

자주쓰지는 않지만
모르면 뺑치는
프로토콜

전체 목차

- Protocol
- SSH
- FTP
 - TFTP
 - SFTP
- SCP
- wget

Protocol

Protocol

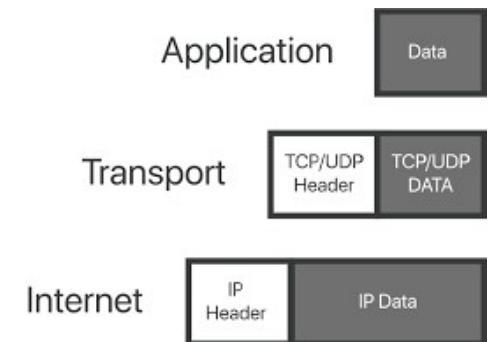
: 디바이스(컴퓨터)간 정보 교환을 위해 서로간에 약속된 통신 방식

- Protocol 예시

- 인터넷 : TCP / UDP / IP
- 웹 : HTTP / HTTPS
- 이메일 : SMTP
- IoT : MQTT

- Internet Protocol (IP, 인터넷 프로토콜)

- 네트워크에 연결되어 있을때 데이터를 전달하는 기술
- 라우팅을 통해 연결되고 있고
- IP주소를 사용해서 받는사람과 보내는 사람이 서로 데이터를 교환한다.
- 패킷구조, 캡슐화 등의 기술을 사용



SSH

SSH 프로토콜 개요

SSH 프로토콜

- 컴퓨터와 컴퓨터가 인터넷과같은 network 통해 서로 통신하는 프로토콜
- 보안측면에서 안전하게 통신하기 위해 사용
- 주로 다른 컴퓨터로 원격 접속시에 사용됨



SSH 를 왜 사용할까

SSH 를 왜 사용할까

- 데이터 전송
 - Github - 소스코드를 원격 저장소인 깃헙에 푸쉬할때 ssh활용해 파일 전송
- 원격제어
 - AWS 클라우드 서비스 : AWS인스턴스 서버에 ssh통해 접속하여 해당 머신에 명령
- 보안
 - SSH는 암호화된 안전한 채널을 구성한 뒤 정보를 교환
 - 중간에 정보가 노출되어도, 정보도 암호화 되어 있어 내용을 알 수 없다
 - 암호화에는 공개키/비공개키 방식 모두 사용 가능

