



LINUX 02

리눅스 개념과 사용



2020.11.

민병훈 / jismin@naver.com

목차

▶ 리눅스01 – 리눅스 시스템 개요 및 활용

일차	내용
1일차(6hrs)	리눅스 개요 가상머신 소개 리눅스 설치
2일차(6hrs)	리눅스 부팅과정 소개 리눅스 파일 및 디렉터리 구조 이해 파일 링크 리눅스 기본 명령 편집기 사용하기 (vi, ed, gedit, 등)
3일차(6hrs)	사용자와 그룹 관리 파일 소유권과 허가권 이해 및 활용 디스크 쿼터 사용
4일차(6hrs)	파이프와 필터 파일 디스크립터와 리디렉션
5일차(6hrs)	프로세스 이해 및 관리 작업 스케줄 관리 시스템 로그 관리
6일차(6hrs)	데이터 묶기와 압축 데이터 백업 및 복구 소프트웨어 패키지 관리
7일차(6hrs)	디스크 관리 및, 파티션의 이해 RAID 와 LVM 개념 및 구축
8일차(6hrs)	셸 프로그래밍

▶ 리눅스02 – 리눅스 네트워크 구축 및 활용

일차	내용
1일차(6hrs)	네트워크 기본 개념 및 관리
2일차(6hrs)	SSH 서버 구축 및 활용 원격 시스템 접속 웹 서버 및 WAS 서버
3일차(6hrs)	파일 공유 시스템(NFS , SAMBA) 구축 및 활용
4일차(6hrs)	DNS 서버 및 백업 서버 구축 및 활용
5일차(6hrs)	데이터베이스 서버 구축과 활용
6일차(6hrs)	Mail server 구축 및 활용
7일차(6hrs)	방화벽 구축 및 활용
8일차(6hrs)	시스템과 네트워크 모니터링 시스템과 네트워크 관리 및 관련 명령어 정리

들어가기

- ▶ xinetd
- ▶ 원격 시스템 연결
- ▶ SSH 서버

xinetd

▶ Daemon

- ▶ 멀티태스킹 운영체제에서 백그라운드에서 돌면서 여러 작업을 처리하는 프로그램
- ▶ 특정 서비스를 위해 백그라운드 상태에서 계속 실행되는 서버 프로세스
- ▶ 시스템 관련 작업을 처리하는 백그라운드 프로세스
- ▶ 메모리 등 기타 자원을 사용
- ▶ 부팅 중 메모리에 로딩되어 종료시까지 메모리에 상주

xinetd

- ▶ Daemon 의 종류

- ▶ 단독으로 실행되는 대몬

- ▶ 클라이언트 요청이 많은 경우 유용
 - ▶ 항상 메모리에 상주하여 빠른 서비스 제공
 - ▶ httpd, named sendmail, crond, rsyslogd dhcp...

xinetd

- ▶ Daemon 의 종류
- ▶ 슈퍼대몬인 inetd 에서 관리하는 대몬
 - ▶ Internet superserver
 - ▶ 자주 실행되지 않는 대몬들을 통틀어 관리하는 슈퍼 대몬
 - ▶ 클라이언트 요청이 있는 경우에만 대몬 실행
 - ▶ 응답 시간 필요. 메모리 절약
 - ▶ 커널 2.4 이후에서는 Xinetd 이름으로 작동
 - ▶ /etc/xinetd.conf 와 /etc/xinetd.d 디렉토리의 각 서비스 파일 관리
 - ▶ telnet, ftp, finger, logi, auth, shell, tcpd...

