OSI7 layer or TCP/IP

client

세션+표현+응용 ( 웹어플리케이션 외부와 통신 process ) chrome 실행후 naver 접속

전송 ( port ) 59321

네트워크 ( IP 주소 ) 10.3.4.123

<->

dns 서버에 naver ip 주소 -> 223.130.200.107

<->

server

네트워크 ( 공인 IP 주소 ) 223.130.200.107

전송 (TCP/UDP )( port ) 80

만약에 사설 IP 서버로 구축 (포트포워딩 테이블 작성)

(사설ip:포트) 192.168.0.9:8080

방화벽 allow (192.168.0.9:8080)

server process

ip 주소(domain name) + 포트번호(80)

https(443) or http(80) 웹서비스 프로토콜

stramlit

127.0.0.1:8080

127.0.0.1:80

127.0.0.1:9999

www.google.com:443

만약 ifconfig 가 안되는 경우 apt-get install net-tools 로 설치

service docker start

터미널 두개

# 터미널 1(server)

sudo docker start test-ubuntu

sudo docker attach test-ubuntu

apt-get update

apt-get install python3

# ifconfig 로 ip 확인 172. ~~~~ (172.17.0.2)

python3

# 터미널 2(client)

sudo docker start test-ubuntu2

sudo docker attach test-ubuntu2

apt-get update

apt-get install python3

# ifconfig 로 ip 확인 172. ~~~~ (172.17.0.3)

python3

# python 을 이용해 소캣 통신을 할거임 python3 설치

```

1. 서버 : from socket import \*

2. 서버 :svrsock = socket(AF\_INET, SOCK\_DGRAM)

3. 서버 : svrsock.bind(('localhost',5000))

4. 클라이언트 : from socket import \*

5. 클라이언트 : csock = socket(AF\_INET, SOCK\_DGRAM)

6. 클라이언트 : csock.sendto('HI'.encode(), ('localhost',5000))

7. 서버 : s, addr =svrsock.recvfrom(1024) # 데이터와 발신 주소를 함께 얻는다. 1024는 버퍼 크기이다.

8. 서버 : s

9. 서버 : svrsock.sendto('Hello'.encode(), addr) # 메시지와 수신 주소와 포트를 입력한다.

10. 클라이언트 : s, addr=csock.recvfrom(1024)

11. 클라이언트 : s

# 소캣 종료

svrsock.close()

```

##################################################################

집에서 server를 구축한다.( 웹서버, jupyter server, ....)

1) 집에서 구축한 server의 ip 주소 확인

ip 가 공인 : 그냥 사용

ip 가 사설 : 공유기의 포트포워딩 테이블 작성

외부에 노출시킬 port, 내부의 ip, 내부의 port

2) server

방화벽 allow (사설,공인)ip, port

service 가동

streamlit 서버를 docker container 로 만들고 외부에 노출