



EDDI

Electronic Design
Development Institute

에디로봇아카데미

임베디드 마스터 Lv1 과정

제 4기

2023. 01. 24

진동민

학습목표 & 15회차 날짜

학습목표

- 도커를 설치할 수 있다

수업 날짜

2023-01-14 (토) 오후 6시~9시

목차

- 1) Docker 설치
- 2) 도커란 무엇인가
- 3) 여담
- 4) 수업내용 사진

Docker 설치

수업에서 AVR 개발을 위해서 Docker 환경을 설정해야한다.

Docker를 설치하기 위해서 노션에서 다음 페이지로 이동한다.

페이지 위치: 엔지니어링 위키 / study materials / Lv1 필수 SW 설정

여기서 'AVR 개발을 위해 설정해야 하는 Docker 설정 진행'을 클릭한다.



Lv1 필수 SW 설정

수업 진행을 위한 최소한의 필수 SW 준비하기

AVR 개발을 위해 설정해야 하는 Docker 설정 진행

링크: <https://cafe.naver.com/eddicorp/1069>

Docker 설치

강사님이 도커를 설치하는 방법을 컬럼으로 잘 적어놓으셨다. 1번부터 시작해서 꼭 따라가면 된다.

링크썸 컬럼 >

[링크썸 컬럼] Docker 기반 Code Blocks 구동법



링크썸 카페메니저 1:1 채팅

2022.11.03. 03:11 조회 38

댓글 0 URL 복사

안녕하세요.

에디로봇아카데미의 링크썸입니다.

이 작업은 Lv1 수업을 진행하는 분들에게 필요할 수도 있는 절차입니다.

원래는 수업 듣는 분중에 맥북 사용자분이 있어서 도입하게 되었습니다만

근래 리눅스 버전이 올라가면서 Code Blocks가 최신 버전에서 동작하지 않는 이슈 또한 발견되었던 것으로 기억합니다.

이 문제 때문에 당시에 Docker에 이미지를 구성하여 문제를 해결하는 방법에 대한 준비를 했던 기억이 있습니다.

또 어떤 노트북은 우분투 버전이 하위 버전이나 상위 버전만 깔리는 경우들이 존재합니다.

버전 문제로 인한 이슈를 회피하고자 도커를 사용하도록 합니다.

문제가 없다면 별도로 Docker 환경을 구성할 필요는 없습니다.

1. 먼저 Docker를 설치할 준비가 되었는지 확인하기 위해 KVM이라는 하이퍼바이저 존재 여부를 파악합니다.

```
(base) oem@ThinkPad:~/proj/eddi/academy/SRS-Radar$ lsmod | grep kvm
kvm_intel          282624  0
kvm                663552  1 kvm_intel
(base) oem@ThinkPad:~/proj/eddi/academy/SRS-Radar$
```

Docker 설치

1. KVM 하이퍼바이저 존재 여부 확인

- lsmod (list module): 현재 OS에 로드된 디바이스 드라이버 리스트를 보여준다

```
(base) try@try-desktop:~$ lsmod | grep kvm
kvm_amd          155648  0
kvm              1011712  1 kvm_amd
ccp              102400   1 kvm_amd
(base) try@try-desktop:~$
```

KVM

- Hypervisor 라고 부르며,
OS 가상화를 지원한다

2. 모든 것을 파일로 관리하는 유닉스의 핵심 철학에 맞게, KVM 드라이버 또한 파일로 맵핑되었는지 확인

```
(base) try@try-desktop:~$ ls -al /dev/kvm
crw-rw----+ 1 root kvm 10, 232 1월 15 00:21 /dev/kvm
(base) try@try-desktop:~$
```

3. 사용자를 kvm 그룹에 추가하여 하이퍼바이저가 제어하는 장치들에 접근할 수 있는 권한을 만들어줌

```
(base) try@try-desktop:~$ sudo usermod -aG kvm $USER
[sudo] try 암호:
(base) try@try-desktop:~$
```

Docker 설치

4. 잘 적용되었는지 확인

```
(base) try@try-desktop:~$ id
uid=1000(try) gid=1000(try) 그룹들=1000(try),4(adm),24(cdrom),27(sudo),30(dip),4
6(plugdev),120(lpadmin),132(lxd),133(sambashare)
(base) try@try-desktop:~$ su - $USER
암호:
(base) try@try-desktop:~$ id
uid=1000(try) gid=1000(try) 그룹들=1000(try),4(adm),24(cdrom),27(sudo),30(dip),4
6(plugdev),108(kvm),120(lpadmin),132(lxd),133(sambashare)
(base) try@try-desktop:~$
```

5. 이후 아래 명령을 입력하여 실행

- `sudo apt install gnome-terminal`

6. 기존의 Legacy 삭제 (만약 Docker를 설치된적 없다면 패키지를 찾을 수 없거나 지울 수 없다는 창이 나올 것이다)

