

# Deployment

충남대학교 데이터네트워크 연구실

# 중간고사 관련

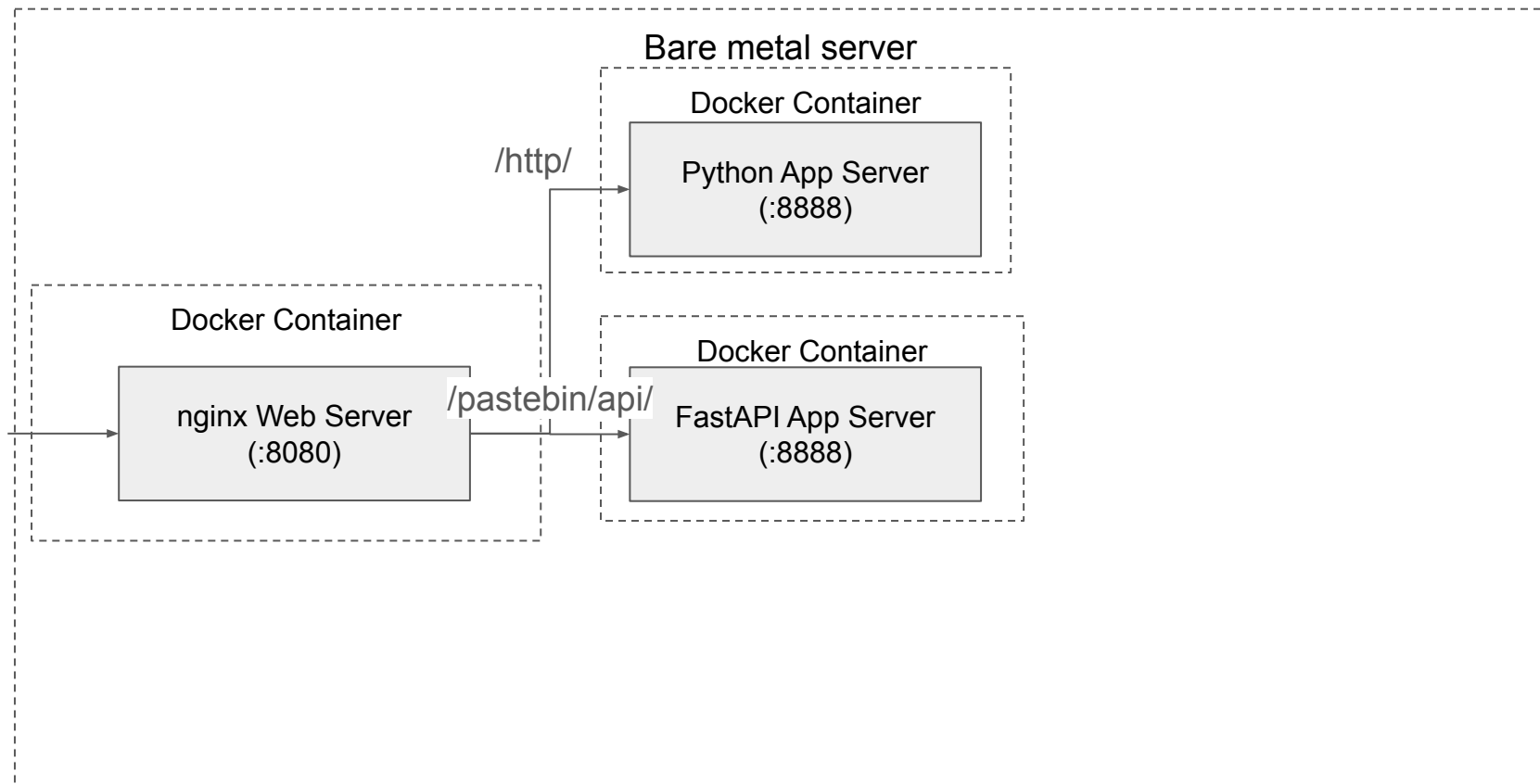
성적 공개 : 11/3 (월)

성적 관련 문의 : 연구실 방문 (11/3 ~ 11/12(수))후 문의

메일로 사전 연락후 방문!

