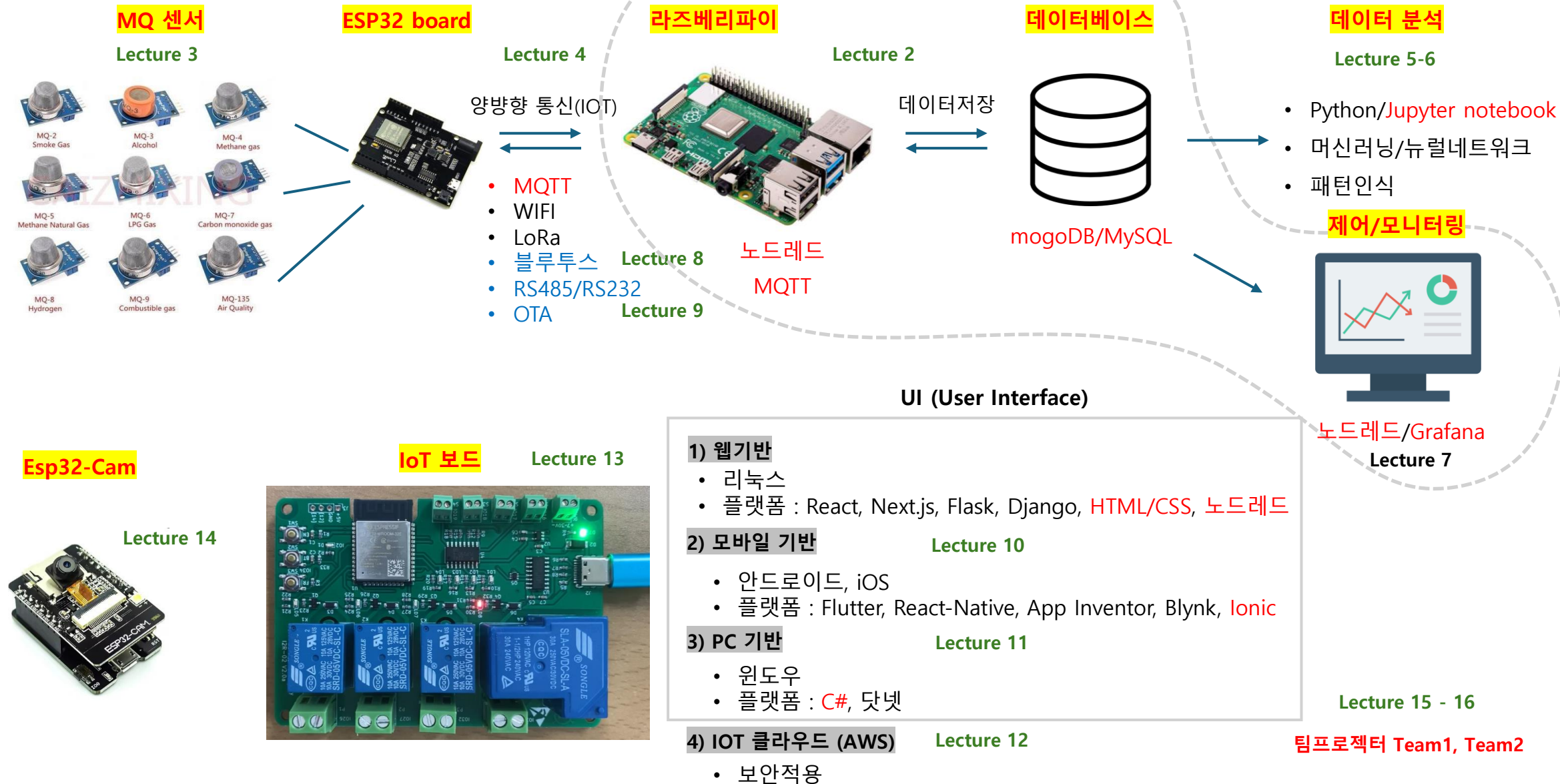


# 강의 Flow & Plan



1) Raspberry Pi에 라즈비안 설치

- SSH 접속/리눅스제어

2) User Account 생성/제거

3) Docker 설치 후 개정 권한 부여

4) Node-Red, mongoDB, MySQL, MQTT 서버 설치

5) VI editor 사용법 익히기

## 1. Raspberry Pi에 Rasbian 설치

<https://www.raspberrypi.com/software/>



라즈베리파이(Raspberry Pi) 디바이스

X

No filtering

Show every possible image



Raspberry Pi 5

The latest Raspberry Pi, Raspberry Pi 5



Raspberry Pi 4

Models B, 400, and Compute Modules 4, 4S



Raspberry Pi Zero 2 W

The Raspberry Pi Zero 2 W

운영체제

X



Raspberry Pi OS (64-bit)

A port of Debian Bookworm with the Raspberry Pi Desktop (Recommended) (Recommended)

릴리즈: 2024-07-04

온라인 - 1.1 GB 다운로드



Raspberry Pi OS (32-bit)

A port of Debian Bookworm with the Raspberry Pi Desktop

릴리즈: 2024-07-04

온라인 - 1.2 GB 다운로드



Raspberry Pi OS (other)