- `sudo apt remove docker-desktop`
- `rm -r $HOME/.docker/desktop`
- `sudo rm /usr/local/bin/com.docker.cli`
- `sudo apt purge docker-desktop`

Docker 설치

7. 다운 받을 패키지 정보를 갱신

- `sudo apt-get update`

8. 도커 설치에 필요한 필수 패키지들을 설치

- `sudo apt-get install ca-certificates curl gnupg lsb-release`

9. 도커 다운로드를 위한 GPG 키를 설정

- `sudo mkdir -p /etc/apt/keyrings`
- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg`

10. 다운 받을 패키지 정보(도커 관련)들과 관련한 설정을 진행

- `echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >
/dev/null`

11. 다시 패키지 저장소를 갱신

- `sudo apt-get update`

Docker 설치

12. 만약 **GPG 에러가 발생한다면** 아래 명령어를 입력하여 실행 (노트북과 데스크탑 실습 환경에서는 오류가 발생하지 않았다)

- `sudo chmod a+r /etc/apt/keyrings/docker.gpg`
- `sudo apt-get update`

13. 도커 엔진을 설치

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

14. 동작이 잘 되는지 확인하기 위한 기본 출력을 시도

- `sudo docker run hello-world`

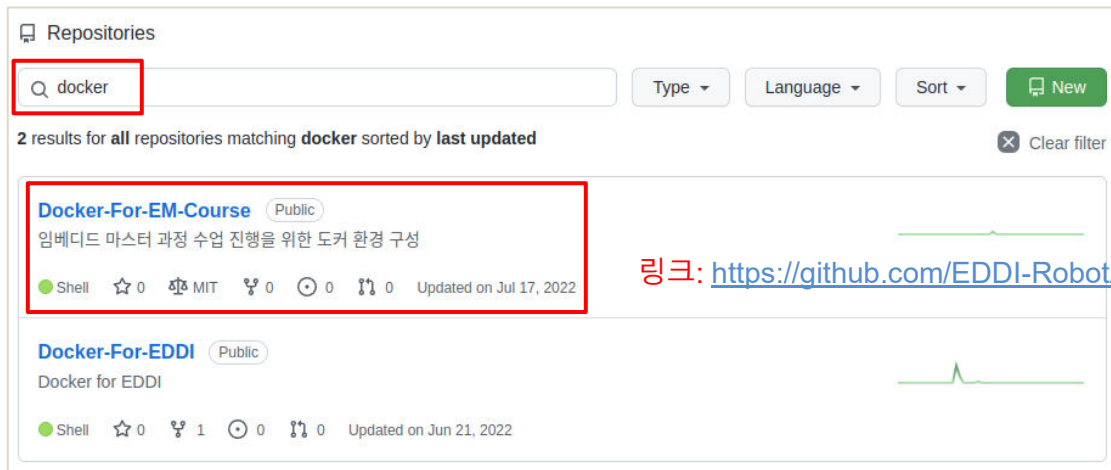
15. 추가로 도커 버전 확인

```
(base) try@try-desktop:~$ docker --version
Docker version 20.10.22, build 3a2c30b
(base) try@try-desktop:~$
```

Docker 설치

16. 다음으로 Code::Blocks를 구동하기 위한 도커파일을 생성

- 학원 GitHub 저장소에서 docker를 검색
- Docker-For-EM-Course 저장소를 찾아서 그 저장소를 git clone



The screenshot shows the GitHub search interface. The search bar contains the text 'docker'. Below the search bar, it says '2 results for all repositories matching docker sorted by last updated'. Two repositories are listed:

- Docker-For-EM-Course** (Public)
임베디드 마스터 과정 수업 진행을 위한 도커 환경 구성
● Shell ☆ 0 📄 MIT 🚀 0 🕒 0 🔗 0 Updated on Jul 17, 2022
- Docker-For-EDDI** (Public)
Docker for EDDI
● Shell ☆ 0 🚀 1 🕒 0 🔗 0 Updated on Jun 21, 2022

링크: <https://github.com/EDDI-RobotAcademy/Docker-For-EM-Course>

Docker 설치

17. emlv1 디렉토리에 있는 docker-lv1-image-install.sh 파일을 실행

- 만약 바로 아래와 같이 '지금부터 docker를 등록합니다!'만 뜨면 재부팅 후 다시 실행하도록 한다.

```
(base) try@try-desktop:~/Desktop/real_last/Docker-For-EM-Course/emlv1$ ./docker-  
lv1-image-install.sh  
지금부터 docker를 등록합니다!  
[sudo] try 암호:  
(base) try@try-desktop:~/Desktop/real_last/Docker-For-EM-Course/emlv1$ ./docker-  
lv1-image-install.sh  
지금부터 docker를 등록합니다!  
(base) try@try-desktop:~/Desktop/real_last/Docker-For-EM-Course/emlv1$ █
```

