

- Ch\_1 \_ 데이터베이스 생성과 운영
- Ch\_2 \_ Visual studio 와 MariaDB 연동
- Ch\_3 \_ Oracle 설치 및 운영
- Ch\_4 \_ 리눅스 서버에 USB 마운트
- Ch\_5 \_ SCP 파일 전송과 SMB 공유 파일 실습

- 데이터베이스 생성과 운영 (MariaDB)

```
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql       |
| performance_schema |
+-----+
3 rows in set (0.000 sec)
```

➔ SHOW DATABASES 데이터베이스 확인

```
MariaDB [(none)]> CREATE DATABASE shopping_db;
Query OK, 1 row affected (0.000 sec)
```

➔ 데이터베이스 생성 (shopping\_db)

```
MariaDB [shopping_db]> CREATE TABLE customer (
  -> id VARCHAR(10) NOT NULL PRIMARY KEY,
  -> name VARCHAR(5),
  -> age INT,
  -> address VARCHAR(5) );
Query OK, 0 rows affected (0.016 sec)
```

➔ shopping\_db 안에 customer TABLE 을 생성

```

MariaDB [shopping_db]> CREATE TABLE purchase (
  -> no INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  -> cust_id VARCHAR(10),
  -> date CHAR(8),
  -> product VARCHAR(5) );
Query OK, 0 rows affected (0.010 sec)

MariaDB [shopping_db]> █

```

➔ purchase TABLE 도 생성

- AUTO\_INCREMENT 사용하여 no 에 숫자가 순서대로 자동 부여됨
- 총 데이터베이스 1 개 테이블 2 개 생성함

(shopping\_db, customer, purchase)

```

MariaDB [(none)]>
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| shopping_db |
+-----+
4 rows in set (0.010 sec)

```

➔ shopping\_db 생성이 잘 적용된 모습

```

MariaDB [shopping_db]> INSERT INTO customer VALUES ('hong','홍길동',22,'경기');
Query OK, 1 row affected (0.002 sec)

MariaDB [shopping_db]> INSERT INTO customer VALUES ('dang','당탕이',23,'충북');
Query OK, 1 row affected (0.001 sec)

MariaDB [shopping_db]> INSERT INTO customer VALUES ('ppuni','이뿌니',30,'서울');
Query OK, 1 row affected (0.001 sec)

MariaDB [shopping_db]> INSERT INTO customer VALUES ('john','존앤이',28,'강원');
Query OK, 1 row affected (0.001 sec)

```

➔ customer table 에 내용 삽입

```

MariaDB [shopping_db]>
MariaDB [shopping_db]> INSERT INTO purchase VALUES (null,'hong',20210122,'TV');
Query OK, 1 row affected (0.001 sec)

MariaDB [shopping_db]> INSERT INTO purchase VALUES (null,'ppuni',20210211,'TV');
Query OK, 1 row affected (0.001 sec)

MariaDB [shopping_db]> INSERT INTO purchase VALUES (null,'john',20210211,'냉장고');
Query OK, 1 row affected (0.001 sec)

MariaDB [shopping_db]> INSERT INTO purchase VALUES (null,'hong',20210222,'세탁기');
Query OK, 1 row affected (0.001 sec)

MariaDB [shopping_db]> INSERT INTO purchase VALUES (null,'john',20210311,'베디오');
Query OK, 1 row affected (0.001 sec)