Other Raspberry Pi OS based images



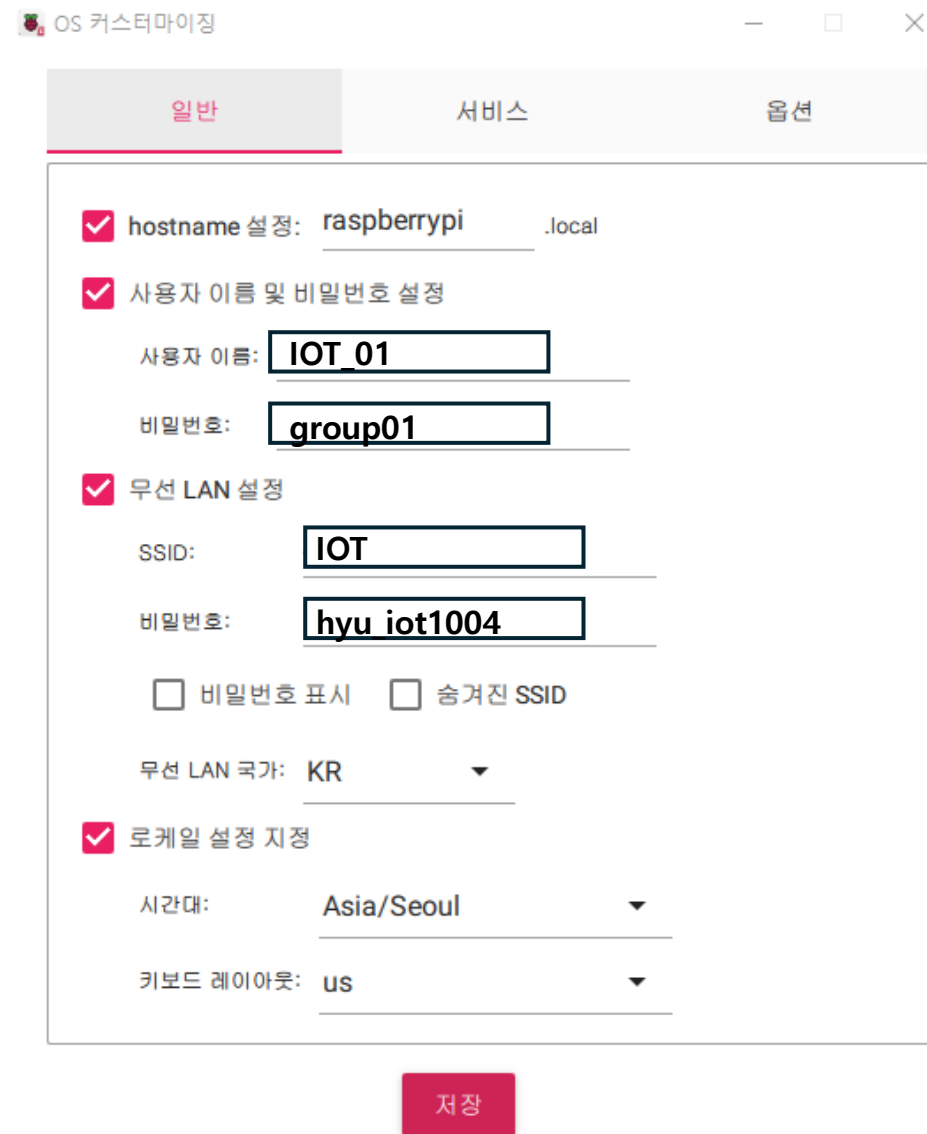
Other general-purpose OS

Other general-purpose operating systems



Media player OS





일반

서비스

옵션

☒ SSH 사용

☒ 비밀번호 인증 사용

☐ 공개 키만 인증 허용

'pi' 인증키 설정:

