오픈소스 SW 입문

w5 docker_2

2023-0504, 강의실(과학관213)

강사 : 박노헌

https://github.com/nparkcourage/2023-kau-0504

한국항공대학교

W5 docker_2

목차

7교시 15:00	◆ **네트워크** • PC 네트워크 • WSL 네트워크 • 도커 네트워크
8교시 16:00	◆ **네트워크 매핑** • 네트워크 연결 확인 • 네트워크 매핑 • 네트워크 매핑 컨테이너 실행
9교시 17:00	◆ **영구 스토리지** • 도커의 영구 스토리지 • 바인드 마운트하기 • 영구 스토리지 마운트 컨테이너 실행

W5 docker_2

첫째 시간

7교시 15:00	◆ **네트워크** • PC 네트워크 • WSL 네트워크 • 도커 네트워크
8교시 16:00	◆ **네트워크 매핑** • 네트워크 연결 확인 • 네트워크 매핑 • 네트워크 매핑 컨테이너 실행
9교시 17:00	◆ **영구 스토리지** • 도커의 영구 스토리지 • 바인드 마운트하기 • 영구 스토리지 마운트 컨테이너 실행

PC 네트워크

IP Address

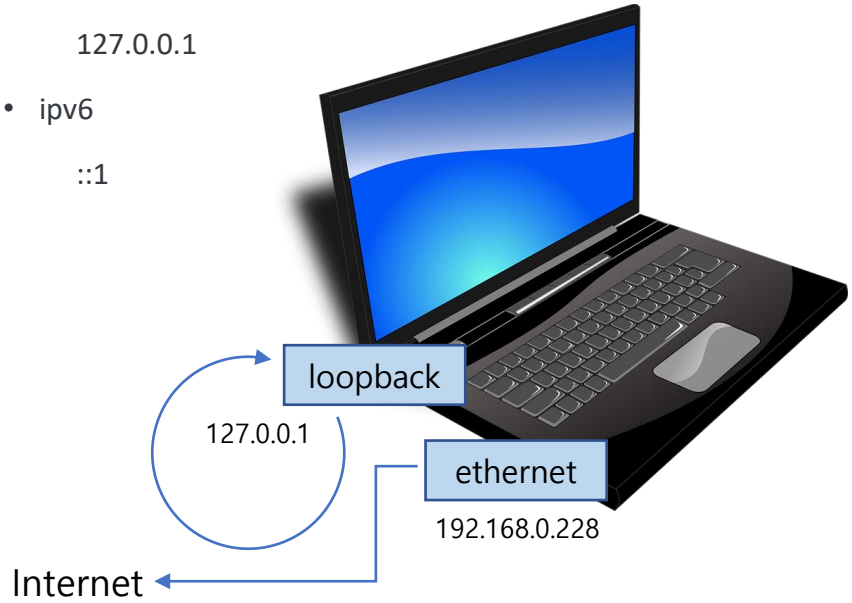
- 이더넷 어댑터의 주소
- 외부와의 통신에 사용됨

Loopback Address

- 내부에서만 작동되는 특수 주소, 외부와는 통신 불가
- localhost
- ipv4 : 127.으로 시작

  127.0.0.1
- ipv6

  ::1



네트워크 확인(파워셸에서 관리자 권한으로 실행)

```
ipconfig
nslookup localhost
ping localhost
ping 127.0.0.1
```

```
PS C:\> nslookup localhost
서버:    bns1.hananet.net
Address:  210.220.163.82

권한 없는 응답:
이름:    localhost
Address:  127.0.0.1
```

```
PS C:\> ping 127.0.0.1

Ping 127.0.0.1 32바이트 데이터 사용:
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128
127.0.0.1의 응답: 바이트=32 시간<1ms TTL=128

127.0.0.1에 대한 Ping 통계:
    패킷: 보냄 = 4, 받음 = 4, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
    최소 = 0ms, 최대 = 0ms, 평균 = 0ms
```

WSL 네트워크

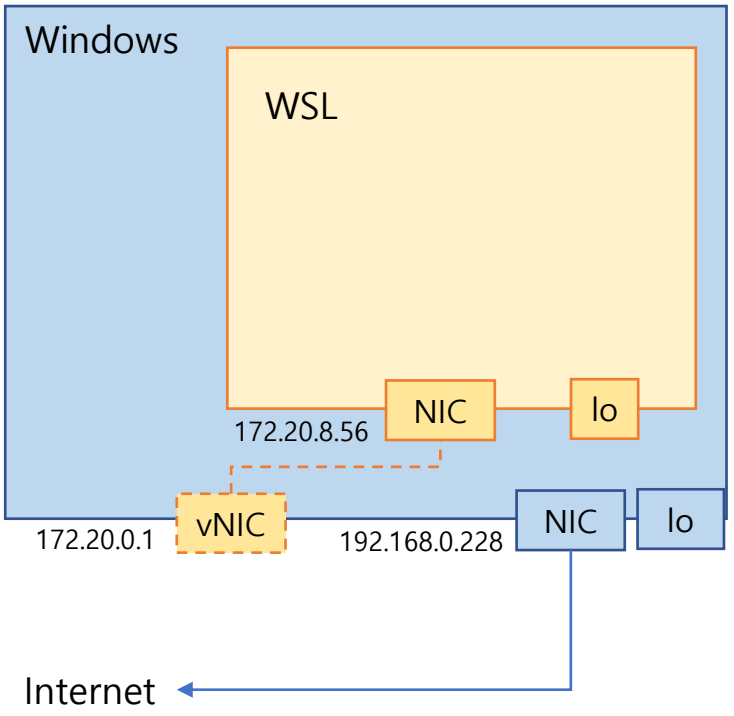
WSL 네트워크

- WSL은 이더넷 네트워크 설정

  ip addr

- 호스트에서는 WSL용 가상의 네트워크 생성

  ipconfig



W5 docker_2 : 네트워크

WSL 네트워크

WSL에서 네트워크 확인

`ip addr`

(없으면, `apt install iproute2` 설치)

```
esu@npark-dell:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 9a:86:cd:8e:38:16 brd ff:ff:ff:ff:ff:ff
3: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether fa:da:a2:4b:bc:b2 brd ff:ff:ff:ff:ff:ff
4: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
5: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
6: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:69:13:99 brd ff:ff:ff:ff:ff:ff
    inet 172.20.8.56/20 brd 172.20.15.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe69:1399/64 scope link
       valid_lft forever preferred_lft forever
```