```

➔ purchase table 에 내용 삽입

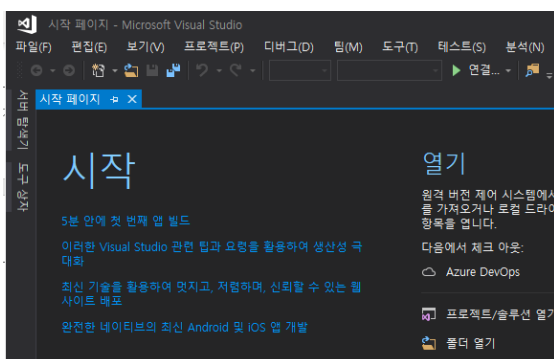
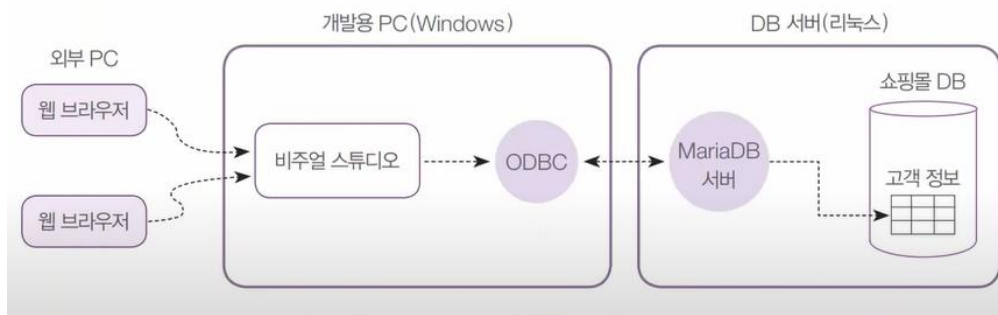
- no 부분은 AUTO\_INCREMENT 이기 때문에 null 로 비워 두면 자동으로 삽입

```
MariaDB [shopping_db]> SELECT * FROM customer;
+-----+-----+-----+-----+
| id | name | age | address |
+-----+-----+-----+-----+
| dang | 당 탕 이 | 23 | 충 북 |
| hong | 홍 길 동 | 22 | 경 기 |
| john | 존 변 이 | 28 | 강 원 |
| ppuni | 이 부 니 | 30 | 서 울 |
+-----+-----+-----+-----+
4 rows in set (0.000 sec)

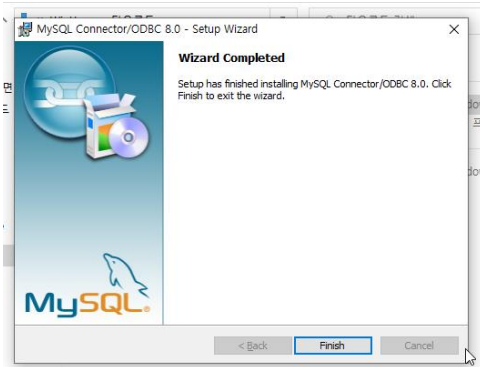
MariaDB [shopping_db]> SELECT * FROM purchase;
+-----+-----+-----+-----+
| no | cust_id | date | product |
+-----+-----+-----+-----+
| 1 | hong | 20210122 | TV |
| 2 | ppuni | 20210211 | TV |
| 3 | john | 20210211 | 냉 장 고 |
| 4 | hong | 20210222 | 세 탁 기 |
| 5 | john | 20210311 | 비 디 오 |
+-----+-----+-----+-----+
5 rows in set (0.000 sec)
```

➔ select 으로 내용 확인

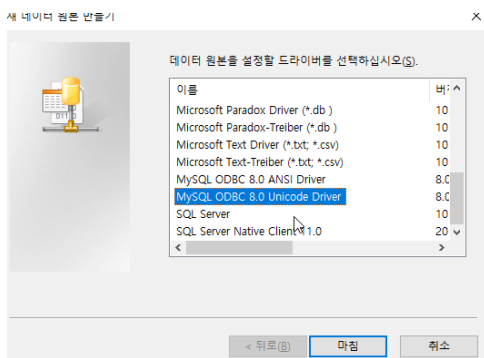
## ● Visual Studio 와 MariaDB 연동



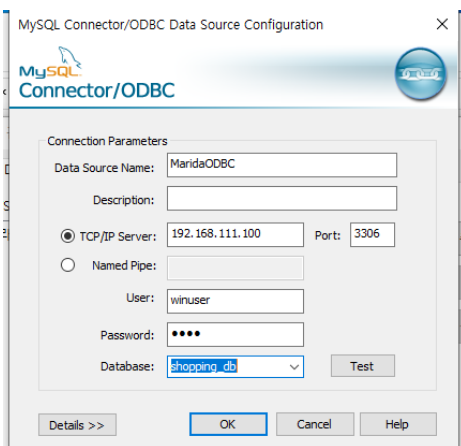
➔ winclient 에서 visual studio 설치 (본인은 2017 버전)



