May 13 09:05 cron.service -> /lib/systemd/system/cron.service
lrwxrwxrwx  1 root root  40 May 13 09:10 cups-browsed.service -> /lib/systemd/system/cups-browsed.service
lrwxrwxrwx  1 root root  29 May 13 09:10 cups.path -> /lib/systemd/system/cups.path
lrwxrwxrwx  1 root root  32 May 13 09:10 cups.service -> /lib/systemd/system/cups.service
lrwxrwxrwx  1 root root  42 May 13 09:07 dphys-swapfile.service -> /lib/systemd/system/dphys-swapfile.service
lrwxrwxrwx  1 root root  40 May 13 09:04 e2scrub_reap.service -> /lib/systemd/system/e2scrub_reap.service
lrwxrwxrwx  1 root root  39 May 13 09:09 glamor-test.service -> /lib/systemd/system/glamor-test.service
lrwxrwxrwx  1 root root  40 May 13 09:07 ModemManager.service -> /lib/systemd/system/ModemManager.service
lrwxrwxrwx  1 root root  42 May 13 09:07 NetworkManager.service -> /lib/systemd/system/NetworkManager.service
lrwxrwxrwx  1 root root  37 May 13 09:07 nfs-client.target -> /lib/systemd/system/nfs-client.target
lrwxrwxrwx  1 root root  36 May 13 09:05 remote-fs.target -> /lib/systemd/system/remote-fs.target
lrwxrwxrwx  1 root root  36 May 13 09:09 rpi-test.service -> /lib/systemd/system/rpi-test.service
lrwxrwxrwx  1 root root  45 May 13 09:06 rpi-eeprom-update.service -> /lib/systemd/system/rpi-eeprom-update.service
lrwxrwxrwx  1 root root  31 May 23 06:37 ssh.service -> /lib/systemd/system/ssh.service
lrwxrwxrwx  1 root root  37 May 13 09:07 sshswitch.service -> /lib/systemd/system/sshswitch.service
lrwxrwxrwx  1 root root  40 May 13 09:06 triggerhappy.service -> /lib/systemd/system/triggerhappy.service
lrwxrwxrwx  1 root root  34 May 23 07:29 wayvnc.service -> /lib/systemd/system/wayvnc.service
lrwxrwxrwx  1 root root  42 May 13 09:07 wpa_supplicant.service -> /lib/systemd/system/wpa_supplicant.service
```



II. SFTP

II. sftp

- sftp란 무엇인가?

- FTP(File Transfer Protocol)
- 파일을 전송하는 통신 규약, 보안이 되지 않는 단점
- **Secure FTP**를 활용하여 이를 해결
(ssh를 이용해서 정보를 안전하게 보호한다)

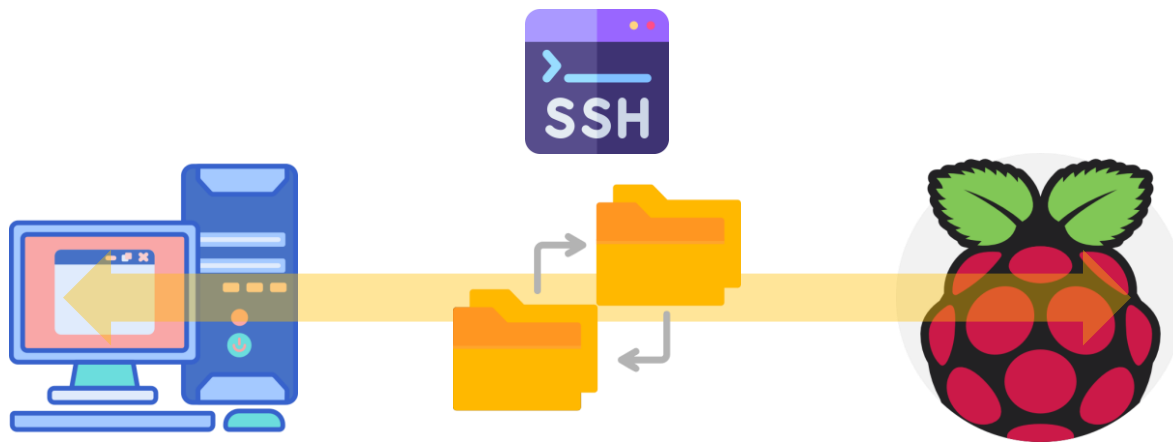
- Hands-on

- sftp를 이용해서 Host와 라즈베리파이 간에 파일을 전송 해보자!