host(Windows)에서 네트워크 확인

- powershell에서 다음 명령 실행

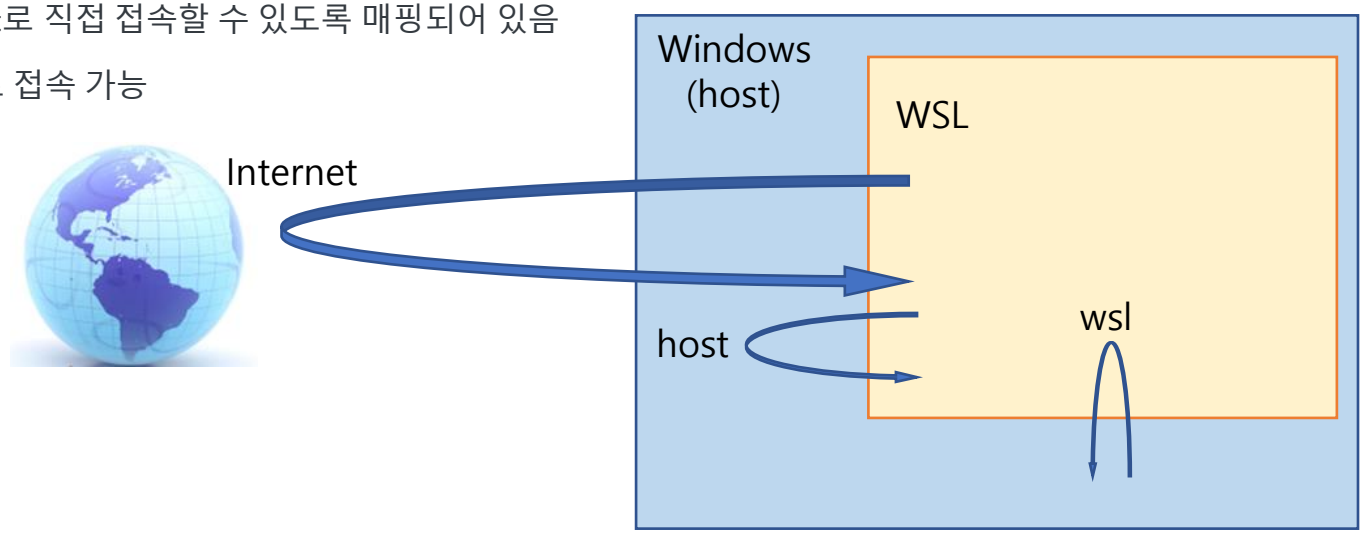
  `ipconfig`

```
PS C:\Users\npark> ipconfig

Windows IP 구성


무선 LAN 어댑터 로컬 영역 연결* 11:

   미디어 상태 . . . . . . . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . . :

무선 LAN 어댑터 로컬 영역 연결* 12:

   미디어 상태 . . . . . . . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . . :

무선 LAN 어댑터 Wi-Fi:

   연결별 DNS 접미사. . . . . :
   링크-로컬 IPv6 주소 . . . . . : fe80::e728:7003:6b5c:535d%12
   IPv4 주소 . . . . . . . . . . . : 192.168.0.228
   서브넷 마스크 . . . . . . . . . : 255.255.255.0
   기본 게이트웨이 . . . . . . . . : 192.168.0.1

이더넷 어댑터 Bluetooth 네트워크 연결:

   미디어 상태 . . . . . . . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . . :

이더넷 어댑터 vEthernet (Default Switch):

   연결별 DNS 접미사. . . . . :
   링크-로컬 IPv6 주소 . . . . . : fe80::f73:2c17:35a9:6eab%21
   IPv4 주소 . . . . . . . . . . . : 172.23.96.1
   서브넷 마스크 . . . . . . . . . : 255.255.240.0
   기본 게이트웨이 . . . . . . . . :

이더넷 어댑터 vEthernet (WSL):

   연결별 DNS 접미사. . . . . :
   링크-로컬 IPv6 주소 . . . . . : fe80::1c93:2b87:2146:7864%49
   IPv4 주소 . . . . . . . . . . . : 172.20.0.1
   서브넷 마스크 . . . . . . . . . : 255.255.240.0
   기본 게이트웨이 . . . . . . . . :
```

WSL 네트워크

라우팅

- WSL -> 외부 네트워크(인터넷)
  - Hyper-V의 버추얼 스위치를 통하여 보통의 공유기와 같이 외부 네트워크와 통신
- WSL -> 호스트(Windows) 네트워크
  - Windows의 LAN 어댑터 네트워크로 연결
  - 호스트의 버추얼 네트워크(vEthernet(WSL))로 연결
  - ex) 192.168.0.228, 172.20.0.1
- 호스트(Windows) -> WSL
  - 호스트에서 localhost로 직접 접속할 수 있도록 매핑되어 있음
  - localhost, 127.0.0.1로 접속 가능



Windows: 174. WSL 2의 네트워크 통신 방법 (sysnet.pe.kr)

WSL 네트워크

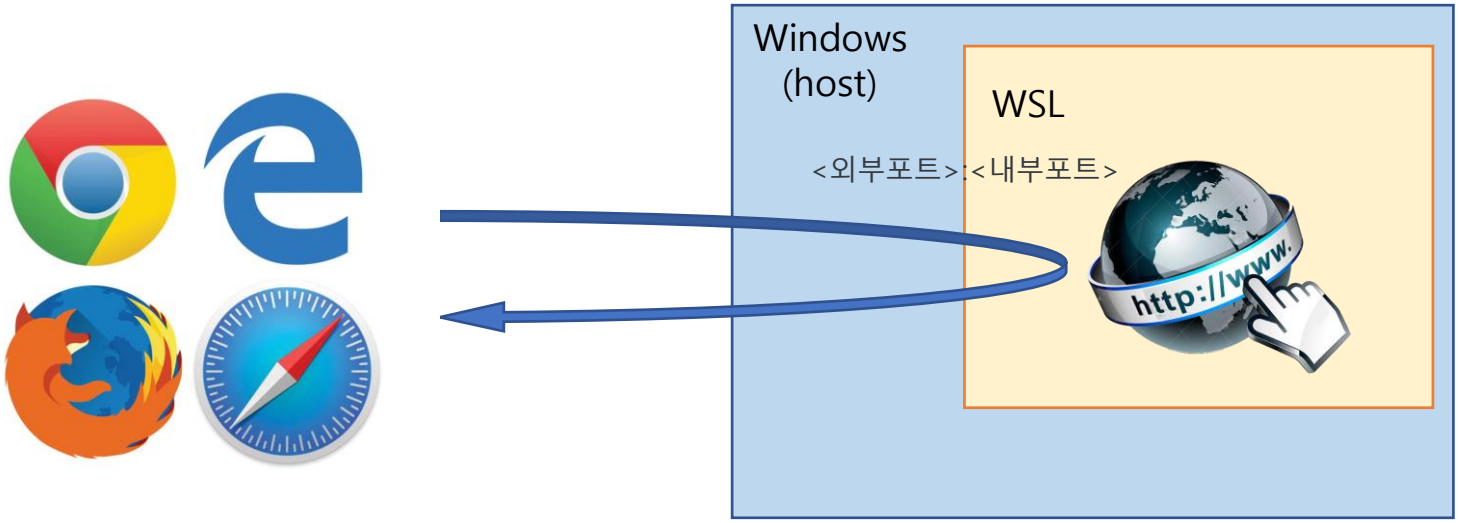
라우팅

- 외부 네트워크(인터넷) -> WSL
  - 외부에서 WSL에 접속할 수 있도록 포트 포워딩 필요(파워셸에서 관리자 권한으로)

    netsh interface portproxy add v4tov4 listenport=<포트번호> listenaddress=0.0.0.0 connectport= <포트번호> \
    connectaddress=<WSL의 ip address>
  - 설정 보기

    netsh interface portproxy show v4tov4
  - 포트 포워딩 제거하기

    netsh interface portproxy delete v4tov4 listenport=<포트 번호> listenaddress=0.0.0.0

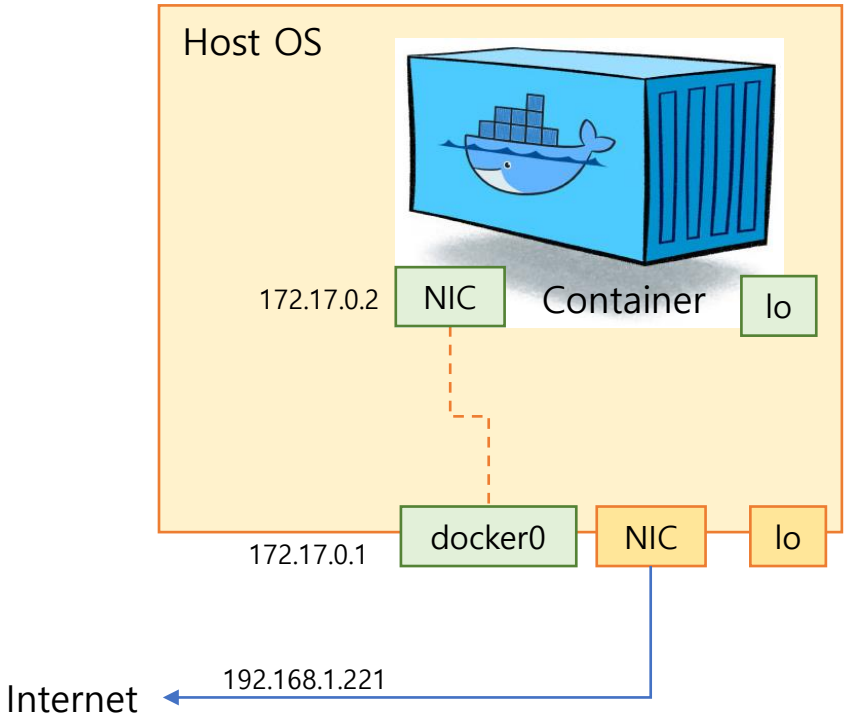


Windows: 174. WSL 2의 네트워크 통신 방법 (sysnet.pe.kr)

