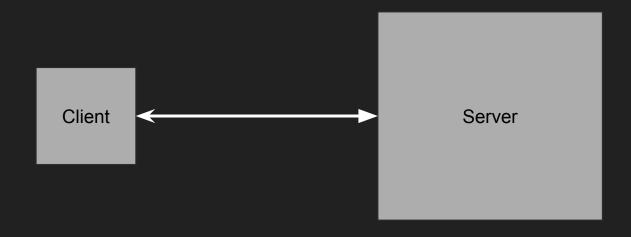
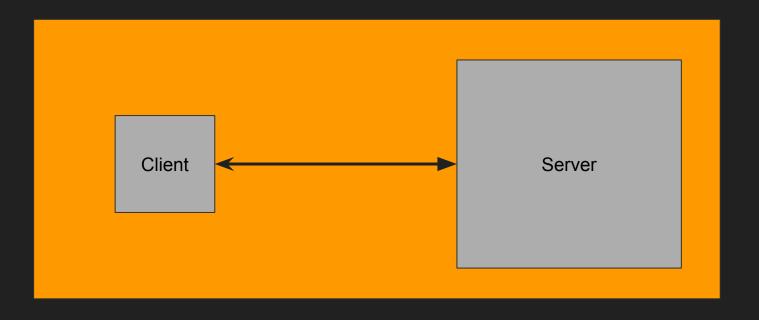
#### Docker

