

## 직접 도커 이미지를 만들어 보기

### 도커 이미지 생성하는 순서

현재까지는 도커 이미지를 항상 도커 허브에 이미 있던 것들만

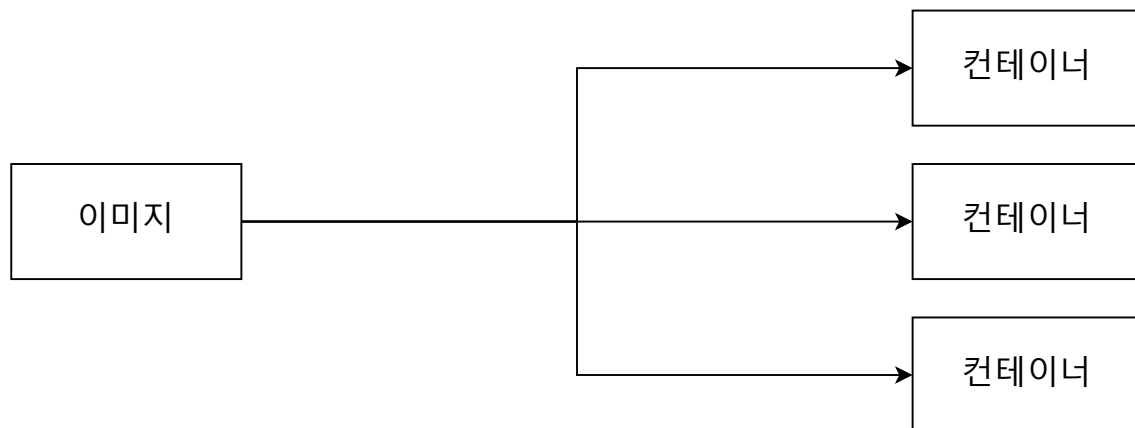
가져와서 사용했습니다. 하지만 저희가 직접 도커 이미지를 만들어서 사용할 수도 있고 직접 만든 도커 이미지를 도커 허브에 올려서 공유할 수도 있는데요.

### 간단하게 도커 이미지 다시 복습하기

1. 도커 이미지는 컨테이너를 만들기 위해 필요한 설정이나 종속성들을 갖고 있는 소프트웨어 패키지입니다.
2. 지금까지 해왔듯이 도커 이미지는 Dockerhub에 이미 다른 사람들이 만들어 놓은 것을 이용할 수도 있으며, 직접 도커 이미지를 만들어서 사용할 수도 있고 직접 만든 것을 Dockerhub에 업로드할 수도 있습니다.

도커 이미지를 이용해서 도커 컨테이너를 생성.