xinetd

- ▶ inetd 와 xinetd

- ▶ inetd

- ▶ Internet Services Daemon
 - ▶ 네트워크 접속을 제어
 - ▶ 접속 요청 → inetd → tcpd → 규칙에 따라 허락여부 결정
→ 서버 프로세스 시작

xinetd

- ▶ inetd 와 xinetd

- ▶ Xinetd

- ▶ Extended Internet Services Daemon
- ▶ 외부 프로그램과 내부 프로그램 연결, 서비스 처리
- ▶ TCP, UDP, RCP 서비스에 대한 접근 제어
- ▶ 시간별 접근 제어 가능
- ▶ 접속 성공/실패 로그
- ▶ RFC1413 기반 사용자 관리
- ▶ 서비스 거부 공격(DoS) 능률적 견제
- ▶ 서버 개수 제한 / 로그 파일 크기 제한
- ▶ 접속 시도 → 인증 → 서비스 프로그램 연결
- ▶ 즉, xinetd 가 /etc/xinetd.conf 에 등록되어 있는 서비스 호출

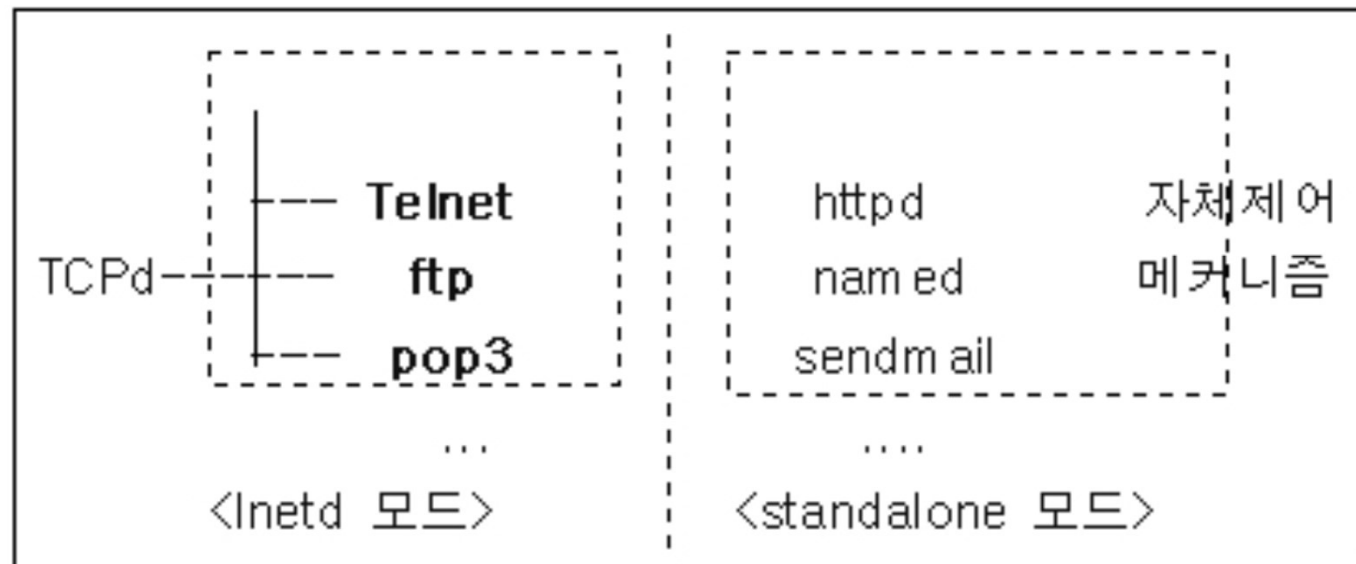
xinetd

- ▶ Standalone Daemon VS inetd, xinetd
 - ▶ Standalone daemon
 - ▶ 독립적으로 실행. 메모리 상주.
 - ▶ 서비스 즉각 응답
 - ▶ 빠른 반응 속도
 - ▶ 메모리 점유율 상승, 서버 부하 증가
 - ▶ Inetd, xinetd
 - ▶ 슈퍼대몬이 일반 대몬 관리
 - ▶ 서비스 요청시에만 메모리에 적재
 - ▶ 서버 부하 감소, 응답 속도 느림

xinetd

▶ Standalone Daemon VS inetd, xinetd

inetd 와 Standalone 모드에서의 데몬 접근제어



xinetd

▶ xinetd 설치

```
[root@LNXTTEST01 ijoo]# yum install xinetd
```

```
Loaded plugins: fastestmirror, langpacks
```

```
Loading mirror speeds from cached hostfile
```

```
~~~~~
```