SSH-KEYGEN 실행

저장

일반

서비스

옵션

☒ 완료되면 효과음으로 알림

☒ 완료되면 미디어 꺼내기

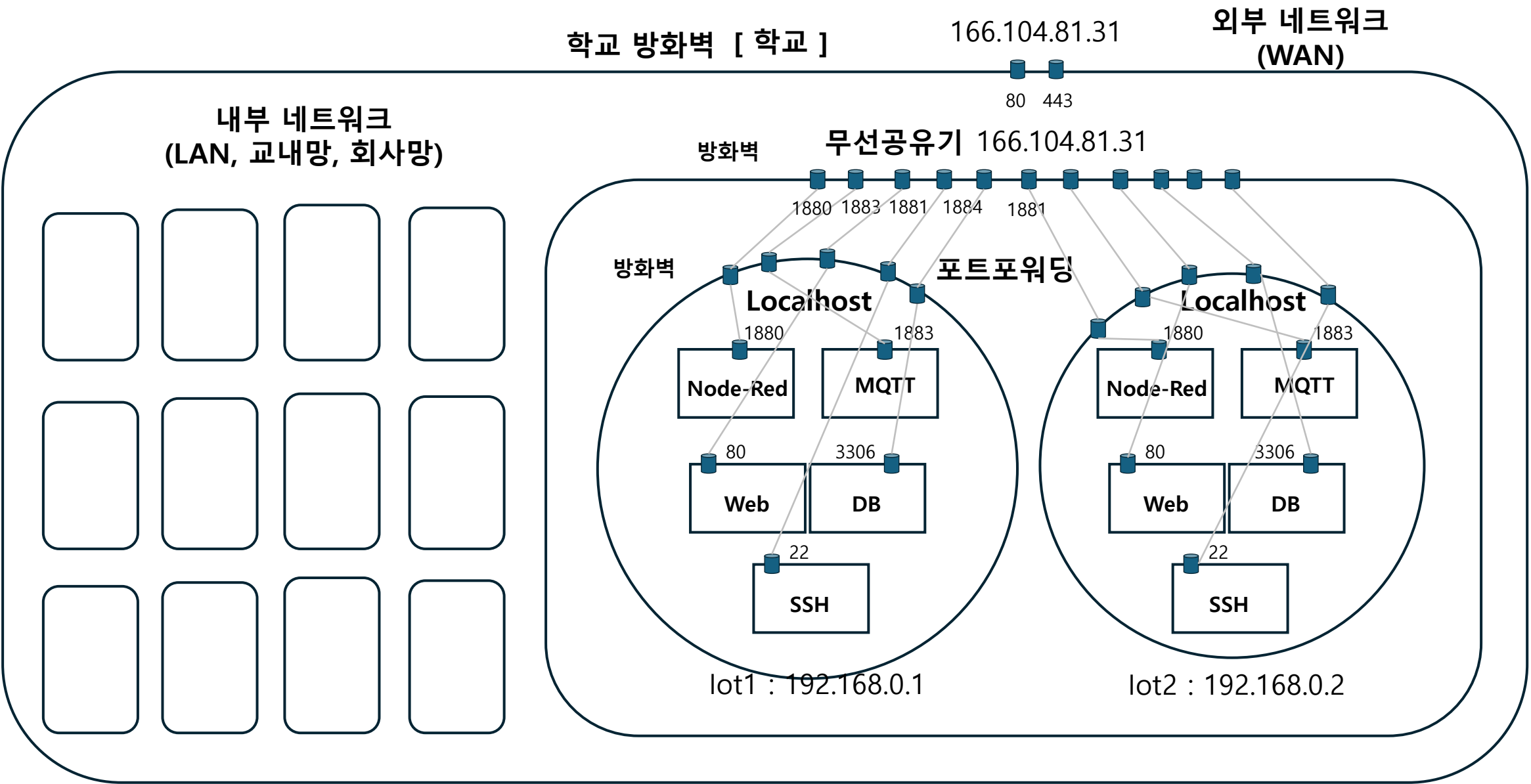
☒ 이미지 다운로드 통계 관하여 정보 수집 허용

