Database Server

RDBMS

- ❖ MariaDB
  - ✓ 설치
    - MariaDB Server: 패키지 이름은 mariadb-server

```
user1@myubuntu:~$ sudo apt-get install mariadb-server
[sudo] user1의 암호:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
(생략)
다음 새 패키지를 설치할 것입니다:
  galera-3 gawk libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl
  libjemalloc1 libsigsegv2 libterm-readkey-perl mariadb-client-10.1
  mariadb-client-core-10.1 mariadb-common mariadb-server mariadb-server-10.1
  mariadb-server-core-10.1 socat
0개 업그레이드, 16개 새로 설치, 0개 제거 및 0개 업그레이드 안 함.
24.1 M바이트 아카이브를 받아야 합니다.
이 작업 후 185 M바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까? [Y/n]
(생략)
user1@myubuntu:~$
```

RDBMS

- MariaDB
  - ✓ mariadb.service를 활성화
    - sudo systemctl start mariadb
    - sudo systemctl enable mariadb
  - ✓ 서비스 확인
    - systemctl status mariadb

```
mariadb.service - MariaDB 10.11.13 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: >
     Active: active (running) since Sun 2026-01-11 23:36:54 UTC; 5min ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 2023 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 9 (limit: 14598)
     Memory: 80.1M (peak: 83.0M)
        CPU: 267ms
     CGroup: /system.slice/mariadb.service
             └─2023 /usr/sbin/mariadbd
```

RDBMS

- MariaDB
  - ✓ sudo를 사용하여 root 권한으로 MariaDB에 접속: sudo mysql

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 6
Server version: 10.1.25-MariaDB-1 Ubuntu 17.10

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

  - ✓ MariaDB의 기본 프롬프트는 MairaDB [(none)]>
  - ✓ MariaDB에서 종료하려면 exit를 입력

RDBMS

❖ MariaDB

mysqladmin

기능	MariaDB 서버를 관리한다.
형식	mysqladmin 명령
명령	version : MariaDB 서버의 버전 정보를 출력한다. status : MariaDB 서버의 현재 상태 정보를 출력한다. password 암호 : 계정의 암호를 지정한다.
사용 예	mysqladmin password "123456"

RDBMS

- MariaDB
  - ✓ 데이터베이스 상태 확인

```
user1@myubuntu:~$ sudo mysqladmin status
Uptime: 2145   Threads: 1   Questions: 51   Slow queries: 0   Opens: 25   Flush
tables: 1  Open tables: 13  Queries per second avg: 0.023
user1@myubuntu:~$
```

RDBMS

- MariaDB
  - ✓ 데이터베이스 버전 확인

```
user1@myubuntu:~$ sudo mysqladmin version
mysqladmin  Ver 9.1 Distrib 10.1.25-MariaDB, for debian-linux-gnu on x86_64
Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Server version          10.1.25-MariaDB-1
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /var/run/mysqld/mysqld.sock
Uptime:                 37 min 42 sec

Threads: 1   Questions: 52   Slow queries: 0   Opens: 25   Flush tables: 1   Open
tables: 13  Queries per second avg: 0.022
user1@myubuntu:~$
```

RDBMS

- MariaDB
  - ✓ 관리자 비밀번호 변경
    - 변경: sudo mysqladmin -u root password 'wnddkd'
    - 변경된 비밀번호로 접속: sudo mysql -u root –p
  - ✓ 사용자 생성 및 권한 설정
    - 유저 생성: CREATE USER 'itstudy'@'%' IDENTIFIED BY 'wnddkd';
      - %는 모든 외부 IP로부터의 접속을 허용한다는 의미로 특정 IP에서만 접속하게 하려면 % 대신 '123.123.123.123'과 같이 IP를 적어주면 됨
    - 접속 권한 설정: GRANT ALL PRIVILEGES ON *.* TO 'itstudy'@'%';
      - *.*은 모든 데이터베이스와 모든 테이블에 대한 권한을 의미
    - 변경 사항 적용: FLUSH PRIVILEGES;
    - 변경된 비밀번호로 접속: sudo mysql -u itstudy -p

RDBMS

- MariaDB
  - ✓ 외부 접속 허용
    - 설정 파일에서 바인딩 부분 수정: sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

      bind-address = 0.0.0.0
    - 서비스 재시작: sudo systemctl restart mariadb
    - 방화벽에 3306 번 포트를 외부에서 허용하도록 설정: ufw allow 3306/tcp
    - 다른 Ubuntu Computer에서 수행
      - Client 설치: sudo apt install -y mysql-client
      - Client 확인: mysql --version
      - 외부에서 접속: mysql -h <원격_서버_IP_또는_호스트명> -P <포트번호> -u <사용자명> -p

RDBMS

- MariaDB
  - ✓ 백업과 복원
    - mysqldump -u [사용자 계정] -p [패스워드] [원본 데이터베이스명] > [생성할 백업 DB명].sql
    - mysql -u [사용자 계정] -p [패스워드] [복원할 DB] < [백업된 DB].sql