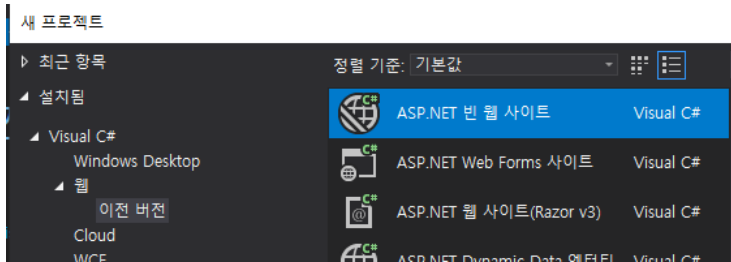
➔ ODBC 도 기본값으로 설치



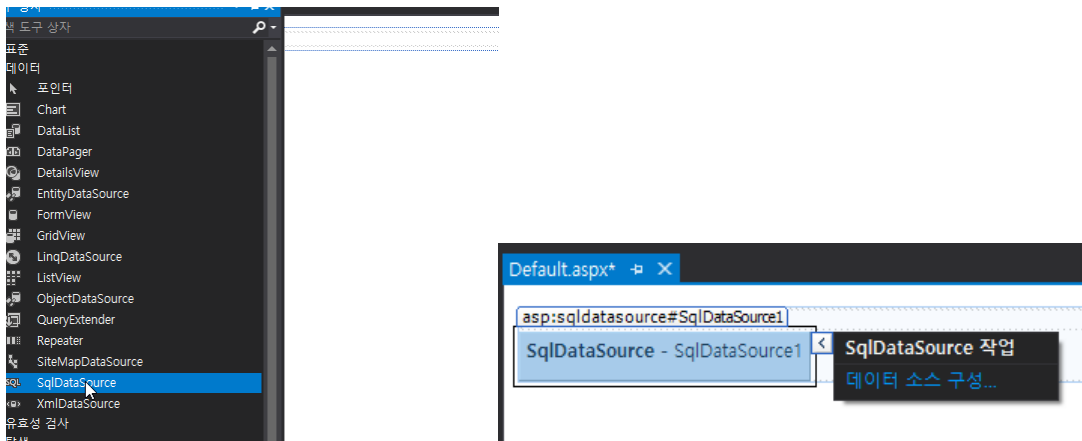
➔ 제어판 > 시스템 및 보안 > 관리도구 > ODBC 데이터 원본 관리자 > 새 데이터 원본 만들기 에서 MYSQL ODBC ... 선택