 - `sftp <옵션> <라즈베리파이ID>@<라즈베리파이IP>`
- 접속 방법, get/put 명령어 사용법
- 폴더를 보내고 싶다면? (-r 옵션)
- 폴더를 묶어서 파일처럼 보내보기 (tar 실습)



III. SCP

III. scp



- scp란 무엇인가?
 - **Secure Copy**
 - 원격지에 파일 또는 디렉토리를 보내거나 가져올 때 사용하는 파일 전송 프로토콜
- 로컬(Ubuntu) -> 원격지(라즈베리파이)
 - `scp <옵션> <파일명> <원격지ID>@<원격지IP>:<저장경로>`
- 원격지(라즈베리파이) -> 로컬(Ubuntu)
 - `scp <옵션> <원격지ID>@<원격지IP>:<원본파일> <저장경로>`

III. scp vs sftp

	SCP	SFTP
기능	파일 전송만 허용하는 간단한 프로토콜	원격 파일 관리를 위한 광범위한 작업을 제공 원격으로 파일 제거, 중단된 전송 재개 등의 추가 기능을 가짐
플랫폼	Unix 기반	Windows, Mac, Linux, Unix 가능
속도	SFTP 보다 빠름	패킷의 암호화 및 일치 확인 해야 하므로 SCP 보다 느림
파일 전송 용량	4GB 이상 전송 불가	대용량 전송 가능
세션 유지	파일 전송 취소시, 세션을 취소 해야 됨	파일 전송 취소 시, 세션 유지 가능
전송 재개 가능	불가능	가능

VI. VNC

VI. VNC



- VNC란?
 - Virtual Network Computing
 - 컴퓨터 환경에서 다른 컴퓨터를 원격으로 제어 하는데 사용되는 **그래픽 데스크톱 공유 시스템**
 - Remote Frame Buffer(RFB) 프로토콜 사용
 - 원격 컴퓨터의 데스크톱에 접속 하여 마치 직접 사용하는 것처럼 조작 가능!

VI. VNC 설치 (in Target)

- 범용적인 OS에서 사용되고 있는 Real VNC를 설치해보자!
 - https://www.realvnc.com/en/?lai_vid=53l2VP5zbf2V&lai_sr=40-44&lai_sl=m
 - 원격 접속 SW를 제공하는 회사
 - VNC 프로토콜을 통해 화면을 원격으로 제어하는 서버(VNC 서버)와 클라이언트(VNC 뷰어)로 구성됨
- Raspberry Pi OS에서 기본 apt로 설치 가능
 - `$ sudo apt install -y realvnc-vnc-server realvnc-vnc-viewer`
- But, 현재 설치된 GNU/Linux 12(bookworm) 기반 Raspberry Pi OS에서는 지원 안됨!
 - 라즈베리파이 터미널 셸에서
`$ cat /etc/os-release`
 - <https://www.realvnc.com/en/connect/download/vnc/raspberrypi>



VI. VNC 설치 (in Target, cont)

1. Raspberry Pi 선택
2. amd64 선택(64bit 시스템)
