



인하공업전문대학
INHA TECHNICAL COLLEGE

TCP/IP 네트워크 프로그래밍 6주차

인하공업전문대학 컴퓨터 정보과
김한결 강사

- Docker 설치
 - ✓ 도커 설치
 - ✓ yml 환경 파일 수정
 - ✓ mysql 설치 및 워드프레스 설치
 - ✓ 워드프레스 실행
 - ✓ 워드프레스 수정

• 도커란?

도커 (소프트웨어)

위키백과, 우리 모두의 백과사전.

도커(Docker)는 리눅스의 응용 프로그램들을 소프트웨어 컨테이너 안에 배치시키는 일을 자동화하는 오픈 소스 프로젝트이다. 도커 웹 페이지의 기능을 인용하면 다음과 같다:

도커 컨테이너는 일종의 소프트웨어를 소프트웨어의 실행에 필요한 모든 것을 포함하는 완전한 파일 시스템 안에 감싼다. 여기에는 코드, 런타임, 시스템 도구, 시스템 라이브러리 등 서버에 설치되는 무엇이든 아우른다. 이는 실행 중인 환경에 관계 없이 언제나 동일하게 실행될 것을 보증한다.^[5]

도커는 리눅스에서 운영 체제 수준 가상화의 추상화 및 자동화 계층을 추가적으로 제공한다.^[6] 도커는 cgroups와 커널 이름공간과 같은 리눅스 커널, 또 aufs와 같은 유닉스 파일 시스템의 리소스 격리 기능을 사용하며,^[7] 이를 통해 독립적인 "컨테이너"가 하나의 리눅스 인스턴스 안에서 실행할 수 있게 함으로써 가상 머신을 시작하여 유지보수해야 하는 부담을 없애준다.^[8]

리눅스 커널의 이름공간 지원은 대체적으로^[9] 프로세스 트리, 네트워크 사용자 ID, 마운트된 파일 시스템을 포함한 운영 환경에 대한 응용 프로그램의 관점을 격리시키지만, 커널의 cgroup들은 CPU, 메모리, 블록 입출력, 네트워크를 포함한 리소스 제한을 제공한다. 버전 0.9부터 도커는 libvirt, LXC (리눅스 컨테이너), systemd-nspawn을 통한 추상화된 가상화 인터페이스를 사용하는 것 뿐 아니라 리눅스 커널이 제공하는 가상화 기능을 직접 사용하기 위한 유일한 수단으로 libcontainer 라이브러리를 포함하고 있다.^{[10][11][12]}

목차 [숨기기]

- 1 개요
 - 1.1 통합
- 2 역사
- 3 도구
 - 3.1 Docker Compose
 - 3.2 Docker Swarm
- 4 같이 보기
- 5 각주
- 6 외부 링크

도커 Docker



원저자	솔로몬 하익스(Solomon Hykes)
개발자	도커
발표일	2013년 3월 13일 (8년 전)
안정화 버전	20.10.2 / 2021년 1월 4일 (2개월 전) ^[1]
저장소	github.com/moby/moby
프로그래밍 언어	고 ^[2]
운영 체제 플랫폼	리눅스 ^[3] , 윈도우 현대의 리눅스 커널이 포함된 x86-64, ARM (실험적), 하이퍼-V 기능이 포함된 x86-64 윈도우
종류	운영 체제 수준 가상화
라이선스	아파치 라이선스 2.0
웹사이트	www.docker.com

출처 :

[https://ko.wikipedia.org/wiki/%EB%8F%84%EC%BB%A4_\(%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4\)](https://ko.wikipedia.org/wiki/%EB%8F%84%EC%BB%A4_(%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4))

• 일반설치환경 vs 도커

```
brew tap microsoft/mssql-release https://github.com/Microsoft/homebrew-mssql-release
brew update
HOMEBREW_NO_ENV_FILTERING=1 ACCEPT_EULA=y brew install mssql-tools
brew install unixodbc msodbcsql17
```

- Ubuntu

```
sudo apt install python3.9-dev
```

PostgreSQL 설치

[참고] <https://brownwarf.tistory.com/58>

설치 후, 다음과 같이 DB 및 DB 사용자를 생성한다. (Ubuntu 기준; Mac 은 조금 다를 TODO)

```
sudo -u postgres psql # PostgreSQL 실행
create user yolo password 'yolo';
create database "yolo" owner yolo;
grant all privileges on database yolo to yolo;
```

소스코드 다운

```
git clone https://github.com/321core/kefi-yolo
```

- /home/kefi_admin/kefi-yolo 에 설치, 이후 이 디렉토리에서 작업.
- 이미지 저장 위치 : 백스락 재설정 (/home/places/bexco/pir)

python 3.9로 설치 참고 URL

```
sudo apt install python3-pip
```

python3.9

pipenv 설치

```
sudo apt-get install -y python3.9-distutils
sudo -H pip3 install -U pipenv
pipenv sync
```

- 다음 명령으로 DB 스키마를 생성한다.

```
pipenv run python manage.py migrate
```

version: '3.1'

services:

wordpress:

image: wordpress

restart: always

ports:

- 8080:80

environment:

WORDPRESS_DB_HOST: db

WORDPRESS_DB_USER: exampleuser

WORDPRESS_DB_PASSWORD: examplepass

WORDPRESS_DB_NAME: exampledb

db:

image: mysql:5.7

restart: always

environment:

MYSQL_DATABASE: exampledb

MYSQL_USER: exampleuser

MYSQL_PASSWORD: examplepass

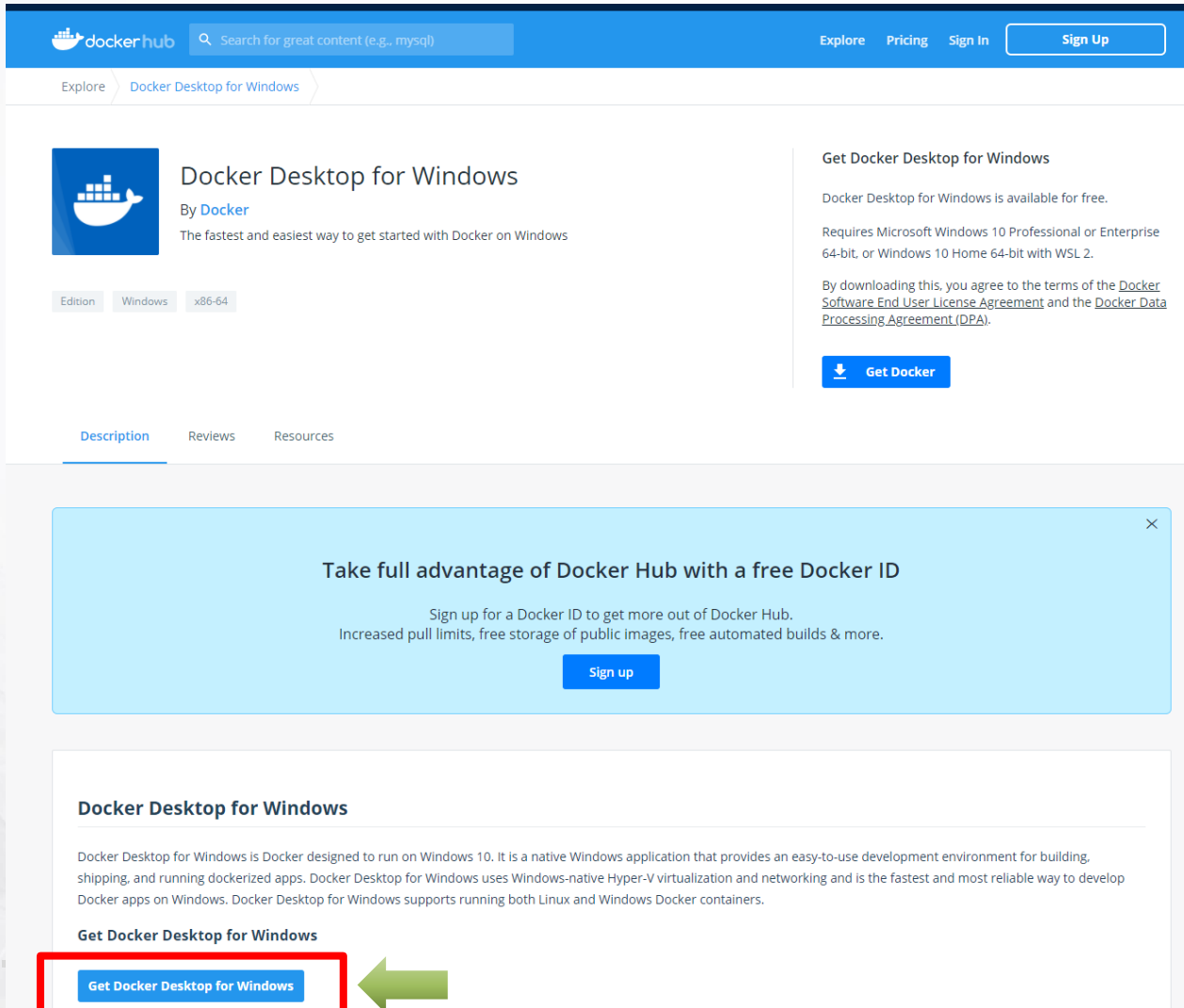
MYSQL_RANDOM_ROOT_PASSWORD: '1'



Docker 설치(윈도우)

• 도커 설치 - 1


다운로드 주소 : <https://hub.docker.com/editions/community/docker-ce-desktop-windows>



The screenshot shows the Docker Hub page for Docker Desktop for Windows. The page has a blue header with the Docker Hub logo and a search bar. Below the header, there's a section for 'Docker Desktop for Windows' with a Docker logo icon. The text 'Docker Desktop for Windows' is followed by 'By Docker' and 'The fastest and easiest way to get started with Docker on Windows'. There are tabs for 'Edition', 'Windows', and 'x86-64'. On the right, there's a section titled 'Get Docker Desktop for Windows' with text stating it's available for free and requires Microsoft Windows 10 Professional or Enterprise 64-bit, or Windows 10 Home 64-bit with WSL 2. It also mentions that by downloading, you agree to the terms of the Docker Software End User License Agreement and the Docker Data Processing Agreement (DPA). A blue 'Get Docker' button is present. At the bottom, there's a light blue banner with the text 'Take full advantage of Docker Hub with a free Docker ID' and a 'Sign up' button. Below the banner, there's a section titled 'Docker Desktop for Windows' with a description of the application. At the very bottom, there's a blue button labeled 'Get Docker Desktop for Windows' which is highlighted with a red box and a green arrow pointing to it.

dockerhub Search for great content (e.g., mysql) Explore Pricing Sign In Sign Up

Explore Docker Desktop for Windows

 **Docker Desktop for Windows**
By [Docker](#)
The fastest and easiest way to get started with Docker on Windows

Edition Windows x86-64


[Description](#) [Reviews](#) [Resources](#)

Get Docker Desktop for Windows

Docker Desktop for Windows is available for free.