Docker 네트워크

Docker 네트워크 구성

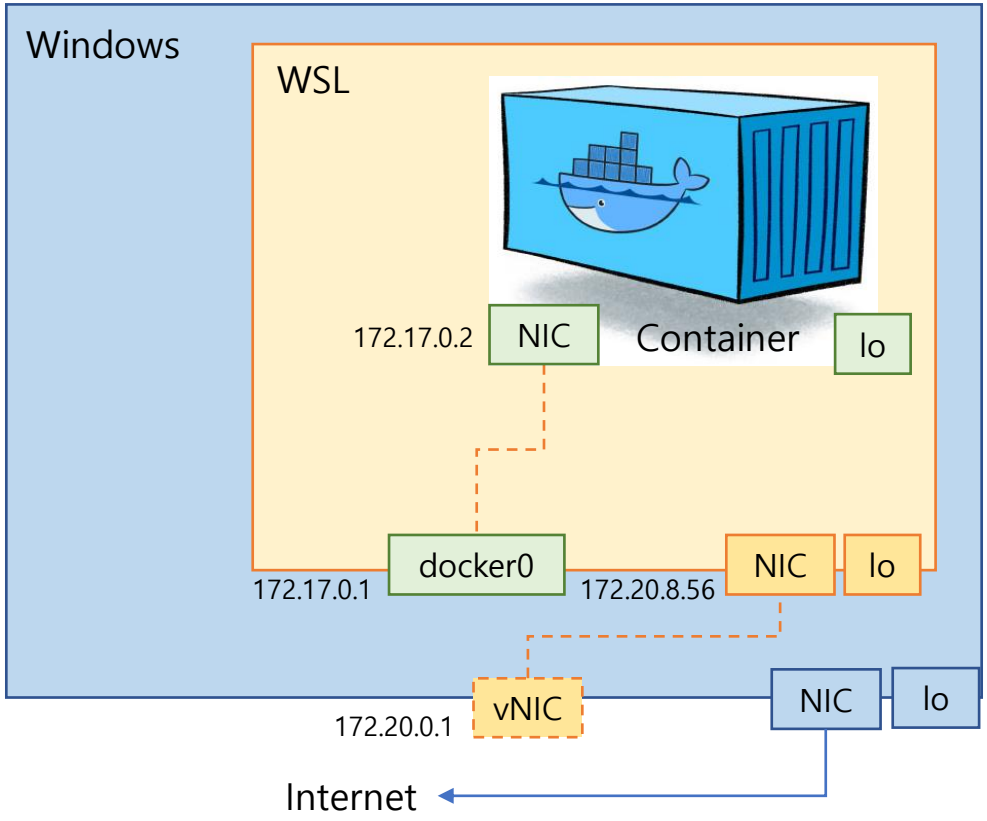
- 도커 엔진이 시작될 때 docker0 네트워크 어댑터
- 생성컨테이너가 시작될 때마다 veth 가상 네트워크 어댑터 생성



Docker 네트워크

WSL의 Docker 네트워크 구성

- 도커가 WSL 내부에 있으므로 도커의 네트워크가 호스트로 노출이 되어도 WSL의 가상 네트워크에 연결됨
- 도커는 WSL의 가상 네트워크를 통하여 외부 네트워크로 연결됨



Docker 네트워크

wsl에서 docker container 네트워크 확인

```
ip addr
```

```
esu@npark-dell:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 1e:73:26:3e:1a:8c brd ff:ff:ff:ff:ff:ff
3: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 86:9c:2d:d4:c5:91 brd ff:ff:ff:ff:ff:ff
4: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
5: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
6: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:e1:e1:60 brd ff:ff:ff:ff:ff:ff
    inet 172.20.8.56/20 brd 172.20.15.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fee1:e160/64 scope link
       valid_lft forever preferred_lft forever
7: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:74:3c:27:f6 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:74ff:fe3c:27f6/64 scope link
       valid_lft forever preferred_lft forever
13: vethaa5da14@if12: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0 state UP group default
    link/ether 16:8e:20:d8:27:65 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::148e:20ff:fed8:2765/64 scope link
       valid_lft forever preferred_lft forever
```

Docker 네트워크

docker container 내부에서 네트워크 확인

```
sudo service docker start
sudo docker ps -a
sudo docker start <container name>
sudo docker exec -it <container name> bash
ip addr
```

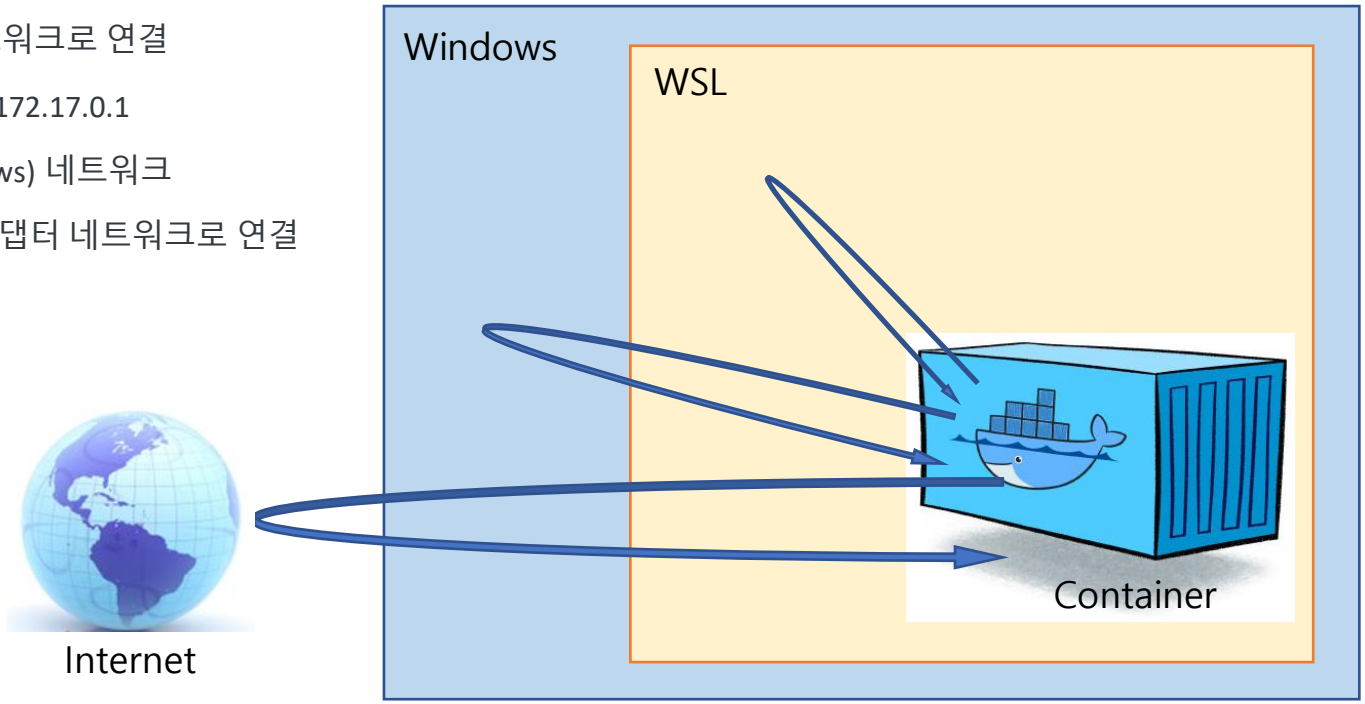
(없으면, `apt install iproute2` 설치)

```
esu@npark-dell:~$ docker exec -it pedantic_driscoll bash
root@414b77cbb242:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
3: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
12: eth0@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
```

Docker 네트워크

라우팅

- 컨테이너 -> 외부 네트워크(인터넷)
  - 컨테이너 네트워크 WSL의 Hyper-V의 버추얼 스위치를 통하여 보통의 공유기와 같이 외부 네트워크와 통신
- 컨테이너 -> wsl 네트워크
  - wsl의 eth 네트워크로 연결
  - wsl의 docker0 네트워크로 연결
  - ex) 192.168.0.228, 172.17.0.1
- 컨테이너 -> host(Windows) 네트워크
  - Windows의 LAN 어댑터 네트워크로 연결
  - ex) 192.168.0.228