SSH 실습

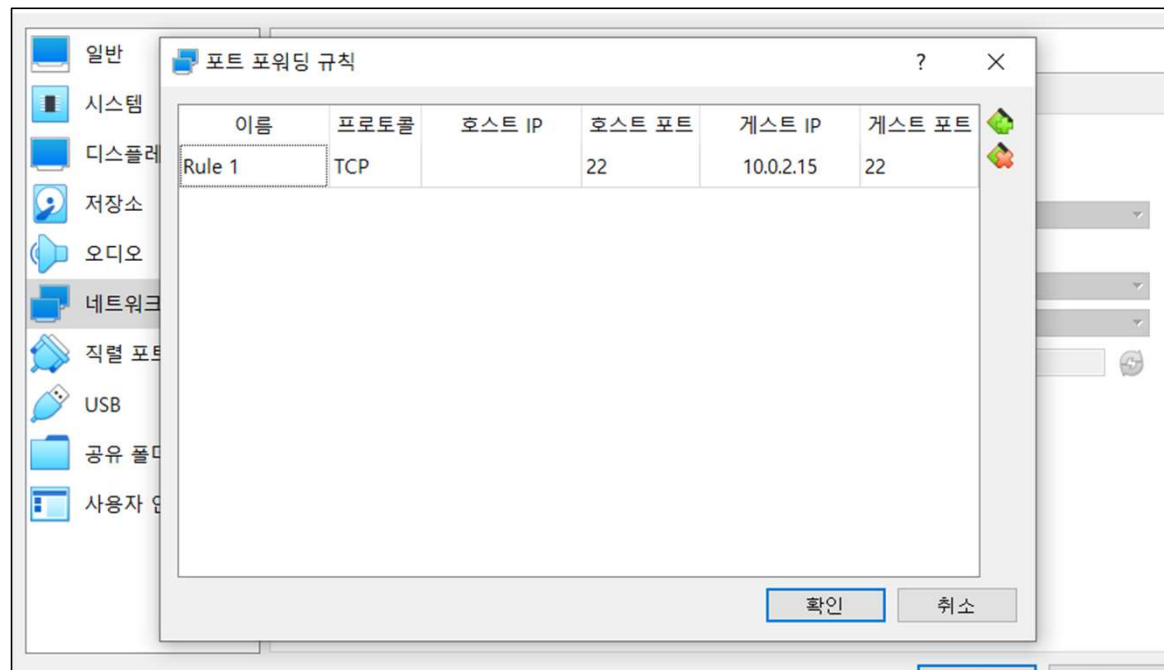
- SSH 실습을 위해서는 동작중인 OS 두개가 필요
 - 여기에서는 Linux를 대상으로(이하 게스트) Windows 에서 제어하는(이하 호스트) 구조로 실습을 진행
- Ubuntu GUEST : SSH server 설치필요, ip확인
 - \$ sudo apt install -y openssh-server
 - \$ ifconfig
- Windows HOST : mobaxterm 필요
 - 링크 : <https://mobaxterm.mobatek.net/>

```
ubuntu@ubuntu:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.35.131 netmask 255.255.255.0 broadcast 192.168.35.255
    inet6 fe80::dea6:32ff:fe55:1681 prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:55:16:81 txqueuelen 1000 (Ethernet)
    RX packets 3911 bytes 5403407 (5.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1669 bytes 127585 (127.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



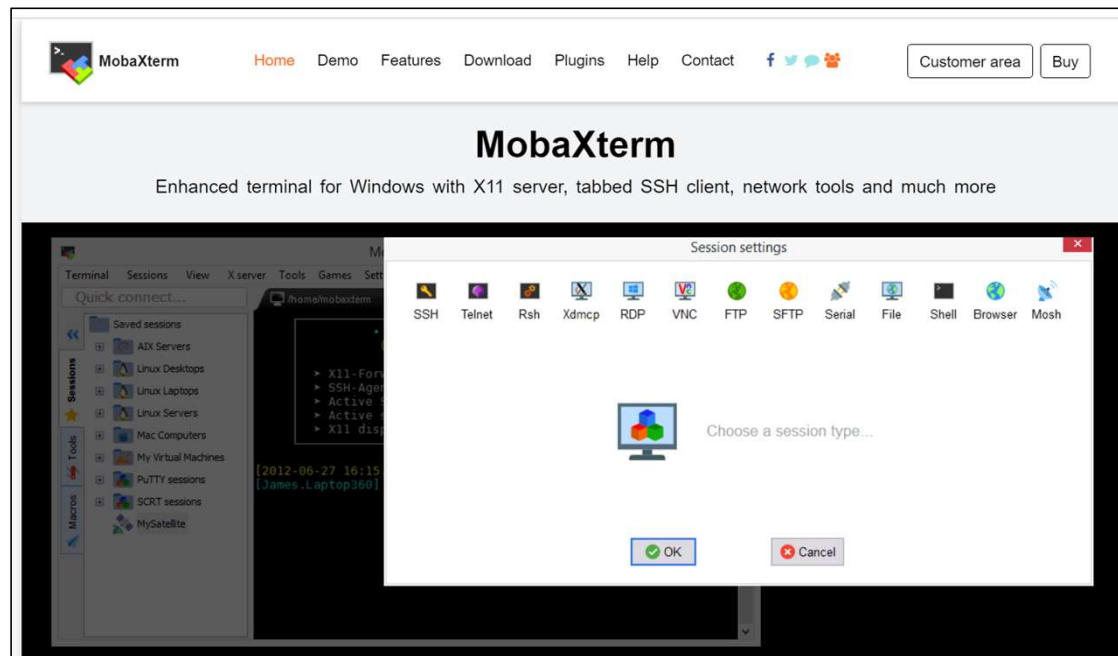
SSH 실습 – **virtual box** 포트포워딩

- 우분투 가상머신을 종료하고 virtual box 설정을 변경
- virtual box 설정 > 네트워크 > 고급 > 포트포워딩 규칙
- GUEST IP 항목을 우분투 ip주소로 지정한다



SSH 실습 – virtual box 포트포워딩

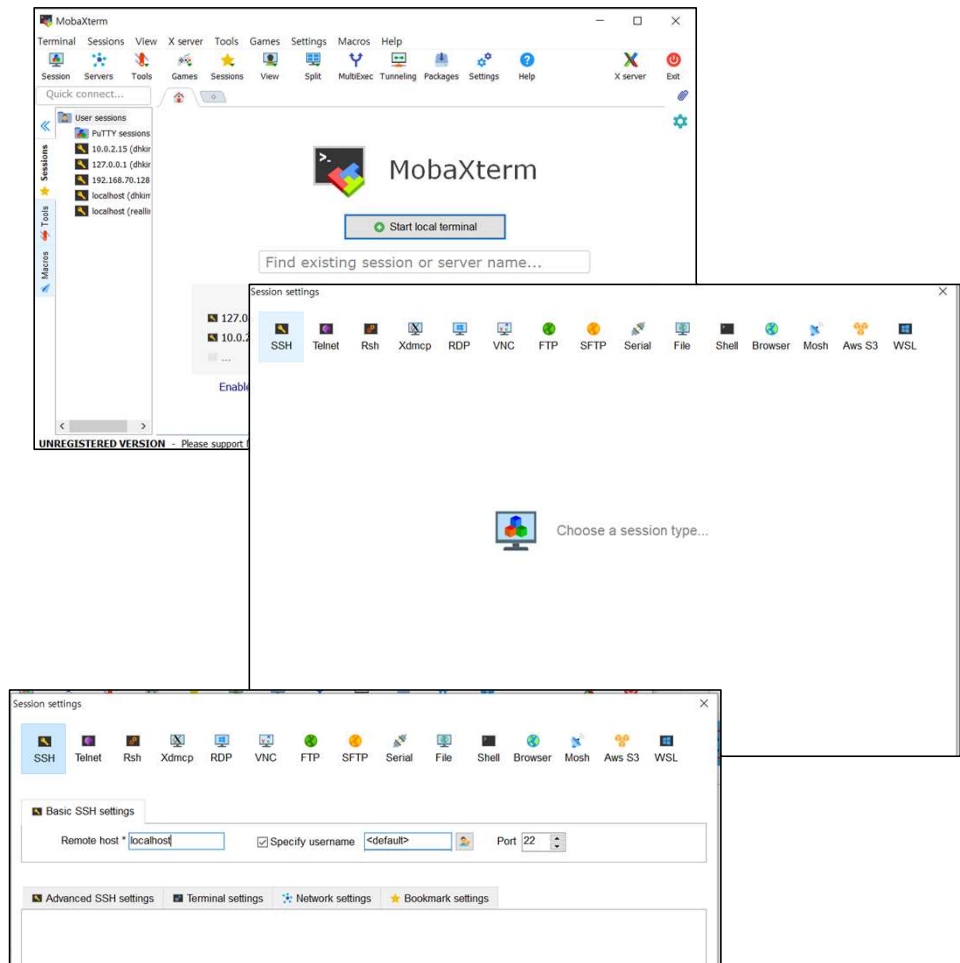
- 윈도우즈 host 에서 우분투로 원격 접속을 위해서 프로그램 설치
- 여기서는 mobaxterm 프로그램을 사용



링크 : <https://mobaxterm.mobatek.net/>

SSH 실습 – virtual box 포트포워딩

- MobaXterm 실행
 - 프로그램 좌상단의 session 버튼 클릭
- session 윈도우에서 접속
 - 좌측 상단의 SSH 글자 클릭
- IP와 계정명 입력
 - Remote host : localhost
 - username : 리눅스 계정



SSH 실습 – virtual box 포트포워딩

- **접속 성공!**

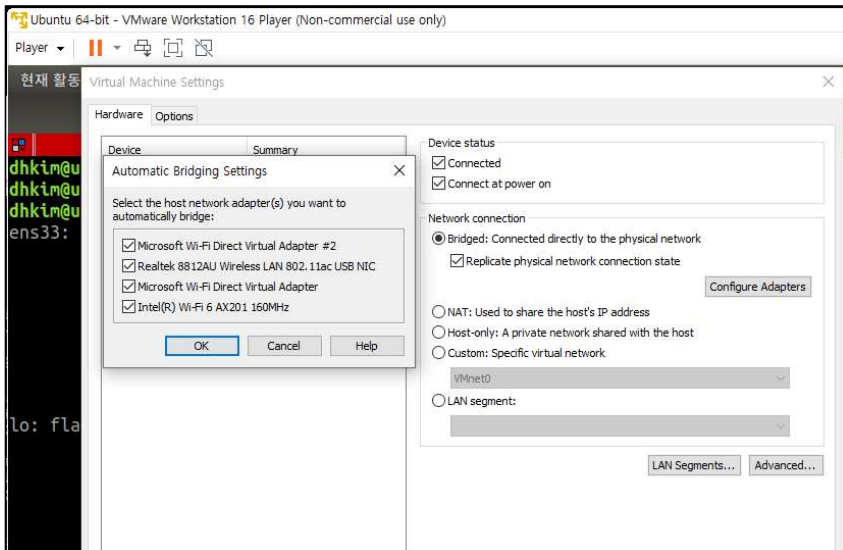
The screenshot shows a terminal window titled "192.168.35.131 (ubuntu)". The top menu bar includes Terminal, Sessions, View, X server, Tools, Games, Settings, Macros, and Help. Below the menu is a toolbar with icons for Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. A sidebar on the left contains a "Quick connect..." field and a list of sessions: "Sessions" (with a star icon), "Tools" (with a wrench icon), "Macros" (with a document icon), and "Sftp" (with a folder icon). The main terminal area displays the following commands and output:

```
ubuntu@ubuntu:~$ uname -a  
Linux ubuntu 5.4.0-1026-raspi #29-18.04.1-Ubuntu SMP PREEMPT Tue Dec 15 17:58:09 UT C 2020 aarch64 aarch64 aarch64 GNU/Linux  
ubuntu@ubuntu:~$ screenfetch  
          ./.+o+-      ubuntu@ubuntu  
        yyyyyy- -yyyyyy+    OS: Ubuntu 18.04 bionic  
       ://+//////-yyyoyyo   Kernel: aarch64 Linux 5.4.0-1026-raspi  
     .++ .:/+++++/-.-sss`    Uptime: 2h 21m  
   .:++o: /+++++++/:--:/-    Packages: 664  
 o:+o+:++. `..`..`-/oo++++/+ Shell: bash 4.4.20  
.:+o:+o/.      +sssoo+/-    WM: Not Found  
.++/+:+oo+o:`      /sssooo. CPU: BCM2835 @ 4x 1.5GHz [47.7°C]  
/+++//+:`oo+o      /:-::-.. GPU: simple  
BCM2708  
\+/+o+++`o++o      ++///// RAM: 216MiB / 3791MiB  
..+.o+++o++:`      /dddhhh.  
..+.o+oo:.      `oddhhhhh+  
 \+.++o+o`-`..`..`..ohdhhhhh+  
   :o+++ `ohhhhhhhhhho++os:  
   .o:`syhhhhhhh/.oo++o`  
         /osyyyyyyo++ooo++/  
           +oooo++\`  
             `oo++.  
ubuntu@ubuntu:~$ cat /proc/cpuinfo | tail  
CPU implementer : 0x41  
CPU architecture: 8  
CPU variant      : 0x0  
CPU part         : 0xd08  
CPU revision     : 3  
  