Requires Microsoft Windows 10 Professional or Enterprise 64-bit, or Windows 10 Home 64-bit with WSL 2.

By downloading this, you agree to the terms of the [Docker Software End User License Agreement](#) and the [Docker Data Processing Agreement \(DPA\)](#).

 **Get Docker**

Take full advantage of Docker Hub with a free Docker ID

Sign up for a Docker ID to get more out of Docker Hub.
Increased pull limits, free storage of public images, free automated builds & more.

[Sign up](#)

Docker Desktop for Windows

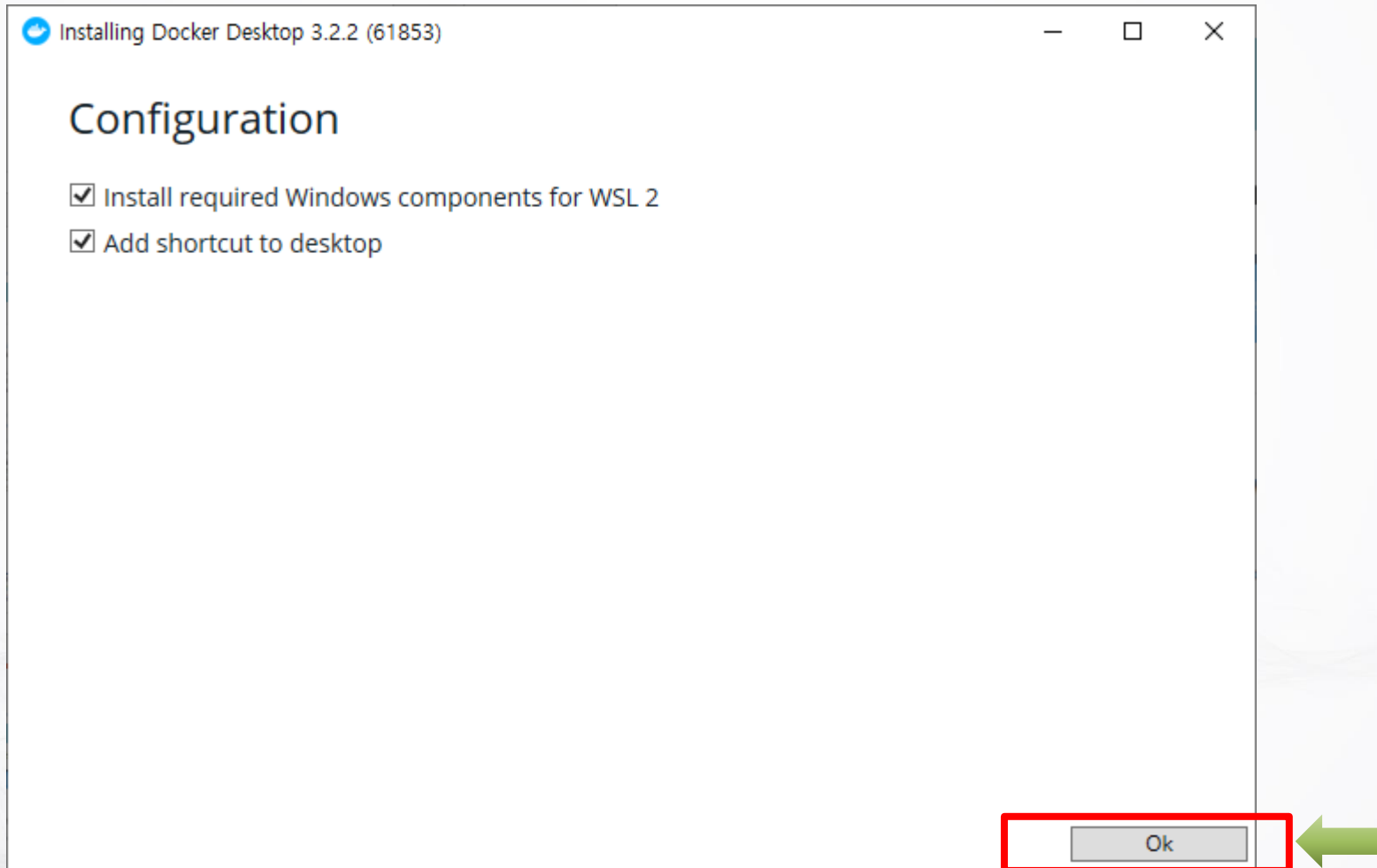
Docker Desktop for Windows is Docker designed to run on Windows 10. It is a native Windows application that provides an easy-to-use development environment for building, shipping, and running dockerized apps. Docker Desktop for Windows uses Windows-native Hyper-V virtualization and networking and is the fastest and most reliable way to develop Docker apps on Windows. Docker Desktop for Windows supports running both Linux and Windows Docker containers.

Get Docker Desktop for Windows

[Get Docker Desktop for Windows](#)

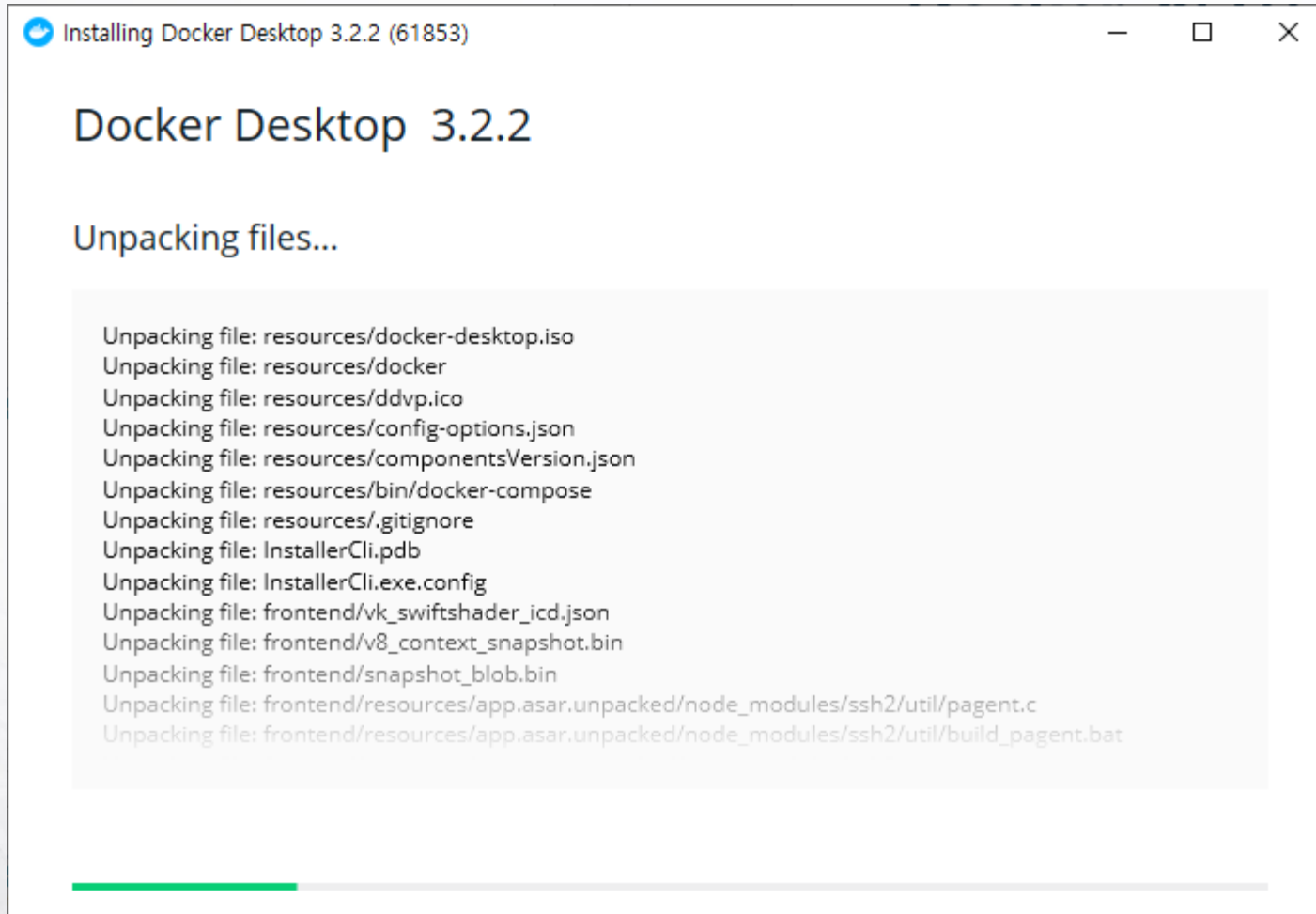
Docker 설치(윈도우)