저장



USB SD카드

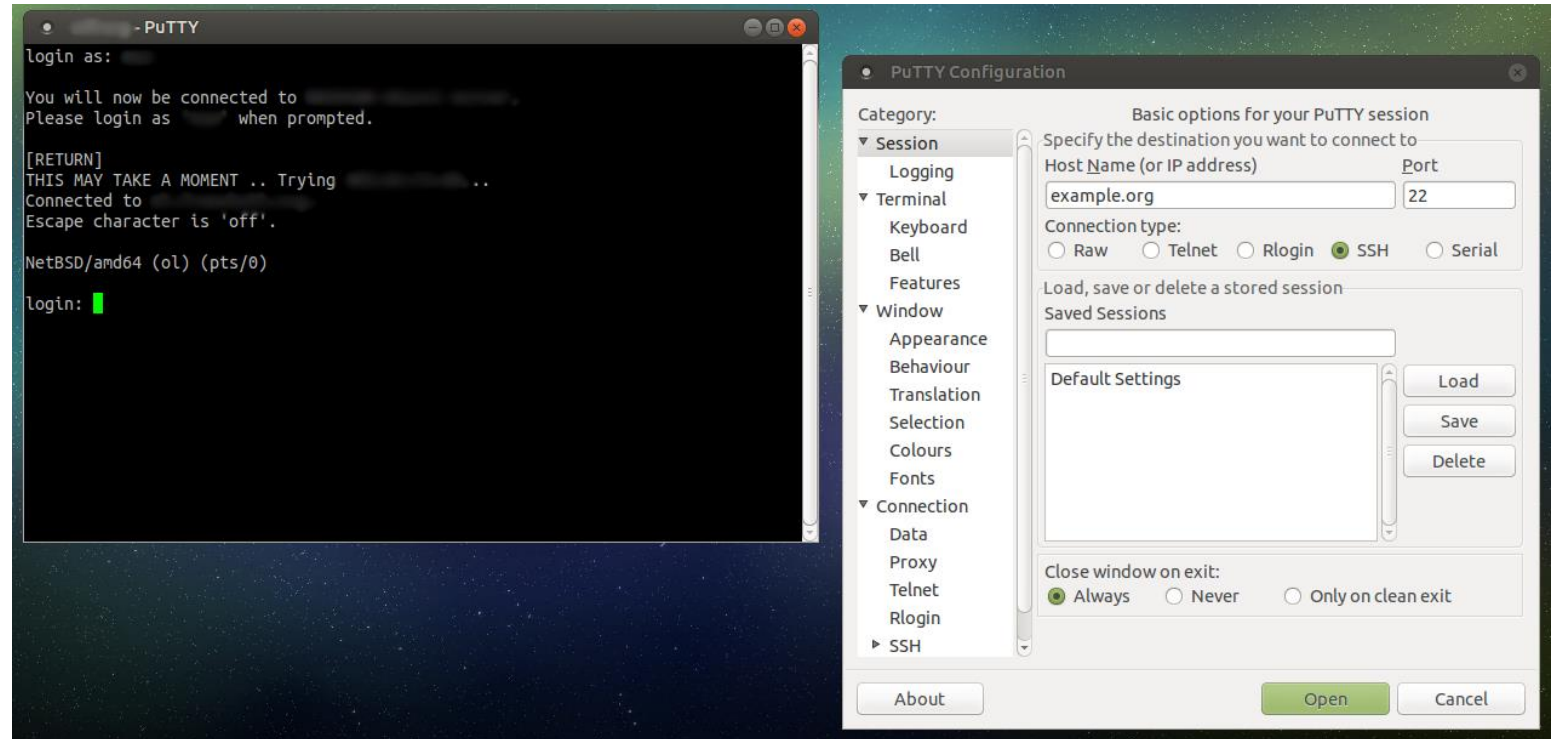
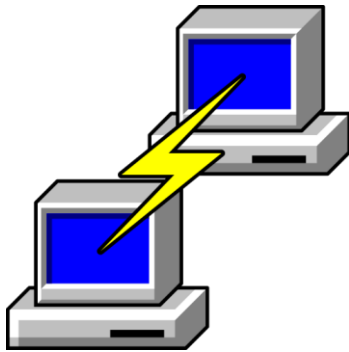
네트워크 구조