# Python 3 HTTP Server on Docker

# 그림으로 보는 오늘의 서비스 구조



# 첨부파일 활용! (실습5 내용)



```
1 <!DOCTYPE html>$
2 <html>$
3   <head>$
4     <meta charset="utf-8">$
5     <link href="style.css" rel="stylesheet" />$
6     <title>내 TCP 웹 서버 테스트</title>$
7   </head>$
8   <body>$
9     <h1>202301234 성이름</h1>$
10    <h2></h2>$
11    <img id="jsimage" src="" width="50%" height="auto">$
12  $
13    <script src="main.js"></script>$
14  </body>$
15 </html>$
```

# Docker network: mybridge 생성

- `docker network create --driver=bridge mybridge`
- mybridge 라는 이름의 docker bridge network 생성
  - 이 network 에 있는 container 끼리는 IP, container name 으로 통신 가능

```
ubuntu@ip-172-31-3-160:~/11-internalrouting/http_server$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
e8189dbb1859        bridge             bridge             local
937cbb9889d3        host              host              local
d609ec1d5ae3        none             null              local
ubuntu@ip-172-31-3-160:~/11-internalrouting/http_server$ docker network create -
-driver=bridge mybridge
a0ac598dc6199b13e8d44ae775539678f5d9fccd971fb2acdea2fd2773dcc691
ubuntu@ip-172-31-3-160:~/11-internalrouting/http_server$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
e8189dbb1859        bridge             bridge             local
937cbb9889d3        host              host              local
a0ac598dc619        mybridge           bridge             local
d609ec1d5ae3        none             null              local
ubuntu@ip-172-31-3-160:~/11-internalrouting/http_server$
```

# Ubuntu Docker Image [\[링크\]](#)

- `docker pull ubuntu:24.04`

```
ubuntu@ip-172-31-3-160:~/11-internalrouting/http_server$ docker pull ubuntu:24.04
24.04: Pulling from library/ubuntu
9c704ecd0c69: Already exists
Digest: sha256:2e863c44b718727c860746568e1d54afd13b2fa71b160f5cd9058fc436217b30
Status: Downloaded newer image for ubuntu:24.04
docker.io/library/ubuntu:24.04
```

# Docker run

- `docker run -itd --name myhttp --network mybridge -v [HTTP_SERVER_PATH]:/media ubuntu:24.04`

```
ubuntu@ip-172-31-4-155:~/ComputerNetwork2024/Week11-Deployment$ docker run --int  
eractive --tty --detach --name myhttp --network mybridge --volume /home/ubuntu/C  
omputerNetwork2024/Week11-Deployment/python3_http:/media ubuntu:24.04  
00df16eaac2495e34b964e95b5a271c5de38991fbafa008085717185d661aba0
```



# Ubuntu container Python 3 실행 환경 설정

```
# docker exec -it myhttp bash
```

```
$ apt update
```

```
$ apt install net-tools vim python3-dev python3-venv python3-pip
```

```
root@05f1129f05d2:~# python
bash: python: command not found
root@05f1129f05d2:~# python3
bash: python3: command not found
root@05f1129f05d2:~# apt update
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1192
kB]
root@dc7dcf02b625:/# apt install python3-dev python3-venv python3-pip net-tools
vim
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

# Execute TCP HTTP Server inside container (Linux)

```
$ cd /  
$ python3 -m venv venv  
$ source venv/bin/activate  
(venv) $ cd /media  
(venv) $ python3 http_server.py
```