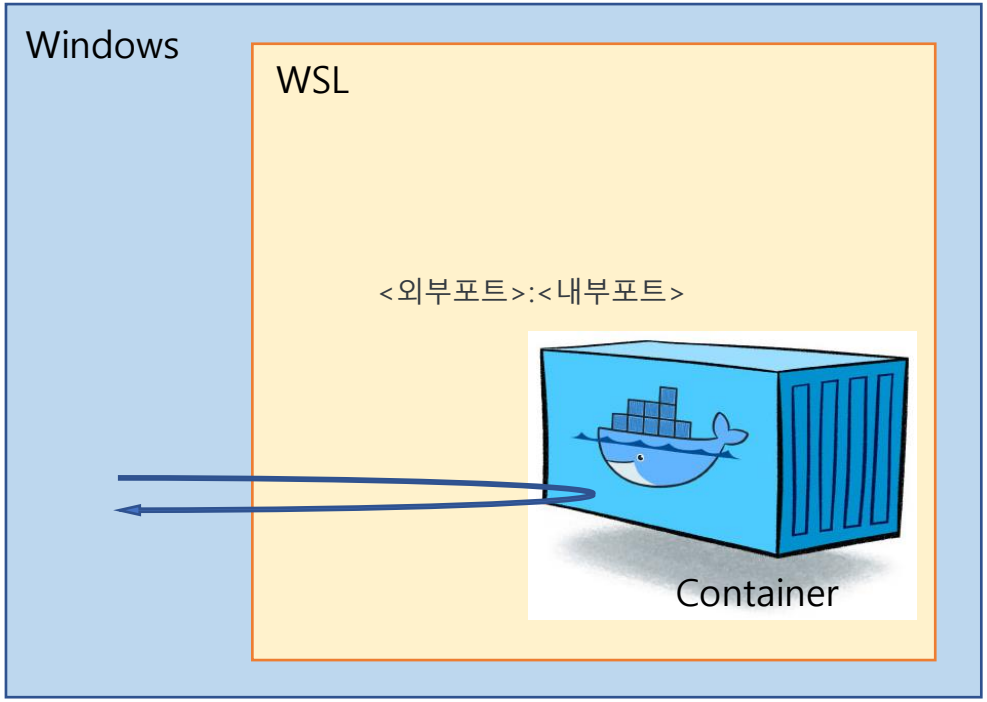


Windows: 174. WSL 2의 네트워크 통신 방법 (sysnet.pe.kr)

Docker 네트워크

라우팅

- 호스트(Windows) -> 컨테이너
  - 호스트에서 컨테이너로 접속할 수 있는 경로가 없으므로 연결할 수 없음
  - 컨테이너의 포트를 매핑하여 호스트(WSL)의 포트로 노출시켜야 함



Windows: 174. WSL 2의 네트워크 통신 방법 (sysnet.pe.kr)

Docker 네트워크

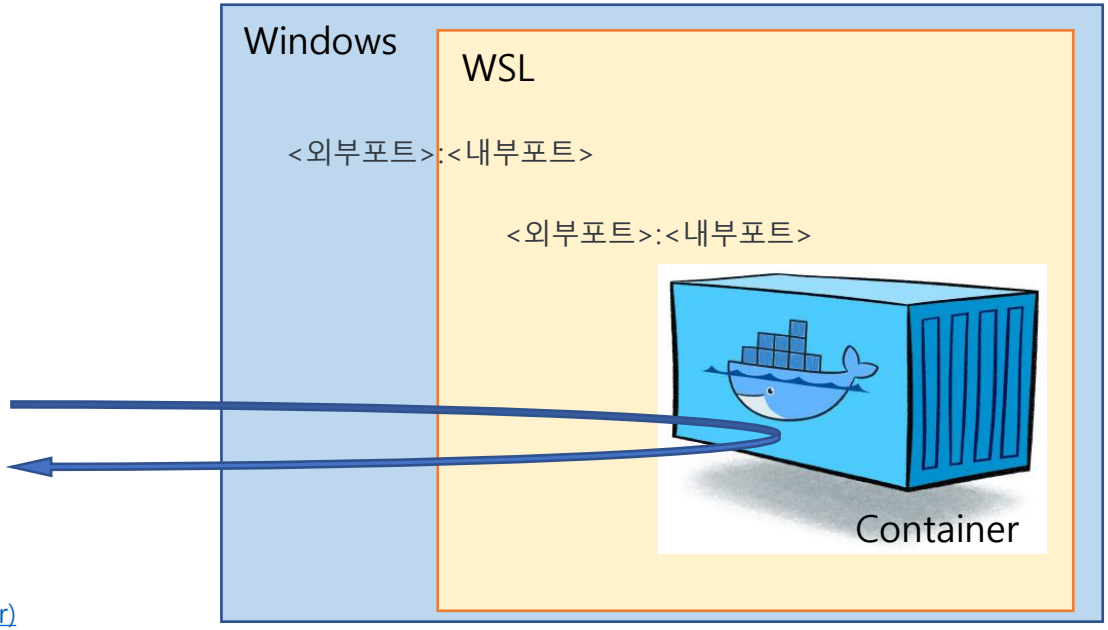
라우팅

- 외부(인터넷) -> 컨테이너
  - 컨테이너의 포트를 매핑하여 WSL의 포트로 노출시키고 노출된 WSL 포트를 윈도우에서 포워딩 설정

    netsh interface portproxy add v4tov4 listenport=<포트번호> listenaddress=0.0.0.0 connectport= <포트번호> \
    connectaddress=<WSL의 ip address>
  - 설정 보기

    netsh interface portproxy show v4tov4
  - 포트 포워딩 제거하기

    netsh interface portproxy delete v4tov4 listenport=<포트 번호> listenaddress=0.0.0.0



Windows: 174. WSL 2의 네트워크 통신 방법 (sysnet.pe.kr)

둘째 시간

7교시 15:00	◆ 네트워크 • PC 네트워크 • WSL 네트워크 • 도커 네트워크
8교시 16:00	◆ **네트워크 매핑** • 네트워크 연결 확인 • 네트워크 매핑 • 네트워크 매핑 컨테이너 실행
9교시 17:00	◆ 영구 스토리지 • 도커의 영구 스토리지 • 바인드 마운트하기 • 영구 스토리지 마운트 컨테이너 실행

네트워크 연결 확인

web server를 이용한 연결 확인

python web server 실행

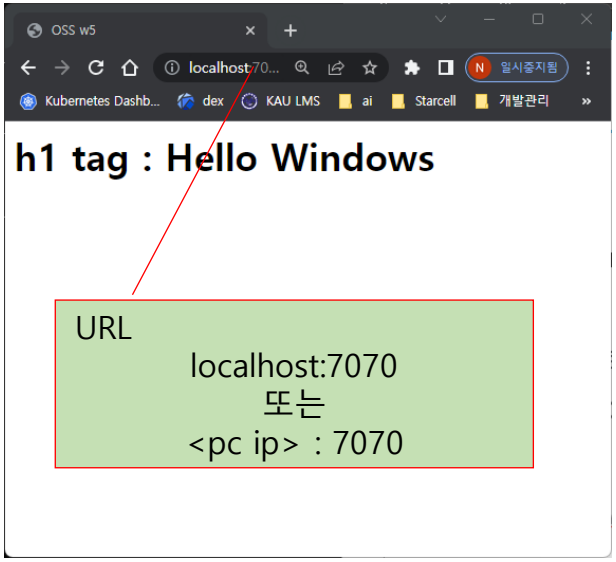
- python에는 web server 모듈이 내장되어 있음
- 이 내장 web server 모듈을 이용하여 간단하게 웹서버를 실행시킬 수 있으며 이 웹서버에 접속 가능한 지를 보고 네트워크 연결을 확인할 수 있음
- 사용 방법 실행하는 현재 디렉토리에 index.html 파일을 준비하고 다음과 같은 명령을 실행하면 웹서버가 index.html을 첫페이지로 보여 주면서 실행 됨
    - python3 -m http.server <포트 번호>
- CLI로 웹 접속 확인
    - curl <url>
    - curl이 없는 경우 설치 : sudo apt install curl

W5 docker_2 : 네트워크 매핑

네트워크 연결 확인

Windows에서 python web server 실행

- 현재 디렉토리에 index.html 준비
- Powershell에서 python 설치(관리자 권한으로 실행)
- python : python 설치 안내 표시 -> 설치
- python3 -m http.server <port number>



powershell에서 실행

```
notepad index.html

python3 -m http.server 7070
```

index.html

```
<html>
    <head>
        <title>OSS w5</title>
    </head>
    <body>
        <h1>h1 tag : Hello Windows</h1>
    </body>
</html>
```

powershell에서 실행

```
curl http://localhost:7070

(curl http://192.168.0.228:7070)
```