Hardware        : BCM2835  
Revision        : c03112  
Serial          : 1000000012eefad6  
Model           : Raspberry Pi 4 Model B Rev 1.2
```

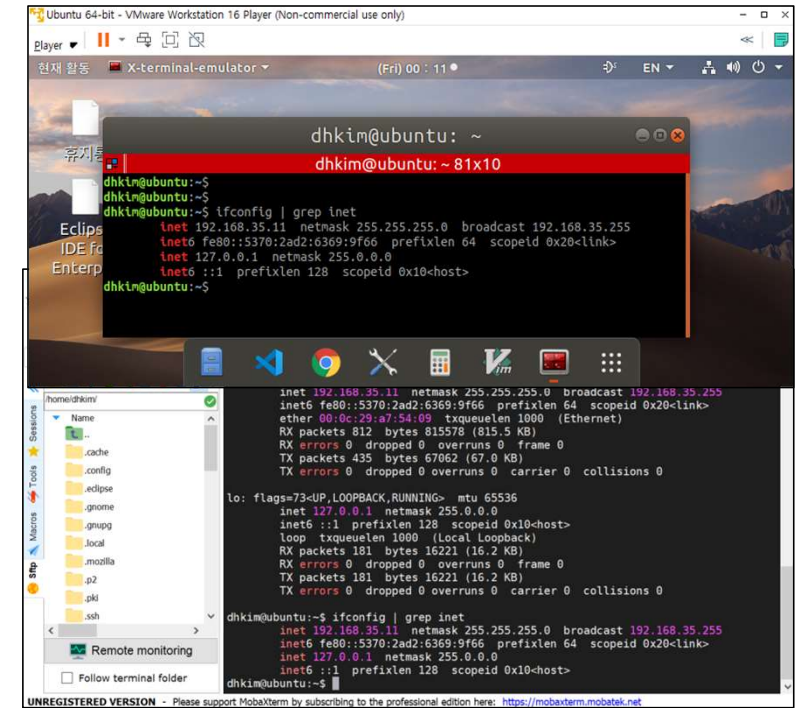
SSH 실습 – VMWare NAT

- VMWare에서 NAT설정
 - menu> Virtual Machine Settings 항목에서 네트워크 브릿지를 연결해줌

설정



접속성공



SSH 실습 - 설정파일

SSH 설정 파일을 보면서 확인해보기!