• 도커 설치 - 2



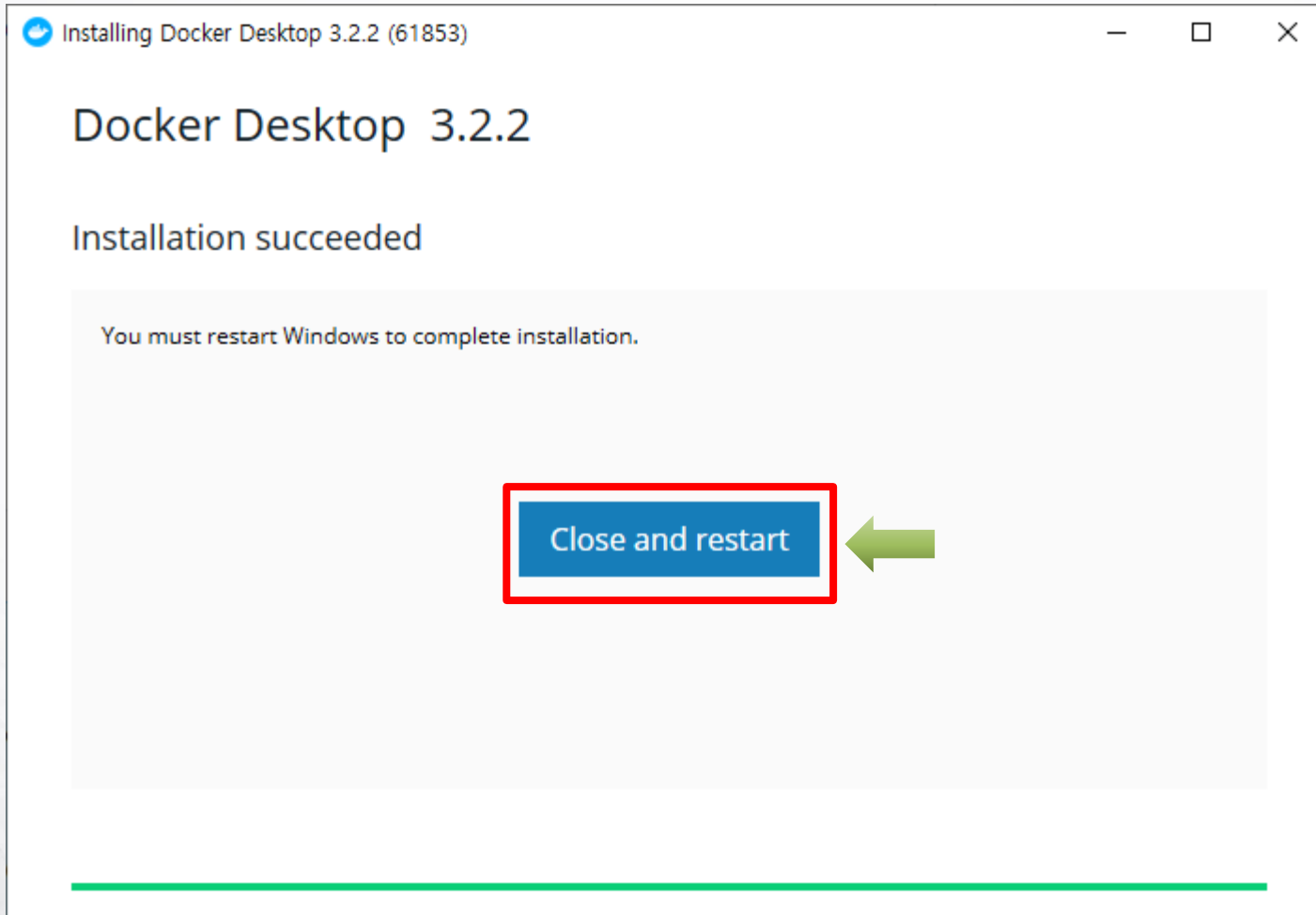
Docker 설치(윈도우)

• 도커 설치 - 3



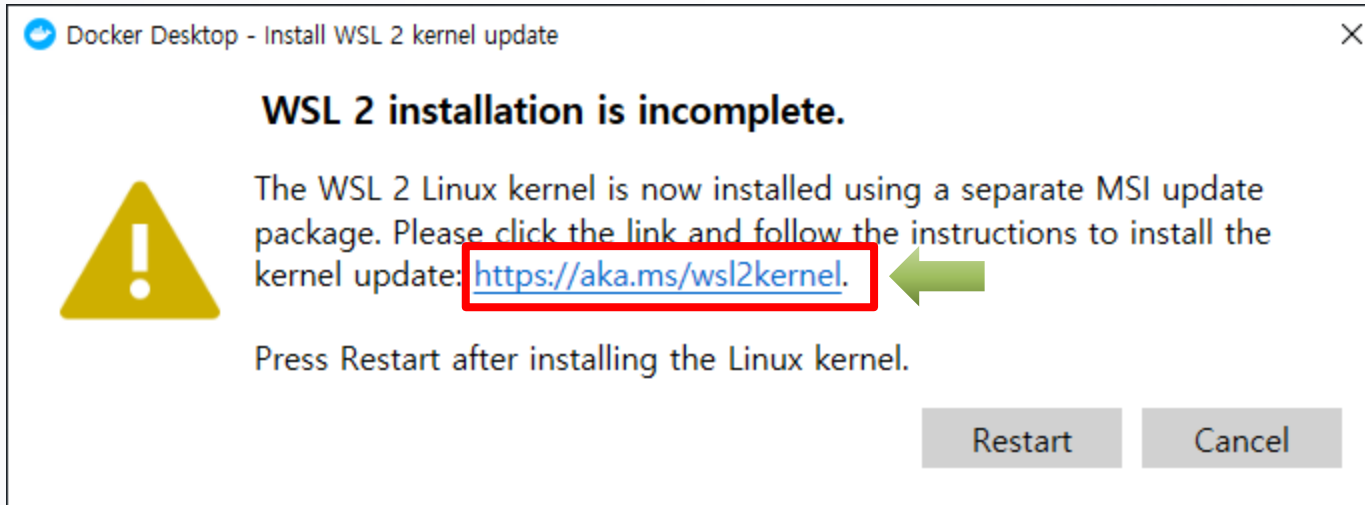
Docker 설치(윈도우)

• 도커 설치 - 4



Docker 설치(윈도우)

• 도커 설치 - 5



Docker 설치(윈도우)

• 도커 설치 - 6

Windows 10에 WSL(Linux용 Win...) x +

→ ↻ 🏠 <https://docs.microsoft.com/ko-kr/windows/wsl/install-win10#step-4---download-the-linux-kernel-update-package>

Waver Baekjoon Online Ju... YBM 토익인강 공식... 이거 빌트블건 1:2... Docker Desktop - Install WSL 2 kernel update

WSL 2 installation is incomplete.

The WSL 2 Linux kernel is now installed using a separate MSI update package. Please click the link and follow the instructions to install the kernel update: <https://aka.ms/wsl2kernel>.

Press Restart after installing the Linux kernel.

Restart Cancel

PowerShell

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

머신을 다시 시작하여 WSL 설치를 완료하고 WSL 2로 업데이트합니다.

4단계 - Linux 커널 업데이트 패키지 다운로드

최신 패키지를 다운로드합니다.

- x64 머신용 최신 WSL2 Linux 커널 업데이트 패키지

① 참고

ARM64 머신을 사용하는 경우 ARM64 패키지 를 대신 다운로드하세요. 사용하고 있는 머신의 종류를 잘 모르는 경우 명령 프롬프트 또는 PowerShell을 열고 `systeminfo | find "System Type"`을 입력합니다.

2. 이전 단계에서 다운로드한 업데이트 패키지를 실행합니다. (실행하려면 두 번 클릭 - 관리자 권한을 요구하는 메시지가 표시되면 '예'를 선택하여 이 설치를 승인합니다.)

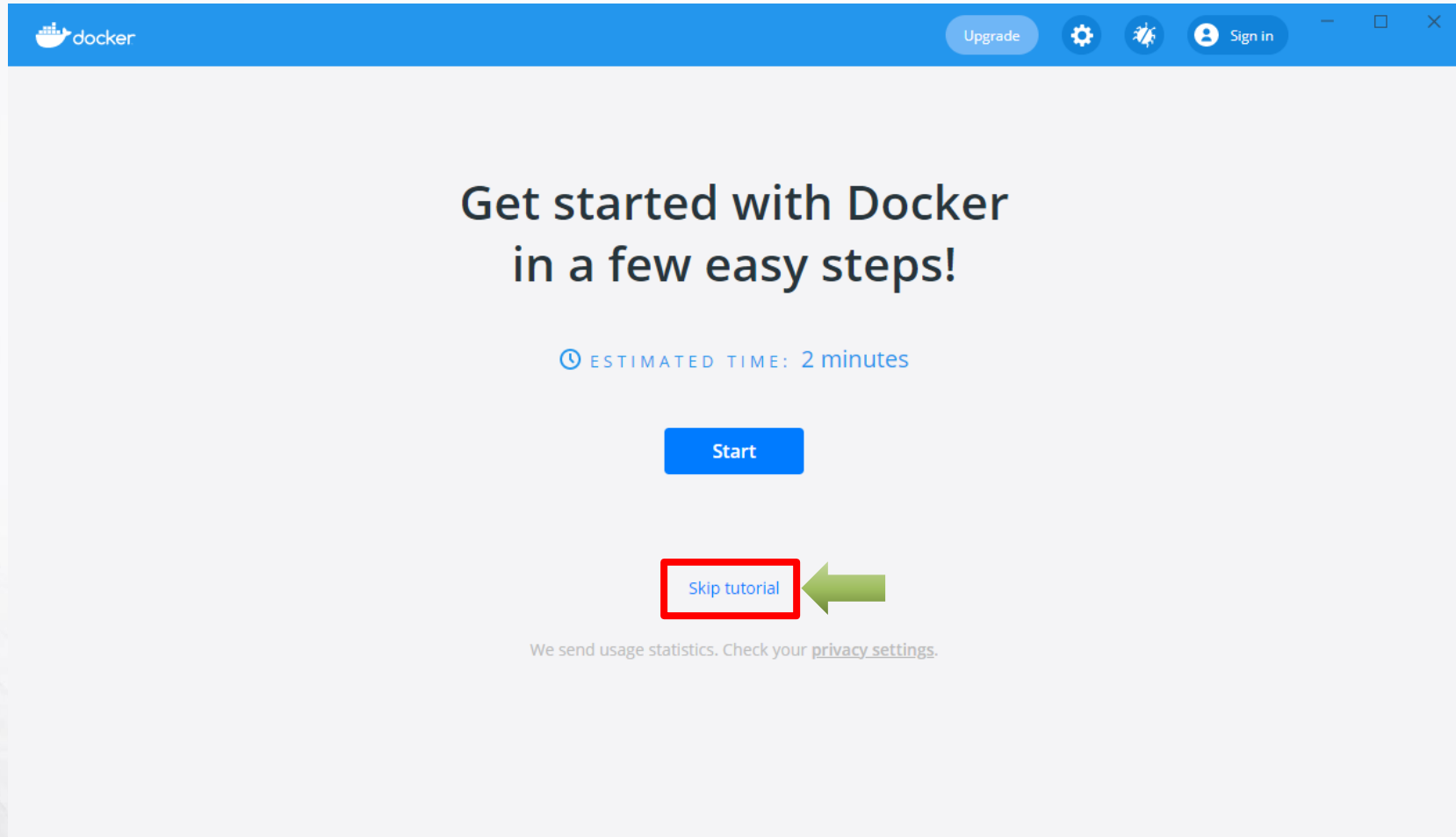
설치가 완료되면 새 Linux 배포를 설치할 때 WSL 2를 기본 버전으로 설정하는 다음 단계로 이동합니다. (새 Linux 설치를 WSL 1로 설정하려면 이 단계를 건너뛸니다.)

① 참고

자세한 내용은 Windows 명령줄 블로그 에서 사용할 수 있는 WSL2 Linux 커널업데이트 변경 를 문서를 참조하세요.