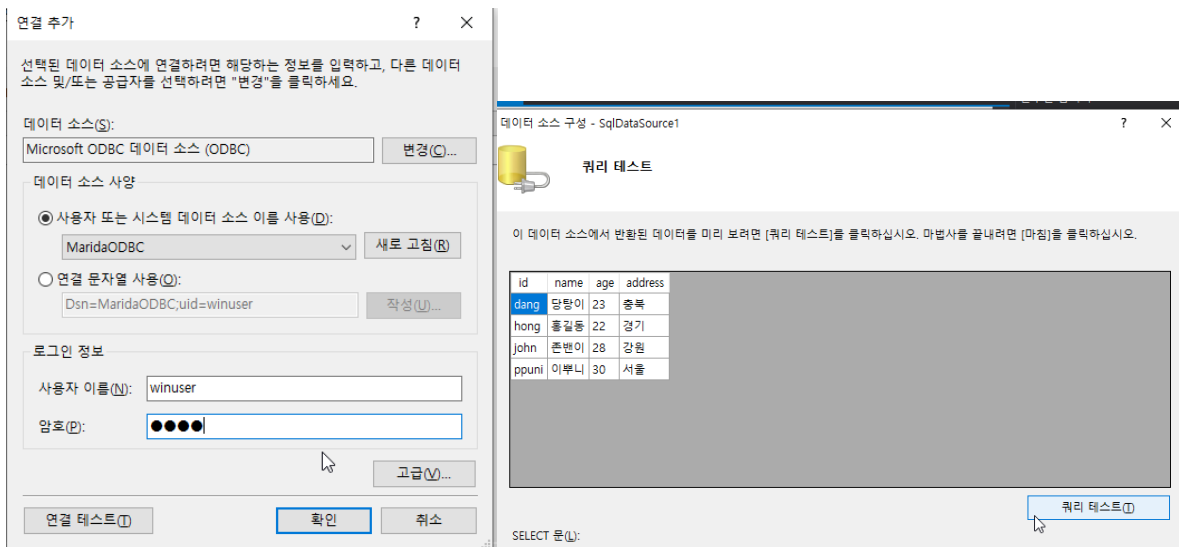
➔ ODBC 설정



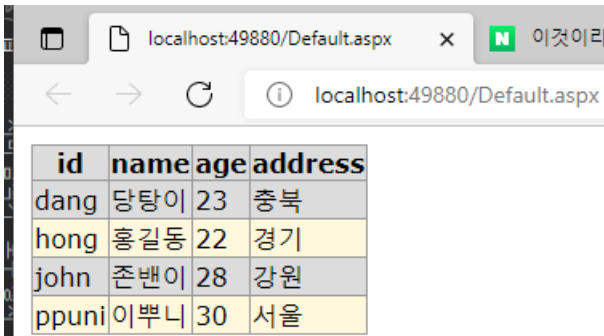
→ visual studio 실행 후 프로젝트 생성



→ 프로젝트에 sqldatasource 추가



→ sqldatasource 에 odbc 연결 추가 후 쿼리 테스트 진행

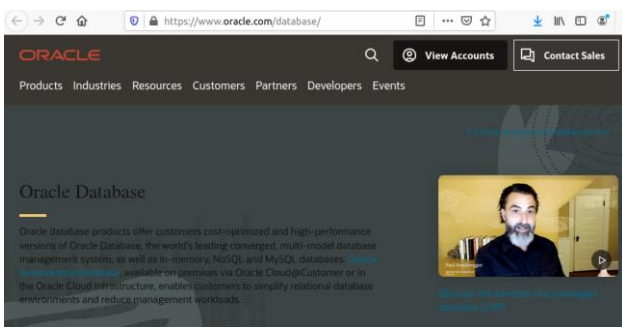


| id    | name | age | address |
|-------|------|-----|---------|
| dang  | 당당이  | 23  | 충북      |
| hong  | 홍길동  | 22  | 경기      |
| john  | 존벤이  | 28  | 강원      |
| ppuni | 이뿌니  | 30  | 서울      |

➔ 생성 확인 후 실제 웹 브라우저 테스트

➤ server 의 Mariadb 의 정보들이 winclient 브라우저에서 확인이 가능한 모습이다

## ● Oracle 설치 및 운영



➔ oracle database 다운로드

```
ubuntu@server:/oracledb/Disk1$
ubuntu@server:/oracledb/Disk1$ ls
oracle-xe-11.2.0-1.0.x86_64.rpm response upgrade
ubuntu@server:/oracledb/Disk1$
ubuntu@server:/oracledb/Disk1$
```

➔ 다운로드받은 oracle 파일 압축 해제

- 파일 확장명이 rpm 인 것을 확인 가능 (rpm 파일은 redhat 계열 리눅스)
- 해당 파일을 변환해야 사용 가능 (deb 파일로 변경)