```
$ sudo vim /etc/ssh/sshd_config
```

- Port 22
 - SSH의 기본 포트는 22번입니다. 변경하면 접속할 때 명시적으로 지정
- ListenAddress 0.0.0.0
 - ListenAddress의 주석을 지우고 특정 IP 주소를 넣으면 해당 주소에서만 접속가능
 - 기본값인 0.0.0.0 모두 허용을 의미함
- Protocol 2
 - SSH 프로토콜 1의 개정판인 2를 사용합니다. 1과는 호환불가, 구형장비를
 - 보안 문제가 있어 1을 사용하지 않음

SSH 실습 – 설정파일

- SSH 접속에 사용하는 서버의 키의 위치
 - 클라이언트 접속 시 아래 네 가지 방식으로 암호화된 키 중 하나 사용
 - 기본값은 ECDSA 방식으로 암호화(=호스트 공개키)함
 - 클라이언트의 홈 디렉터리/.ssh/known_hosts 파일에 저장
 - HostKey /etc/ssh/ssh_host_rsa_key
 - HostKey /etc/ssh/ssh_host_dsa_key
 - HostKey /etc/ssh/ssh_host_ecdsa_key
 - HostKey /etc/ssh/ssh_host_ed25519_key
- PermitRootLogin prohibit-password
 - 관리자 계정인 root로 로그인을 허용하면 yes, 아니면 no
 - 기본값은 공개키 인증 방식이 아닌 아이디와 비밀번호로 로그인할 때만 금지
- PubkeyAuthentication yes
 - 공개키 인증 방식을 사용하려면 기본값
 - 아이디와 비밀번호로만 로그인하려면 주석 처리

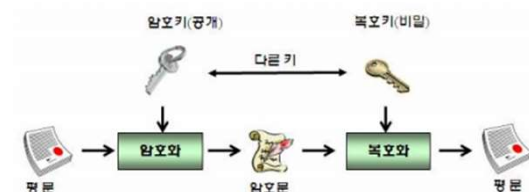
[참고] 공개 키 & 비공개 키 방식

암호화 방식의 두가지 종류

- 대칭 키(비공개 키) : 전송 시 암호키를 같이 넘겨줌
 - 속도가 매우 빠르지만, 키 분배가 어렵다
 - 금고와 열쇠를 같이 전달해주는 방식이므로 열쇠를 도둑맞는다면 큰일남
 - 대칭키 : AES, DES 등등..
- 공개키 방식(비대칭 키) : 개인이 저마다의 해독용 키를 가지고 있음
 - 공인인증서..
 - 비대칭키 원리는 수학 난제를 기반으로 설계됨 (암호 알고리즘)
 - RSA(Rivestm, Shamir, Adleman) : 큰 수는 소인수분해가 어렵다(시간이 오래걸린다)
 - ECC(Ecliptic Curve Cryptography) : 타원곡선상에 존재하는 유한체의 크기 문제
 - ElGaml : 유한체에서의 이산대수 문제



대칭키(비밀키) 암호 시스템



비대칭키(공개키) 암호시스템

[참고] 포트포워딩 VS NAT

	포트포워딩	NAT(network address translation)
접속할때 적어야 할 IP	127.0.0.1 (localhost)	타겟(리눅스)의 ip를
그게 뭔데	포트(Port)를 전달(Forwarding)해주는 작업	네트워크 주소 변환(=NAT) 은 IP 패킷의 TCP/UDP 포트 숫자와 소스 및 목적지의 IP 주소 등을 재 기록하면서 라우터를 통해 네 트워크 트래픽을 주고 받는 기술

FTP

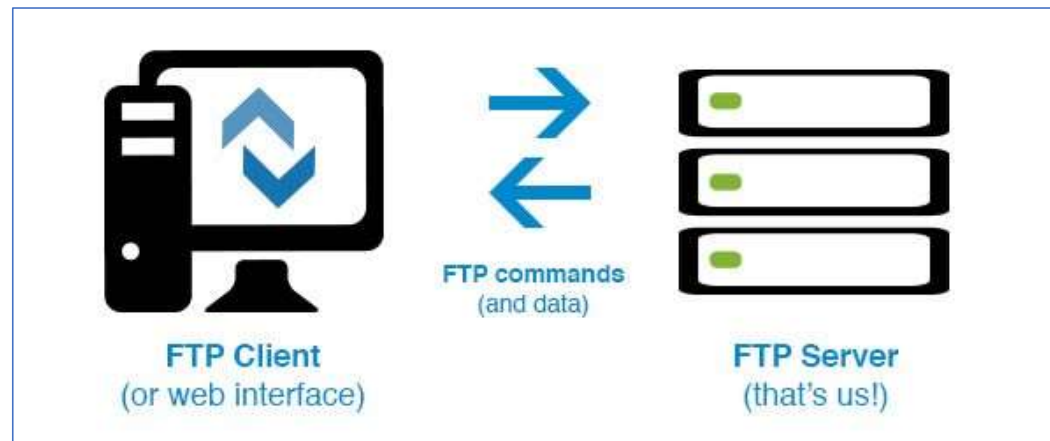
FTP

SFTP

TFTP

FTP

- 파일 전송 프로토콜(File Transfer Protocol, FTP)
- TCP/IP 프로토콜을 가지고 서버와 클라이언트 사이의 파일 전송을 하기 위한 프로토콜



FTP

FTP 보안문제

- FTP에서는 암호가 암호화되지 않아 위험해유
- FTP는 보안 프로토콜로 계획되지 않음, 수많은 보안 취약점이 존재
 - 무차별 대입 공격
 - FTP 바운스 어택
 - 패킷 가로채기
 - 포트 훔치기 (다음에 열릴 포트를 추측하여 적절한 연결을 빼앗는 것)
 - 스푸핑 공격
 - 사용자 이름 열거
- 결론: 보안을 위해서는 다른거 쓰세요 - SFTP 혹은 FTPS

FTP 실습

- 실습을 위해 설치
\$ sudo apt install -y vsftpd
- 설치 후 서비스 데몬을 재시작
\$ /etc/init.d/vsftpd restart
- ps 명령어를 통해 서비스가 제공되는지 확인
\$ ps -ef | grep vsftpd

```
ubuntu@ubuntu:~$ ps -ef | grep vsftpd
root      3744      1   0 06:42 ?        00:00:00 /usr/sbin/vsftpd /etc/vsftpd.conf
ubuntu    3765    3337   0 06:42 pts/0    00:00:00 grep --color=auto vsftpd
```

- ftp 서비스 상태를 출력
\$ service vsftpd status