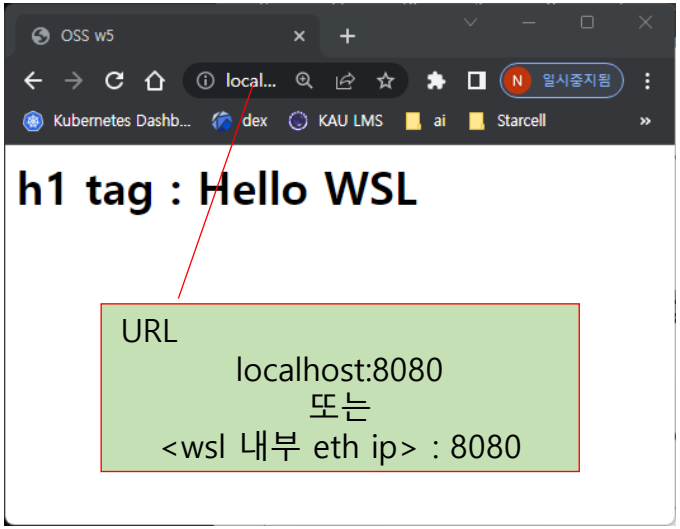
```
<html>
    <head>
        <title>OSS w5</title>
    </head>
    <body>
        <h1>h1 tag : Hello Windows </h1>
    </body>
</html>
```

W5 docker_2 : 네트워크 매핑

네트워크 연결 확인

Unix에서 python web server 실행

- Unix(Linux, Mac OS, WSL)에서 실행
  - 현재 디렉토리에 index.html 준비
  - python3 -m http.server <port number>



wsl에서 실행

```
mkdir -p ~/projects/w5
cd ~/projects/w5
nano index.html
python3 -m http.server 8080
```

wsl에서 실행

```
curl localhost:8080
(curl 172.20.8.56:8080)
```

index.html

```
<html>
    <head>
        <title>OSS w5</title>
    </head>
    <body>
        <h1>h1 tag : Hello WSL</h1>
    </body>
</html>
```

```
esu@npark-dell:~$ curl localhost:8080
<html>

    <head>
        <title>OSS w5</title>
    </head>

    <body>
        <h1>h1 tag : Hello WSL</h1>
    </body>

</html>
```

W5 docker_2 : 네트워크 매핑

네트워크 연결 확인

container에서 web server 실행

- container에 로그인
  - 현재 디렉토리에 index.html 준비
  - python3 -m http.server <port number>

container에서 실행

```
docker exec -it <container name> bash
mkdir -p ~/projects/w5
cd ~/projects/w5
nano index.html
python3 -m http.server 9090 &
```

index.html

```
<html>
    <head>
        <title>OSS w5</title>
    </head>
    <body>
        <h1>h1 tag : Hello Container 1</h1>
    </body>
</html>
```

container에서 실행

```
curl localhost:9090
(curl 172.17.0.2:9090)
```

```
root@414b77cbb242:~/projects/w5# curl localhost:9090
127.0.0.1 - - [04/Apr/2023 01:33:22] "GET / HTTP/1.1" 200 -
<html>
        <head>
                <title>OSS w5</title>
        </head>
        <body>
                <h1>h1 tag : Hello Container 1</h>
        </body>
</html>
```

wsl에서 실행

```
curl 172.17.0.2:9090
```

```
esu@npark-dell:~$ curl 172.17.0.2:9090
<html>
        <head>
                <title>OSS w5</title>
        </head>
        <body>
                <h1>h1 tag : Hello Container 1</h>
        </body>
</html>
```

네트워크 매핑

container port mapping

- 도커 실행 시 -p 옵션을 사용하여 포트 매핑을 설정 가능
  - ex) docker run -p <host port>:<container port>

container image commit

- 실행 중인 컨테이너에 포트 매핑을 추가할 수 없음
- 컨테이너의 이미지를 커밋하고 새로운 컨테이너 실행
- 컨테이너 이미지를 커밋하기 위해 컨테이너를 stop
- 다음과 같은 명령으로 컨테이너 이미지 커밋

  docker commit <container name> <image name:tag>
- docker run으로 새로운 컨테이너 실행
- docker run 옵션으로 포트 매핑 추가

  -p <host port>:<container port>

wsl에서 실행

```
docker stop <container name>
docker commit <container name> ubuntu:w5
docker images
docker run -p 9090:9090 -itd ubuntu:w5 bash
```

네트워크 매핑

container port mapping

- 웹서버를 실행하여 네트워크 연결 확인

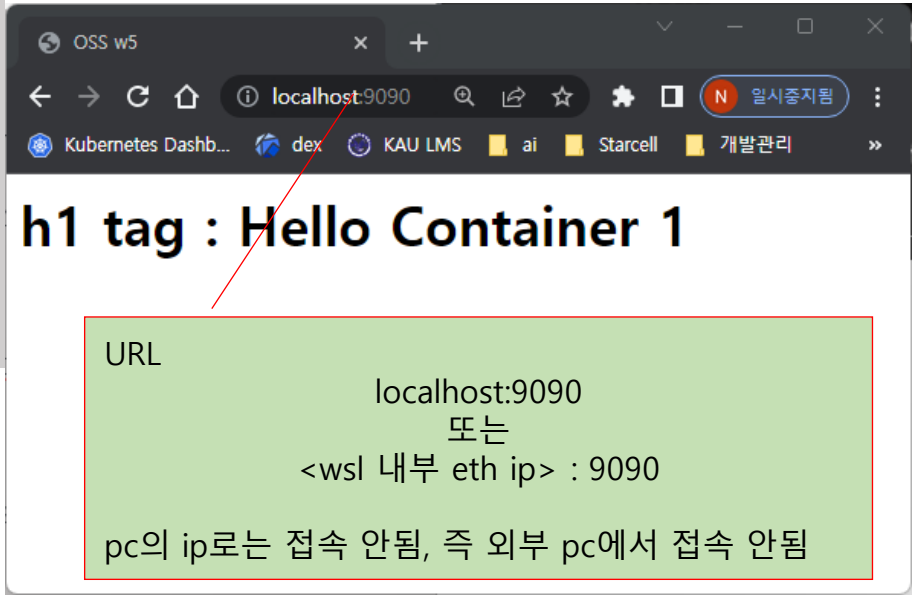
wsl에서 실행

```
curl localhost:9090
```

```
esu@npark-dell:~$ curl localhost:9090
<html>
        <head>
                <title>OSS w5</title>
        </head>
        <body>
                <h1>h1 tag : Hello Container 1</h>
        </body>
</html>
```

wsl에서 실행

```
docker ps

docker exec -it <new container name> bash

cd ~/projects/w5

python3 -m http.server 9090 &
```



네트워크 매핑

wsl port mapping

- Windows에서는 wsl 가상 머신의 서비스로 외부 기기가 접속할 수 있도록 통신 패킷을 전달(forward)하는 프락시 지원
- TCP Packet만 지원하므로 UDP 통신에는 사용할 수 없음

설정 : netsh interface portproxy add v4tov4 listenport=<포트번호> listenaddress=0.0.0.0 connectport= <포트번호> \
connectaddress=<WSL의 ip address>

확인 : netsh interface portproxy show v4tov4

제거 : netsh interface portproxy delete v4tov4 listenport=<포트 번호> listenaddress=0.0.0.0

포트 포워드 설정(파워셸에서 관리자 권한으로 실행)

- 설정하기

netsh interface portproxy add v4tov4 listenport=9090 listenaddress=0.0.0.0 connectport=9090 connectaddress=172.20.8.56