# Server Virtual Machine Container

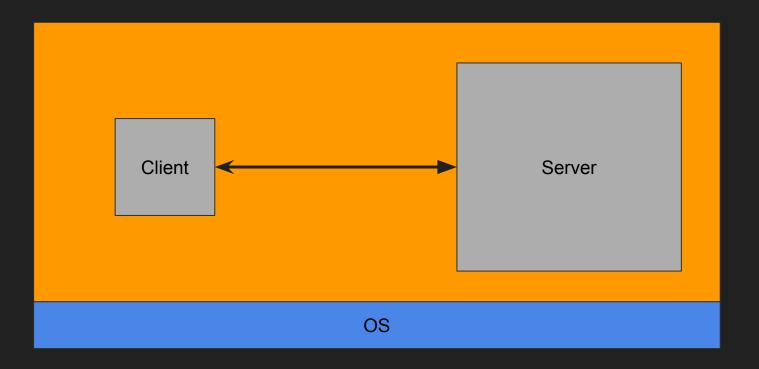
# Server

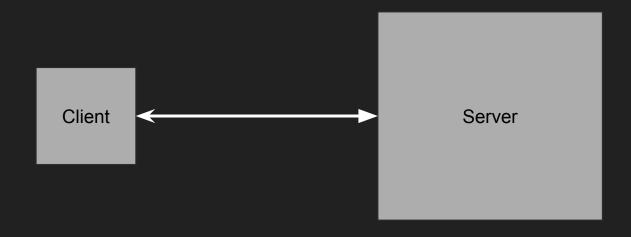


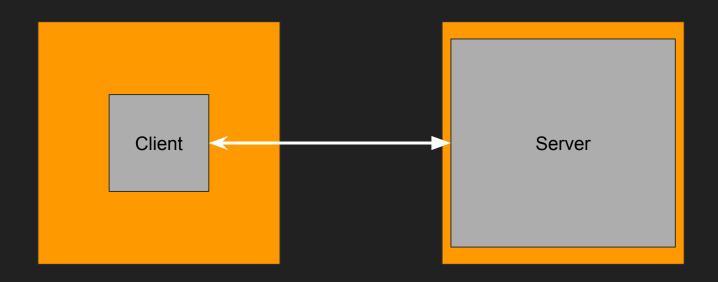
### Server

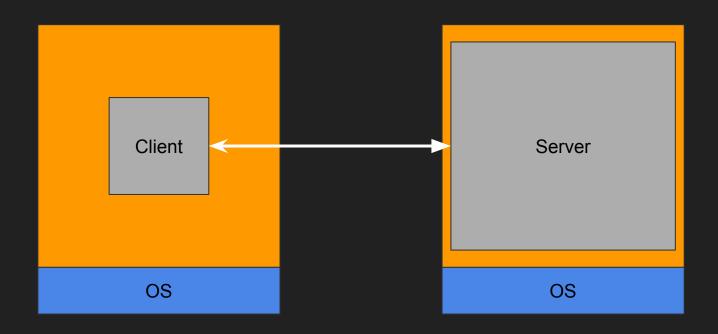


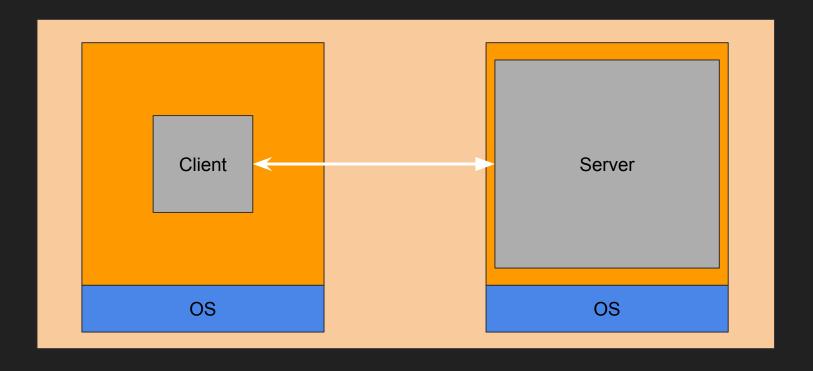
### Server

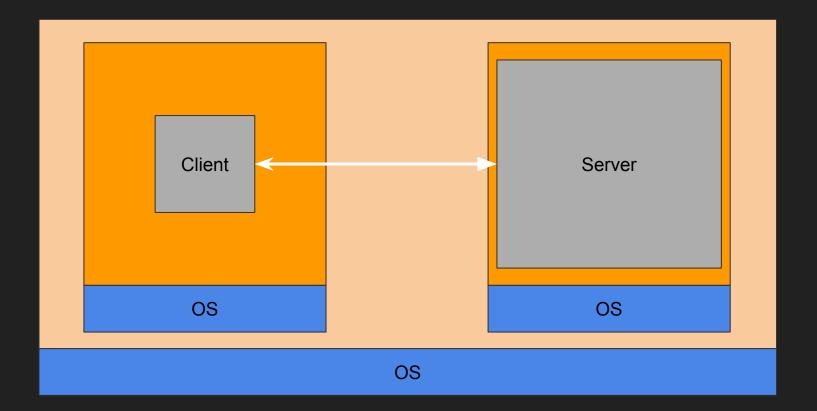


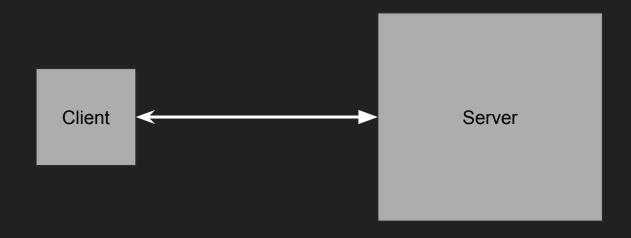


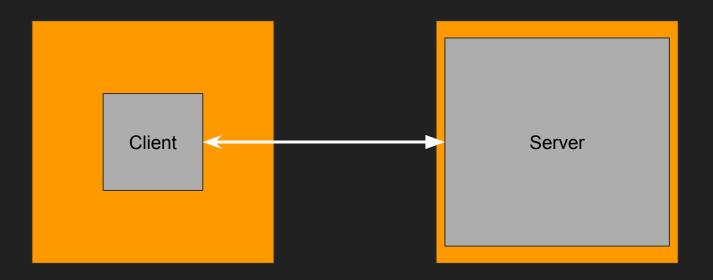


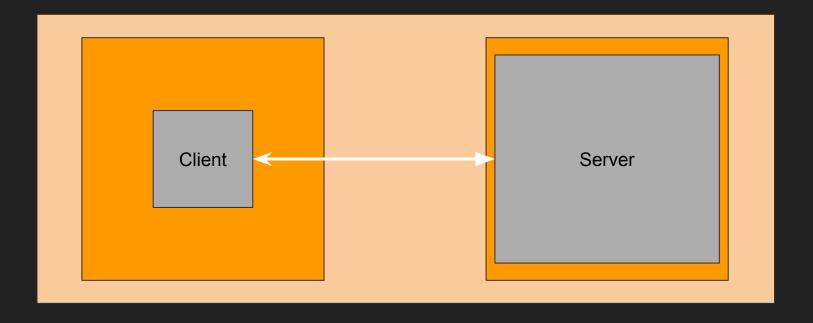


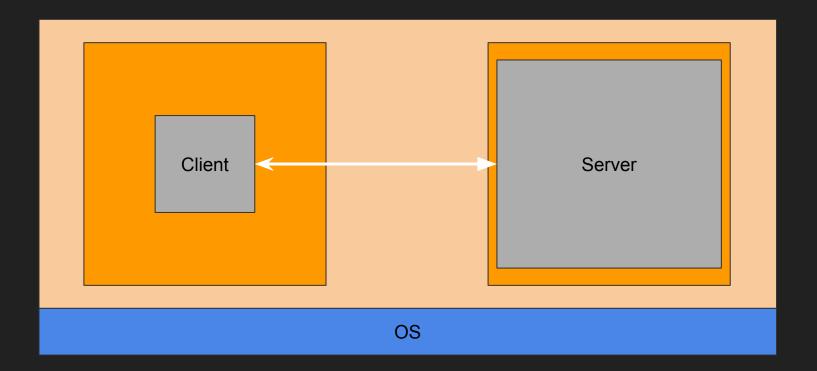








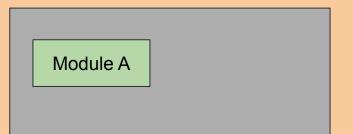