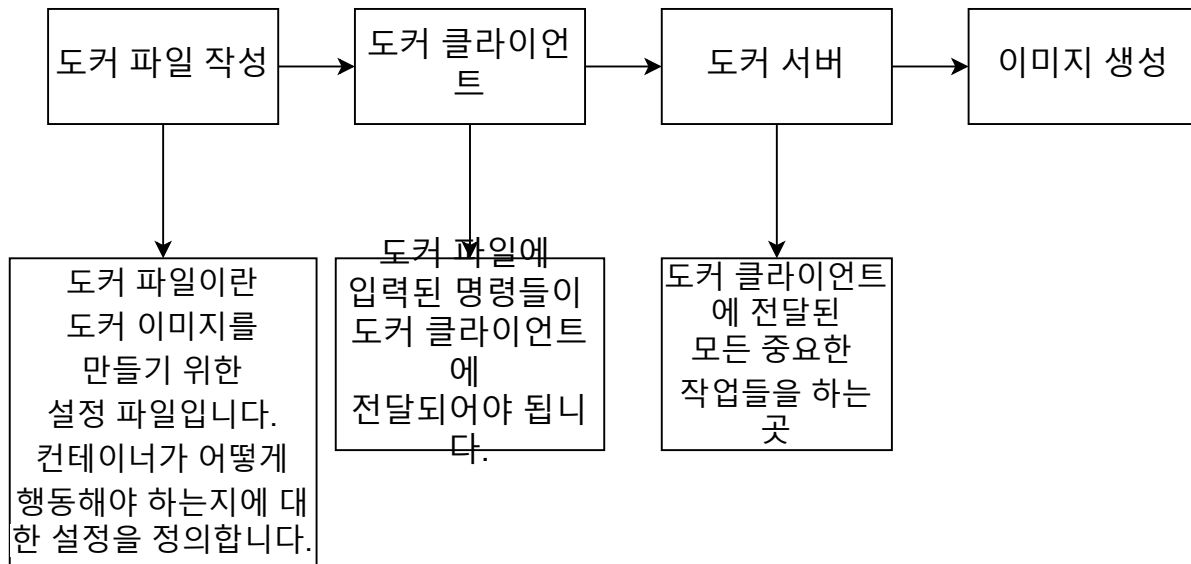
ex) `docker create <이미지 이름>`



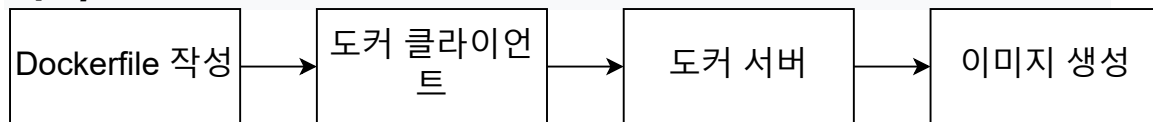
컨테이너는 도커 이미지로 생성...  
그럼 도커 이미지는 어떻게 생성할까요?



## 도커 이미지 생성하는



**그럼 이제 먼저 *Dockerfile*을 만드는 법을 알아보겠습니다**





# Dockerfile 만들기

## 도커 파일(Docker file)이란 ?

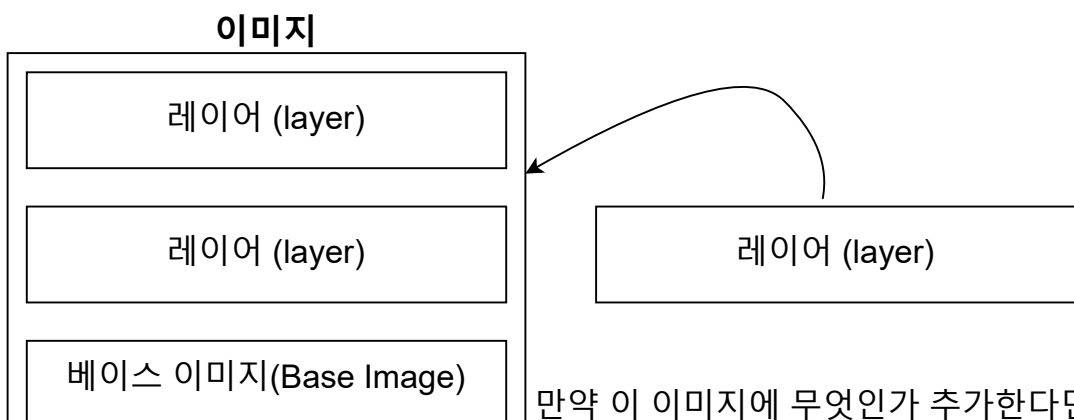
- 도커 이미지를 만들기 위한 설정 파일이며, 컨테이너가 어떻게 행동해야 하는지에 대한 설정들을 정의해 주는 곳입니다.

## 도커 파일 만드는 순서 (도커 이미지가 필요한 것이 무엇인지를 생각하기)

1. 베이스 이미지를 명시해준다. (파일 스냅샷에 해당)
2. 추가적으로 필요한 파일을 다운 받기 위한 몇 가지 명령어를 명시해준다. (파일 스냅샷에 해당)

## 베이스 이미지는 무엇인가?

- 도커 이미지는 여러 개의 레이어들로 되어 있습니다.  
그중에서 베이스 이미지는 이 이미지의 기반이 되는 부분입니다.



만약 이 이미지에 무엇인가 추가한다면?  
가다하게 OS라고 생각하며 되니 위에 보이는 레이어가 이미지에 추가됩니다.

시작  
어

파일  
코코O

코아톡 이미지

시 실행할 명령

...