## [1.1] SSH 접속 제어

<https://www.putty.org/>



## 2. 사용자 계정 만들기

- 계정 생성 : `sudo adduser {name}`
- 계정 제거 : `sudo deluser --remove-all {name}`

## 1. 파일 및 디렉토리 관련 명령어

- **ls**: 현재 디렉토리의 파일 및 디렉토리 목록을 표시
  - ls: 기본 파일 목록 표시
  - ls -l: 자세한 정보와 함께 목록 표시
  - ls -a: 숨김 파일 포함 목록 표시
- **cd**: 디렉토리 이동
  - cd /path/to/directory: 지정한 디렉토리로 이동
  - cd.. : 상위 디렉토리로 이동
  - cd ~: 홈 디렉토리로 이동
- **pwd**: 현재 작업 중인 디렉토리 경로 출력
  - pwd: 현재 디렉토리의 전체 경로 출력
- **mkdir**: 새 디렉토리 생성
  - mkdir directory\_name: 디렉토리 생성
- **rmdir**: 빈 디렉토리 삭제
  - rmdir directory\_name: 디렉토리 삭제
- **rm**: 파일 또는 디렉토리 삭제
  - rm filename: 파일 삭제
  - rm -r directory\_name: 디렉토리 및 그 내부 파일/디렉토리 삭제
- **cp**: 파일 복사
  - cp source\_file destination\_file: 파일 복사
  - cp -r source\_directory destination\_directory: 디렉토리 복사
- **mv**: 파일 또는 디렉토리 이동/이름 변경
  - mv source\_file destination\_file: 파일 이름 변경 또는 이동

## 2. 파일 내용 관련 명령어

- **cat**: 파일 내용 출력
  - cat filename: 파일 내용 출력
- **more / less**: 파일 내용 페이지 단위로 보기
  - more filename: 파일 내용 표시(스크롤 가능)
  - less filename: more보다 많은 기능 제공, 스크롤 가능
- **head**: 파일의 첫 부분만 출력
  - head -n 10 filename: 파일의 처음 10줄 출력
- **tail**: 파일의 마지막 부분만 출력
  - tail -n 10 filename: 파일의 마지막 10줄 출력
  - tail -f filename: 파일의 마지막 부분을 실시간으로 출력 (로그 파일 모니터링에 유용)

### 3. 시스템 관리 관련 명령어

- **sudo**: 관리자 권한으로 명령 실행
  - `sudo command`: 해당 명령을 관리자 권한으로 실행
- **chmod**: 파일 권한 변경
  - `chmod 755 filename`: 파일 권한 변경 (읽기/쓰기/실행 권한 설정)
- **chown**: 파일 소유자 변경
  - `chown user:group filename`: 파일 소유자 및 그룹 변경
- **df**: 디스크 사용량 확인
  - `df -h`: 디스크 사용량을 사람이 읽기 쉬운 형식으로 표시
- **du**: 디렉토리 크기 확인
  - `du -h`: 디렉토리 크기를 사람이 읽기 쉬운 형식으로 표시
- **top**: 시스템 프로세스 모니터링
  - `top`: 현재 실행 중인 프로세스와 시스템 리소스 사용량 표시
- **ps**: 현재 실행 중인 프로세스 목록 출력
  - `ps aux`: 시스템 전체 프로세스 목록 출력
- **kill**: 프로세스 종료
  - `kill PID`: 지정한 PID의 프로세스 종료
  - `kill -9 PID`: 강제 종료