```
ubuntu@ubuntu:~$ service vsftpd status
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: enab
   Active: active (running) since Fri 2021-01-08 06:42:36 UTC; 1min 37s ago
     Process: 3743 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, stat
   Main PID: 3744 (vsftpd)
      Tasks: 1 (limit: 4443)
    CGroup: /system.slice/vsftpd.service
            └─3744 /usr/sbin/vsftpd /etc/vsftpd.conf

Jan 08 06:42:36 ubuntu systemd[1]: Starting vsftpd FTP server...
Jan 08 06:42:36 ubuntu systemd[1]: Started vsftpd FTP server.
lines 1-11/11 (END)
```

FTP

ftp 환경설정

- 설정파일을 변경한다
\$ sudo vi /etc/vsftpd.conf
- 파일 수정 후 한번 더
\$ service vsftpd restart

안될때?

- 방화벽
- 설정파일(컴퓨터는 거짓말을 하지않음)

```
listen=YES
listen_ipv6=NO
anonymous_enable=NO
port_enable=NO
pasv_enable=YES
local_enable=YES
write_enable=YES
use_localtime=YES
xferlog_enable=YES
chroot_local_user=YES
allow_writeable_chroot=YES
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
ftpd_banner=Welcome to My FTP Server!
ssl_enable=YES
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-
snakeoil.key
listen_port=21
pasv_min_port=60020
pasv_max_port=60030
#utf8_filesystem=YES
#local_umask=022
```

TFTP

TFTP란?

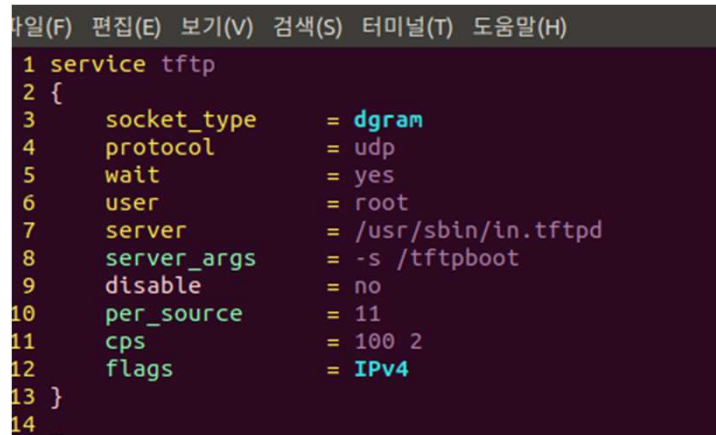
- 간단한 파일 전송 프로토콜 (TFTP, Trivial File Transfer Protocol)
- FTP와 마찬가지로 파일을 전송하기 위한 프로토콜이지만, FTP보다 더 단순한 방식으로 파일을 전송
 - 단점 : 데이터 전송 과정에서 데이터가 손실될 수 있는 등 불안정
 - 장점 : FTP처럼 복잡한 프로토콜을 사용하지 않기 때문에 구현이 간단
 - 사용 : 임베디드 시스템에서 운영 체제 업로드로 주로 사용



TFTP 실습

TFTP 실습

- tftp 를 설치하기 위해 필요한 다른 패키지와 함께 설치
\$ sudo apt install xinetd tftp tftpd
- 설치 후에는 tftp를 사용하기 위한 /etc/xinetd.d/tftp 파일을 생성
\$ sudo vi /etc/xinetd.d/tftp



```
1 service tftp
2 {
3     socket_type    = dgram
4     protocol       = udp
5     wait           = yes
6     user           = root
7     server         = /usr/sbin/in.tftpd
8     server_args    = -s /tftpboot
9     disable        = no
10    per_source      = 11
11    cps              = 100 2
12    flags           = IPv4
13 }
14
```

복사해서 붙여넣기, 타이핑하면 더 좋고

```
service tftp
{
    socket_type    = dgram
    protocol       = udp
    wait           = yes
    user           = root
    server         = /usr/sbin/in.tftpd
    server_args    = -s /tftpboot
    disable        = no
    per_source      = 11
    cps              = 100 2
    flags           = IPv4
}
```

TFTP 실습

TFTP 실습

- tftp에서 파일을 제공할 때 사용하는 디렉토리를 생성
 - \$ sudo mkdir /tftpboot
 - \$ sudo chmod 777 /tftpboot
- tftp 서버 재실행하고 테스트
 - \$ sudo /etc/init.d/xinetd restart
- /tftpboot에 파일을 만들고 다른 디렉토리로 이동하여 파일을 받기
 - \$ cd /tftpboot
 - \$ vi test.txt
 - <파일 내용 입력>
 - "hello tftp" + 이름 + 아무거나 적고 후 저장하고 닫기

TFTP

- tftp 로컬호스트 다운로드

\$ cd

: 다운로드 테스트를 위해 임시 임시 디렉토리로 이동

\$ tftp localhost

: tftp 서버와 연결

tftp> get test.txt

: test.txt 다운로드

- 결과 : Received 6 bytes in 0.0 seconds

tftp> quit

: tftp 종료