NoSQL

- ❖ Mongo DB
  - ✓ 설치
    - 패키지 업데이트

      sudo apt update && sudo apt install gnupg curl
    - MongoDB 공개 GPG 키 등록

      curl -fsSL https://www.mongodb.org/static/pgp/server-7.0.asc | ₩

      sudo gpg -o /usr/share/keyrings/mongodb-server-7.0.gpg ₩

      --dearmor
    - 저장소 리스트 추가

      echo "deb [arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-7.0.list

NoSQL

- ❖ Mongo DB
  - ✓ 설치
    - 설치

      sudo apt update

      sudo apt install -y mongodb-org
    - 서비스 시작 및 자동 실행 설정

      sudo systemctl start mongod

      sudo systemctl enable mongod

NoSQL

- ❖ Mongo DB
  - ✓ 외부 접속 허용
    - sudo nano /etc/mongod.conf

      net:

        port: 27017

        bindIp: 0.0.0.0 # 127.0.0.1에서 0.0.0.0으로 변경 (모든 IP 허용)
    - 서비스 재시작

      sudo systemctl restart mongod
    - 방화벽에서 포트 연결 허용

      sudo ufw allow 27017/tcp

NoSQL

- Mongo DB
  - ✓ 외부에서 접속
    - gnupg 및 curl 설치: sudo apt install gnupg curl
    - GPG 키 등록: curl -fsSL https://www.mongodb.org/static/pgp/server-7.0.asc | sudo gpg -o /usr/share/keyrings/mongodb-server-7.0.gpg --dearmor --yes
    - 저장소 등록: echo "deb [arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-7.0.gpg] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/7.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-7.0.list
    - 저장소 정보 업데이트: sudo apt update
    - mongosh 설치: sudo apt install -y mongodb-mongosh
    - 인증없이 접속: mongosh --host <원격_IP> --port 27017

NoSQL

- ❖ Mongo DB
  - ✓ 사용자 인증
    - 몽고 쉘 접속

      mongosh
    - 계정 생성
      - use admin
      - db.createUser({ user: "adminUser", pwd: "password123", roles: [{ role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase"]})
    - /etc/mongod.conf 파일을 열어서 인증 모드 사용 설정 코드 추가

      security:

        authorization: enabled
    - 서비스 재시작

      sudo systemctl restart mongod

NoSQL

- ❖ Mongo DB
  - ✓ 사용자 인증
    - ● 외부에서 인증없이 접속
      - ◆ mongosh --host 192.168.0.100 --port 27017

```
Current Mongosh Log ID:         69e5c87f808a5c33a73d88b2
Connecting to:
        mongodb://192.168.0.100:27017/?directConnection=true&appName=mongosh+2.8.2
Using MongoDB:                  7.0.31
Using Mongosh:                  2.8.2

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
```

      - ◆ test> show dbs

```
MongoServerError[Unauthorized]: Command listDatabases requires authentication
```

NoSQL

- ❖ Mongo DB
  - ✓ 사용자 인증
    - ● 외부에서 인증과 함께 접속
      - ◆ mongosh --host 192.168.0.100 --port 27017 -u adminUser -p password123

```
Current Mongosh Log ID:         69e5ca090d218bf0b33d88b2
Connecting to:
            mongodb://<credentials>@192.168.0.100:27017/?directConnecti
    on=true&appName=mongosh+2.8.2
Using MongoDB:                  7.0.31
Using Mongosh:                  2.8.2
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
```

      - ◆ test> show dbs

```
admin   132.00 KiB
config  108.00 KiB
local    72.00 KiB
```

NoSQL

- ❖ Redis
  - ✓ 설치
    - sudo apt update
    - sudo apt install redis-server -y
  - ✓ 상태 확인
    - sudo systemctl status redis-server
  - ✓ 로컬에서 접속
    - redis-cli

      127.0.0.1:6379>

NoSQL

- ❖ Redis
  - ✓ 인증
    - 설정 파일 수정: sudo nano /etc/redis/redis.conf

      # bind 127.0.0.1 ::1 <- 주석 처리하거나

      bind 0.0.0.0 # <- 모든 IPv4 접속 허용

      requirepass 비밀번호
    - 서비스 재시작

      sudo systemctl restart redis-server
    - 방화벽 설정

      sudo ufw allow 6379/tcp

NoSQL

- Redis
  - ✓ 인증
    - 로컬에서 확인

      redis-cli

      > AUTH 비밀번호

      > PING

      # 결과로 PONG이 나오면 성공
    - 외부에서 접속

      redis-cli -h IP주소 -p 포트번호 -a 비밀번호
    - Production 환경
      - 특정 IP만 허용: 가능하면 bind 0.0.0.0보다는 접속할 클라이언트의 특정 IP만 허용하는 것이 안전
      - 포트 변경: 기본 포트인 6379 대신 다른 포트를 사용하는 것도 자동화된 공격을 피하는 좋은 방법