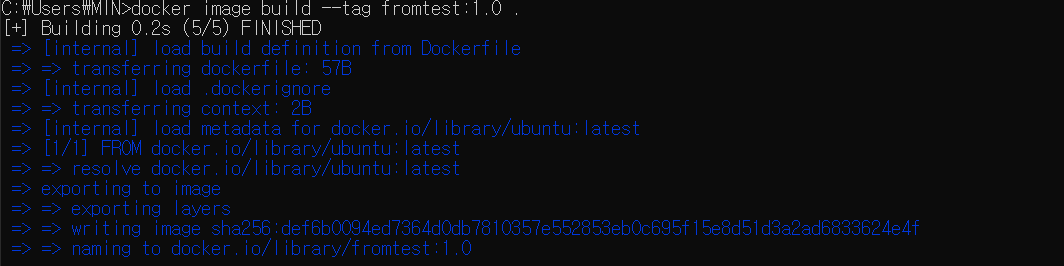
docker container run --volume C:\Users\MIN\docker\_volume:/var/lib/mysql run -d -p 23306:3306 -e MYSQL\_ALLOW\_EMPTY\_PASSWORD=true --name mysql\_db mysql:5.7

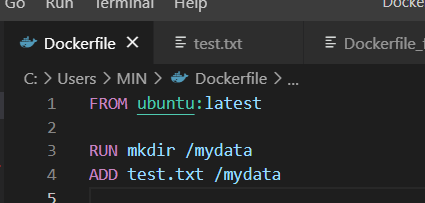
볼륨마운트



이미지 빌드(프롬테스트/버전 1.0 의 이미지 빌드)

run하고 나면실행 중인 목록에 없다…

이유는 계속 실행하게 만들만한 명령이 없어서 그런것임



도커파일을 만들고

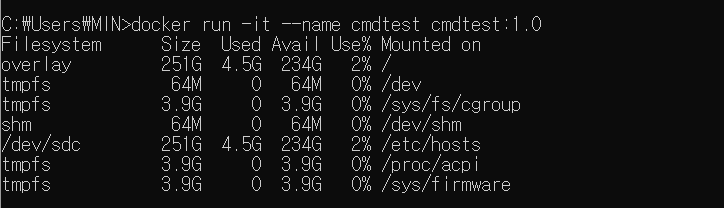
이미지 빌드한다음

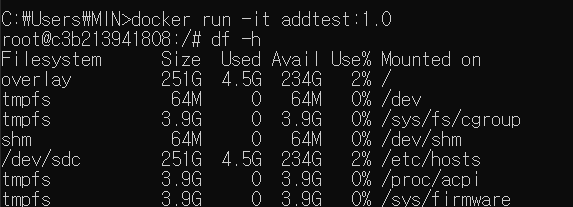


들어가보면 저장한 내용이 나온다



업데이트!

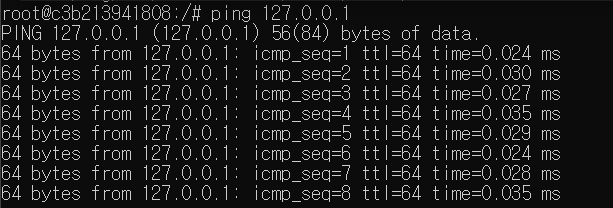


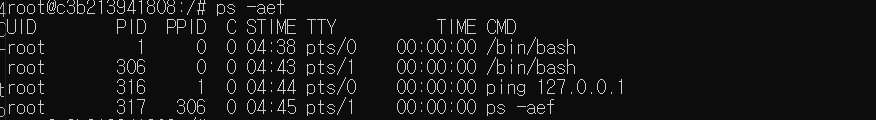
이렇게 하는거랑 내용이 같다!

Ping은 하나의 커맨드에서 다른 곳으로 연결이 되는지 확인해주는 커맨드핑 받을 수 있다

Exec는 추가적인 작업하는 것

Run은 실행만 하는 것

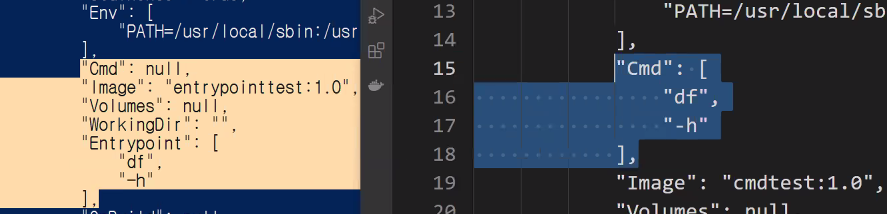


핑 실행한 채로 

확인해보면 핑 뜨는게 보인다.