- 다음은 재부팅하고 다시 실행한 결과이다. 이와 같이 뜨면 ok이다.

```
(base) try@try-desktop:~/Desktop/real_last/Docker-For-EM-Course/emlv1$ ./docker-  
lv1-image-install.sh  
docker가 이미 등록되어 있습니다!  
도커 이미지 설치!  
20.04: Pulling from library/ubuntu  
846c0b181fff: Pull complete  
Digest: sha256:0e0402cd13f68137edb0266e1d2c682f217814420f2d43d300ed8f65479b14fb  
Status: Downloaded newer image for ubuntu:20.04  
docker.io/library/ubuntu:20.04  
[sudo] try 암호:  
access control disabled, clients can connect from any host  
root@try-desktop:~/workspace# █
```

18. workspace 디렉토리에 있는 inner-init.sh 파일 실행

- 만약 파일 이름을 일부 작성한 상태에서, tab키를 눌러도 자동완성이 되지 않는다면 실행권한이 없다는 것입니다

```
root@try-desktop:~/workspace# ls
inner-init.sh
root@try-desktop:~/workspace# ./i
```

- chmod를 이용하여 실행권한을 주고 실행한다.
 - chmod => change mode (권한 설정)
 - x = eXecute
 - (참고) inner-init.sh 파일을 들여다보면 apt-get 명령어를 사용하여 패키지를 설치합니다.

```
root@try-desktop:~/workspace# chmod +x inner-init.sh
root@try-desktop:~/workspace# ./inner-init.sh
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [
979 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2423
kB]
0% [2 InRelease 162 kB/265 kB 61%] [4 Packages 4133 B/2423 kB 0%]
```

- 18.
- 패키지 설치 중 지역, 시간대와 키보드 설정이 필요한데, 이는 GitHub의 README.md 파일에서 다음을 참고하면 된다.

라이브러리 설치시 (시간 및 키보드 설정)

시간 설정은 한국 시간, 키보드 설정은 미국입니다.

6, 69, 31, 1 순서로 넣으면 됩니다.

혹시 모르니 확인 하시길 바랍니다.

```
Please select the geographic area in which you live. Subsequent configuration
questions will narrow this down by presenting a list of cities, representing
the time zones in which they are located.
```

- | | | |
|-----------------|-------------------|------------------------|
| 1. Africa | 6. Asia | 11. System V timezones |
| 2. America | 7. Atlantic Ocean | 12. US |
| 3. Antarctica | 8. Europe | 13. None of the above |
| 4. Australia | 9. Indian Ocean | |
| 5. Arctic Ocean | 10. Pacific Ocean | |

```
Geographic area: 
```

Docker 설치

18.

- 다음과 같이 뜨면 ok이다.

```
Setting up avrdude (6.3-20171130+svn1429-2) ...  
Setting up libx32ubsan1 (10.3.0-1ubuntu1~20.04) ...  
Setting up lib32ubsan1 (10.3.0-1ubuntu1~20.04) ...  
Setting up lib32gcc-9-dev (9.4.0-1ubuntu1~20.04.1) ...  
Setting up libx32gcc-9-dev (9.4.0-1ubuntu1~20.04.1) ...  
Setting up gcc-9-multilib (9.4.0-1ubuntu1~20.04.1) ...  
Setting up gcc-multilib (4:9.3.0-1ubuntu2) ...  
Processing triggers for man-db (2.9.1-1) ...  
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...  
root@try-desktop:~/workspace#
```

19. (도커 가상환경 터미널이 아닌) 새 터미널을 열고 다음 명령어를 입력하여 구동 중인 컨테이너를 감지한다.

- docker ps -a

```
(base) try@try-desktop:~/Desktop$ docker ps -a  
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES  
ddf365d4df7f   ubuntu:20.04   "/bin/bash"             31 minutes ago Up 31 minutes                test-ubuntu  
da90104071e7   hello-world    "/hello"                 9 hours ago   Exited (0) 9 hours ago                nifty_solomon  
(base) try@try-desktop:~/Desktop$
```

Docker 설치

20. 앞에서 나온 결과를 참고하여 현재 상태를 저장합니다.