3. Download 클릭!
4. 라즈베리 파이 터미널 쉘에서,

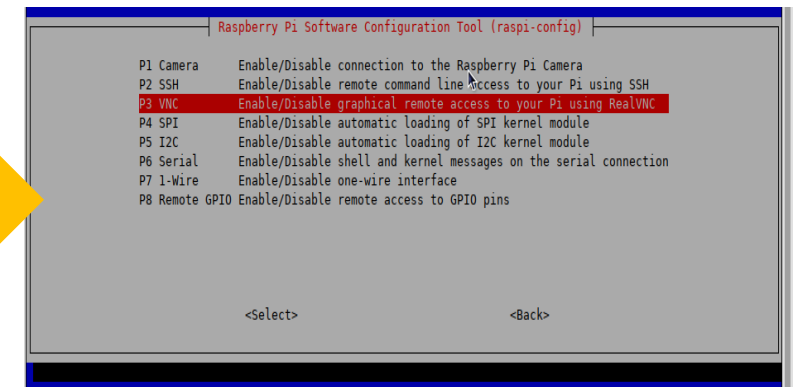
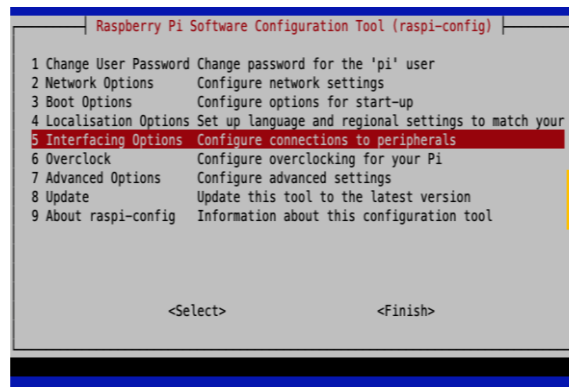
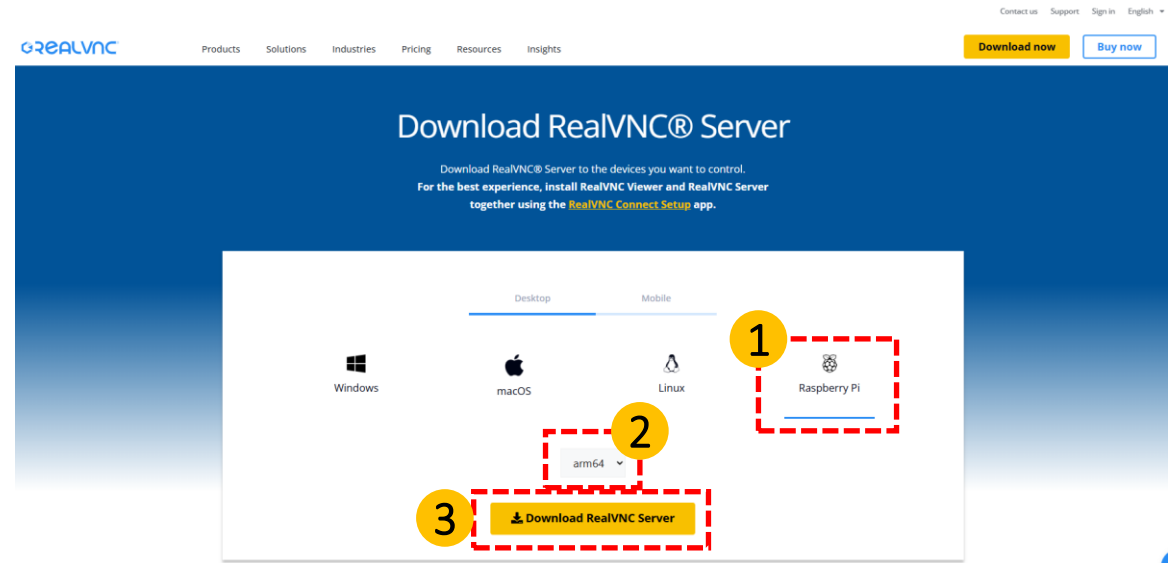
```
$ cd ~/Downloads
```

```
$ sudo apt install -y ./VNC-Server-xxx-Linux-ARM64.deb
```
5. Config에서 VNC 활성화