```
Installed:
```

```
  xinetd.x86_64 2:2.3.15-14.el7
```

```
Complete!
```

```
[root@LNXTTEST01 ijoo]#
```

xinetd

▶ 네트워크 프로그래밍과 소켓

▶ 네트워크 프로그래밍

- ▶ 네트워크로 연결되어 있는 서로 다른 두 컴퓨터가 데이터를 주고받을 수 있도록 하는 것

▶ 소켓

- ▶ 네트워크상에서 데이터 송수신에 사용할 수 있는 소프트웨어 장치
- ▶ 소켓 생성 과정

소켓 생성	socket 함수 호출
IP 주소와 PORT 번호 할당	bind 함수 호출
연결요청 가능상태로 변경	listen 함수 호출
연결요청에 대한 수락	accept 함수 호출

xinetd

- ▶ 네트워크 프로그래밍과 소켓

- ▶ 소켓 구현 함수

```
# include <sys/socket.h>
```

```
int socket(int domain, int type, int protocol)
```

```
int bind(int sockfd, struct sockaddr *myaddr, socketlen_t addrlen);
```

```
int listen(int sockfd, int backlog);
```

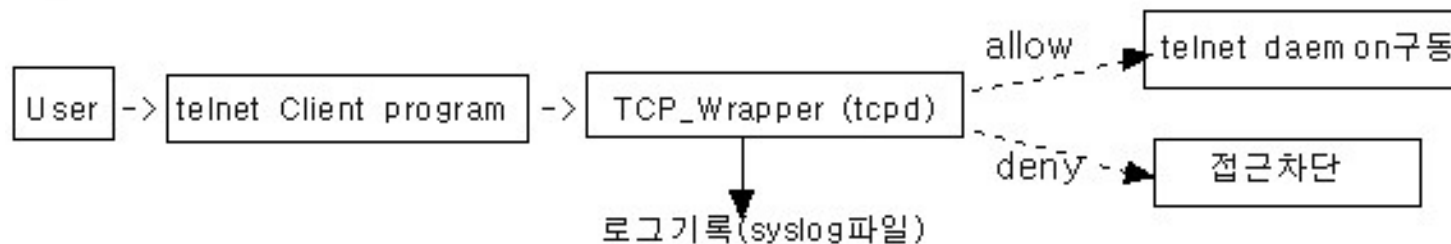
```
int accept(int sockfd, struct sockaddr *addr, socketlen_t *addrlen);
```

TCP Wrapper

▶ TCP wrapper

- ▶ 호스트 기반 네트워킹 ACL 시스템
- ▶ 인터넷 프로토콜에서 네트워크 접근을 필터링하기 위해 사용

tcp_wrapper에 의한 telnet서비스의 초기 데몬 구동 과정



TCP Wrapper

- ▶ TCP wrapper 에 의해 제어되는 서비스
 - ▶ FTP, telnet, ssh, xinetd 기반 서비스 접근 제어 (ACL) 가능
 - ▶ 파일 설정으로 특정 IP 나 대역에 대한 접근 제어
 - ▶ /etc/hosts.allow : 서비스 허용
 - ▶ /etc/hosts.deny : 서비스 차단

TCP Wrapper

- ▶ TCP wrapper 에 의해 제어되는 서비스 확인

```
[root@C7SVR01 fmin]# which xinetd
/usr/sbin/xinetd
[root@C7SVR01 fmin]# ldd /usr/sbin/xinetd
linux-vdso.so.1 => (0x00007fff05694000)
libselinux.so.1 => /lib64/libselinux.so.1 (0x00007fe8e17db000)
libwrap.so.0 => /lib64/libwrap.so.0 (0x00007fe8e15d0000)
~~~~~
[root@C7SVR01 fmin]# which sshd
/usr/sbin/sshd
[root@C7SVR01 fmin]# ldd /usr/sbin/sshd | grep libwrap
libwrap.so.0 => /lib64/libwrap.so.0 (0x00007f1857411000)
[root@C7SVR01 fmin]#
```