- `docker commit ddf365d4df7f ubuntu:20.04`
- 지금 보고 있는 실습에서는 `docker commit ddf365d4df7f ubuntu:20.04` 을 입력하였습니다.

```
(base) try@try-desktop:~/Desktop$ docker commit ddf365d4df7f ubuntu:20.04  
sha256:03f8e00a38acca326ca52c891bd3129ddc158a2d4fcee9df73e1242faf5a21d0  
(base) try@try-desktop:~/Desktop$
```

21. (도커 가상환경 터미널로 돌아와서) 다음 명령어를 실행합니다. (도커를 재실행하여 커밋이 되었는지 확인)

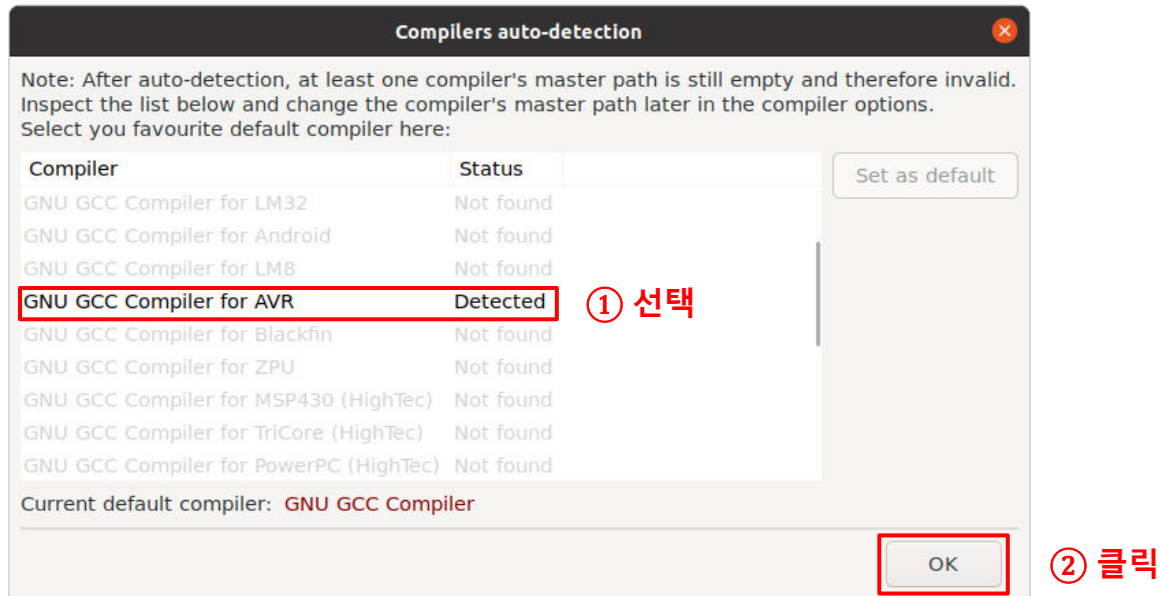
- `exit`
- `./docker-lv1-image-install.sh`
- `codeblocks`

```
root@try-desktop:~/workspace# exit  
exit  
(base) try@try-desktop:~/Desktop/real_last/Docker-For-EM-Course/emlv1$ ./docker-lv1-image-install.sh  
docker가 이미 등록되어 있습니다!  
이미 설치되어 있습니다!  
ubuntu 20.04 컨테이너를 내립니다.  
ddf365d4df7f  
[sudo] try 암호:  
access control disabled, clients can connect from any host  
root@try-desktop:~/workspace# codeblocks
```