[우분투는 데비안(Debian)계열]

```
ubuntu@server:/oracledb/Disk1$ sudo apt -y install alien libaio1 unixodbc
[sudo] ubuntu의 암호 :
패키지 목록을 읽는 중입니다 ... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다 ... 완료
패키지 libaio1는 이미 최신 버전입니다 (0.3.112-5).
libaio1 패키지는 수동설치로 지정합니다.
다음의 추가 패키지가 설치될 것입니다 :
autoconf automake autopoint autotools-dev binutils binutils-common
```

➔ alien / libaio1 / unixodbc 총 3 개의 패키지 설치

➤ alien = 파일 확장자 변환

➤ libaio , unixodbc = db,odbc 관련 패키지

```
ubuntu@server:/oracledb/Disk1$ sudo alien --scripts -d oracle*
oracle-xe_11.2.0-2_amd64.deb generated
ubuntu@server:/oracledb/Disk1$ ls -l
합계 579752
-rw-rw-r-- 1 root root 317320273 8월 29 2011 oracle-xe-11.2.0-1.0.x86_64.rpm
-rw-r--r-- 1 root root 276332728 3월 25 12:17 oracle-xe_11.2.0-2_amd64.deb
drwxr-xr-x 2 root root 4096 8월 29 2011 response
drwxrwxr-x 2 root root 4096 8월 29 2011 upgrade
ubuntu@server:/oracledb/Disk1$
```

➔ alien 명령어로 deb 파일 생성(변환) 완료

```
ubuntu@server:/oracledb/Disk1$
ubuntu@server:/oracledb/Disk1$ sudo dpkg --install oracle*.deb
Selecting previously unselected package oracle-xe.
(데이터베이스 읽는 중 ...현재 191984개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack oracle-xe_11.2.0-2_amd64.deb ...
Unpacking oracle-xe (11.2.0-2) ...
■
```

➔ dpkg 명령어로 oracle database install

```
ubuntu@server:/oracledb/Disk1$
ubuntu@server:/oracledb/Disk1$ sudo /etc/init.d/oracle-xe configure

Oracle Database 11g Express Edition Configuration
-----
This will configure on-boot properties of Oracle Database 11g Express
Edition. The following questions will determine whether the database should
be starting upon system boot, the ports it will use, and the passwords that
will be used for database accounts. Press <Enter> to accept the defaults.
Ctrl-C will abort.

Specify the HTTP port that will be used for Oracle Application Express [8080]:

Specify a port that will be used for the database listener [1521]:

Specify a password to be used for database accounts. Note that the same
password will be used for SYS and SYSTEM. Oracle recommends the use of
different passwords for each database account. This can be done after
initial configuration:
Password can't be null. Enter password:
Confirm the password:

Do you want Oracle Database 11g Express Edition to be started on boot (y/n) [y]:y
Starting Oracle Net Listener...Done
Configuring database...■
```

➔ configure 명령어로 /etc/init.d/oracle-xe 설정 진입

➤ password 설정 ('1234')

```
ubuntu@server:/oracledb/Disk1$ sudo systemctl status oracle-xe
● oracle-xe.service - SYSV: This is a program that is responsib
   Loaded: loaded (/etc/init.d/oracle-xe; generated)
   Active: active (exited) since Fri 2022-03-25 13:29:50 KST;
     Docs: man:systemd-sysv-generator(8)
   Process: 20877 ExecStart=/etc/init.d/oracle-xe start (code=
  3월 25 13:29:50 server systemd[1]: Starting SYSV: This is a pro
  3월 25 13:29:50 server oracle-xe[20877]: Oracle Database 11g E
  3월 25 13:29:50 server systemd[1]: Started SYSV: This is a pro
lines 1-9/9 (END)
```

➔ systemctl start / status 명령어로 서비스 실행/확인

```
fi
}
fi

./u01/app/oracle/product/11.2.0/xe/bin/oracle_env.sh
```