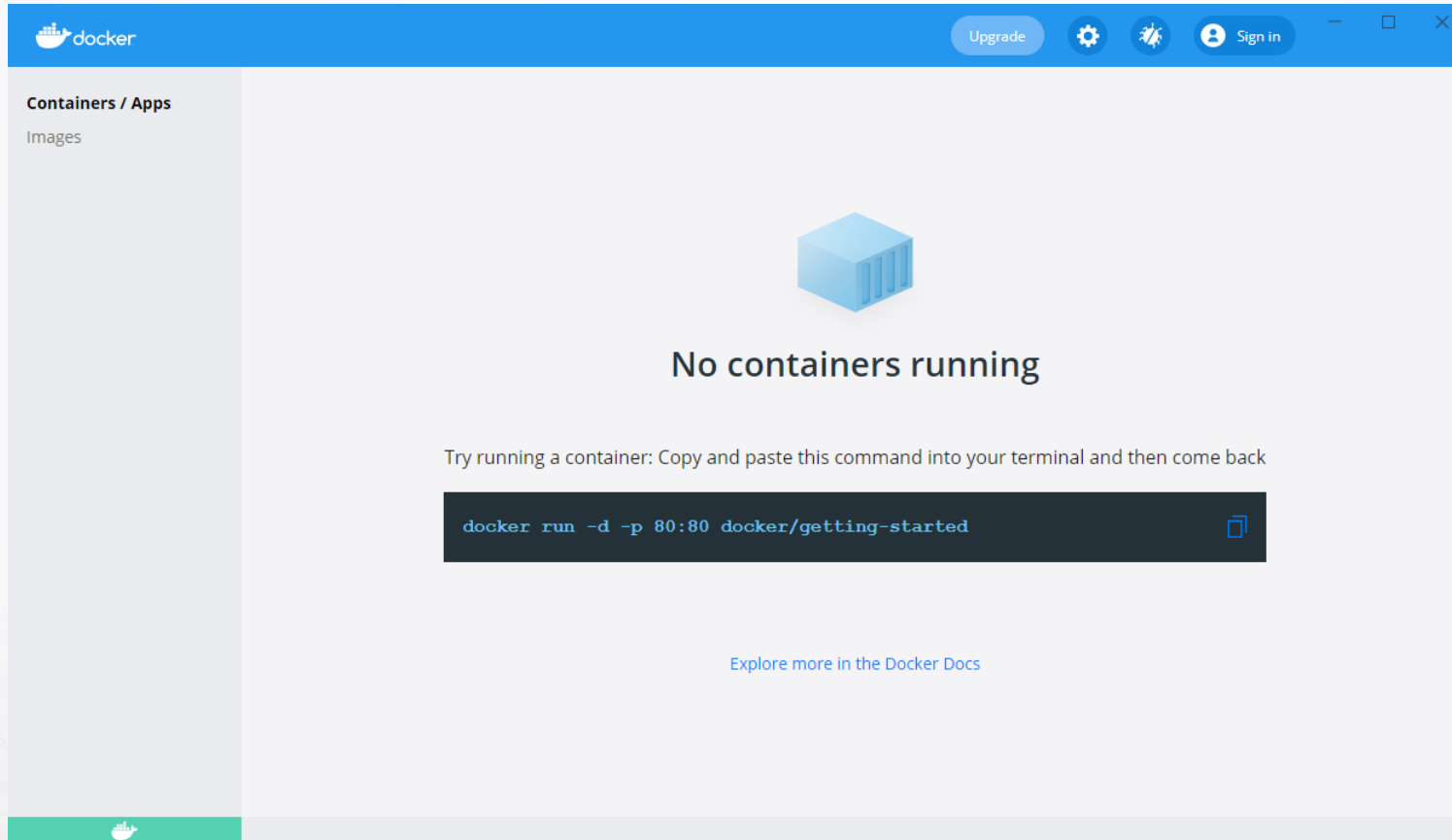
Docker 설치(윈도우)

• 도커 실행 화면



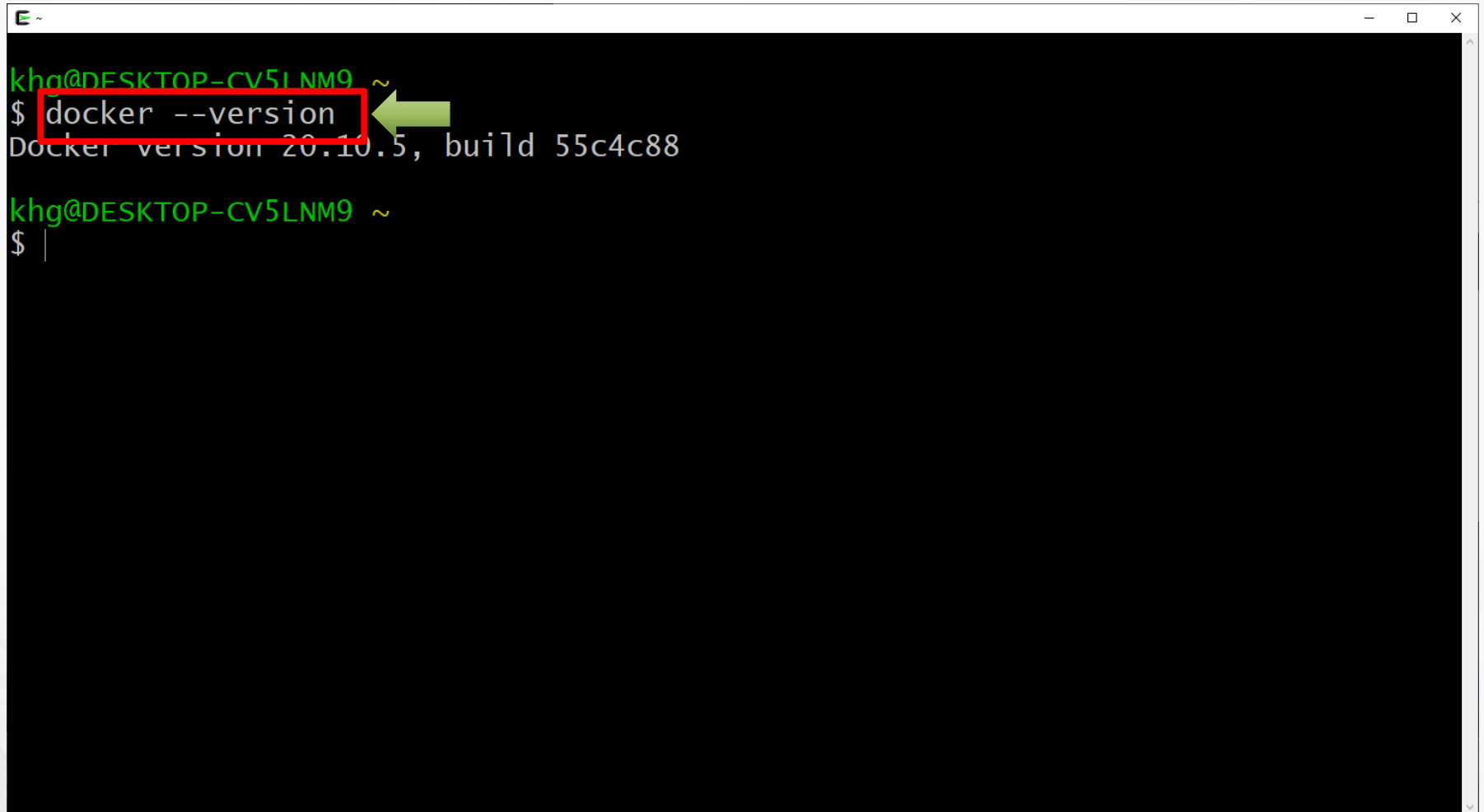
Docker 설치(윈도우)

- 도커 실행 화면



Docker 설치(윈도우)

- 시그윈 실행

A terminal window with a black background and green text. The prompt is 'khg@DESKTOP-CV5LNM9 ~'. The command 'docker --version' is entered and highlighted with a red rectangle. A green arrow points to the command. The output is 'Docker version 20.10.5, build 55c4c88'. The prompt returns to '\$ |' on the next line.

```
khg@DESKTOP-CV5LNM9 ~  
$ docker --version  
Docker version 20.10.5, build 55c4c88  
  
khg@DESKTOP-CV5LNM9 ~  
$ |
```

- docker-compose.yml

```
version: '3.1'

services:
  wordpress:
    image: wordpress
    restart: always
    ports:
      - 8080:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exampledb
    volumes:
      - wordpress:/var/www/html

  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_DATABASE: exampledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql

volumes:
  wordpress:
  db:
```

<https://github.com/sonnonet/inhatc/>

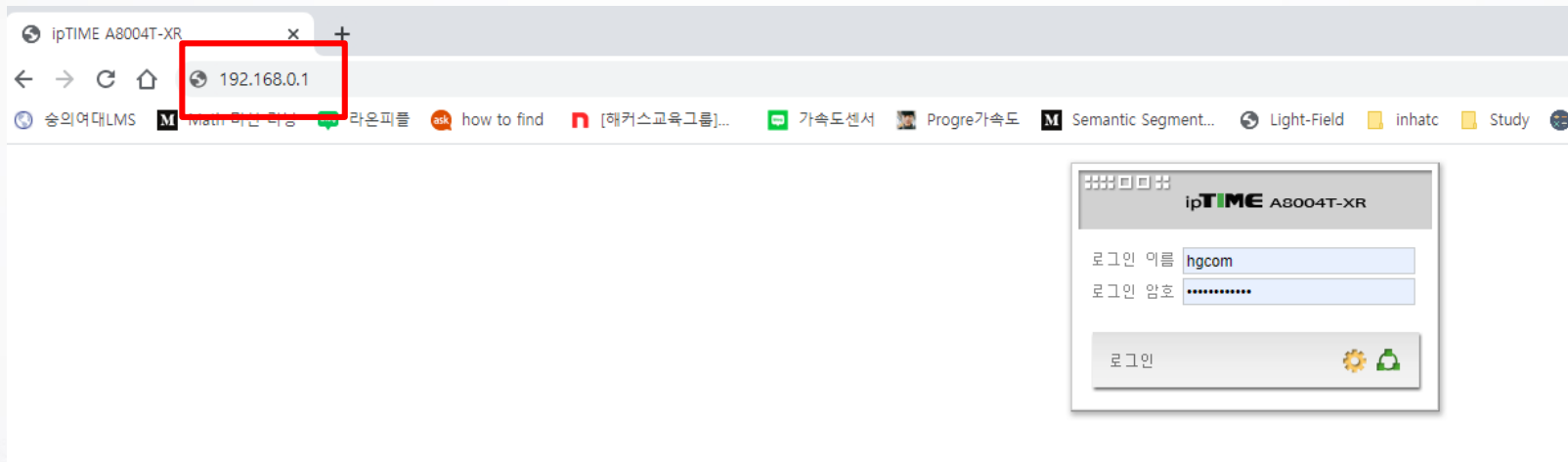
- docker-compose.yml 내용 복사
- Vim 으로 파일 오픈
 - > vim docker-compose.yml
- 붙여넣기
- 실행 : docker-compose up -d
- 웹브라우저 실행 후
- 주소창에 127.0.0.1:8080 or 자신의 컴퓨터 IP주소:8080
 - ipconfig