Docker 설치

22. Code::Blocks 구동 확인

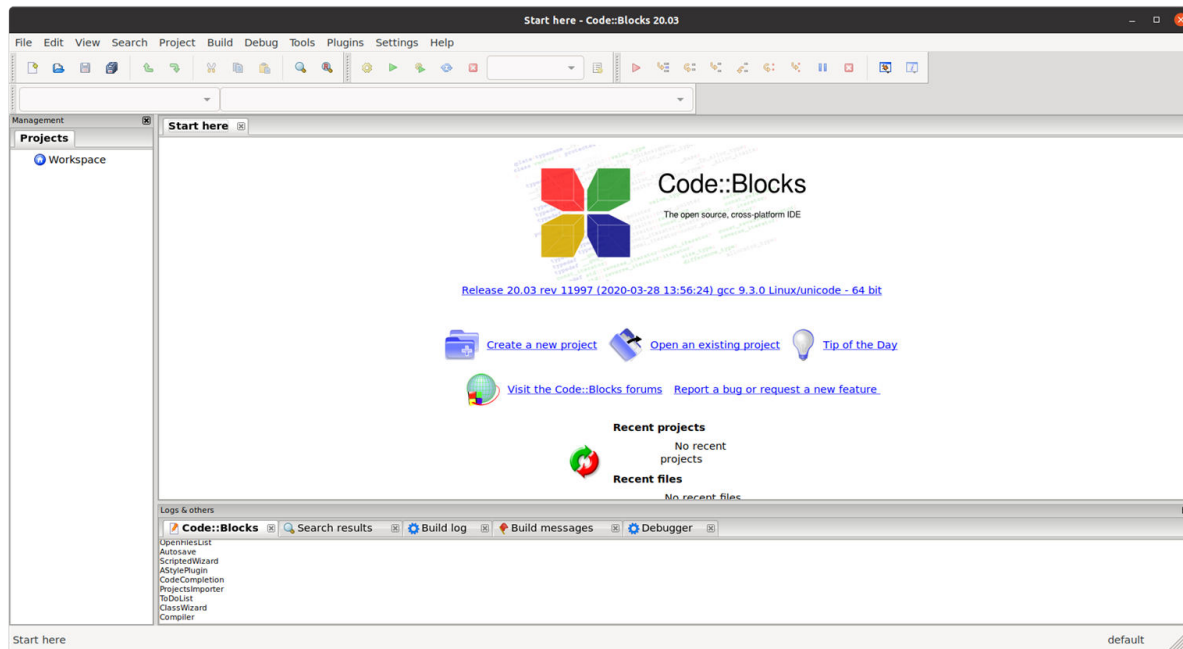
- 강사님이 Set as default는 누르지 말라고 하셨다



Docker 설치

마지막.

- 잘 구동되는 것을 확인할 수 있다. 끝!



도커란 무엇인가

DevOps와 비슷한 것이라고 한다.

데이터 센터를 구축한다고 생각한다면 구축에 필요한 초기 투자 비용이 엄청 많이 들어간다.
이는 예전에 리츠 투자를 공부해봤어서 어느 정도 알고있는 내용이다.
또, 서버를 구축하는 과정에서 엄청난 노가다가 필요하다.

사람이면 실수를 하게 되는데...

이를 자동화할 수 있는 스마트한 방법이 없을까를 고민하다가 생각한 것이 클라우드!

사람이 직접하는게 아니라 자동화하자!

마치 앞에서 Docker 설치에서 강사님이 만들어둔 쉘 스크립트처럼 명령어를 한 번만 치며 전부 세팅될 수 있도록 말이다.

그래서 서버를 구성하는 하드웨어가 달라서 다른 환경에 있어도 Docker를 사용하면 같은 환경을 공유할 수 있어
자유도도 높이면서 개발환경을 서포트해줄 수 있다.

도커를 사용하여 환경을 통일시켰다고 보면된다.

결론적으로 도커는 인프라 구축 툴이다.

VMWare나 VirtualBox라고 생각하면 된다.

Docker 설치 실습의 20번과 관련하여...

- 학원 카페의 게시글에는 commit이 아닌 pull로 적혀있어서 계속 docker pull [containerID] 명령어를 실행해도 이미지가 저장되지 않았다.
- 결국은 이슈를 해결하기 위해 Docker Hub에 회원가입했는데 그래도 안되길래 GitHub README.md 파일에 적혀있는 commit 명령어가 아닌가 싶어서 강사님에게 물어봤는데 한 번 실행해보시니 ㅋㅋㅋㅋㅋㅋ 이 명령어가 맞는듯 싶었다.
- 근데 docker commit containerID를 실행하고 나서 도커를 꺾다가 다시 켜보니 저장이 되지 않은 상태여서 inner-init.sh 파일을 재실행해서 설치를 반복했다. (또 다른 이슈 발생...)
- 이번에는 명령어 맨 뒤에 이미지 이름을 적어야 했는데 적지 않아서 저장이 제대로 되지 않았던 것이다.