```
root@00df16eaac24:~# cd /  
root@00df16eaac24:/# ls  
bin          dev          lib          mnt          root         sbin.usr-is-merged  tmp  
bin.usr-is-merged  etc          lib64        opt          run          srv                usr  
boot         home         media        proc         sbin         sys                var  
root@00df16eaac24:/# python3 -m venv venv  
root@00df16eaac24:/# source venv/bin/activate  
(venv) root@00df16eaac24:/# cd /media/  
(venv) root@00df16eaac24:/media# python3 tcp_http_server_answer.py --debug  
Start server  
█
```

하지만 아직 접속 안 됨... 왜?

# Deployment using nginx

# nginx on Docker

- nginx container 를 띄우며 설정!

해야할 동작 (우리가 세팅할 것)

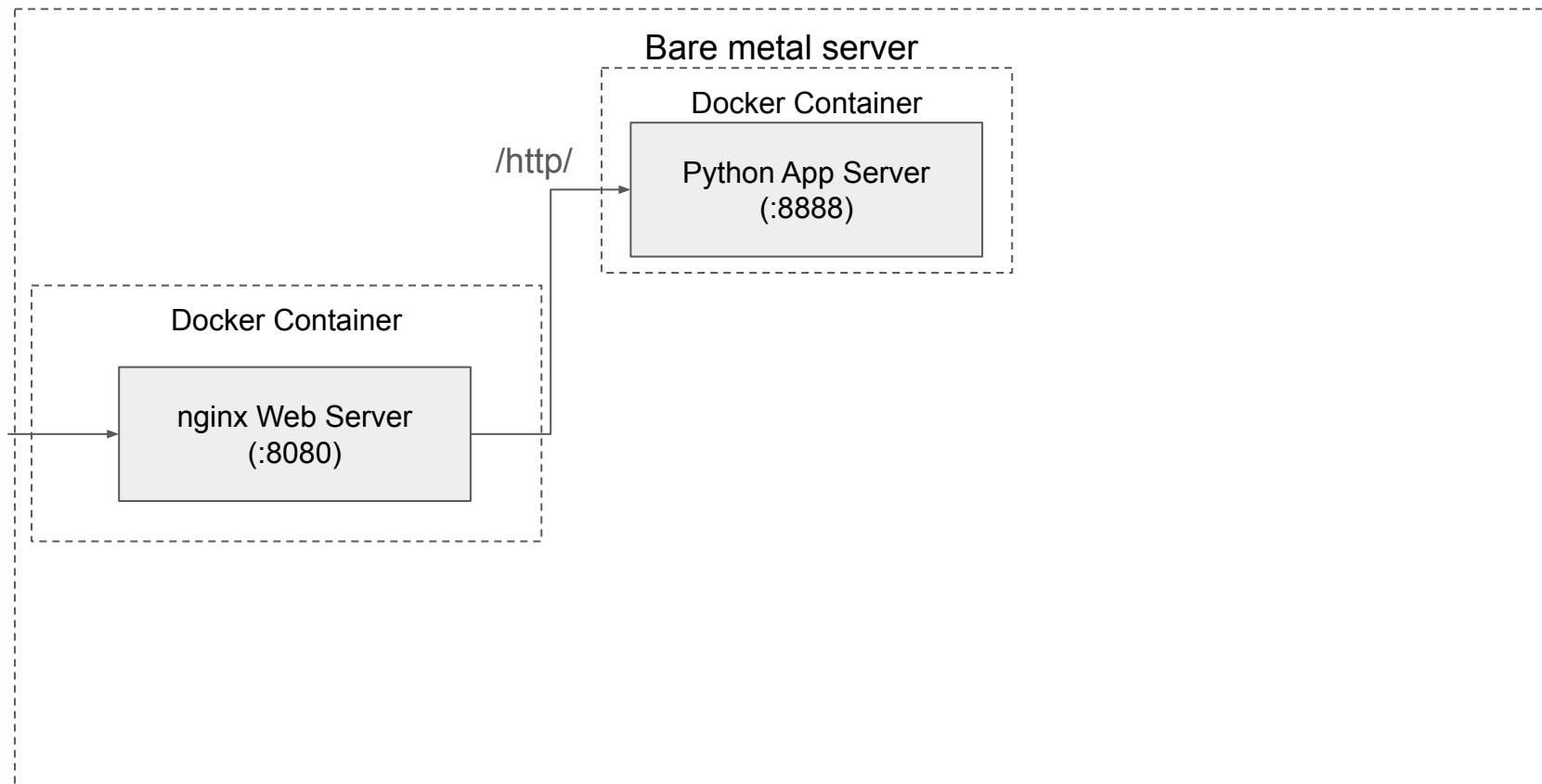
- nginx 로 들어온 요청을 App server 로 전달 (reverse proxy)
- App server 의 응답을 nginx 에게 전달 (HTTP/RESTful 등)
- nginx 가 클라이언트에게 응답
  - Header 가 바뀐!
  - 데이터만 전송함