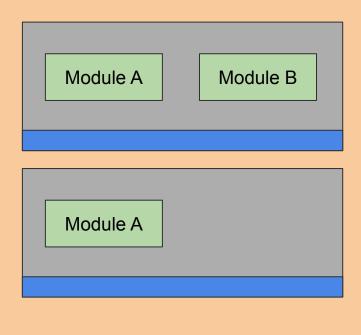


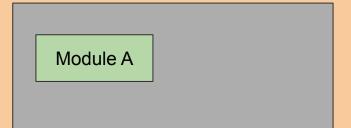
# Virtual Machine Container

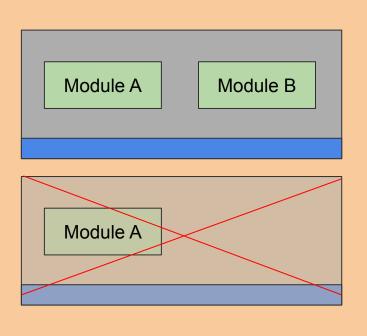
#### Container

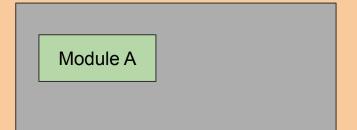
Module A

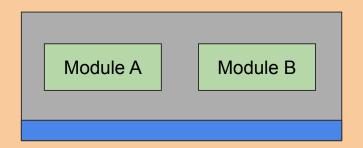


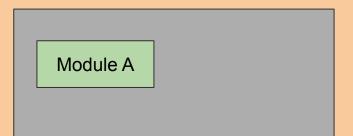


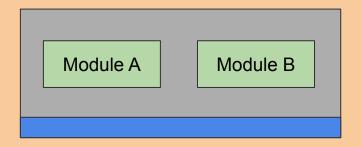


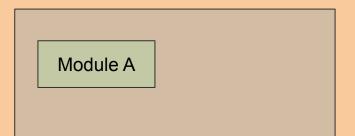


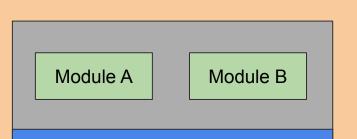


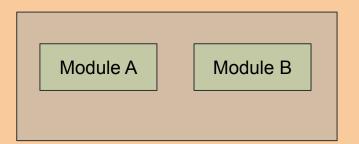




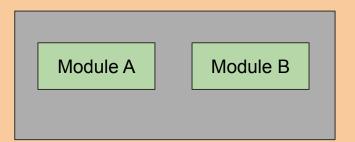