➔ /etc/bash.bashrc 파일 수정 (맨 밑줄 추가)

➤ ./u01/app/oracle/product/11.2.0/xe/bin/oracle\_env.sh 파일을 재부팅시에도 자동으로 실행시키기 위함

```
ubuntu@server:/oracledb/Disk1$ sudo ufw allow 8080
규칙이 추가되었습니다
규칙이 추가되었습니다 (v6)
ubuntu@server:/oracledb/Disk1$ sudo ufw allow 1521
규칙이 추가되었습니다
규칙이 추가되었습니다 (v6)
ubuntu@server:/oracledb/Disk1$
```

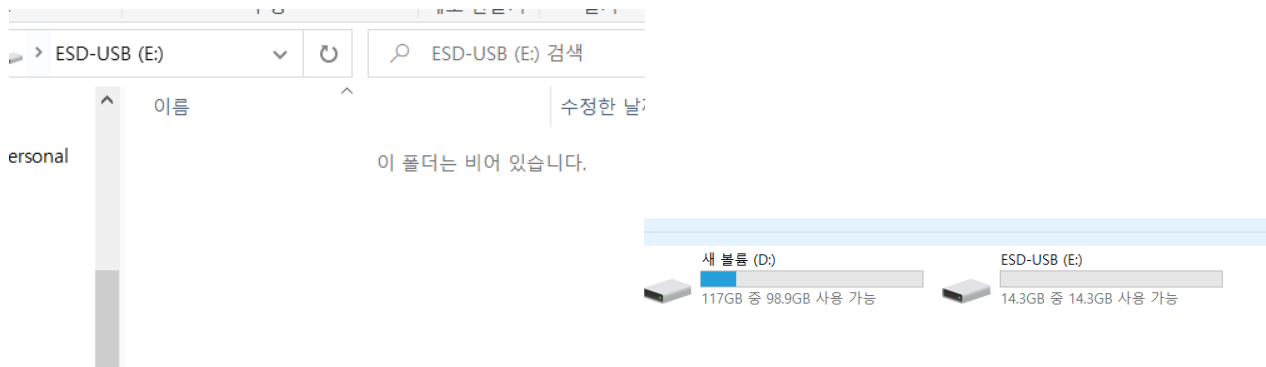
➔ ufw 명령어로 방화벽 포트 허용 (8080, 1521)

● 서버 간의 파일 전송 및 usb 마운트

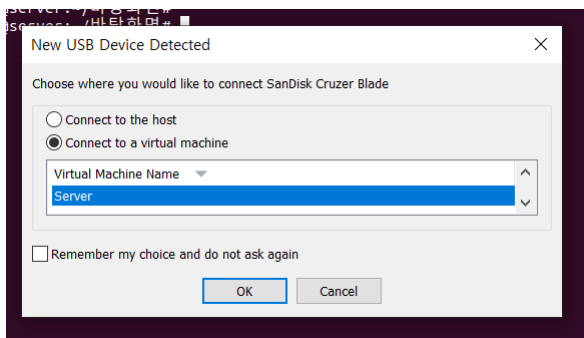
➤ 실습 내용



- server(1)에 usb 를 마운트한 뒤, server(2)의 nmon 분석 데이터를 scp 명령어로 파일 전송하여 usb 로 받아올 예정



➔ 비어있는 USB 상태 확인



```
Disk /dev/sdb: 14.33 GiB, 15376318464 bytes, 30031872 sectors
Disk model: Cruzer Blade
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xb2f83f57
```

➔ server(1) usb 연결 확인 // fdisk -l 명령어

```
root@server:~# mkdir /home/usb
root@server:~# mount -t vfat /dev/sdb1 /home/usb
```

➔ mount 시킬 디렉토리를 하나 생성한 뒤, usb 폴더와 mount

```
root@server-b:~# mkdir /home/nmon
root@server-b:~# nmon -f -s 10 -c 10 -m /home/nmon
```