Entrypoint 랑 CMD랑 결과 내용은 같게 나온다

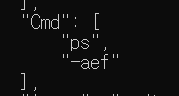


왼쪽인 엔트리포인트

엔트리포인트에는 CMD가 null로 나온다. cmd에는 동적인 파라미터로 만들 수 있지만 엔트리포인트는 고정적인 값이 된다. 변경될 가능성이 있으면 cmd커맨드로 진행하고 가변가능성 없으면 entrypoint 사용할것

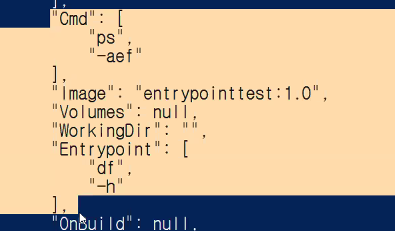
**docker run -it cmdtest:1.0 ps -aef**

실행하면서 부가적인 커맨드 넣기

그러고 나서 인스펙트 하면 

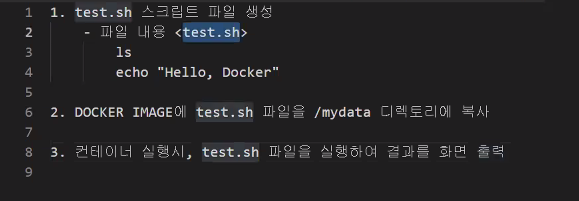
이렇게 된다

Entrypoint:1.0 ps -aef하면 오류가 뜬다



다 CMD에 들어갔다!

새롭게 입력한 데이터는 가변 파라미터식으로 들어갈 수 있다



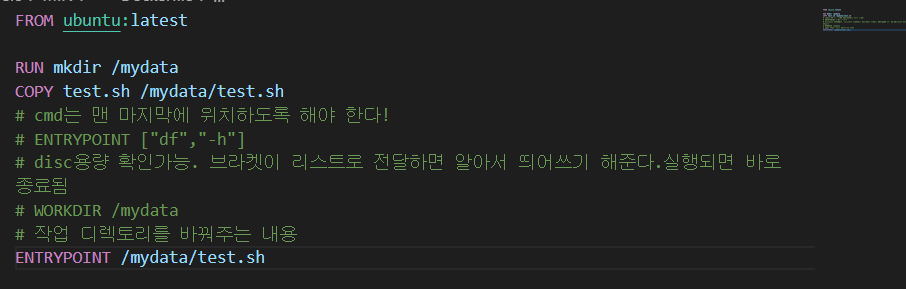
먼저 저렇게 내용 바꿔준다음에



빌드한다



Run 하고



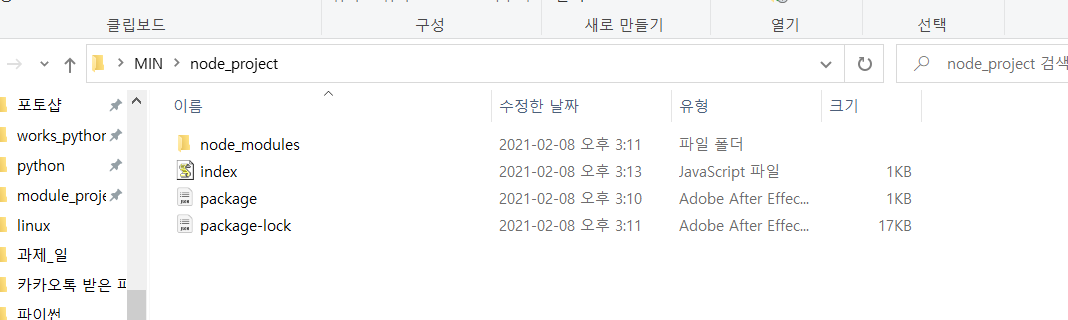
이렇게 넣어준다음에 빌드하고 실행하묘ㅕㄴ

리스트 뽑은다음

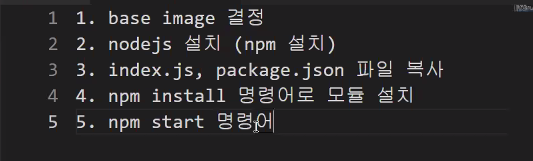


이렇게 하단에 글자가 출력된다





노드 프로젝트 폴더 만들어서 하단에 파일들 작성해서 넣어준다.