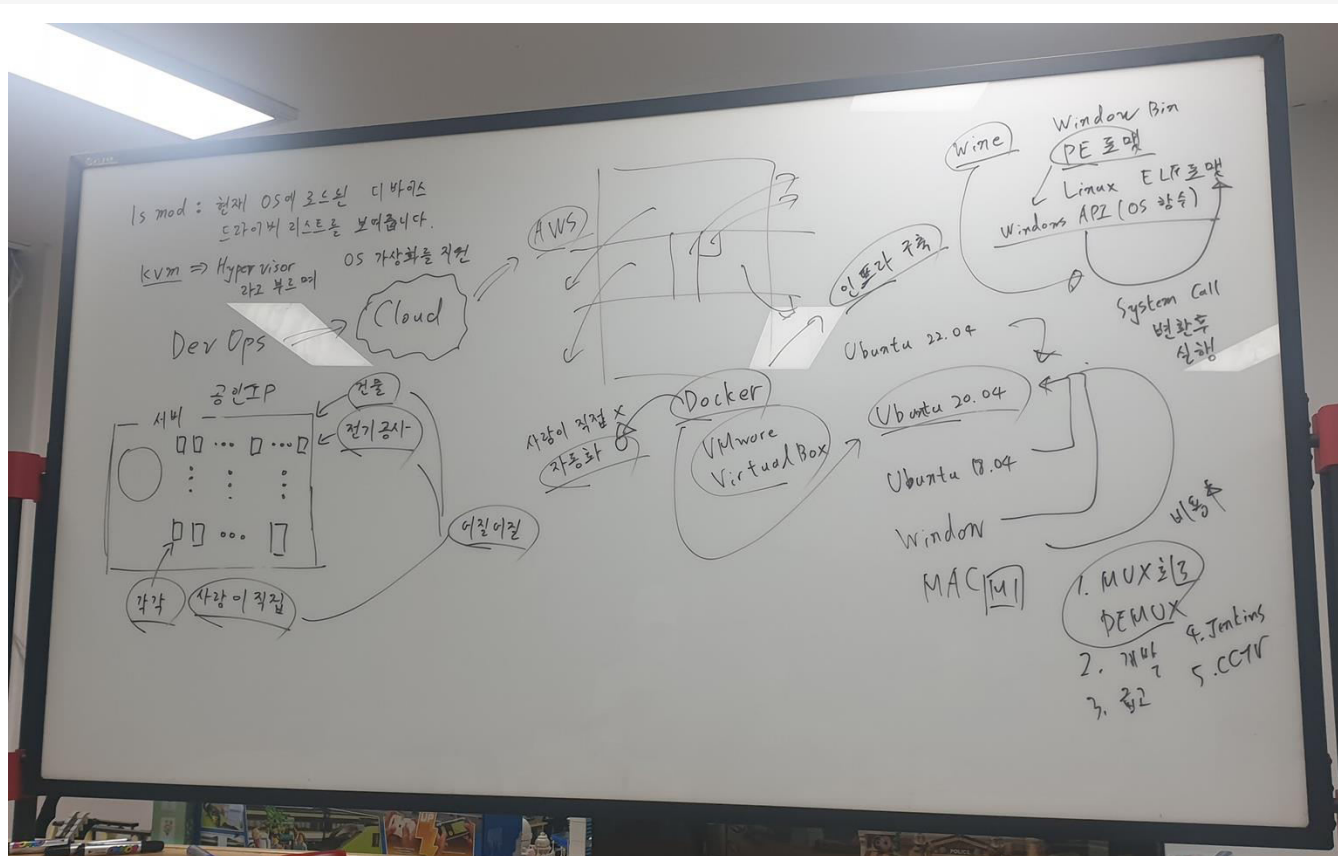
위의 경험으로 얻은 점은 일단 환경설정은 매번이 아닌 프로젝트 초기에 딱 한 번만 진행하기 때문에 그냥 시간 지나면 잊는다고 보면된다. 그렇기에 문서화가 필요한 이유다. 또한 명령어의 예시가 있으면 좋을 것 같다는 생각이 들었다.