- 웹서버 호스팅
 - ✓ 공유기 접속
 - ✓ 외부 IP 주소 확인
 - ✓ 포트포워딩
 - ✓ 외부접속
 - ✓ 도메인 서비스

웹서버 호스팅

• 내 컴퓨터 웹서버 호스팅 서비스

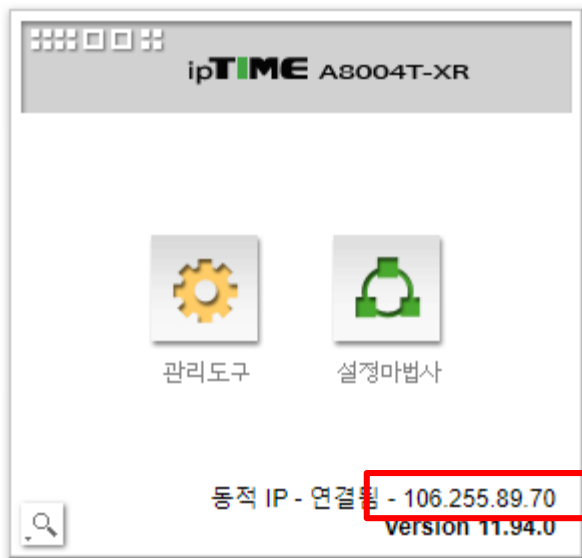
- IP 공유기 주소 – 192.168.0.1 ex) IP 공유기
 - 로그인



웹서버 호스팅

• 내 컴퓨터 웹서버 호스팅 서비스 – 외부IP

- 외부 IP 주소 확인



• 내 컴퓨터 웹서버 호스팅 서비스 – 포트포워딩

• 고급설정 -> 포트포워딩 설정

ipTIME A8004T-XR

메뉴탐색기

- 기본 설정
 - 시스템 요약 정보
 - 인터넷 설정 정보
 - 무선 설정/보안
 - 편웨어 업그레이드
 - Easy Mesh
- 고급 설정
 - 네트워크 관리
 - 무선랜 관리
 - NAT/라우터 관리
 - 포트포워딩 설정**
 - 고급 NAT 설정
 - 라우팅 테이블 관리
 - 보안 기능
 - 특수기능
 - 트래픽 관리
 - 시스템 관리
 - USB/서비스 관리

포트포워딩 설정

순위	사용자 규칙	내부 IP	외부 포트	내부 포트	삭제
1	sncloud	192.168.0.49	TCP(30225~30225)	TCP(30225~30225)	<input type="checkbox"/>
2	sncloud_web	192.168.0.55	TCP(80~80)	TCP(80~80)	<input type="checkbox"/>
3	sncloud_we...	192.168.0.49	TCP(443~443)	TCP(443~443)	<input type="checkbox"/>
4	infoRPI	192.168.0.119	TCP(6667~6667)	TCP(22~22)	<input type="checkbox"/>
5	infoRPI_AI	192.168.0.79	TCP(7777~7777)	TCP(22~22)	<input type="checkbox"/>
6	sncloud_por...	192.168.0.49	TCP(9000~9000)	TCP(9000~9000)	<input type="checkbox"/>
7	sncloud_influx	192.168.0.49	TCP(30226~30226)	TCP(8086~8086)	<input type="checkbox"/>
8	sncloud_gra...	192.168.0.49	TCP(30227~30227)	TCP(3000~3000)	<input type="checkbox"/>
9	fitpc_55	192.168.0.55	TCP(5502)	TCP(22)	<input type="checkbox"/>
10	fitpc_55_5503	192.168.0.55	TCP(5503)	TCP(3389)	<input type="checkbox"/>
11	sncloud_web1	192.168.0.49	TCP(8080~8080)	TCP(80~80)	<input type="checkbox"/>
12	web_service	192.168.0.25	TCP(9090~9090)	TCP(9090~9090)	<input type="checkbox"/>

새규칙 추가

규칙이름: 포트포워딩 사용자정의 ☐ 규칙 비활성화

내부 IP주소: 192.168.0. 현재 접속된 IP 주소 ☐

프로토콜: TCP 외부 포트: ~ 내부 포트: ~

순위:

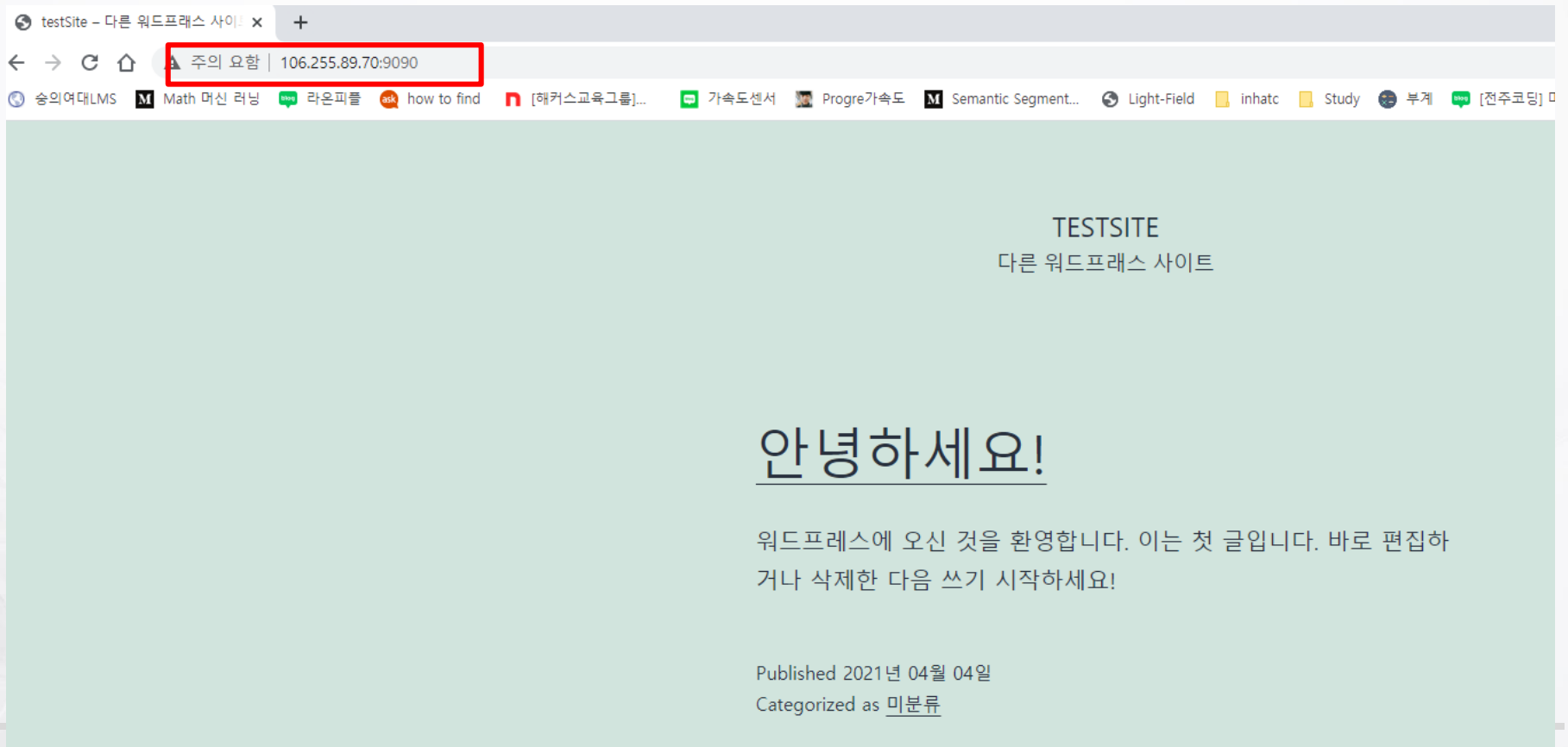
▲순위높임 ▼순위낮춤

PC<-규칙저장 PC->규칙복원 파일 선택 선택된 파일 없음 새규칙 적용 취소

웹서버 호스팅

• 내 컴퓨터 웹서버 호스팅 서비스 – 외부접속

- 외부 접속
 - 외부IP:포트번호 ex) 106.255.89.70:9090



• 내 컴퓨터 웹서버 호스팅 서비스 – 도메인등록

WH?IS 도메인

대한민국 도메인 등록 1위
국내 유일 일반최상위도메인 관리기관

SMS 최대 47% 할인!
저렴하고 편리하게 문자 발송하세요.

[신규등록](#) | [등록기간 연장](#) | [변경 / 이전](#) | [도메인 활용/부가서비스](#) | [브랜드 관리/컨설팅](#) | [내 도메인 자산 관리](#) | [1:1 친절상담](#) | [전체보기](#)

[로그인](#) | [회원가입](#) | 로그인 없이 도메인 연장하기 | 장바구니: 0

'coffee' 등 다양한 신규 도메인을 선택하세요.

Domain Registration
신규등록

[도메인 검색 / 등록](#)

[국가별 도메인 검색](#)

[업종별 도메인 검색](#)

[대량입력 검색](#)

[Whois 정보 검색](#)

도메인 검색 / 등록

도메인 검색결과 확인

도메인 검색 → 도메인 선택 → 등록신청서 작성 → 결제 → 신청완료

- 전세계 실시간 검색결과로, 등록가능 도메인은 누구든지 먼저 등록할 수 있습니다.
- 도메인 선택 후 '등록신청' 버튼을 눌러 도메인별 **할인가격**을 확인하고 등록하세요.

HOME > 신규등록 > 도메인 검색 / 등록

전화한 동안엔 신청에서 결제까지 끝!
1588-4259 (ARS 1번)

30개 도메인 더 보기

총 410개 도메인 중 7개 검색결과입니다.

☐ 등록가능 ☒ 검색중 ☐ 검색불가 ☒ 등록불가 ☐ 가격유류

도메인 2개 이상 등록하시고
스타벅스 카피드세요.