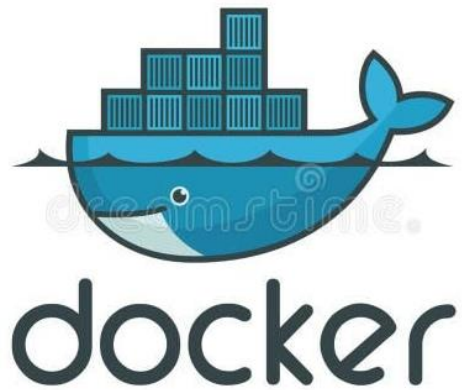
### 4. 네트워크 관련 명령어

- **ping**: 네트워크 연결 확인
  - `ping example.com`: 대상 서버로 패킷을 보내 연결 상태 확인
- **ifconfig**: 네트워크 인터페이스 설정 확인 및 변경
  - `ifconfig`: 네트워크 설정 정보 표시
- **netstat**: 네트워크 상태 확인
  - `netstat -an`: 네트워크 연결 및 포트 상태 표시

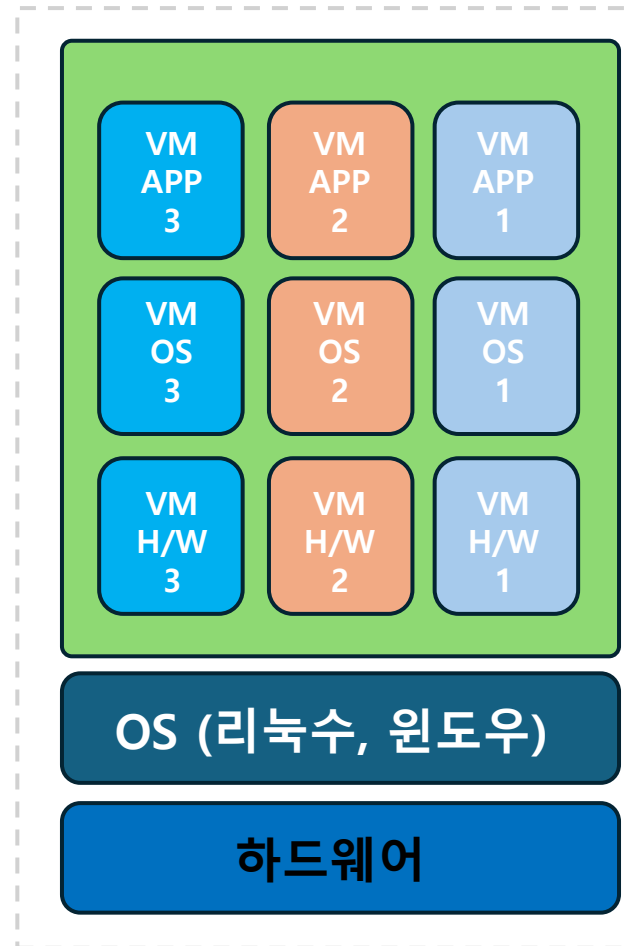
### 5. 패키지 관리 명령어 (Ubuntu)

- **apt-get update**: 패키지 목록 업데이트
- **apt-get upgrade**: 설치된 패키지 업그레이드
- **apt-get install package\_name**: 새로운 패키지 설치
- **apt-get remove package\_name**: 패키지 삭제