TCP Wrapper

▶ TCP wrapper 설정

▶ /etc/hosts.deny

- ▶ TCP wrapper 에 의해 제어되는 모든 서비스에 대해, 모든 IP 차단

```
[root@C7SVR01 fmin]# cat /etc/hosts.deny
```

```
#
```

```
# hosts.deny          This file contains access rules which are used to  
#                     deny connections to network services that either use
```

```
~~~~~
```

```
ALL:ALL
```

TCP Wrapper

▶ TCP wrapper 설정

▶ /etc/hosts.allow

- ▶ 접근 허용할 IP 허가 리스트
- ▶ /etc/hosts 파일에 등록된 호스트들에 대하여 모든 서비스 허가하고,
- ▶ sshd 에 대해서는 등록 IP 에 대해서만 허가

```
[root@C7SVR01 fmin]# cat /etc/hosts.allow
```

```
~~~~~
```

```
ALL:LOCAL
```

```
sshd:192.168.8.201, 192.168.8.202
```

TCP Wrapper

- ▶ TCP wrapper 설정
 - ▶ /etc/hosts
 - ▶ 등록된 IP 의 경우 서비스 접근 가능

```
[root@C7SVR01 fmin]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1      localhost localhost.localdomain localhost6 localhost6.localdomain6
lnxsvr01 192.168.8.1
lnxsvr02 192.168.8.2
lnxsvr03 192.168.8.3
[root@C7SVR01 fmin]#
```

원격시스템 연결

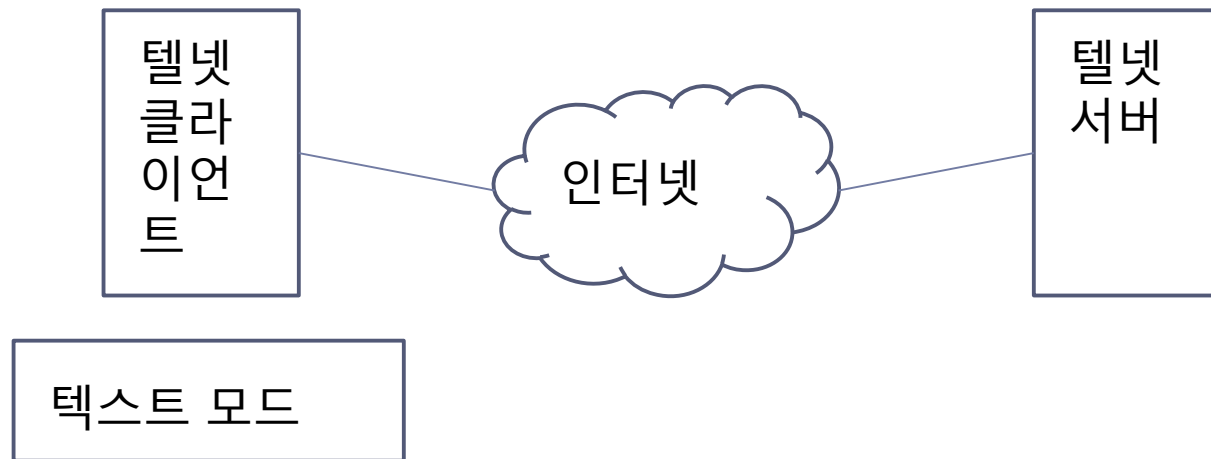
▶ 원격 시스템 연결

- ▶ Telnet 서버
- ▶ RDP 서버
- ▶ VNC 서버
- ▶ SSH 서버

원격시스템 연결

▶ Telnet 서버

- ▶ 전통적인 원격 접속
- ▶ 보안 취약
- ▶ 연결 프로그램
 - ▶ 텔넷 서버
 - ▶ 텔넷 클라이언트



원격시스템 연결

▶ Telnet 서버 구축

```
# yum install telnet-server  
# systemctl start telnet.socket  
# useradd ~~~ teluser  
# passwd teluser  
# firewall-config
```

```
c:\> telnet server-ip
```