이런 식으로 실행하면 될 것

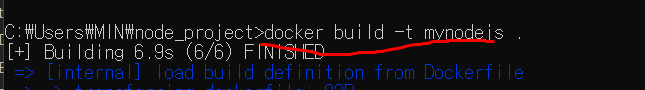
알파인 리눅스 다운받기

알파인 리눅스는 가볍고 간단한, 보안성을 목적으로 개발한 리눅스 배포판입니다.

용량을 줄이기 위해 시스템의 기본 C runtime을 [glibc](https://ko.wikipedia.org/wiki/GNU_C_%EB%9D%BC%EC%9D%B4%EB%B8%8C%EB%9F%AC%EB%A6%AC) 대신 [musl libc](https://en.wikipedia.org/wiki/Musl) 를 사용하며 다양한 쉘 명령어는 GNU util 대신 [busybox](https://en.wikipedia.org/wiki/BusyBox) 를 탑재하였습니다.

용량이 80M인 초경량화된 배포판이므로 Embbeded 나 네트웍 서버등 특정 용도에 적합하며 특히 도커(docker)에 채택되어 5M 크기의 리눅스 이미지로 유명합니다.

베이스 이미지 먼저 다운받고(리눅스 먼저)

이미 운영체제 위에 우리가 설치해 둔 것을 가지고 베이스이미지로 활용하면 훨씬 사용하기 편하다. 이미지 생성한다



실행하면 이렇게 뜬다

노드js의 프롬프트 창으로 이동된것임

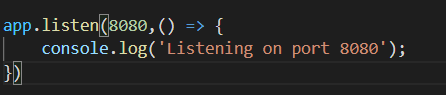
알파인 버전에는 bin/bash가 없다

하지만 모든 리눅스에는 bin/sh는 있기 떄문에 이건 된다!

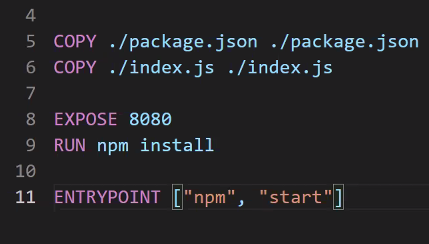
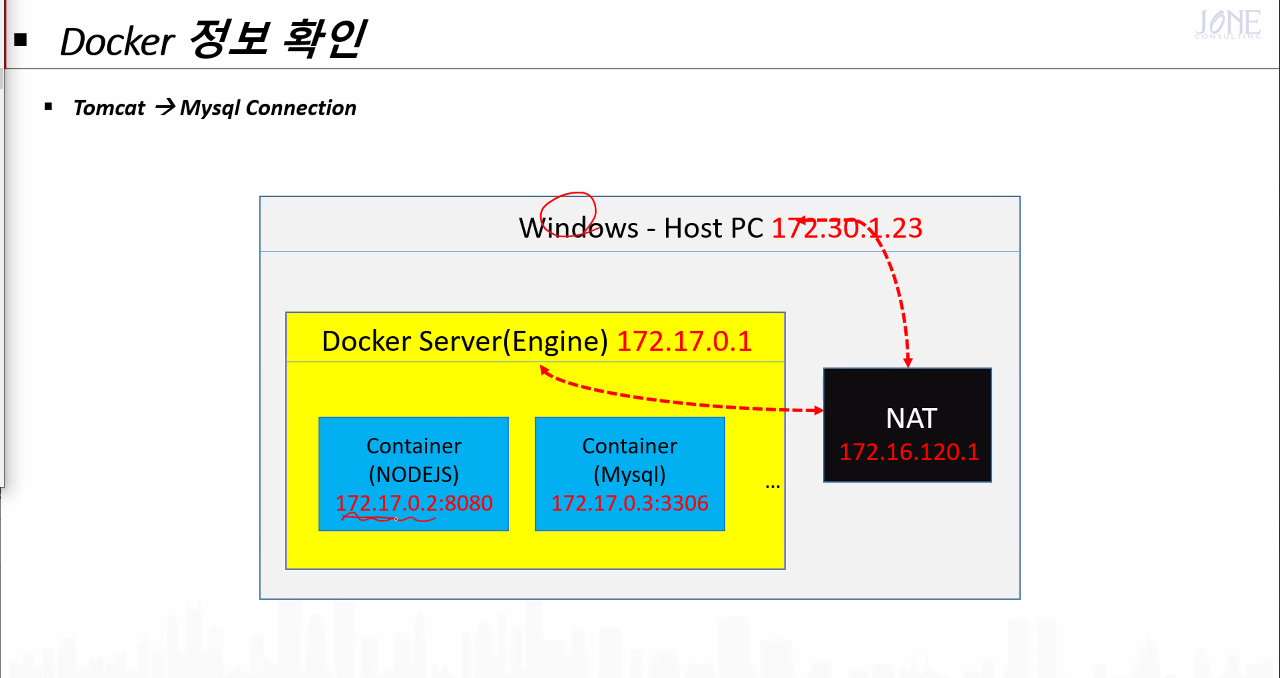