```
$ sudo raspi-config
```

```
Interfacing Options -> VNC -> Enable
```
6. VNC 서버 확인

```
$ sudo systemctl status wayvnc
```



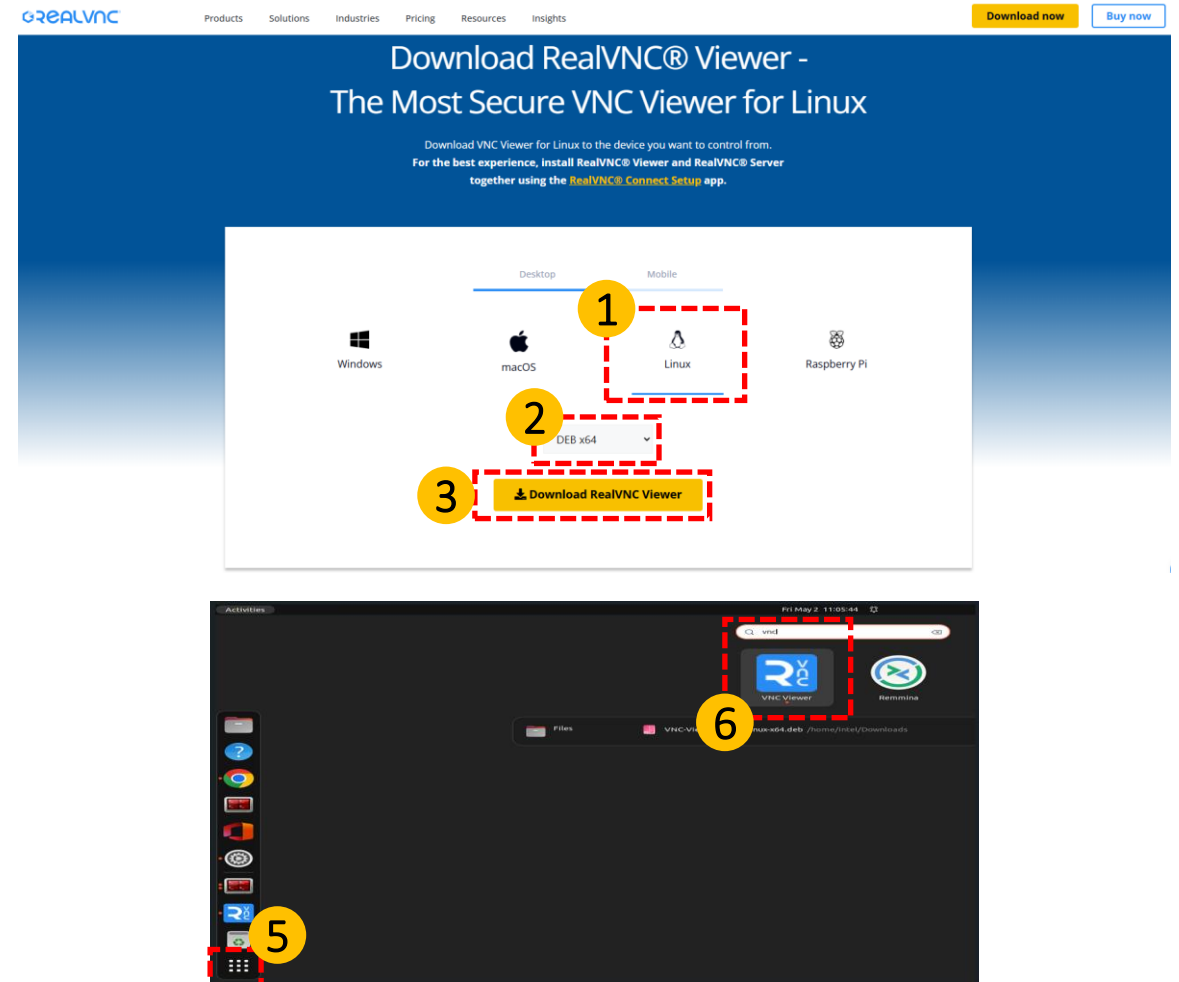
VI. VNC 설치 (in Host)

<https://www.realvnc.com/en/connect/download/viewer/linux>