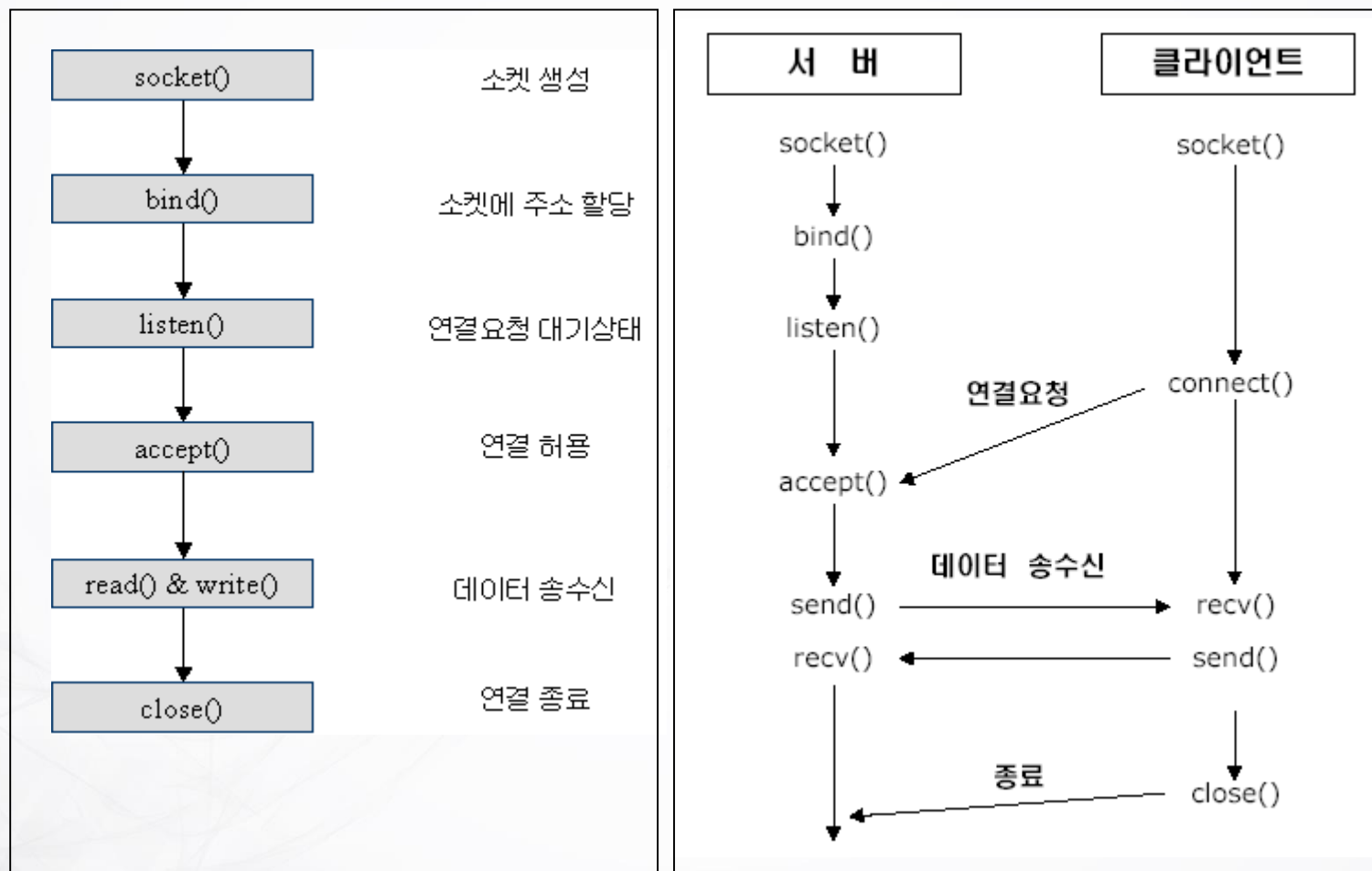
[장바구니 담기](#) [견적서 신청](#) [등록신청 >](#)



- TCP기반 서버, 클라이언트 구현
- 네트워크 데이터 송수신 주체
- 프로토콜
- TCP/IP 프로토콜
- 클라우드 서비스

TCP기반 서버, 클라이언트 구현

- TCP 서버에서의 기본적인 함수 호출 순서



리눅스 파일 디스크립터

- 리눅스에서의 소켓조작은 파일조작과 동일
- 소켓을 파일의 일종으로 구분
- 리눅스에서 제공
- 표준 파일 디스크립터는 별도의 생성과정을 거치지 않아도 프로그램을 실행하면 자동으로 할당되는 파일 디스크립터

파일 디스크립터	대 상
0	표준입력: Standard Input
1	표준출력 : Standard Output
2	표준에러 : Standard Error

파일 디스크립터 생성 예제 - 1

- low_open.c

```
~/socket_programing
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <fcntl.h>
4 #include <unistd.h>
5
6 void error_handling(char * message);
7
8 int main(void)
9 {
10     int fd;
11     char buf[]="Let's go!\n";
12     fd=open("data.txt", O_CREAT|O_WRONLY|O_TRUNC);
13     if(fd==-1)
14         error_handling("open() error!");
15     printf("File descriptor : %d \n", fd);
16
17     if(write(fd, buf, sizeof(buf))==-1)
18         error_handling("write() error!");
19     close(fd);
20     return 0;
21 }
22
23 void error_handling(char *message)
24 {
25     fputs(message, stderr);
26     fputc('\n', stderr);
27     exit(1);
28 }
```

12,1-4

꼭 대 기

파일 디스크립터 생성 예제 - 2

- low_read.c

```
~/socket_programing
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <fcntl.h>
4 #include <unistd.h>
5 #define BUF_SIZE 100
6
7 void error_handling(char * message);
8
9 int main(void)
10 {
11     int fd;
12     char buf[BUF_SIZE];
13
14     fd=open("data.txt", O_RDONLY);
15     if(fd==-1)
16         error_handling("open() error!");
17     printf("File descriptor : %d \n", fd);
18
19     if(read(fd, buf, sizeof(buf))==-1)
20         error_handling("read() error!");
21     printf("file data: %s", buf);
22     close(fd);
23     return 0;
24 }
25
26 void error_handling(char *message)
27 {
28     fputs(message, stderr);
29     fputc('\n', stderr);
30     exit(1);
31 }
32
33
```

파일 디스크립터 생성 예제 - 3

- fd_seri.c

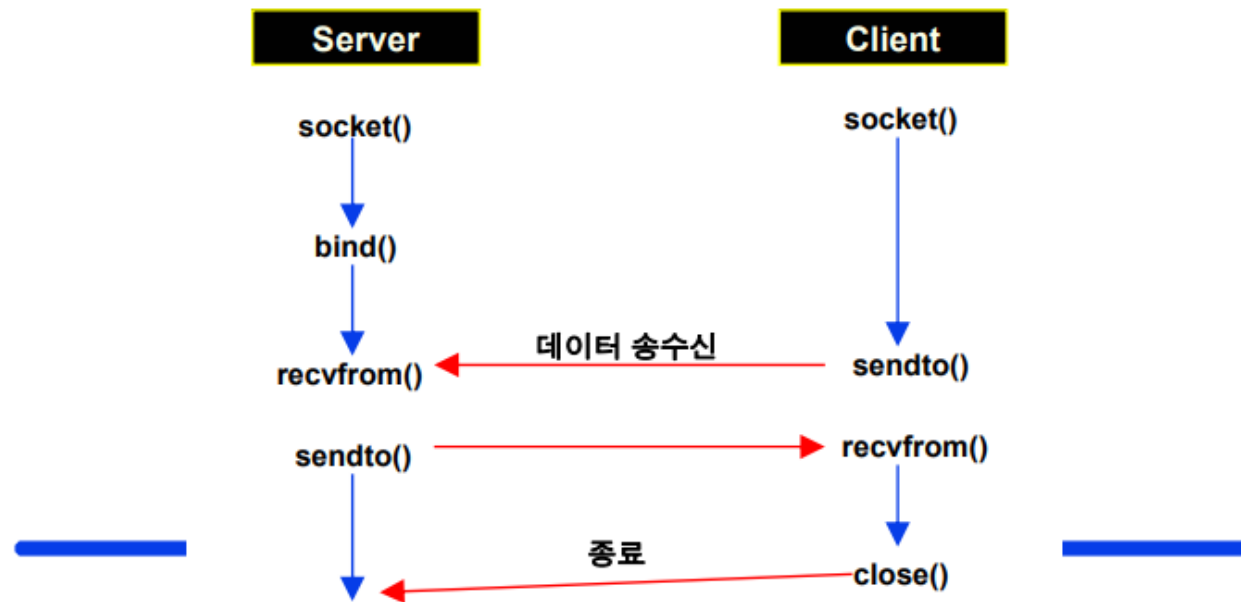
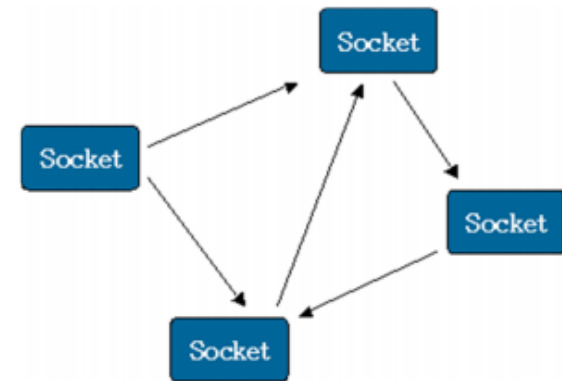
```
~/socket_programing
1 #include <stdio.h>
2 #include <fcntl.h>
3 #include <unistd.h>
4 #include <sys/socket.h>
5
6 int main(void)
7 {
8     int fd1,fd2,fd3;
9     fd1 = socket(PF_INET, SOCK_STREAM, 0);
10    fd2 = open("test.dat",O_CREAT|O_WRONLY|O_TRUNC);
11    fd3 = socket(PF_INET, SOCK_DGRAM, 0);
12
13    printf("file descriptor 1: %d\n", fd1);
14    printf("file descriptor 2: %d\n", fd2);
15    printf("file descriptor 3: %d\n", fd3);
16
17    close(fd1);
18    close(fd2);
19    close(fd3);
20
21    return 0;
22 }
23 }
```