하면 하단에 프롬프트 창으로 이동한 것을 볼수있음

start하면

index.js의 

리스닝 내용이 뜬다



이렇게 내용수정하고

**docker run -d -p 8080:8080 mynodejs**

실행하면

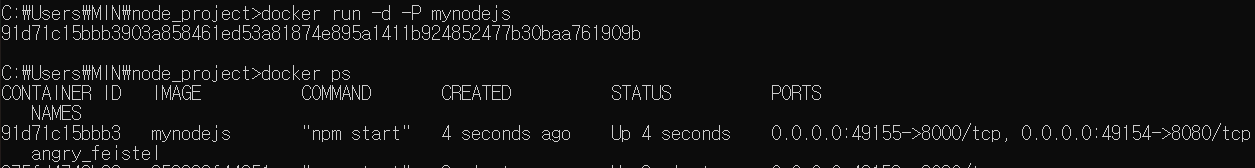
내용 나온다

**docker build --no-cache=true -t mynodejs .**

빌드할 때 이렇게 바꿔주는게 좋을듯

**docker run -d -P mynodejs**

**이렇게 하면 포트 열어주는것**



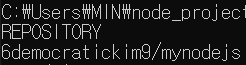
개발자나 운영자들이 다른 서비스에서 포트가 매번 바껴서 매번 빌드해야 할 수도

포트포워딩 안열어주면 포트

Localhost:8000대신 49154 해도 됨

**docker tag mynodejs:latest 6democratickim9/mynodejs:1.0**

**해당 계정으로 이름 바뀐다**

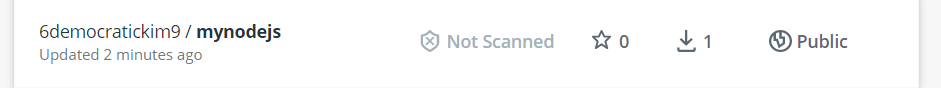
**!** 이렇게

**docker image push 6democratickim9/mynodejs:1.0**

**이렇게 해주고 업로드! 태그 이름까지 꼭 붙여줘야됨**

**docker image rm -f 1fe6ec01189e**

**강제삭제 -f**



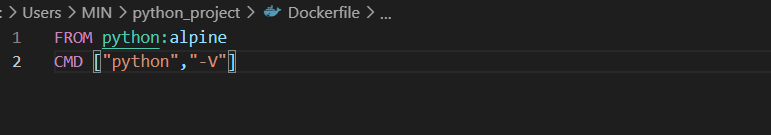
**아까 올려놨던게 허브사이트에 들어가면 뜬다!**

**docker image pull 6democratickim9/mynodejs:1.0**

**이제 pull 해준다.**

**이 떄 원본 파일이 로컬에 반드시 없어야됨----------------------------**

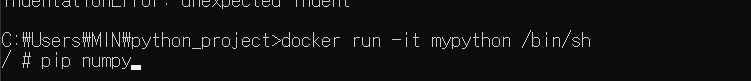
**Mypython 이라는 디렉토리 만들어주고 안에 들어가서**

**이런 내용을 작성한다**

**docker build -t mypython**

**docker run -it mypython**

**하면 버전 출력됨**

**컨테이너 상태에서 numpy 다운받아보자**