- 설정 내용 보기

netsh interface portproxy show v4tov4

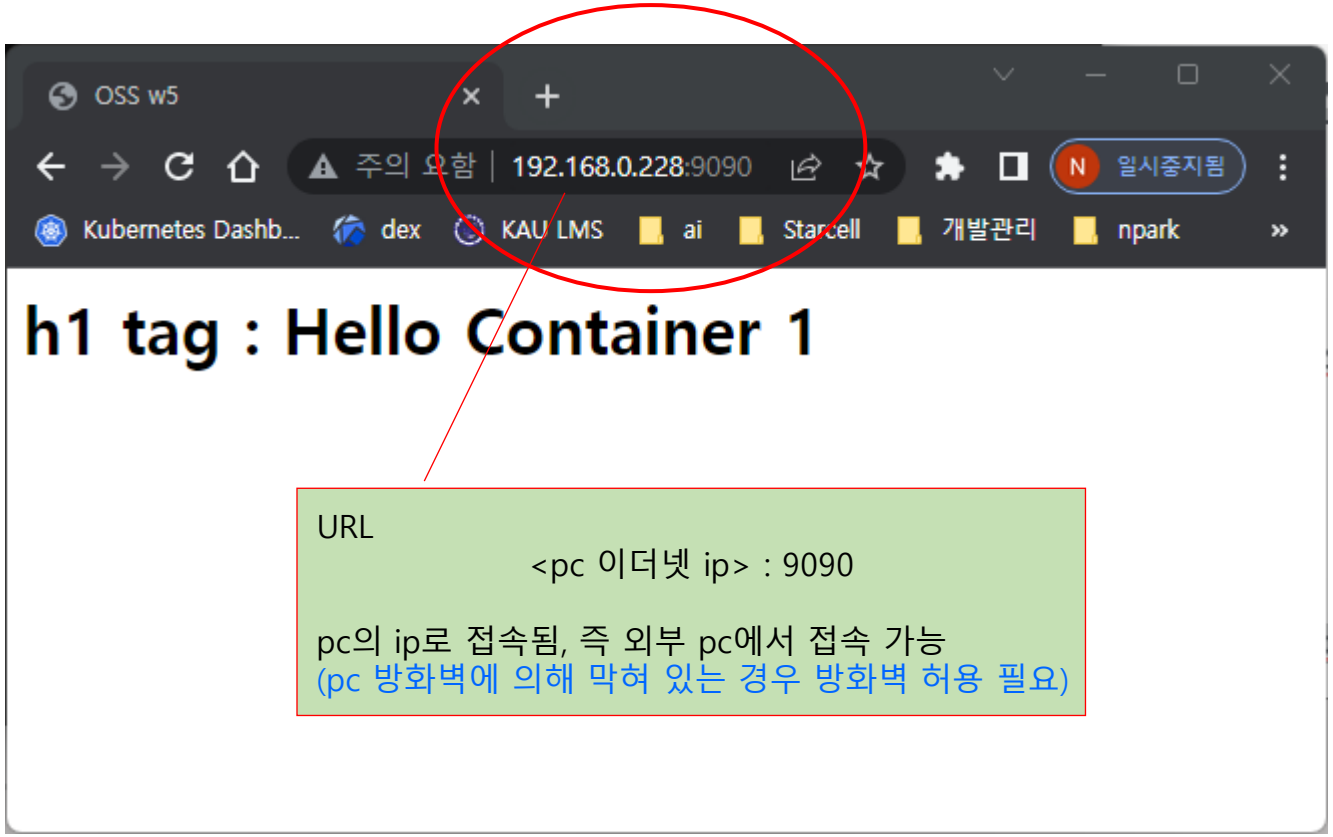
- 제거하기

netsh interface portproxy delete v4tov4 listenport=9090 listenaddress=0.0.0.0

네트워크 매핑

매핑 확인

- 브라우저로 연결하여 매핑을 확인할 수 있음



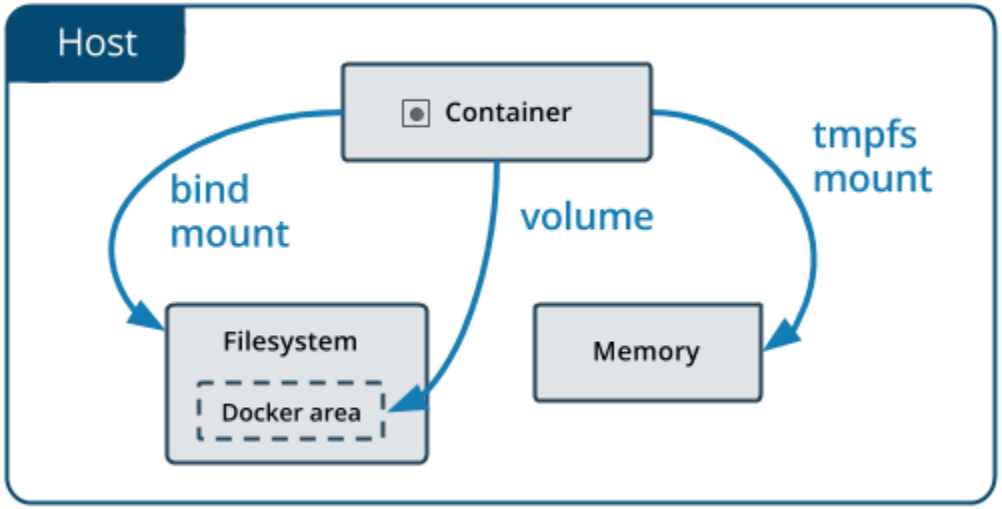
세째 시간

7교시 15:00	◆ 네트워크 • PC 네트워크 • WSL 네트워크 • 도커 네트워크
8교시 16:00	◆ 네트워크 매핑 • 네트워크 연결 확인 • 네트워크 매핑 • 네트워크 매핑 컨테이너 실행
9교시 17:00	◆ 영구 스토리지 • 도커의 영구 스토리지 • 바인드 마운트하기 • 영구 스토리지 마운트 컨테이너 실행

도커의 영구 스토리지

컨테이너의 영구 스토리지 필요성과 사용 방법

- 도커 컨테이너의 저장소는 외부와 단절되어 있으며 컨테이너가 삭제될 때 같이 삭제됨
- 컨테이너가 삭제되어도 데이터를 보관할 수 있는 영구 저장소(permanent storage) 필요
- 컨테이너 외부와 데이터를 주고 받기위해서도 외부에서 접근 가능한 저장소 필요
- 도커에서는 다음과 같은 2가지 영구 저장소 사용 방식을 제공
  - 도커 볼륨(Volume)
  - 바인드 마운트(Bind Mount)
- 도커는 볼륨 사용을 권장하나 본 과정에서는 간편한 바인드 마운트 방법을 사용



https://www.daleseo.com/docker-volumes-bind-mounts/

바인트 마운트하기

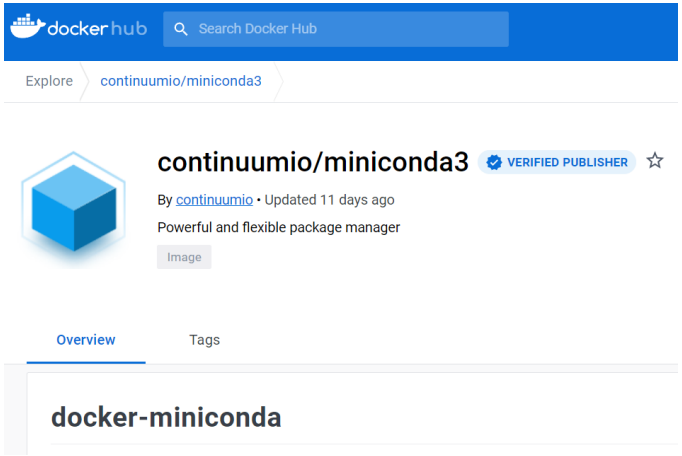
바인드 마운트 방법

- docker run 명령에서 -v 옵션을 사용
  - ex) docker run -v <host dir>:<container dir>

영구 스토리지 마운트 컨테이너 실행

miniconda image를 이용한 실습

- docker hub의 miniconda3 이미지 사용
- 스토리지 바인드 마운트와 포트 매핑을 함께 사용하는 실습
  - ex) docker run -v <host dir>:<container dir> -p <host port>:<container port>



wsl에서 실행

- 컨테이너 실행

```
docker run -itd --name miniconda -v ~/projects/w5:/root/projects/w5 -p 8888:8888 continuumio/miniconda3 /bin/bash

docker exec -it miniconda bash
conda list
conda update conda
conda install jupyter
jupyter lab --notebook-dir=/root/projects/w5 --ip='*' --port=8888 --no-browser --allow-root
```