```
dhkim@dhvb:/tmp$ tftp localhost
tftp> get test.txt
Received 33 bytes in 0.0 seconds
tftp> quit
```

SFTP

- SFTP란?

- Secure File transfer protocol
- FTP와 같이 파일을 전송할 때 사용되며, 내용을 암호화 시켜서 전송합니다.
- SSH에 부가적으로 있는 기능들 중 하나의 기능
- FTP와 같은 역할을 하지만, 사용하는 포트는 FTP 포트(기본 21포트) 가 아닌 SSH 접속시 사용하는 포트를 사용함(기본은 22번)

- 실습을 위한 sftp 설치

- ssh를 설치했다면, 이미 설치된 상태, 한번 더 확인
 - \$ sudo apt install openssh-sftp-server
 - \$ whereis sftp

```
ubuntu@ubuntu:~$ whereis sftp
sftp: /usr/bin/sftp /usr/share/man/man1/sftp.1.gz
ubuntu@ubuntu:~$
```

SFTP

SFTP실습 : 로컬호스트 내부에서 SFTP 파일전송

- sftp 실습을 위해 파일을 생성
\$ seq 1 22 > num.txt
\$ cat num.txt

```
ubuntu@ubuntu:~$ seq 1 22 > num.txt
ubuntu@ubuntu:~$ head num.txt
1
2
3
4
5
6
7
8
9
10
ubuntu@ubuntu:~$
```

- 원격 접속 프로그램을 통해 파일을 복사해보자(cp명령어 사용금지)
 - 테스트를 위해서 /tmp로 이동 (num.txt를 복사해 위치시킬 곳)
\$ sudo cd /tmp

```
ubuntu@ubuntu:~$ cd /tmp
```

- sftp로 접속한다
\$ sftp [계정]@주소
예시) \$ sftp dhkim@localhost

```
ubuntu@ubuntu:/tmp$ ls
systemd-private-7b578a7122484e02a91a2cbe5c
systemd-private-7b578a7122484e02a91a2cbe5c
ubuntu@ubuntu:/tmp$ sftp ubuntu@localhost
ubuntu@localhost's password:
Connected to localhost.
```

SFTP

SFTP실습 : 로컬호스트 내부에서 SFTP 파일전송

- 파일 전송 : 파일전송을 위해 get 명령어를 사용

```
$stfp> get /home/dhkim/num.txt
```

```
$stfp> exit
```

- 이후 cat 명령어로 num.txt파일을 확인해보면 정상적인 파일 전송 확인

```
ubuntu@ubuntu:/tmp$ sftp ubuntu@localhost
ubuntu@localhost's password:
Connected to localhost.
sftp> get /home/dhkim/num.txt
File "/home/dhkim/num.txt" not found.
sftp> get /home/ubuntu/num.txt
Fetching /home/ubuntu/num.txt to num.txt
/home/ubuntu/num.txt 100%
sftp> exit
ubuntu@ubuntu:/tmp$ ls
num.txt
systemd-private-7b578a7122484e02a91a2cbe5c665a72-sys
systemd-private-7b578a7122484e02a91a2cbe5c665a72-sys
ubuntu@ubuntu:/tmp$ head num.txt
1
2
3
4
5
6
7
8
9
10
ubuntu@ubuntu:/tmp$
```

[참고] SFTP 와 SCP 차이

SCP와 SFTP 프로토콜의 차이

- SCP
 - 파일 전송에만 사용할 수 있으며 비 대화식
 - 모든 것이 CLI(셸)에서 작성됨
 - SFTP 대비 전송속도가 빠르다
 - 한번 실행되면 중단이 불가능하다 (Like as 대포알)
- SFTP
 - 정교하며 대화식 명령으로 디렉토리 작성,
 - 디렉토리 및 파일 삭제 (시스템 권한에 따라 다름) 등과 같은 작업을 수행 가능
 - 전송도중 전송중단이 가능하다 (Like as 폭격기)

[참고] SFTP 와 FTPS

- SFTP
 - SSH 를 기반으로 FTP로 파일을 주고받음, 그래서 안전하다
 - MobaXterm이 내부적으로 사용됨
- FTPS
 - FTP를 secure하게 업그레이드 한 버전
 - 둘이 다른거임

SCP

SCP

SCP : secure copy

- 원격지에 있는 파일을 복사하는 프로그램
- 네트워크를 통해 파일을 복사
- 주로 ssh 연결을 통해 접속한 컴퓨터에 파일을 복사하기 위해 사용
- 우분투 배포판은 기본 설치되어있음

- 설치 확인
\$ whereis scp

```
ubuntu@ubuntu:/tmp$ whereis scp
scp: /usr/bin/scp /usr/share/man/man1/scp.1.gz
ubuntu@ubuntu:/tmp$
```

- 좀더 자세히 알아보고 싶다면
\$ man scp

```
SCP(1) BSD General Commands Manual SCP(1)

NAME
    scp - secure copy (remote file copy program)