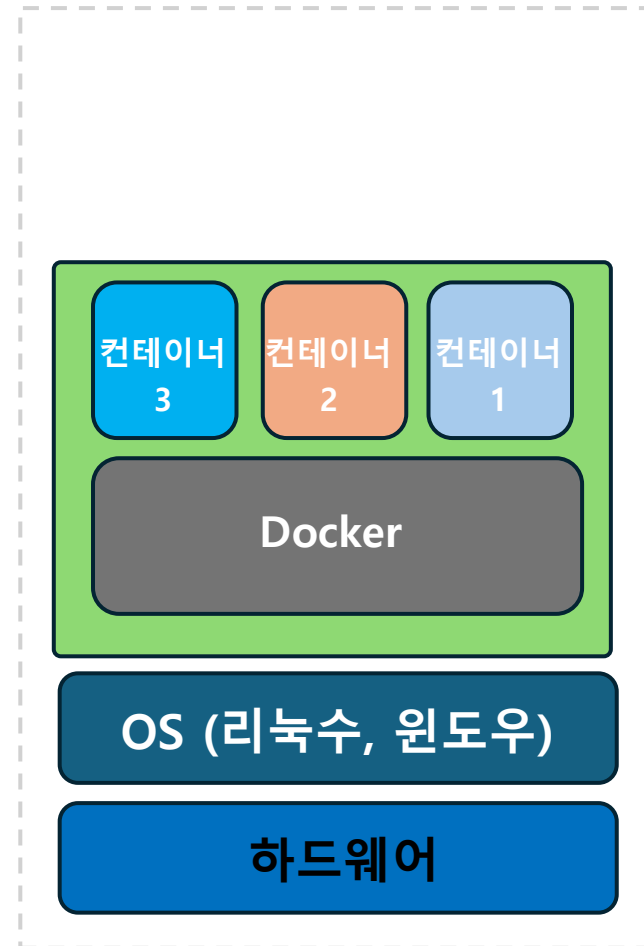
# Docker 설치



Virtual Machine



Docker



## Docker의 장점

- **격리된 환경:** 각 컨테이너는 다른 컨테이너와 독립적인 환경을 제공하므로, 애플리케이션 간 충돌을 방지할 수 있다.
- **이식성:** Docker 이미지를 사용하면 동일한 환경을 어디서나 재현할 수 있어, 개발 환경과 운영 환경 간의 차이로 인한 문제를 줄일 수 있다.
- **효율성:** 컨테이너는 가상 머신보다 가볍고 빠르며, 호스트의 리소스를 더 효율적으로 사용할 수 있다.
- **빠른 배포:** 컨테이너는 빠르게 생성 및 삭제가 가능하므로, 애플리케이션 배포와 스케일링이 매우 용이하다.

# Docker 설치

## # docker 공식 GPG 키 추가

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o
/etc/apt/keyrings/docker.asc sudo chmod a+r /etc/apt/keyrings/docker.asc
```

## # 리파지토리 추가

```
echo \ "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian \ $(. /etc/os-release && echo "$VERSION_CODENAME") stable" |
\ sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

## # 최신버전 설치

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

## # 부팅시 docker 자동 실행

```
sudo systemctl enable docker
```

# Docker 실행

## # sudo 없이 docker 사용하기

```
sudo usermod -aG docker ${USER}
groups ${USER}
```

## # Dockerhub 에서 이미지 가져오기

Dockerhub Site : <https://hub.docker.com/>  
docker image pull {image name:tag}

## # 다운로드된 docker image 확인

```
docker images
```

## # 설치된 Container 확인

```
docker ps -a
```

## # 설치된 Container 삭제

```
docker rm -f ${container}
```

## # Container 실행

```
docker run
    --name {container name}
    -e {ENVIRONMENT=password}
    -p {port:port}
    -v {volum:volume}
    -i 컨테이너 생성
    -t 시작, 접속
    -d {background 실행}
    {container:tag}
```

## # Container 시작

```
docker start ${container}
```

## # Container 멈춤

```
docker stop ${container}
```

## # Container 접속

```
docker exec -it ${container} /bin/bash
```



# Container Application in Docker



## Node-Red 서버

```
$ docker run -itd -p 1880:1880 -v myNodeREDdata:/data --name  
mynodered nodered/node-red
```



## Mysql 서버

<https://dev.mysql.com/downloads/installer/>

```
$ docker run -itd -p 3306:3306 --name some-mysql -e  
MYSQL_ROOT_PASSWORD=my-secret-pw mysql:tag
```