스냅샷

아톡 파일







...다. 그리고 이것을 레이어 캐싱이라고 부릅니다.  
원도우, MacOS, 리눅스등

## 실제로 만들면서 배우기!

도커 이미지의 목표는 "hello 문구" 출력하기

### 순서

1. 도커 파일을 만들 폴더 하나 만들기 ex) dockerfile-folder
2. 방금 생성한 도커 파일 폴더를 에디터를 이용해서 실행 (Visual Studio Code 추천)
3. 파일 하나를 생성, 이름은 dockerfile
4. 그 안에 먼저 어떻게 진행해 나갈지 기본적인 토대를 명시.



**FROM RUN CMD** 등은 도커 서버에

```
# 베이스 이미지를 명시해준다.  
FROM baseImage  
  
# 추가적으로 필요한 파일들을 다운로드 받는다.  
RUN command  
  
# 컨테이너 시작시 실행 될 명령어를 명시해준다.  
CMD [ "executable" ]
```

#### FROM

이미지 생성 시 기반이 되는 이미지 리  
<이미지 이름>:<태그> 형식으로 작성  
태그를 안 붙이면 자동적으로 가장 최  
ex) ubuntu:14.04

#### RUN

도커 이미지가 생성되기 전에 수행할

#### CMD

컨테이너가 시작되었을 때  
실행할 실행 파일 또는 셸 스크립트입  
해당 명령어는 DockerFile 내 1회만 실행

5. 이제 베이스 이미지부터 실제 값으로 추가해주기.
6. 베이스 이미지는 ubuntu를 써도 되고 centos 등을 써도 되지만  
hello를 출력하는 기능은 굳이 사이즈가 큰 베이스 이미지를 쓸 필요가 없기  
에  
사이즈가 작은 alpine 베이스 이미지를 사용.
7. hello 문자를 출력해주기 위해 echo를 사용하여야 하는데  
이미 alpine 안에 echo를 사용하게 할 수 있는 파일이 있기에 RUN 부분은  
생략함.

```
# 베이스 이미지를 명시해준다.  
FROM alpine
```

에 무엇을 하라고 알려주는 것이다.

에 이어입니다.

신 것으로 다운 받음

셸 명령어

입니다.

할 수 있습니다

## FROM

기반이 되는 이미지 레이어입니다.

<이미지 이름>:<태그> 형식으로 작성

ex) ubuntu:14.04

## MAINTAINER

메인테이너 정보입니다.

## RUN

도커이미지가 생성되기 전에 수행할 셸 명령어

## VOLUME

VOLUME은 디렉터리의 내용을 컨테이너에 저장하지 않고 호스트에 기록 설정합니다.

데이터 볼륨을 호스트의 특정 디렉터리와 연결하려면 docker run 명령 옵션을 사용해야 합니다.

ex) -v /root/data:/data

저장하도

에서 -v

```
# 추가적으로 필요한 파일들을 다운로드 받는다.  
# RUN command  
  
# 컨테이너 시작시 실행 될 명령어를 명시해준다.  
CMD [ "echo", "hello" ]
```

이렇게 해서 도커 파일 만들기는 완료되었습니다.  
이제 이미지는 어떻게 만들어야 하는지 더욱 알아보겠습니다.



## **CMD**

컨테이너가 시작되었을 때 실행할 실행 파일 또는 셸 스크립트입니다.

해당 명령어는 DockerFile내 1회만 쓸 수 있습니다.

## **WORKDIR**

CMD에서 설정한 실행 파일이 실행될 디렉터리입니다.

## **EXPOSE**

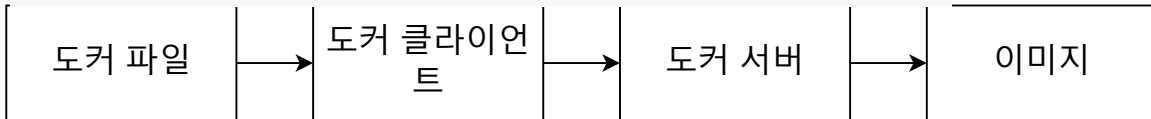
호스트와 연결할 포트 번호입니다.





## 도커 파일로 도커 이미지 만들기

완성된 도커 파일로 어떻게 이미지를 생성하나요?



도커 파일에 입력된 것들이 도커 클라이언트에 전달되어서 도커 서버가 인식하게 하여야 합니다. 그렇게 하기 위해서

**docker build ./ 또는 docker build .**

### Build 명령어는

- 해당 디렉토리 내에서 dockerfile이라는 파일을 찾아서 도커 클라이언트에 전달시켜준다.
- docker build 뒤에 ./ 와 . 는 둘 다 현재 디렉토리를 가리킨다.