원격시스템 연결

- ▶ RDP (Remote Desktop Protocol)
 - ▶ MicroSoft 에서 개발한 원격 데스크톱 프로토콜
 - ▶ GUI 기반 원격시스템 접속



원격시스템 연결

▶ VNC (Virtual Network Computing)

- ▶ X 윈도우 환경으로 원격 접속
- ▶ 그래픽 전송이므로 속도가 아주 느림



원격시스템 연결

▶ SSH 서버

- ▶ 텔넷과 동일한 용도
- ▶ 보안 강화



원격시스템 연결

▶ SSH 서버

- ▶ SSH 서버 설치 확인 및 방화벽 설정

```
# systemctl status sshd  
# firewall-config
```

- ▶ 클라이언트에서 접속
 - ▶ 리눅스 클라이언트

```
# ssh username@server-ip_address
```

- ▶ 윈도우 클라이언트
 - ☐ 윈도우에서 Putty 다운로드
 - ☐ Putty 프로그램 실행

원격시스템 연결

▶ 원격 접속 비교

구분	Telnet	SSH	VNC
속도	빠르다	빠르다	느리다
보안	취약	강화	취약, SSH 와 연동 가능
그래픽지원	No	No	Yes
명령	텍스트 모드	텍스트 모드	제한없음
클라이언트 프로그램	대부분 기본	리눅스는 기본, 윈도우는 별도 설치	별도 설치

원격시스템 연결

▶ SSH (Secure Shell Protocol)

- ▶ Public Network (인터넷)를 통해 서로 통신을 할 때 보안적으로 안전하게 통신을 하기 위해 사용하는 프로토콜
- ▶ 데이터 전송과 원격 제어
- ▶ 파일 전송의 예 : 깃허브
- ▶ 원격 제어의 예 : AWS 등 클라우드 서비스

원격시스템 연결

- ▶ SSH (Secure Shell Protocol) 의 인증 과정
 - ▶ Private Key 와 Public Key 사용
 - ▶ Public Key
 - ▶ 공개 키
 - ▶ 전송전 공개키를 통해 암호화
 - ▶ 단방향, 즉 암호화는 가능하나 복호화는 불가능
 - ▶ Private Key
 - ▶ Public Key 와 쌍을 이룬다.
 - ▶ 절대 노출되면 안된다.
 - ▶ 컴퓨터 내부에 저장
 - ▶ 암호화된 메시지를 복호화

원격시스템 연결

- ▶ SSH (Secure Shell Protocol) 의 인증 과정
 - ▶ Private Key와 Public Key를 통해 다른 컴퓨터와 통신할 때
 - ▶ 먼저 Public Key를 통신 하고자 하는 컴퓨터에 복사하여 저장
 - ▶ 클라이언트에서 접속 요청 시 서버에 저장되어 있는 Public Key와 클라이언트의 해당 Public Key와 쌍을 이루는 Private Key와 비교
 - ▶ 관계 Key 가 증명이 되면 암호화된 채널이 형성
 - ▶ Key를 활용해 메시지를 암호화하고 복호화하며 데이터를 전송

원격시스템 연결

▶ SSH 프로토콜 접속 조건

- ▶ 22번 tcp 포트가 방화벽에서 open
- ▶ ssh 서버 프로그램이 설치 및 구동
- ▶ ssh 클라이언트 필요
- ▶ 최신 리눅스 배포판은 ssh 서버 기본 탑재
- ▶ 설정 파일 : /etc/sshd/sshd_config

```
# rpm -qa | grep openssh-server
```

```
# yum install openssh-server
```

```
# which sshd
```

```
# iptables -A INPUT -p tcp -m tcp --dport 22 -j
```

```
# firewall-cmd --zone=public --add-port=22/tcp --permanent
```

```
# systemctl status sshd
```

```
# ps -aef | grep sshd
```