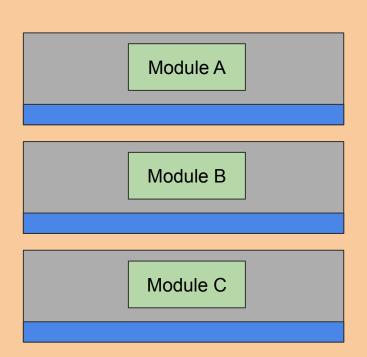


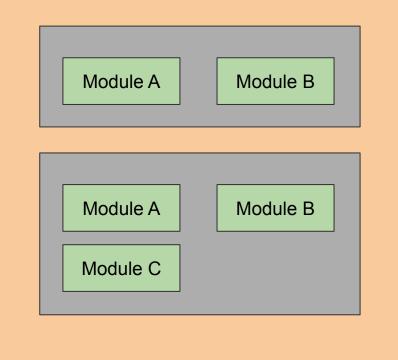












#### Container

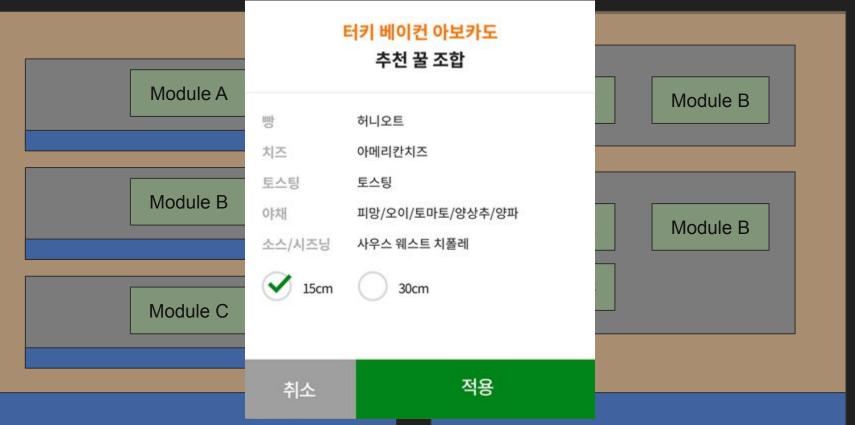
Module A

Module A

Module B

#### ▶ 주문방법

- 1. 메뉴선택 etc) 이탈리안 비엠티, 참치 등
- 2. 빵 선택 화이트, 파마산 오레가노, 위트, 허니오트, 하티, 플랫브레드
- 3. 치즈 선택 아메리칸치즈, 슈레드치즈, 모짜렐라치즈
- 4. 추가 선택 에그마요(+1600원), 페퍼로니(+900원), 베이컨(+1000원), 아보카도(+1300원), 오믈렛(+1300원)
- 5. 야채 선택 양상추, 토마토, 오이, 피망, 양파, 피클, 올리브, 할라피뇨
- 6. 소스 선택 머스타드, 레드와인식초, 스위트 어니언, 허니 머스타드, 스위트 칠리, 스모크 바베큐, 렌치, 마요네즈, 핫 칠리, 사우스 웨스트 치폴레, 홀스래디쉬, 올리브 오일, 소금, 후추 (소스 3가지까지 선택 가능)