그래서 **docker build .** 를 해보면

```
jaewon@Jaewonui-MacBookPro dockerfile-folder % docker build .
Sending build context to Docker daemon 2.048kB
Step 1/2 : FROM alpine
----> a24bb4013296
Step 2/2 : CMD [ "echo", "hello" ]
----> Running in fa6c364fdc39
Removing intermediate container fa6c364fdc39
----> aff4e9afc538
Successfully built aff4e9afc538
jaewon@Jaewonui-MacBookPro dockerfile-folder % docker run aff4e9afc538
hello
```

### build 과정 설명

Step 1/2

- alpine 이미지 가져오기 a24bb~ 는 alpine 이미지 아이디

Step 2/2

Step 2/2

- 임시 컨테이너 생성 후 그 컨테이너에 시작 시 사용할 명령어 포함시키기.
- 그런 후 방금 생성 한 임시 컨테이너를 지우고 새로운 이미지 만들기

## Step 2/2 자세히 보기

- Step 2에 보면 임시 컨테이너를 생성하고 무언가를 하고 다시 지우는 데 왜 그렇게 하는 걸까 ?

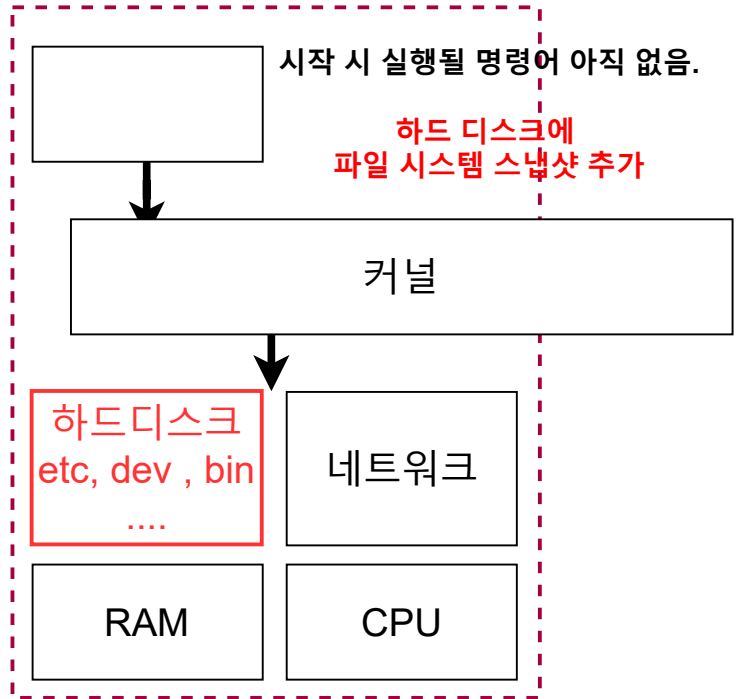
①

### Alpine 이미지 (a24bb4013296~)



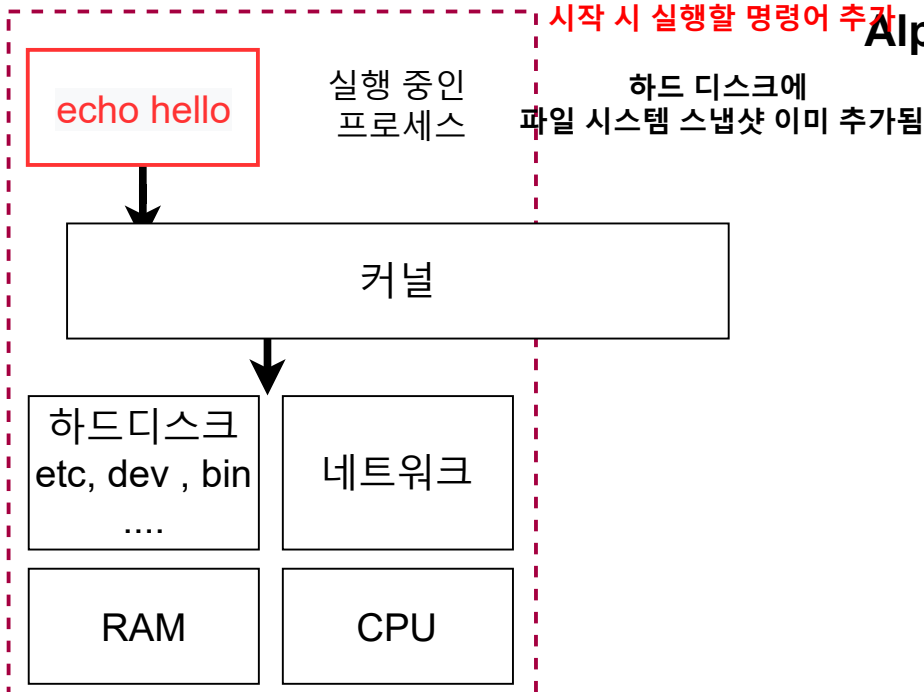
②

### 임시 컨테이너 생성(16c04fc73f76~)



③