4,1 모 두

◆ TCP와 달리 일 대 일 통신에만 사용되지 않음

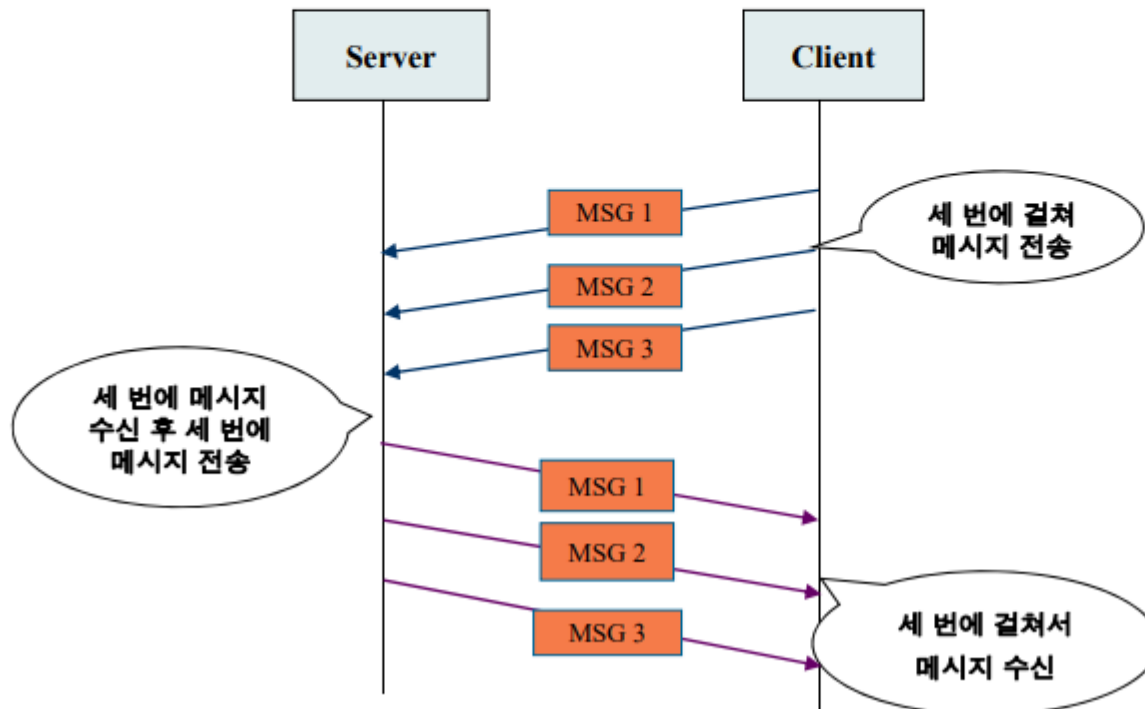
◆ 비연결형 소켓

- 일반적으로 연결 설정 과정을 거치지 않음
- 데이터를 위한 소켓은 하나만 개설
- 소켓 개설 후 바로 상대방과 데이터를 송수신



UDP 소켓 (데이터 경계가 존재함)

- ❑ 한쪽이 `sendto()`를 호출했으면 이 데이터를 받기 위해서 상대방은 반드시 `recvfrom()`을 호출해야 함
- ❑ TCP 소켓에서는 스트림을 이용하므로 `write()`나 `send()`로 연속하여 쓰기를 수행하거나 `read()`나 `recv()`로 연속하여 읽기를 해도 문제가 되지 않음
- ❑ UDP에서는 `sendto()` 와 `recvfrom()` 호출이 서로 짝을 이루도록 순서가 맞아야 함



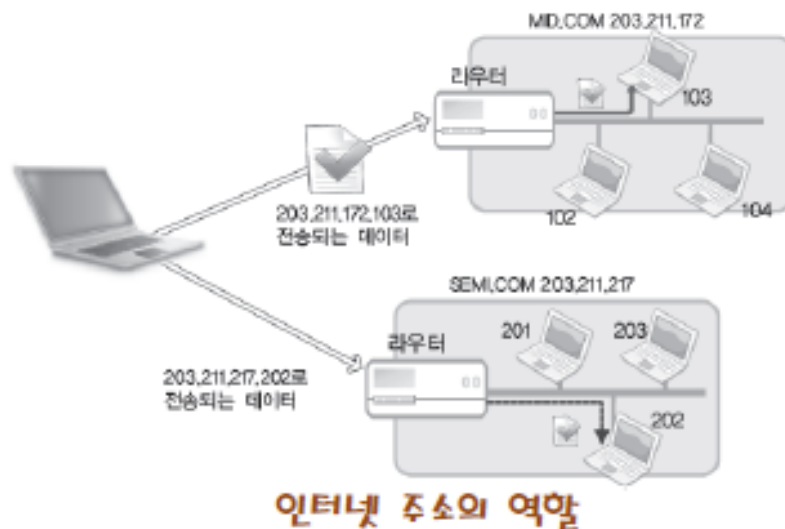
인터넷 주소(Internet Address)

▶ 인터넷 주소란?

- ▶ 인터넷상에서 컴퓨터를 구분하는 목적으로 사용되는 주소.
- ▶ 4바이트 주소체계인 IPv4와 16바이트 주소체계인 IPv6가 존재한다.
- ▶ 소켓을 생성할 때 기본적인 프로토콜을 지정해야 한다.
- ▶ 네트워크 주소와 호스트 주소로 나뉜다. 네트워크 주소를 이용해서 네트워크를 찾고, 호스트 주소를 이용해서 호스트를 구분한다.



IPv4 인터넷 주소의 체계



클래스 별 네트워크 주소와 호스트 주소의 경계

- 클래스 A의 첫 번째 바이트 범위 0이상 127이하
- 클래스 B의 첫 번째 바이트 범위 128이상 191이하
- 클래스 C의 첫 번째 바이트 범위 192이상 223이하



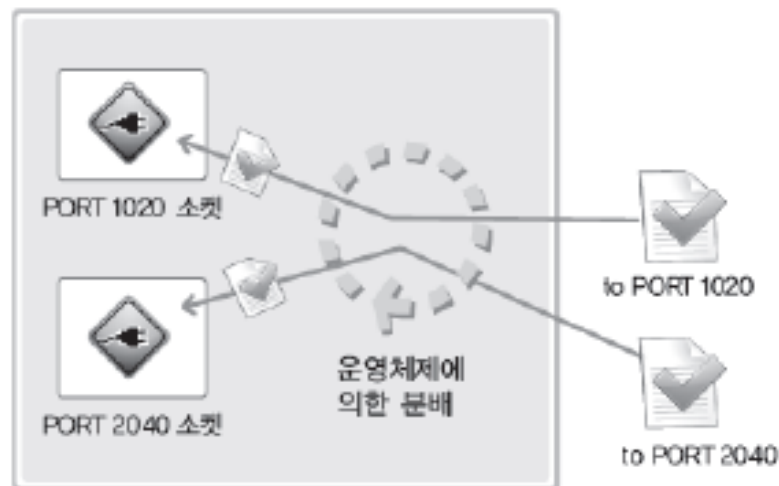
달리 말하면...

- 클래스 A의 첫 번째 비트는 항상 0으로 시작
- 클래스 B의 첫 두 비트는 항상 10으로 시작
- 클래스 C의 첫 세 비트는 항상 110으로 시작

때문에 첫 번째 바이트 정보만 참조해도 IP주소의 클래스 구분이 가능하며,
이로 인해서 네트워크 주소와 호스트 주소의 경계 구분이 가능하다.

▶ PORT번호

- ▶ IP는 컴퓨터를 구분하는 용도로 사용되며, PORT번호는 소켓을 구분하는 용도로 사용된다.
- ▶ 하나의 프로그램 내에서는 둘 이상의 소켓이 존재할 수 있으므로, 둘 이상의 PORT가 하나의 프로그램에 의해 할당될 수 있다.
- ▶ PORT번호는 16비트로 표현, 따라서 그 값은 0 이상 65535 이하
- ▶ 0~1023은 잘 알려진 PORT(Well-known PORT)라 해서 이미 용도가 결정되어 있다.



PORT번호에 의한 소켓의 구분과정

- IPV4 기반의 주소표현을 위한 구조체
- 구조체 Socketaddr_in 활용
- 바이트 순서와 네트워크 바이트 순서
- 과제 (바이트 변환의 예)
- IP주소를 정수로 변환

IPV4 기반의 주소표현을 위한 구조체

```
struct sockaddr_in
{
    sa_family_t    sin_family; 주소체계
    uint16_t       sin_port; PORT번호
    struct in_addr sin_addr; 32비트 IP주소
    char           sin_zero[8]; 사용되지 않음
};
```

IP주소와 PORT번호는 구조체 `sockaddr_in`의 변수에 담아서 표현한다.

```
struct in_addr
{
    in_addr_t      s_addr;
};
```

32비트 IPv4 인터넷 주소

자료형 이름	자료형에 담길 정보
int8_t	signed 8-bit int
uint8_t	unsigned 8-bit int (unsigned char)
int16_t	signed 16-bit int
uint16_t	unsigned 16-bit int (unsigned short)
int32_t	signed 32-bit int
uint32_t	unsigned 32-bit int (unsigned long)
sa_family_t	주소체계(address family)
socklen_t	길이정보(length of struct)
in_addr_t	IP주소정보, uint32_t로 정의되어 있음
in_port_t	PORT번호정보, uint16_t로 정의되어 있음

구조체 sockaddr_in의 활용의 예

```
struct sockaddr_in serv_addr;  
.  
.  
.  
if(bind(serv_sock, (struct sockaddr*) &serv_addr, sizeof(serv_addr))==-1)  
    error_handling("bind() error");  
.  
.  
.
```



구조체 변수 `sockaddr_in`은 `bind` 함수의 인자로 전달되는데, 매개변수 형이 `sockaddr`이므로 형 변환을 해야만 한다.

```
struct sockaddr  
{  
    sa_family_t    sin_family;    // 주소체계(Address Family)  
    char           sa_data[14];   // 주소정보  
};
```



구조체 `sockaddr`은 다양한 주소체계의 주소정보를 담을 수 있도록 정의되었다. 그래서 `IPv4`의 주소정보를 담기가 불편하다. 이에 동일한 바이트 열을 구성하는 구조체 `sockaddr_in`이 정의되었으며, 이를 이용해서 쉽게 `IPv4`의 주소정보를 담을 수 있다.