1. Linux 선택
2. DEB x64 선택 (Ubuntu x86_64 기준)
3. Download 클릭!
4. Ubuntu 터미널 쉘에서

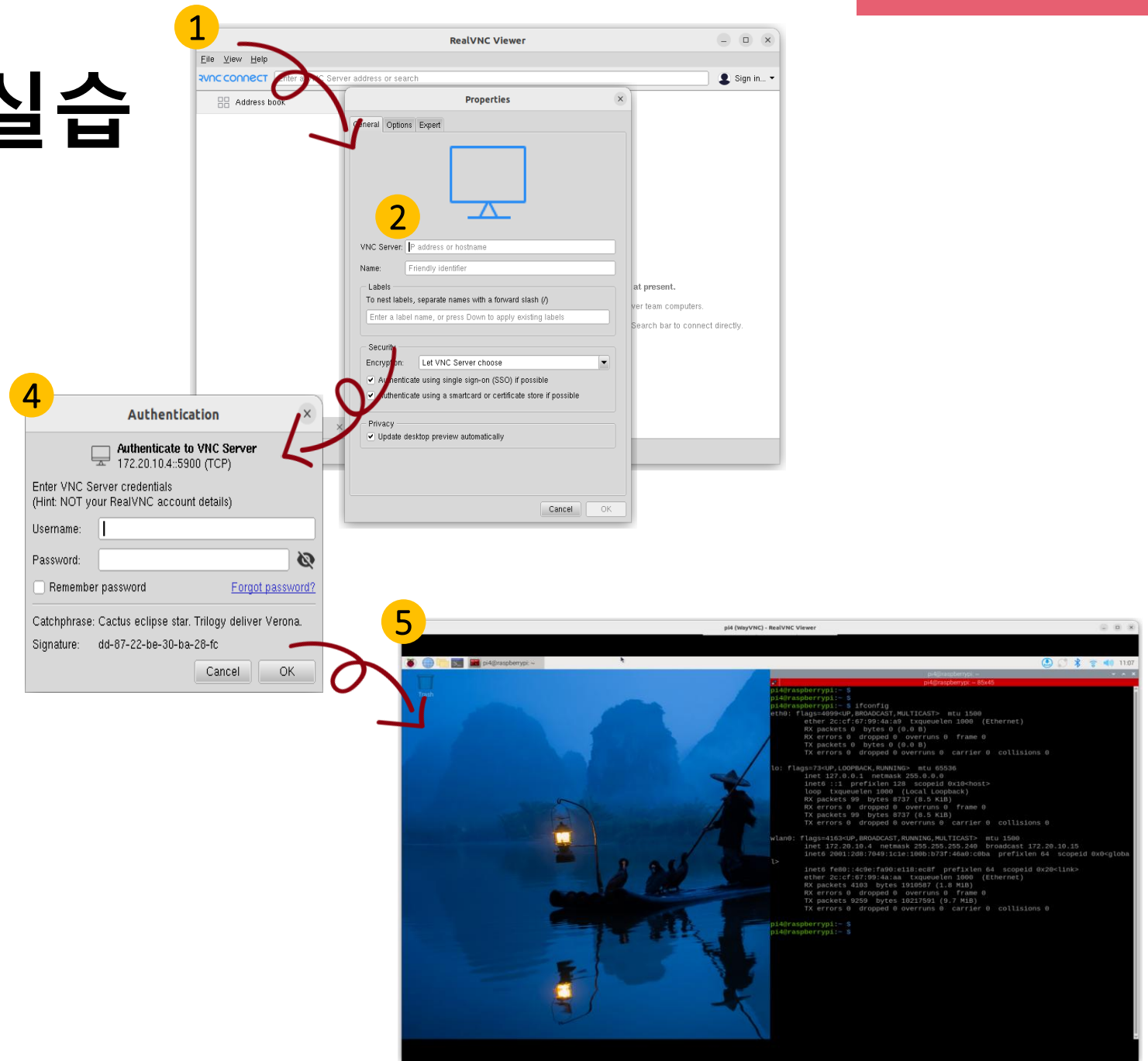
```
$ cd ~/Downloads
```

```
$ sudo apt install -y ./VNC-Viewer-xxx-Linux-x64.deb
```
5. Ubuntu 'Show Application' 메뉴 클릭
6. vnc 검색 후 vnc viewer 실행!



VI. VNC 연결 및 실습

1. File -> New connection
2. VNC Server에 Pi IP주소 입력
Name은 아무거나..
3. 화면에 저장된 서버 더블클릭!
4. Username은 사용자 ID
Password를 입력후 연결
5. Enjoy ☺





THANK YOU