원격시스템 연결

▶ SSH 연결

- ▶ 윈도우에서 연결
 - ▶ 리눅스에서 서비스 확인

```
[fmin@localhost ~]$ systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since 일 2020-08-09 18:34:04 KST; 2h 10min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 1124 (sshd)
    Tasks: 1
   CGroup: /system.slice/sshd.service
           └─1124 /usr/sbin/sshd -D

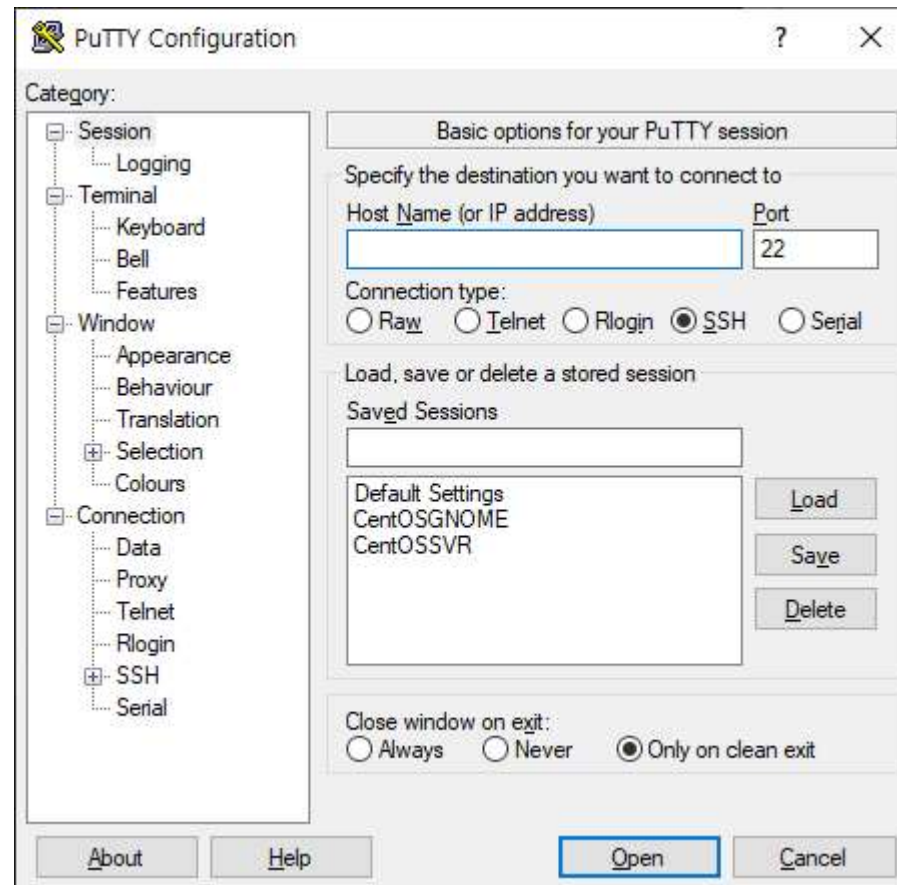
8월 09 18:34:04 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
8월 09 18:34:04 localhost.localdomain sshd[1124]: Server listening on 0.0.0.0 port 22.
8월 09 18:34:04 localhost.localdomain sshd[1124]: Server listening on :: port 22.
8월 09 18:34:04 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
```