➔ server(2) nmon 데이터를 생성할 디렉토리 생성 후 nmon 데이터 생성

```
root@server-b:~# ls -l /home/nmon
total 140
-rw-r--r-- 1 root root 141271 Mar 25 08:05 server-b_220325_0805.nmon
root@server-b:~# _
```

→ nmon 파일이 잘 생성되었다. 이 파일을 server1 에 mount 된 usb 로 전송시키겠다.

```
root@server:~# scp -r ubuntu@192.168.111.200:/home/nmon /home/usb
ubuntu@192.168.111.200's password:
server-b_220325_0805.nmon          100% 143KB 912.0KB/s   00:00
root@server:~# █
```

→ scp 명령어를 사용하여 server1 에서 server2 의 데이터를 가져옴

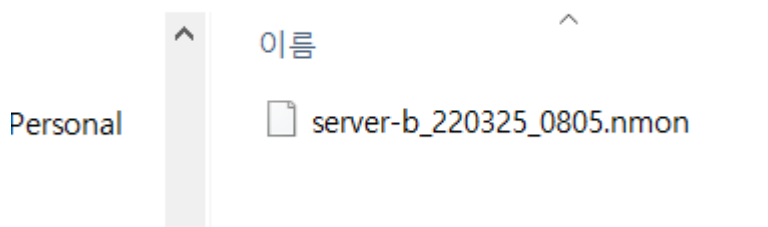
```
root@server:~# ls -l /home/usb
합계 16
drwxr-xr-x 2 root root 8192 3월 26 2022 'System Volume Information'
drwxr-xr-x 2 root root 8192 3월 25 17:11 nmon
root@server:~# ls -l /home/usb/nmon
합계 144
-rw-r--r-- 1 root root 145981 3월 25 17:11 server-b_220325_0805.nmon
root@server:~# █
```

→ usb 폴더에 잘 전송된 모습

```
root@server:~#
root@server:~# umount /home/usb
root@server:~# █
```

→ 작업이 마쳤다면 umount 로 마운트 해제

📁 > ESD-USB (E:) > nmon



→ usb 에 성공적으로 저장되어 다른 컴퓨터에서도 파일 확인이 가능한 모습

## ● SMBBA 서버 사용하여 파일 전송

```
root@server:~#
root@server:~# apt install samba -y
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음의 추가 패키지들이 설치될 것입니다:
```

→ samba 서버 설치

```
root@server:~# smbpasswd -a ubuntu
New SMB password:
Retype new SMB password:
Added user ubuntu.
root@server:~# █
```

➔ 삼바 계정 추가하기 (smbpasswd 명령어)

- 추가하려는 계정은 시스템이 존재하는 계정이여야 한다
- ubuntu / 1234 로 설정 완료

```
root@server-b:~# mkdir /home/backup
root@server-b:~# touch /home/backup/fileX
root@server-b:~# ls /home/backup/
fileX
```

➔ 공유할 디렉터리 생성 후 파일 생성

```
# Please note that you also need to add the path
# to the drivers directory for the share
; write list = root, @lpadmin
[backup]
comment = fileX in~
path = /home
read only = YES
valid users = ubuntu
```

➔ 삼바 설정 파일 확인 및 수정 (사용자별 공유자원 할당)

- /etc/samba/smb.conf 파일을 열고 수정한다.
- [backup] = 섹션을 정의하는 역할. 실제 디렉터리와는 다를 수 있음
- comment = 설명
- path = 공유 디렉터리의 경로
- read only = 공유폴더 안 데이터들을 읽을 수만 있고 수정 불가
- valid users = 공유 디렉터리를 이용할 수 있는 사용자

```
root@server-b:/etc/samba# service smbd restart
root@server-b:/etc/samba# hostname -i
127.0.1.1
root@server-b:/etc/samba# _
```

➔ 재시작 후 서버 ip 정보 확인