## monoDB 서버

```
$ docker run -itd -p 27017:27017 --name mongodb mongodb/mongodb-  
community-server:latest
```



## MQTT 서버

```
$ docker run -it -p 1883:1883 -p 9001:9001 -v  
mosquitto.conf:/mosquitto/config/mosquitto.conf eclipse-mosquitto
```

# 3) VI editor 사용법 익히기

## 1. vi 에디터 모드

- **명령 모드 (Command Mode):** vi 에디터가 기본적으로 시작되는 모드로, 커서를 이동하거나 명령어를 실행할 수 있습니다.
- **입력 모드 (Insert Mode):** 텍스트를 입력할 수 있는 모드입니다. 명령 모드에서 i, a, o 등을 눌러서 입력 모드로 전환할 수 있습니다.
- **명령 입력 모드 (Command-Line Mode):** 명령을 실행할 수 있는 모드입니다. 명령 모드에서 :를 눌러 명령어를 입력할 수 있습니다.

## 2. 모드 전환

- **명령 모드 → 입력 모드:**
  - i: 현재 커서 위치에서 입력 시작
  - a: 현재 커서 다음 위치에서 입력 시작
  - o: 커서 아래 새로운 줄을 추가하고 입력 시작
- **입력 모드 → 명령 모드:** Esc 키를 눌러 입력 모드에서 명령 모드로 전환
- **명령 모드 → 명령 입력 모드:** :를 눌러 명령 입력 모드로 전환

## 3. 명령 모드에서의 주요 명령어

### 커서 이동

- h: 왼쪽으로 한 칸 이동
- j: 아래로 한 줄 이동
- k: 위로 한 줄 이동
- l: 오른쪽으로 한 칸 이동
- w: 다음 단어의 시작으로 이동
- b: 이전 단어의 시작으로 이동
- 0: 현재 줄의 맨 앞으로 이동
- \$: 현재 줄의 끝으로 이동
- G: 파일의 마지막 줄로 이동
- gg: 파일의 첫 번째 줄로 이동
- Ctrl + f: 한 화면 아래로 스크롤
- Ctrl + b: 한 화면 위로 스크롤

### 텍스트 편집

- x: 커서 위치의 문자 삭제
- dw: 커서 위치에서 단어 삭제
- dd: 현재 줄 삭제
- D: 커서 위치부터 줄 끝까지 삭제
- u: 마지막 명령 취소 (Undo)
- Ctrl + r: 취소한 명령 다시 실행 (Redo)
- p: 커서 다음 위치에 복사/잘라내기한 내용 붙여넣기
- y: 현재 줄 복사 (yank)
- yy: 줄 복사
- cw: 단어 수정 (현재 단어 삭제 후 입력 모드로 전환)
- r: 커서 위치의 문자를 다른 문자로 교체

### 텍스트 검색 및 치환

- /search\_term: 파일에서 search\_term 검색
  - 검색 후 n을 눌러 다음 결과로 이동, N을 눌러 이전 결과로 이동
- ?search\_term: 파일에서 search\_term을 위쪽 방향으로 검색
- :s/old/new/: 현재 줄에서 old를 new로 치환
- :%s/old/new/g: 파일 전체에서 old를 new로 모두 치환

## 4. 명령 입력 모드에서의 주요 명령어

- :w: 파일 저장
- :q: vi 종료
- :wq 또는 ZZ: 저장하고 종료
- :q! : 저장하지 않고 강제로 종료
- :e filename: 다른 파일 열기
- :set number: 줄 번호 표시
- :set nonumber: 줄 번호 숨기기
- :w filename: 다른 이름으로 파일 저장

## 5. 빠른 입력 모드 전환

- A: 커서가 위치한 줄의 끝에서 입력 모드 시작
- I: 커서가 위치한 줄의 시작에서 입력 모드 시작
- O: 커서 위에 새로운 줄을 추가하고 입력 모드 시작

## 6. 매크로 사용

- q[a-z]: 매크로 녹화 시작 (a에서 z까지의 레지스터에 저장)
- 매크로 실행할 명령어들 입력
- q: 매크로 녹화 종료
- @[a-z]: 저장된 매크로 실행