구조체 sockaddr_in의 멤버에 대한 분석

- ▶ 멤버 sin_family
 - ▶ 주소체계 정보 저장

- ▶ 멤버 sin_port
 - ▶ 16비트 PORT번호 저장
 - ▶ 네트워크 바이트 순서로 저장

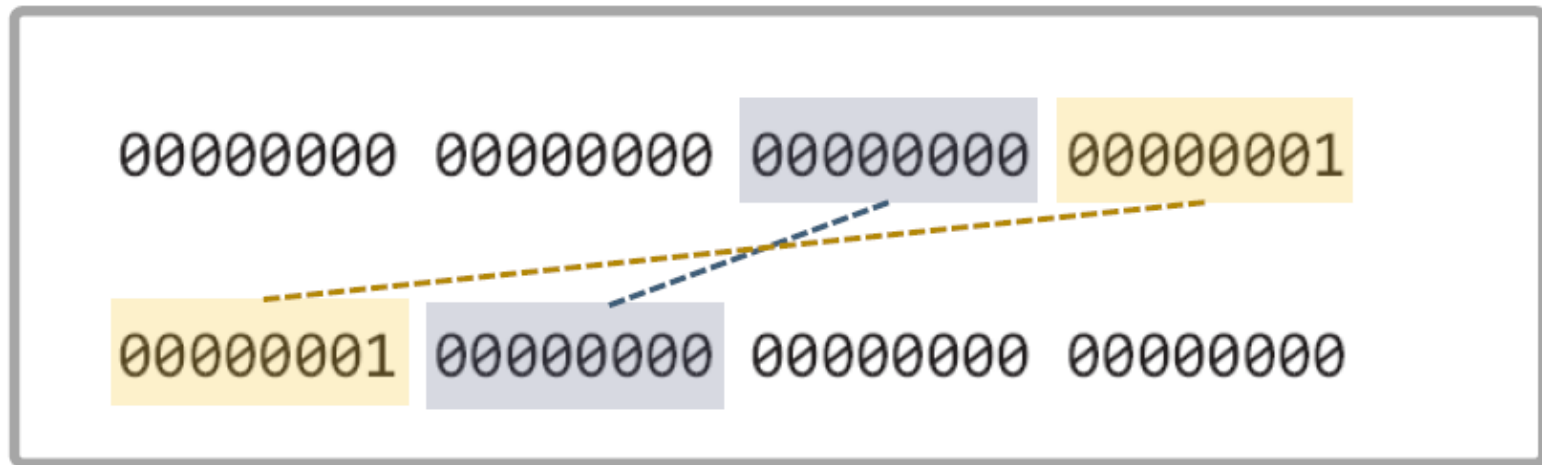
주소체계(Address Family)	의 미
AF_INET	IPv4 인터넷 프로토콜에 적용하는 주소체계
AF_INET6	IPv6 인터넷 프로토콜에 적용하는 주소체계
AF_LOCAL	로컬 통신을 위한 유닉스 프로토콜의 주소체계

- ▶ 멤버 sin_addr
 - ▶ 32비트 IP주소정보 저장
 - ▶ 네트워크 바이트 순서로 저장
 - ▶ 멤버 sin_addr의 구조체 자료형 in_addr 사실상 32비트 정수자료형
- ▶ 멤버 sin_zero
 - ▶ 특별한 의미를 지니지 않는 멤버
 - ▶ 반드시 0으로 채워야 한다.

CPU에 따라 달라지는 정수의 표현

네트워크 바이트 순서와 인터넷 주소 변환

정수 1을 저장하는 두 가지 방법 : CPU가 적용하는....

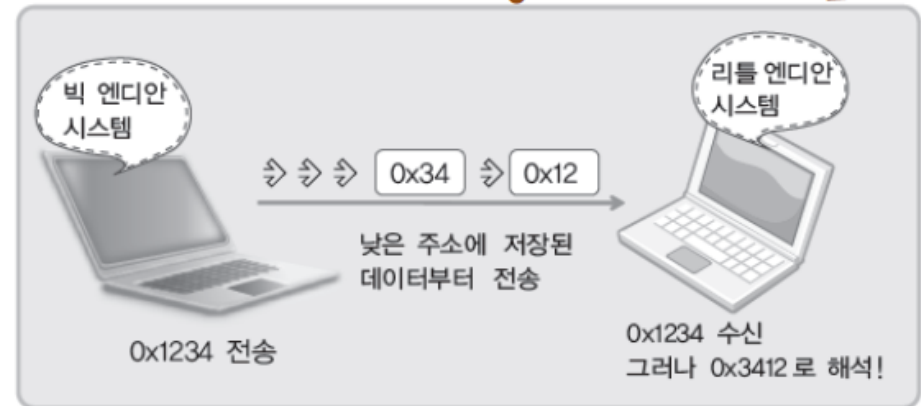


CPU에 따라서 상위 바이트를 하위 메모리 주소에 저장하기도 하고, 상위 바이트를 상위 메모리 주소에 저장하기도 한다. 즉, CPU마다 데이터 표현 및 해석하는 방식이 다르다!

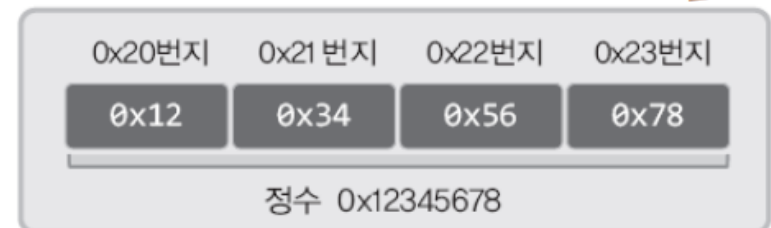
바이트 순서(Order)와 네트워크 바이트 순서

- ▶ 빅 엔디안(Big Endian)
 - ▶ 상위 바이트의 값을 작은 번지수에 저장
- ▶ 리틀 엔디안(Little Endian)
 - ▶ 상위 바이트의 값을 큰 번지수에 저장
- ▶ 호스트 바이트 순서
 - ▶ CPU별 데이터 저장방식을 의미함
- ▶ 네트워크 바이트 순서
 - ▶ 통일된 데이터 송수신 기준을 의미함
 - ▶ 빅 엔디안이 기준이다!

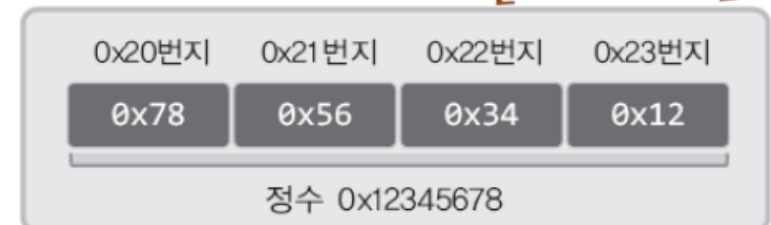
데이터 송수신 과정에서의 문제



빅 엔디안 모델



리틀 엔디안 모델



바이트 순서의 변환

바이트 변환 함수

```
unsigned short htons(unsigned short);  
unsigned short ntohs(unsigned short);  
unsigned long htonl(unsigned long);  
unsigned long ntohl(unsigned long);
```

- htons에서 h는 호스트(host) 바이트 순서를 의미
- htons에서 n은 네트워크(network) 바이트 순서를 의미
- htons에서 s는 자료형 short를 의미
- htonl에서 l은 자료형 long을 의미

이 기준을 적용하면 위 함수가 의미하는 바를 이해할 수 있다.

과제

바이트 변환의 예

```
~/socket_programming
1 #include <stdio.h>
2 #include <arpa/inet.h>
3
4 int main(int argc, char *argv[])
5 {
6     unsigned short host_port=0x1234;
7     unsigned short net_port;
8     unsigned long host_addr=0x12345678;
9     unsigned long net_addr;
10
11     net_port=htons(host_port);
12     net_addr=htonl(host_addr);
13
14     printf("Host ordered port: %#x \n", host_port);
15     printf("Network ordered port: %#x \n", net_port);
16     printf("Host ordered address: %#lx \n", host_addr);
17     printf("Network ordered address: %#lx \n", net_addr);
18     return 0;
19 }
```

문자열 정보를 네트워크 바이트 순서의 정수로 변환

```
#include <arpa/inet.h>
```

```
in_addr_t inet_addr(const char * string);
```

➔ 성공 시 빅 엔디안으로 변환된 32비트 정수 값, 실패 시 INADDR_NONE 반환

“211.214.107.99” 와 같이 점이찍힌 10진수로 표현된 문자열을 전달하면, 해당 문자열 정보를 참조해서 IP주소정보를 32비트 정수형으로 반환!

```
#include <stdio.h>
#include <arpa/inet.h>

int main(int argc, char *argv[])
{
    char *addr1="127.212.124.78";
    char *addr2="127.212.124.256";

    unsigned long conv_addr=inet_addr(addr1);
    if(conv_addr==INADDR_NONE)
        printf("Error occured! \n");
    else
        printf("Network ordered integer addr: %#1x \n", conv_addr);

    conv_addr=inet_addr(addr2);
    if(conv_addr==INADDR_NONE)
        printf("Error occured \n");
    else
        printf("Network ordered integer addr: %#1x \n\n", conv_addr);
    return 0;
}
```


inet_aton

```
#include <arpa/inet.h>
```

```
int inet_aton(const char * string, struct in_addr * addr);
```

→ 성공 시 1(true), 실패 시 0(false) 반환

- string 변환할 IP주소 정보를 담고 있는 문자열의 주소 값 전달.
- addr 변환된 정보를 저장할 in_addr 구조체 변수의 주소 값 전달.

기능상으로 inet_addr 함수와 동일하다. 다만 in_addr형 구조체 변수에 변환의 결과가 저장된다는 점에서 차이를 보인다.

```
#include <stdio.h>
#include <stdlib.h>
#include <arpa/inet.h>
void error_handling(char *message);

int main(int argc, char *argv[])
{
    char *addr="127.232.124.79";
    struct sockaddr_in addr_inet;

    if(!inet_aton(addr, &addr_inet.sin_addr))
        error_handling("Conversion error");
    else
        printf("Network ordered integer addr: %#x \n", addr_inet.sin_addr.s_addr);
    return 0;
}

void error_handling(char *message)
{
    fputs(message, stderr);
    fputc('\n', stderr);
    exit(1);
}
```

inet_ntoa

```
#include <arpa/inet.h>
```

```
char * inet_ntoa(struct in_addr adr);
```

➔ 성공 시 변환된 문자열의 주소 값, 실패 시 -1 반환

inet_aton 함수의 반대기능 제공! 네트워크 바이트 순서로 정렬된 정수형 IP주소정보를 우리가 눈으로 쉽게 인식할 수 있는 문자열의 형태로 변환.

```
~/socket_programming
1 #include <stdio.h>
2 #include <string.h>
3 #include <arpa/inet.h>
4
5 int main(int argc, char *argv[])
6 {
7     struct sockaddr_in addr1, addr2;
8     char *str_ptr;
9     char str_arr[20];
10
11     addr1.sin_addr.s_addr=htonl(0x1020304);
12     addr2.sin_addr.s_addr=htonl(0x1010101);
13
14     str_ptr=inet_ntoa(addr1.sin_addr);
15     strcpy(str_arr, str_ptr);
16     printf("Dotted-Decimal notation1: %s \n", str_ptr);
17
18     inet_ntoa(addr2.sin_addr);
19     printf("Dotted-Decimal notation2: %s \n", str_ptr);
20     printf("Dotted-Decimal notation3: %s \n", str_arr);
21     return 0;
22 }
```

1,1

Top

6주차 수업이 끝났습니다

고생하셨습니다.