영구 스토리지 마운트 컨테이너 실행

영구 스토리지 확인

- 실행한 컨테이너의 마운트 디렉토리가 호스트(wsl)의 디렉토리와 공유되는 것 확인

wsl에서 실행

```
cd ~/projects/w5
ls -al
```

파일 내용 확인

```
docker exec -it miniconda bash
```

컨테이너에서 실행

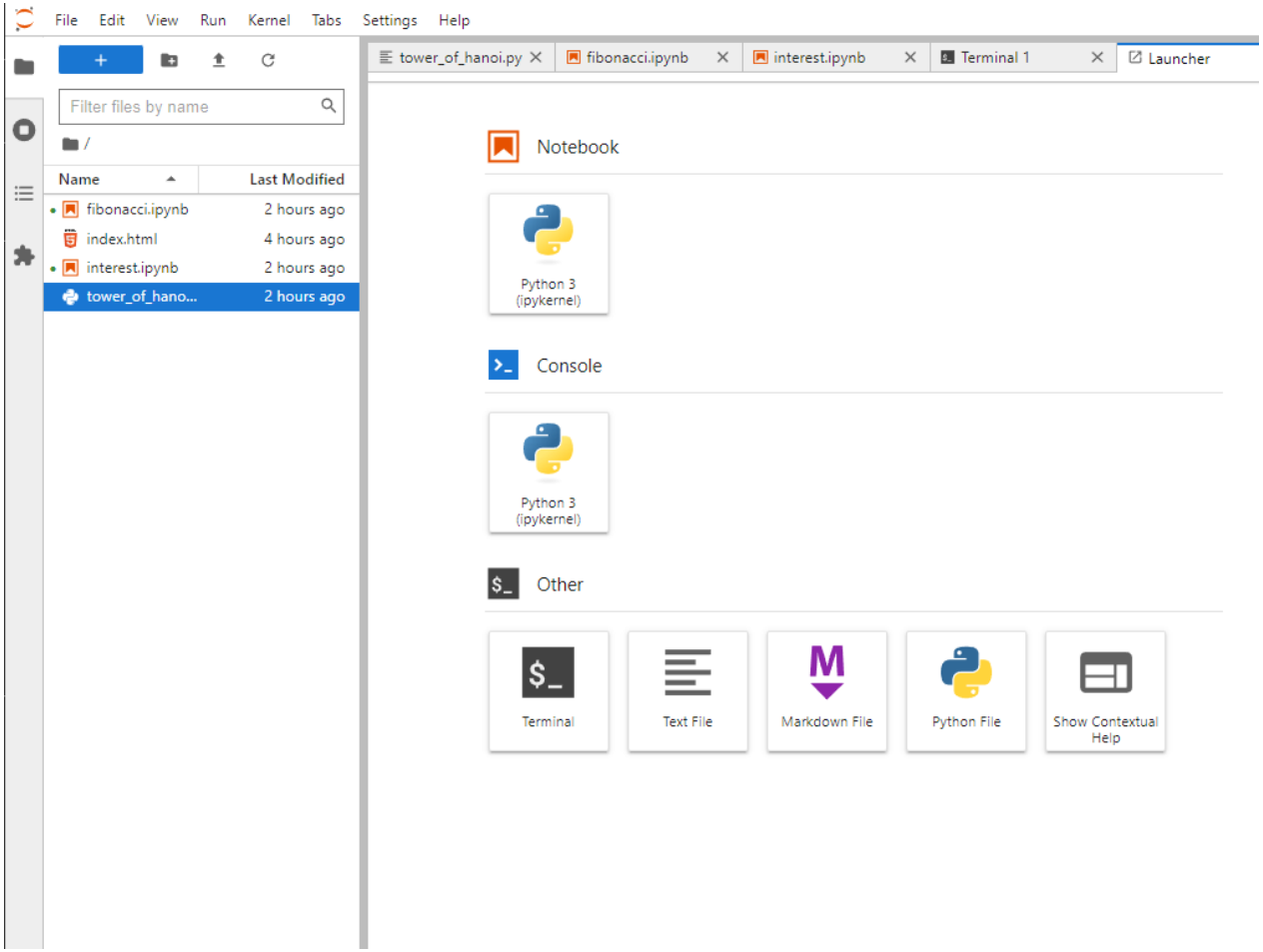
```
cd ~/projects/w5
ls -al
```

파일 내용 확인

영구 스토리지 마운트 컨테이너 실행

네트워크 연결 확인

- jupyter lab에 접속하여 네트워크와 스토리지 모두 확인

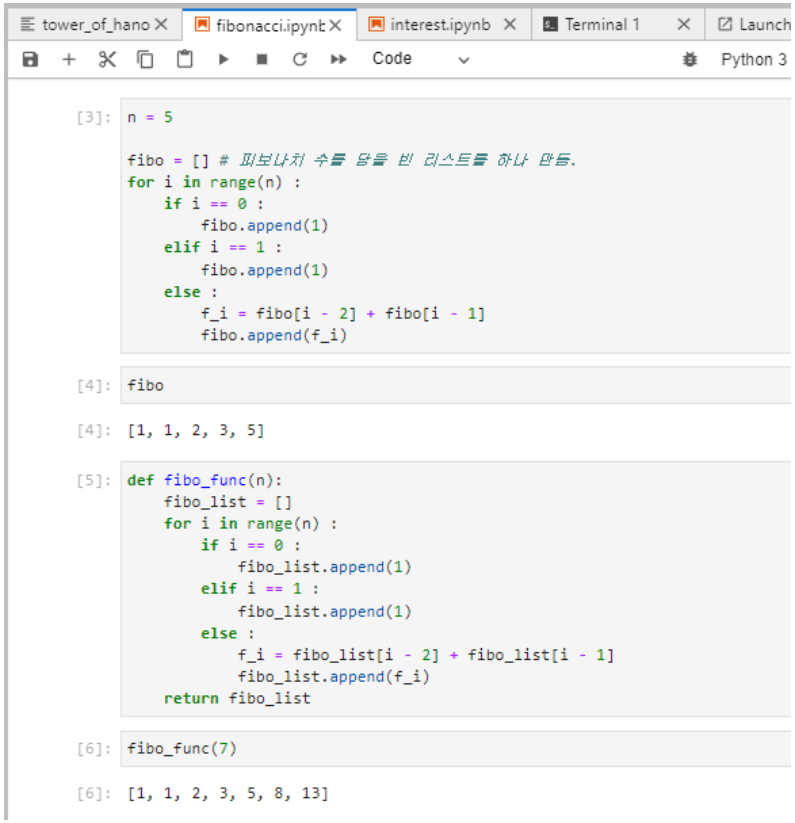


영구 스토리지 마운트 컨테이너 실행

샘플프로그램 실행

- tower_of_hanoi.py 실행
- jupyter notebook 실행

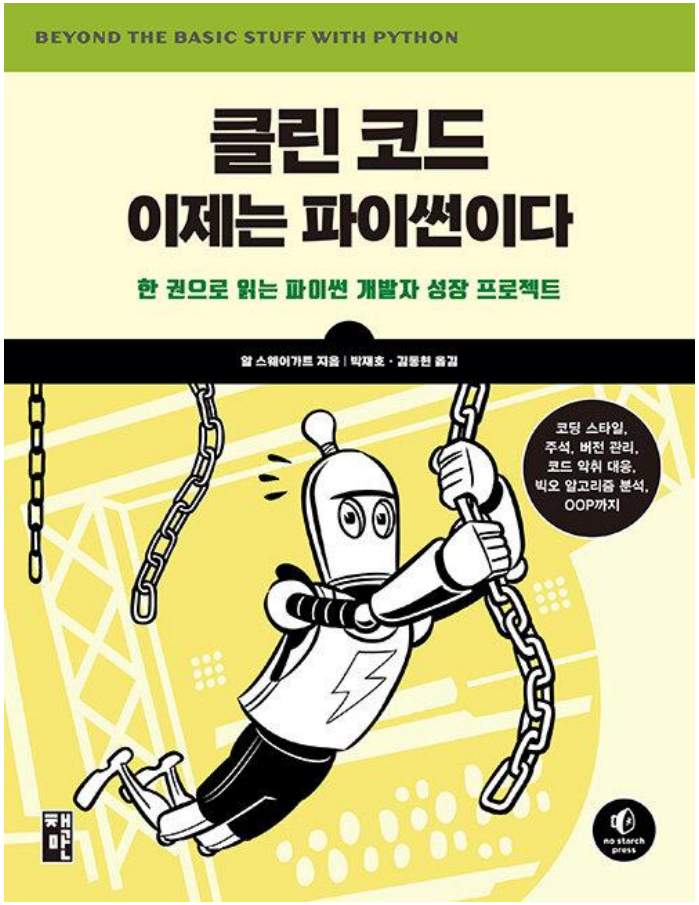
```
탑의 "시작"과 "끝"의 글자 또는 QUIT를 입력하십시오.
(예: 탑 A에서 탑 B로 원판을 이동하려면 AB를 입력합니다.)

> ab
    ||          ||          ||
    ||          ||          ||
    ||          ||          ||
    ||          ||       @@@_3@@@
  @@_2@@      @_1@     @@@@_4@@@@
    A           B           C

탑의 "시작"과 "끝"의 글자 또는 QUIT를 입력하십시오.
(예: 탑 A에서 탑 B로 원판을 이동하려면 AB를 입력합니다.)

> ca
더 작은 원판에 더 큰 원판을 올릴 수 없습니다.
탑의 "시작"과 "끝"의 글자 또는 QUIT를 입력하십시오.
(예: 탑 A에서 탑 B로 원판을 이동하려면 AB를 입력합니다.)

> ac
    ||          ||          ||
    ||          ||          ||
    ||          ||       @@_2@@
    ||          ||      @@@_3@@@
    ||        @_1@     @@@@_4@@@@
     A          B           C

탑의 "시작"과 "끝"의 글자 또는 QUIT를 입력하십시오.
(예: 탑 A에서 탑 B로 원판을 이동하려면 AB를 입력합니다.)

> bc
    ||          ||          ||
    ||          ||        @_1@
    ||          ||       @@_2@@
    ||          ||      @@@_3@@@
    ||          ||     @@@@_4@@@@
     A          B           C

퍼즐을 풀었습니다! 참 잘했습니다!
(base) root@6942f87aa099:~/projects/w5#
```