작업 완료 이후 docker 업데이트

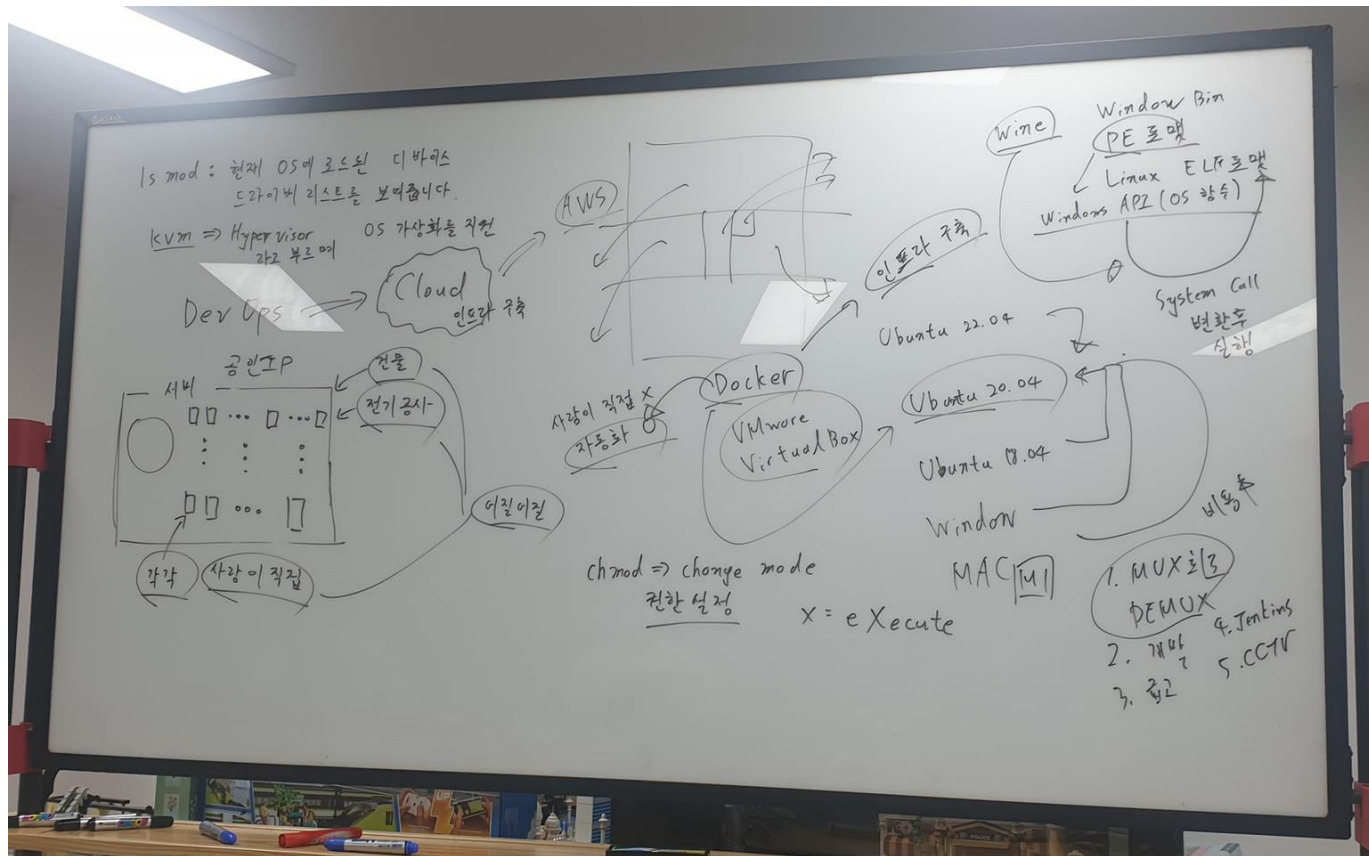
```
docker ps -a로 현재 컨테이너 id와 이미지 이름 파악  
docker commit [container id] [image]
```