SYNOPSIS
    scp [-346BCpqRTv] [-c cipher] [-F ssh_config] [-i identity_file] [-l limit]
        [-o ssh_option] [-P port] [-S program] [[user@]host1:]file1 ...
        [[user@]host2:]file2

DESCRIPTION
    scp copies files between hosts on a network. It uses ssh(1) for data
    transfer, and uses the same authentication and provides the same security
    as ssh(1). scp will ask for passwords or passphrases if they are needed
    for authentication.
```


SCP 실습

SCP 실습 : 다른 위치로 복사(파일 보내기)

- SCP 사용법

\$ scp [파일] [계정]@[서버주소]:[타겟에서 절대 경로]

- SCP 파일 보내기 예시

\$ scp test.txt user@135.79.246.80:/home/user/
: test.txt를 135.79.246.80 서버의 /home/user/ 폴더에 업로드

SCP 실습

SCP 실습 : 다른 위치로 복사(파일 보내기)

- 내컴퓨터간 다른 위치로 scp로 복사, "data.txt" 라는 파일을 이동시킴
 - 출발지 : /tmp
 - 목적지 : /home/ubuntu

```
ubuntu@ubuntu:/tmp$ ls -l | grep data
-rw-rw-r-- 1 ubuntu ubuntu 57 Jan 8 07:08 data.txt
ubuntu@ubuntu:/tmp$ scp data.txt ubuntu@localhost:/home/ubuntu/
ubuntu@localhost's password:
data.txt                                100% 57 65.5KB/s 00:00
ubuntu@ubuntu:/tmp$
```

```
$ ls -l | grep data
$ scp data.txt ubuntu@localhost:/home/ubuntu/
$ cd /home/ubuntu/
$ ls -l | grep data
```

```
ubuntu@ubuntu:/tmp$ cd /home/ubuntu/
ubuntu@ubuntu:~$ ls -l | grep data
-rw-rw-r-- 1 ubuntu ubuntu 57 Jan 8 07:35 data.txt
ubuntu@ubuntu:~$
```

- 처음 1회 사용시 인증을 위한 절차가 필요하며 기본 ESDSA 방식으로 암호화된다.

wget

wget

wget

: Web Get 웹 상의 파일을 다운로드 받을 때 사용하는 명령어

- recursive downloading기능으로 사이트 전체를 로컬로다운가능
- 느리거나 불안정한 네트워크 환경에서도 잘 동작함
- 네트워크 환경이 불안해서 도중에 연결이 끊겼다면, 연결이 끊긴 시점부터 다운로드 받는 기능 있음

wget 사용법

- Wget 사용법

- 기본

- \$ wget [다운받을 URL]

- 다른 이름으로 저장

- \$ wget -O [저장이름] [다운받을 URL]

- 이어받기

- \$ wget -c [다운받을 URL]

- 백그라운드에서 다운로드

- \$ wget -b [다운받을 URL]

Wget 실습

- Wget 실습

- 리눅스 커널 아카이브 사이트에 들어가보자

링크 : <https://www.kernel.org/>

- Longterm > 4.19.165 > [tarball]

- 위 항목에서 마우스 오른쪽 "링크주소복사"

- 명령어 입력

\$ Wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.165.tar.xz

```
ubuntu@ubuntu:~$ wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.165.tar.xz
--2021-01-08 08:00:42-- https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.165.tar.xz
Resolving cdn.kernel.org (cdn.kernel.org)... 151.101.77.176, 2a04:4e42:1a::432
Connecting to cdn.kernel.org (cdn.kernel.org)|151.101.77.176|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 103523764 (99M) [application/x-xz]
Saving to: 'linux-4.19.165.tar.xz'

linux-4.19.165.tar.x 100%[=====] 98.73M 10.9MB/s in 9.2s

2021-01-08 08:00:52 (10.7 MB/s) - 'linux-4.19.165.tar.xz' saved [103523764/103523764]

ubuntu@ubuntu:~$ ls -l |grep xz
-rw-rw-r-- 1 ubuntu ubuntu 103523764 Jan 6 13:52 linux-4.19.165.tar.xz
ubuntu@ubuntu:~$
```

The Linux Kernel Archives

About Contact us FAQ Releases Signatures Site news

Protocol Location
HTTP https://www.kernel.org/pub/
GIT https://git.kernel.org/
RSYNC rsync://rsync.kernel.org/pub/

Latest Release
5.10.5

mainline:	5.11-rc2	2021-01-03	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
stable:	5.10.5	2021-01-06	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
stable:	5.9.16 [EOL]	2020-12-21	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.4.87	2021-01-06	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.19.165	2021-01-06	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.14.213	2020-12-29	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.9.249	2020-12-29	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.4.249	2020-12-29	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	next-20210108	2021-01-08	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]

Other resources

새 탭에서 링크 열기(T)
새 창에서 링크 열기(W)
시크릿 창에서 링크 열기(G)
다른 이름으로 링크 저장(K)...
링크 주소 복사(E)
블록사이트 (BlockSite) - 프록시 와 제어
검색(N) Ctrl+Shift+I

끝