```
Connected by ('172.18.0.3', 48548)
Received: GET / HTTP/1.0
Connected by ('172.18.0.3', 48554)
Received: GET /style.css HTTP/1.0
Connected by ('172.18.0.3', 48568)
Received: GET /main.js HTTP/1.0
Connected by ('172.18.0.3', 48582)
Received: GET /chacha.png HTTP/1.0
```

▼ Response Headers	
	Raw
Connection:	keep-alive
Content-Length:	957
Content-Type:	text/html; charset=utf-8
Date:	Wed, 13 Nov 2024 04:58:23 GMT
Server:	nginx/1.27.2

## 그림으로 보는 지금 설정할 구조



# docker inspect myhttp

- Container의 IP Address 확인

```
ubuntu@ip-172-31-4-155:~$ docker container inspect myhttp | tail -n 15
    "Gateway": "172.18.0.1",
    "IPAddress": "172.18.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "DNSNames": [
        "myhttp",
        "00df16eaac24"
    ]
  }
}
}
```

# nginx.conf

- proxy\_pass 주의 깊게 볼 것!
  - URI or Not [링크](#)
- proxy\_pass 에 들어갈 IP 주소는 HTTP server 의 IP 주소
  - docker exec -it myhttp bash
    - \$ ifconfig 실행

```
$ docker run -dit --name mynginx11 --network mybridge -p 8080:80 -v [경로]/nginx.conf:/etc/nginx/nginx.conf nginx
```

```
user nginx;
worker_processes auto;

error_log /var/log/nginx/error.log notice;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local], "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    keepalive_timeout 65;

    server {
        listen 80;
        server_name localhost;

        location / {
            root /usr/share/nginx/html;
            index index.html index.htm;
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root /usr/share/nginx/html;
        }