- ▶ 윈도우에서 putty 실행

원격시스템 연결

▶ SSH 연결

▶ 윈도우에서 연결



원격시스템 연결

▶ SSH 연결

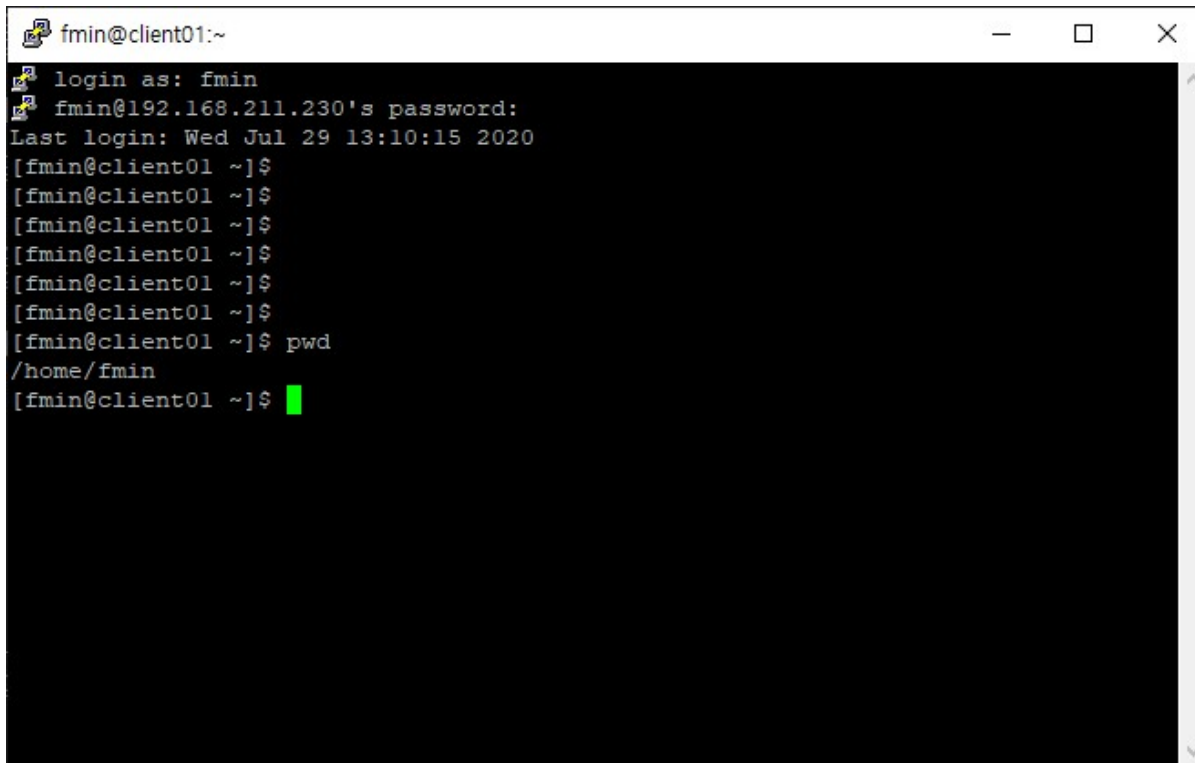
▶ 윈도우에서 연결



원격시스템 연결

▶ SSH 연결

▶ 윈도우에서 연결

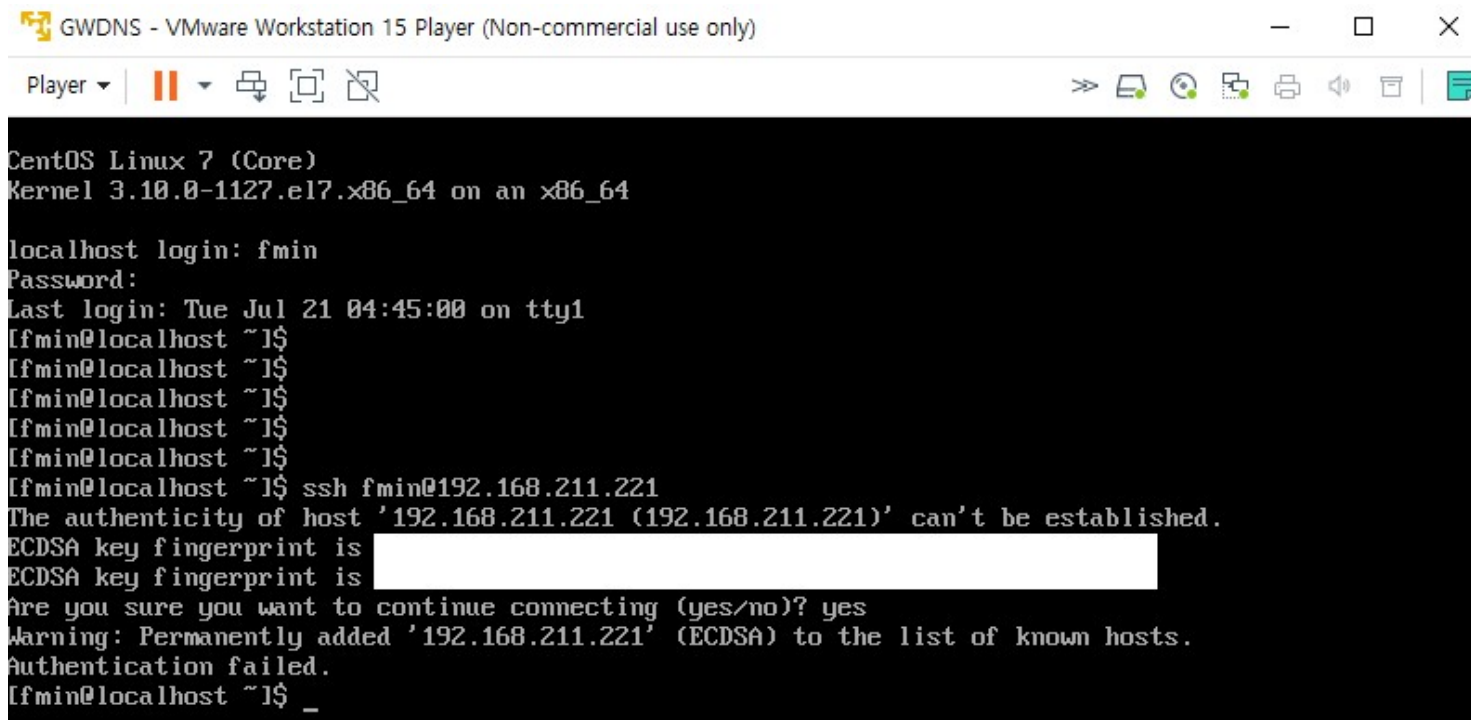


```
fmin@client01:~  
login as: fmin  
fmin@192.168.211.230's password:  
Last login: Wed Jul 29 13:10:15 2020  
[fmin@client01 ~]$  
[fmin@client01 ~]$  
[fmin@client01 ~]$  
[fmin@client01 ~]$  
[fmin@client01 ~]$  
[fmin@client01 ~]$  
[fmin@client01 ~]$  
[fmin@client01 ~]$ pwd  
/home/fmin  
[fmin@client01 ~]$
```

원격시스템 연결

▶ SSH 연결

▶ 리눅스에서 연결



```
GWDNS - VMware Workstation 15 Player (Non-commercial use only)
Player | [Pause] [Full Screen] [Close]
CentOS Linux 7 (Core)
Kernel 3.10.0-1127.el7.x86_64 on an x86_64

localhost login: fmin
Password:
Last login: Tue Jul 21 04:45:00 on tty1
[fmin@localhost ~]$
[fmin@localhost ~]$
[fmin@localhost ~]$
[fmin@localhost ~]$
[fmin@localhost ~]$
[fmin@localhost ~]$
[fmin@localhost ~]$ ssh fmin@192.168.211.221
The authenticity of host '192.168.211.221 (192.168.211.221)' can't be established.
ECDSA key fingerprint is [REDACTED]
ECDSA key fingerprint is [REDACTED]
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.211.221' (ECDSA) to the list of known hosts.
Authentication failed.
[fmin@localhost ~]$ _
```

Q & A