참고 문헌 및 파이썬 추천 서적

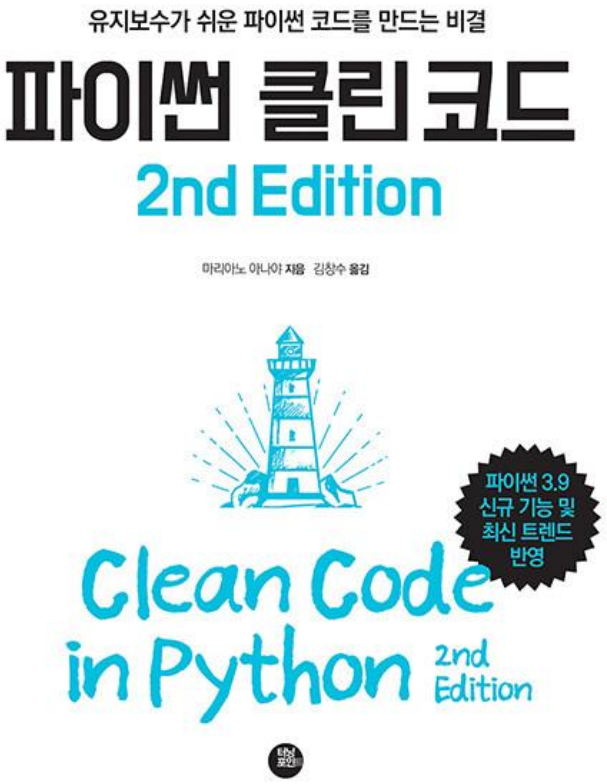
클린코드, 이제는 파이썬이다.

- https://www.onlybook.co.kr/entry/clean-python
- https://inventwithpython.com/beyond/chapter14.html



파이썬 클린코드 2nd Edition

- https://www.aladin.co.kr/shop/wproduct.aspx?ItemId=303080968



과제

1. 네트워크 포트와 스토리지 매핑 컨테이너 실행

- 실행 확인 화면 캡처하여 제출(3점)

  sudo docker ps 또는 docker ps

- miniconda 컨테이너가 보이고 PORT 매핑이 보이면 됨

```
esu@npark-dell:~$ docker ps
CONTAINER ID   IMAGE                    COMMAND       CREATED       STATUS       PORTS                                                NAMES
6942f87aa099   continuumio/miniconda3   "/bin/bash"   2 hours ago   Up 2 hours   0.0.0.0:8888->8888/tcp, :::8888->8888/tcp            miniconda
03ed6ed0960e   ubuntu:w5                "bash"        3 hours ago   Up 3 hours   0.0.0.0:9090->9090/tcp, :::9090->9090/tcp            happy_faraday
```

과제

2. 컨테이너에서 실행한 jupyter lab에 접속

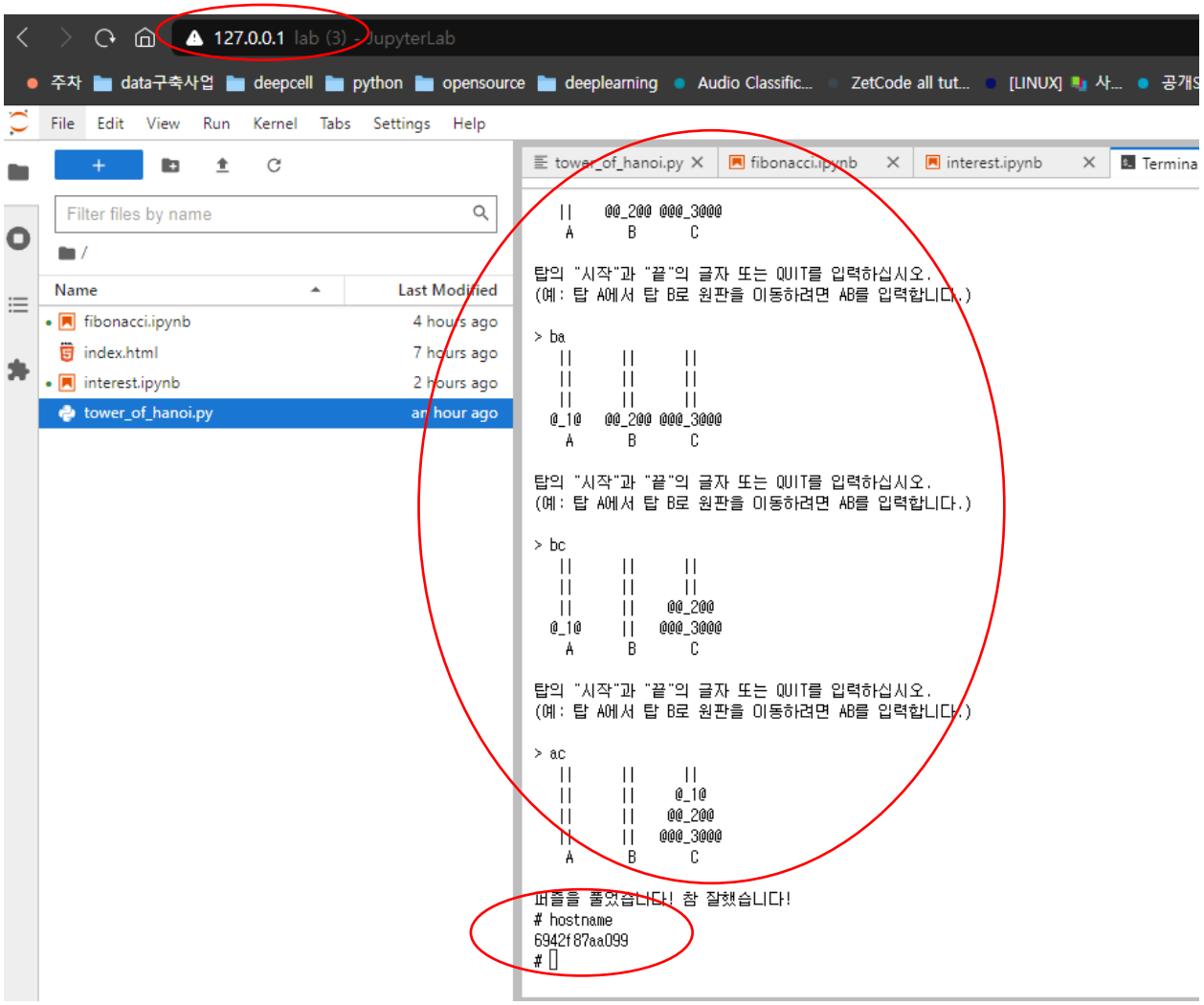
- 실행 확인 화면 캡처하여 제출(2점)

  웹 브라우저로 접속

  하노이탑 실행

  hostname 표시

- 다음 내용들 표시
  - url : 127.0.0.1 또는 localhost
  - 하노이탑 실행
  - hostname



과제

제출

- 다음 주 월요일(4월10일) 자정(24시)까지(KST;한국 표준시)
- 과제 내용 2개 캡처하여 보고서 붙여서 LMS에 제출
- LMS 제출은 LMS의 공지 확인하고 공지내용에 맞게 제출(조교에게 문의)