        location /http/ {
            proxy_pass http://172.18.0.2:8888/;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header Host $host;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        }
    }
}
```

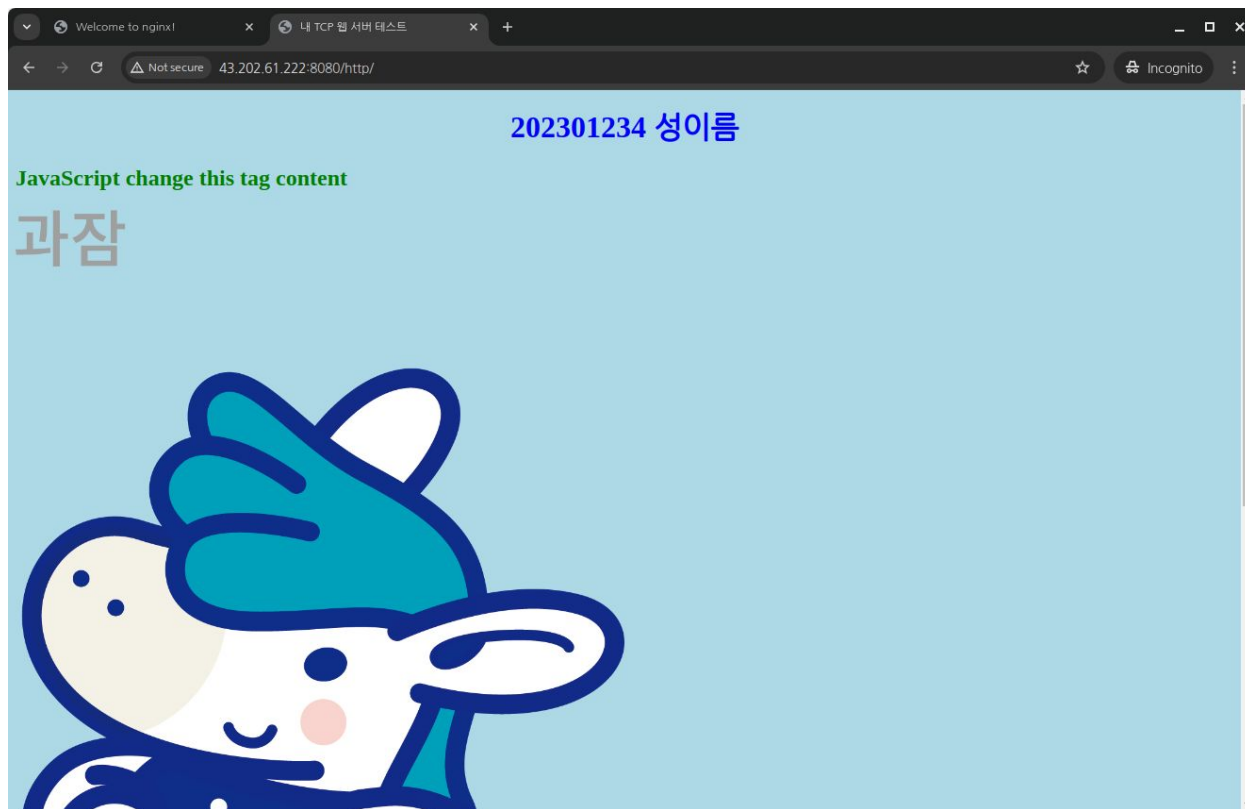


# nginx with docker routing

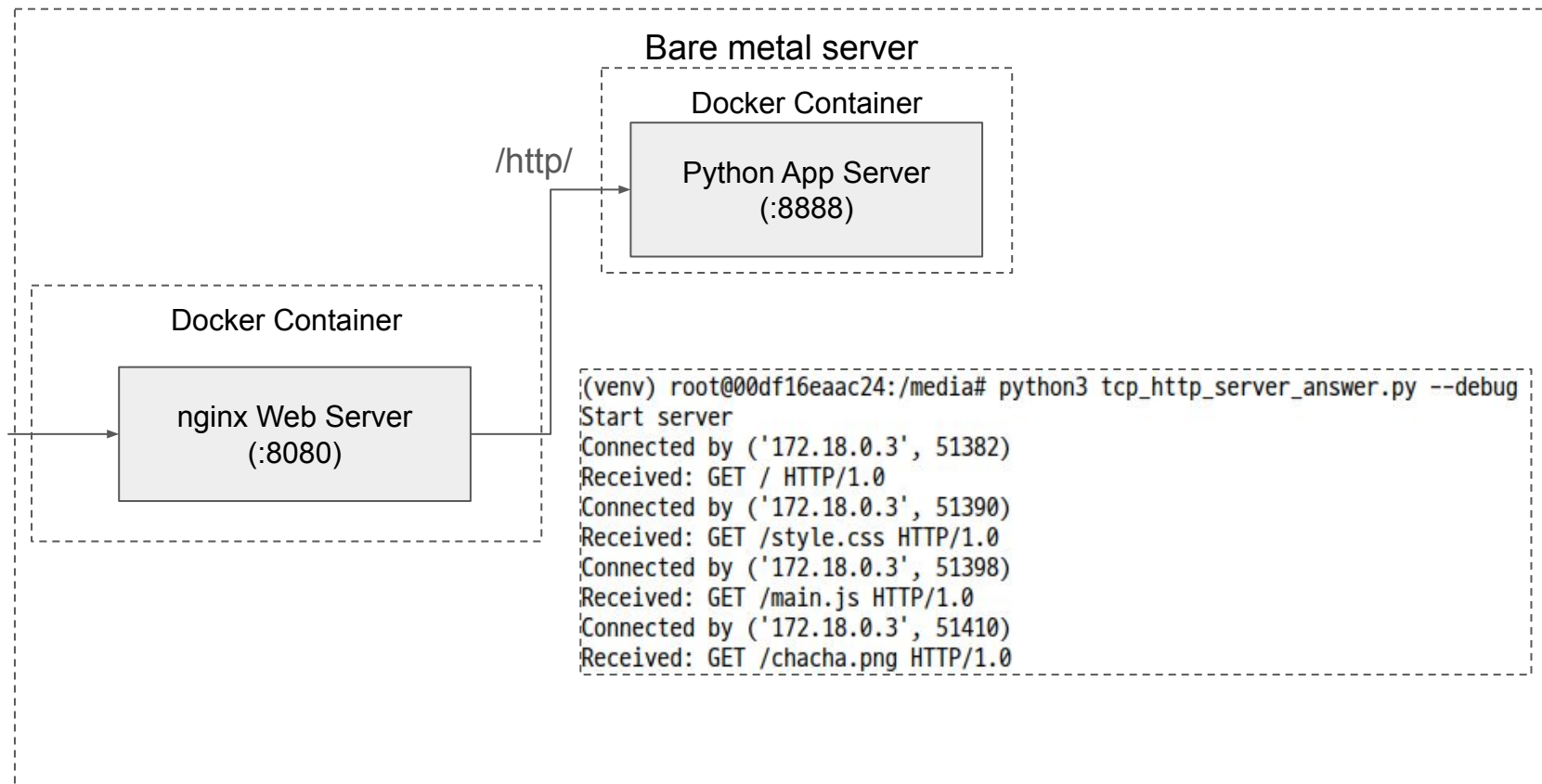
```
ubuntu@ip-172-31-4-155:~$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
2d429b9e73a6: Pull complete
9b1039c85176: Pull complete
9ad567d3b8a2: Pull complete
773c63cd62e4: Pull complete
1d2712910bdf: Pull complete
4b0adc47c460: Pull complete
171eebbdf235: Pull complete
Digest: sha256:bc5eac5eafc581aeda3008b4b1f07ebba230de2f27d47767129a6a905c84f470
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

```
ubuntu@ip-172-31-4-155:~$ docker run --interactive --tty --detach --name mynginx
11 --network mybridge --publish 8080:80 --volume /home/ubuntu/ComputerNetwork202
4/Week11-Deployment/nginx.conf:/etc/nginx/nginx.conf:ro nginx
91bed1935f697629a25b082a1dd864838fa873f242202396f1bd117e798d1a15
```