- 수업 중간에 임베디드 개발자가 재택 근무를 하기 위한 방법으로 강사님이 구상한 것을 칠판에 적으면서 설명하셨는데, 하드웨어와 관련된 내용이 나와서 이해를 포기하였다.

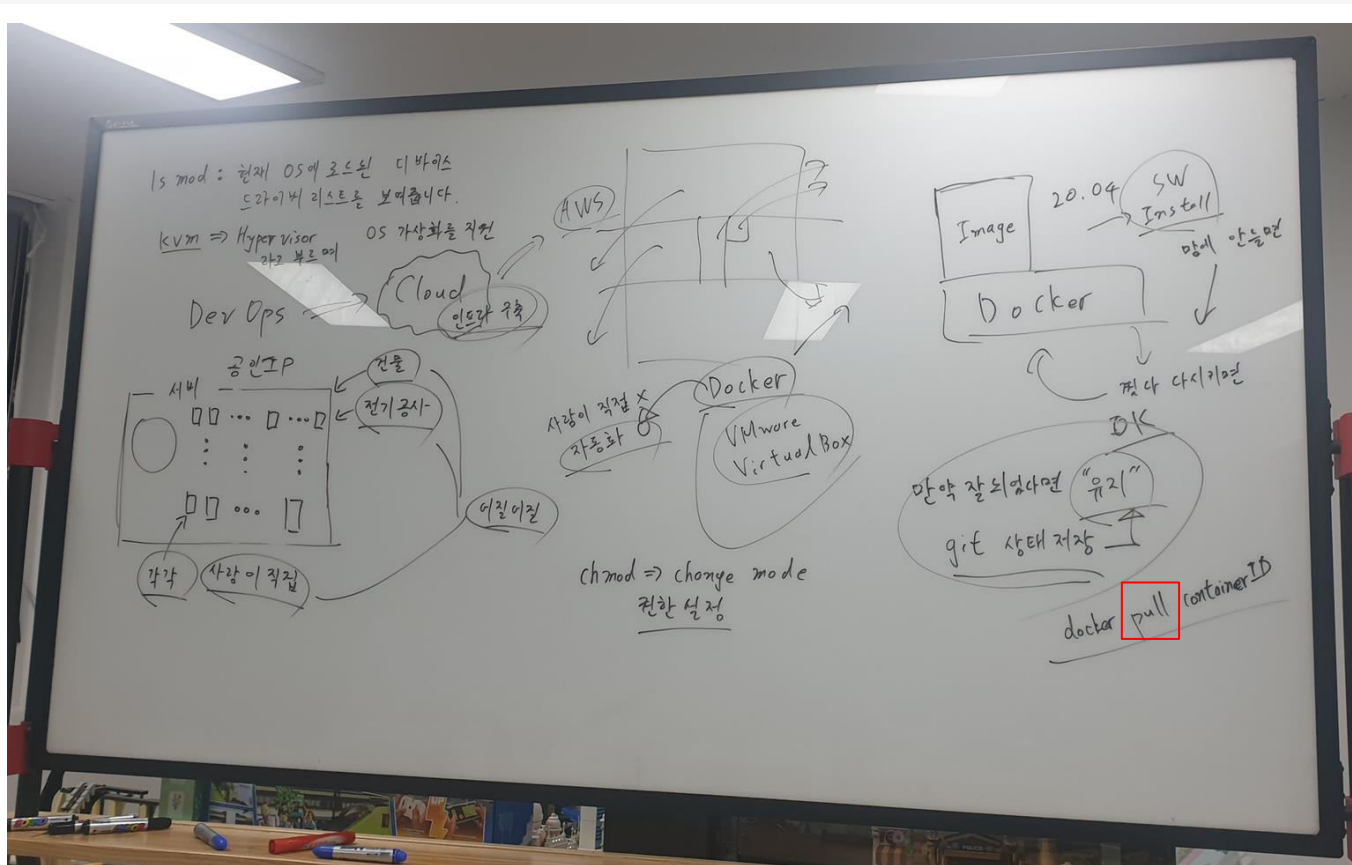
수업내용 사진



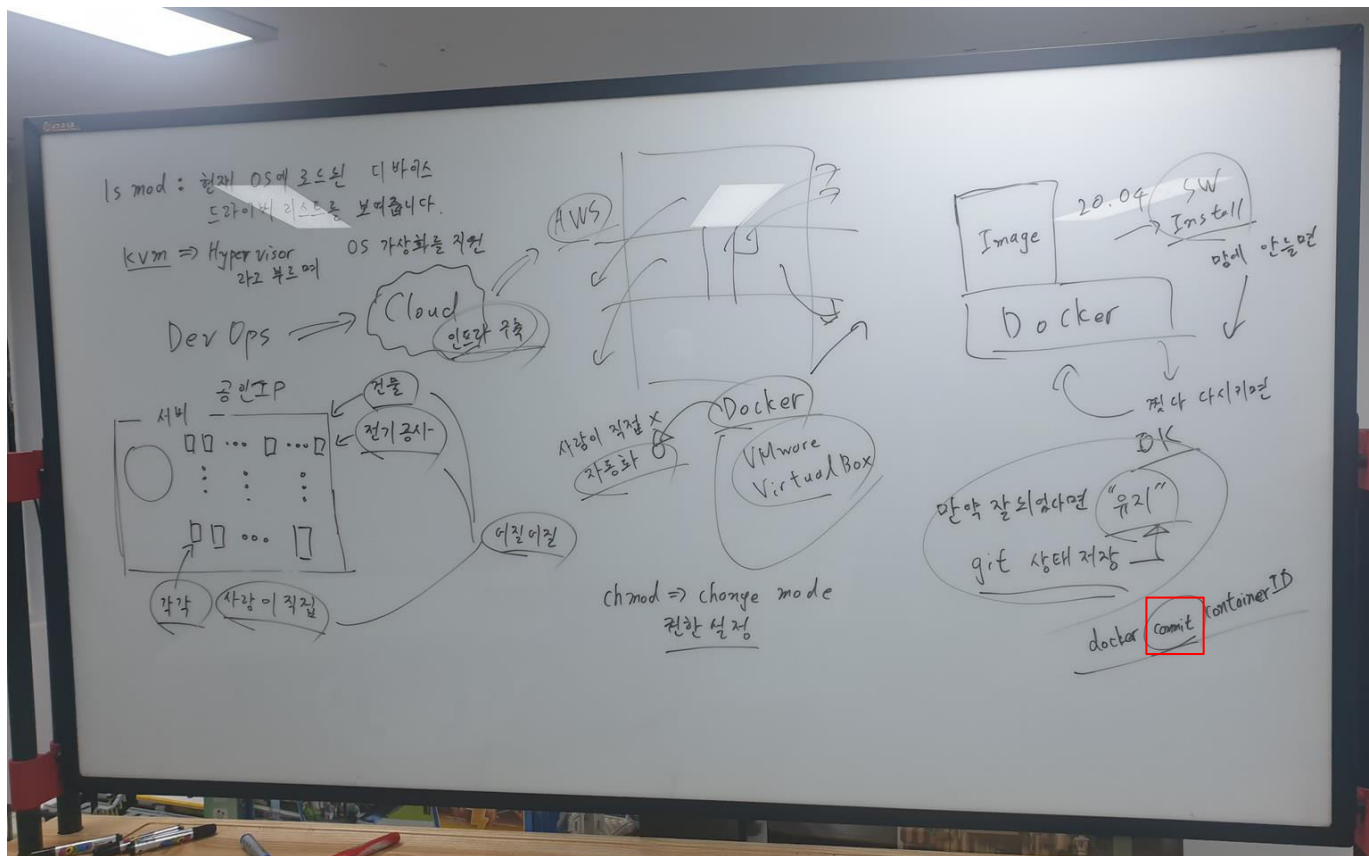
수업내용 사진



수업내용 사진



수업내용 사진



CONTENTS

1) 형식은 자유롭게~~~

<공부 내용을 적어주세요.>