### 임시 컨테이너



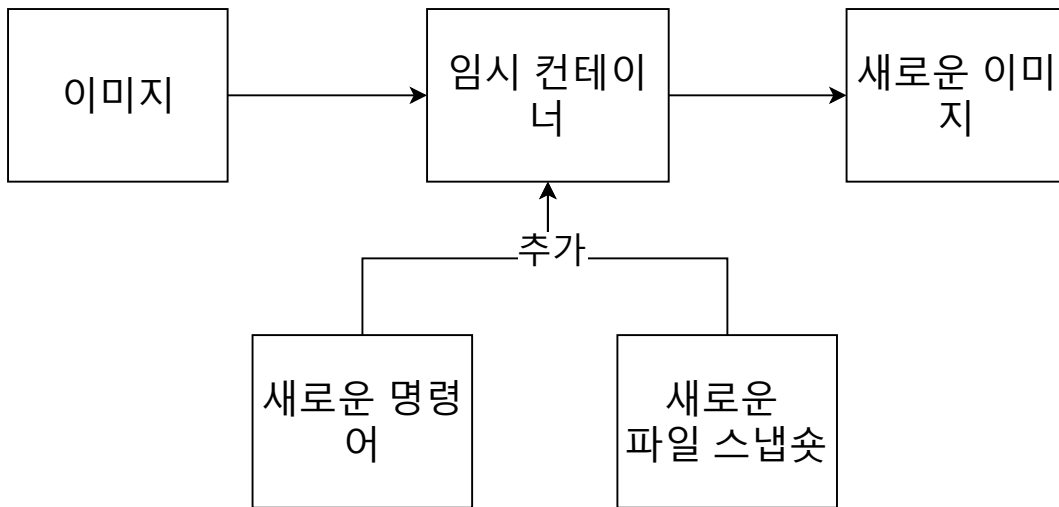
④

### Alpine 이미지 (872457464cf6 ~)



## 결론

- 베이스 이미지에서 다른 종속성이나 새로운 커맨드를 추가할 때는  
임시 컨테이너를 만든 후 그 컨테이너를 토대로 새로운 이미지를 만든다.  
그리고 그 임시 컨테이너는 지워준다.



## 내가 만든 이미지 기억하기 쉬운 이름 주기

### 도커 파일로 만든 이미지를 컨테이너에서 실행하려면 ?

```
jaewon@Jaewonui-MacBookPro dockerfile-folder % docker build .  
Sending build context to Docker daemon 2.048kB  
Step 1/2 : FROM alpine  
----> a24bb4013296  
Step 2/2 : CMD [ "echo", "hello" ]  
----> Running in abe830c610db  
Removing intermediate container abe830c610db  
----> 26254532e71a  
Successfully built 26254532e71a
```

도커 파일로 만든 이미지 아이디가 26254532...  
이런 식이다....그래서 이 아이디로 컨테이너를 실행하려면

`docker run 26254532 ....`

이런 식으로 너무 길고 복잡한 이미지 아이디를 써줘야 한다...

IP Address가 기억하기 힘들기에  
Domain 이름을 새로 만들어서 이용하는 것과 비슷하게  
우리가 만드는 도커 이미지에 이름을 줄 수 있다.

### 도커 이미지에 이름 주는 방법