# WA~! 접속된다!!



# 설정된 구조 돌아보기



# \$ docker logs (--follow) mynginx11

- Log 를 확인해서 제대로 upstream 되는지 확인!!
- Python 3 서버에 upstream 요청이 와야함
  - 이 요청은 누가 보냈을까?

```
211.216.239.73 - - [12/Nov/2024:21:42:35 +0000] "GET /http/ HTTP/1.1" 200 331 "-"
"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
/130.0.0.0 Safari/537.36" "-"
211.216.239.73 - - [12/Nov/2024:21:42:35 +0000] "GET /http/style.css HTTP/1.1" 2
00 111 "http://43.202.61.222:8080/http/" "Mozilla/5.0 (X11; Linux x86_64) AppleW
ebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Safari/537.36" "-"
211.216.239.73 - - [12/Nov/2024:21:42:35 +0000] "GET /http/main.js HTTP/1.1" 200
156 "http://43.202.61.222:8080/http/" "Mozilla/5.0 (X11; Linux x86_64) AppleWeb
Kit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Safari/537.36" "-"
211.216.239.73 - - [12/Nov/2024:21:42:35 +0000] "GET /http/chacha.png HTTP/1.1"
200 110960 "http://43.202.61.222:8080/http/" "Mozilla/5.0 (X11; Linux x86_64) Ap
pleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Safari/537.36" "-"
```

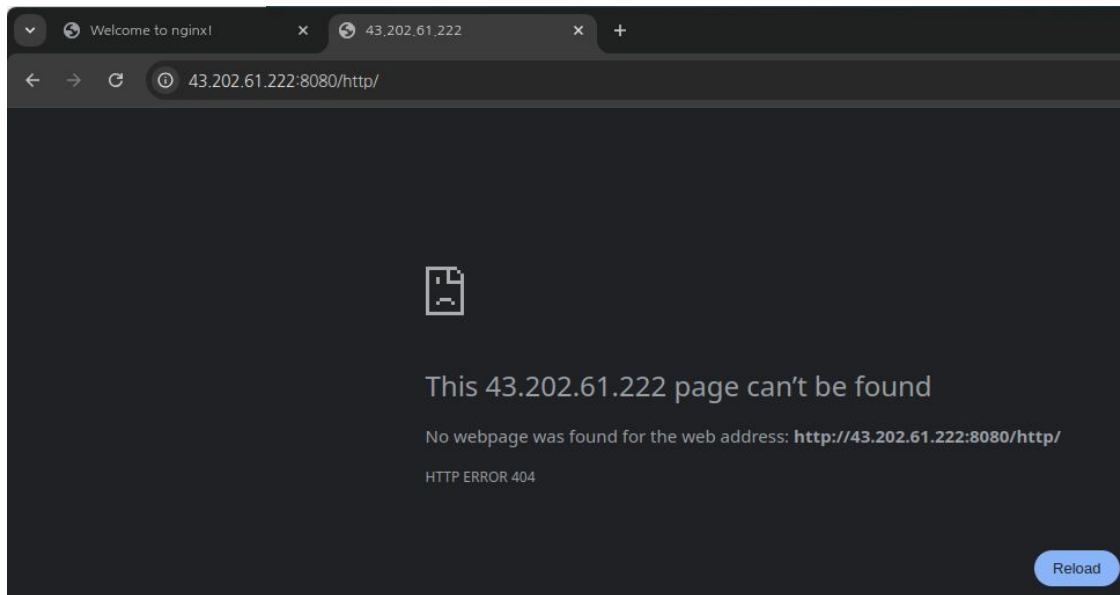
```
Connected by ('172.18.0.3', 48548)
Received: GET / HTTP/1.0
Connected by ('172.18.0.3', 48554)
Received: GET /style.css HTTP/1.0
Connected by ('172.18.0.3', 48568)
Received: GET /main.js HTTP/1.0
Connected by ('172.18.0.3', 48582)
Received: GET /chacha.png HTTP/1.0
```

# 만약 proxy\_pass 를 URI로 안 하면...

```
41 location /http/ {$  
42     proxy_pass http://172.18.0.2:8888;$  
43     proxy_set_header X-Real-IP $remote_addr;$  
44     proxy_set_header Host $host;$  
45     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;$  
46 }$
```

```
Connected by ('172.18.0.3', 40954)  
Received: GET /http/ HTTP/1.0  
Connected by ('172.18.0.3', 40970)  
Received: GET /http/ HTTP/1.0
```

```
ubuntu@ip-172-31-4-155:~$ docker restart mynginx11  
mynginx11
```



# 출석대체과제

- 과제 점수 따로 X (출결 파악 용도)
- 해당 실습을 실행하고,  
**202301234 성이름을 본인 학번 + 이름으로 변경!**
- 아래 사진처럼 주소 + 학번 + 이름 + 사진 이 나오게  
캡처해서 '캡처사진'만 제출! (9주차-202301234-  
성이름.jpg)