#### Docker

·d redis 5 docker run --name redis -p 6379 \$ docker exec -d -v //var/run/d -it mariadb bash n root jenkin \$ docker run --name jenkins-docker -p 7979:8080 -p 50. ocker.sock:/var/run/docker.sock -v jenkins\_home:/var/jenk. \$ docker tag test:latest asdf:asdf s/jenkins:lts \$ docker logs feds docker stop fed? \$ docker build -t test .

5 docker logs feds docker stop fed2

\$ docker build -t test .

#### Dockerfile

FROM node:18.13.0 AS build RUN mkdir -p /app

WORKDIR /app COPY . .

RUN mkdir -p dist RUN npm i

RUN npm run build

FROM nginx

COPY --from=build /app/dist /usr/share/nginx/html

COPY nginx.conf /etc/nginx/conf.d 13 EXPOSE 80 443 14 CMD ["nginx", "-g", "daemon off;"]

RUN rm /etc/nginx/conf.d/default.conf

- FROM node:18.13.0 AS build RUN mkdir -p /app
  - WORKDIR /app
  - COPY . . RUN mkdir -p dist
  - RUN npm i
  - RUN npm run build

- FROM nginx

13 EXPOSE 80 443

- COPY --from=build /app/dist /usr/share/nginx/html

RUN rm /etc/nginx/conf.d/default.conf COPY nginx.conf /etc/nginx/conf.d

14 CMD ["nginx", "-g", "daemon off;"]

FROM node:18.13.0 AS build WORKDIR /app

COPY . .

RUN mkdir -p dist

13 EXPOSE 80 443

RUN npm i

RUN npm run build

FROM nginx

14 CMD ["nginx", "-g", "daemon off;"]

RUN rm /etc/nginx/conf.d/default.conf COPY nginx.conf /etc/nginx/conf.d

COPY --from=build /app/dist /usr/share/nginx/html

RUN mkdir -p /app 컨테이너 루트 위치에 app폴더 생성





```
FROM node:18.13.0 AS build
RUN mkdir -p /app
WORKDIR /app 뒤쪽 작업 모두 app폴더 안에서 할 거임
COPY . .
RUN mkdir -p dist
```

- RUN npm i
- RUN npm run build
- - FROM nginx
  - COPY -- from = build /app/dist /usr/share/nginx/html RUN rm /etc/nginx/conf.d/default.conf
- COPY nginx.conf /etc/nginx/conf.d 13 EXPOSE 80 443
- 14 CMD ["nginx", "-g", "daemon off;"]

FROM node:18.13.0 AS build RUN mkdir -p /app WORKDIR /app copy · · (로컬) 위치에 있는 것을 /app 폴더 안으로 복사

RUN npm i

RUN npm run build

FROM nginx

RUN mkdir -p dist

COPY --from=build /app/dist /usr/share/nginx/html RUN rm /etc/nginx/conf.d/default.conf COPY nginx.conf /etc/nginx/conf.d

13 EXPOSE 80 443

14 CMD ["nginx", "-g", "daemon off;"]

```
1 FROM node:18.13.0 AS build
2 RUN mkdir -p /app
3 WORKDIR /app
4 COPY . .
5 RUN mkdir -p dist
6 RUN npm i
7 RUN npm run build
```

COPY --from=build /app/dist /usr/share/nginx/html

RUN rm /etc/nginx/conf.d/default.conf COPY nginx.conf /etc/nginx/conf.d

14 CMD ["nginx", "-g", "daemon off;"]

FROM nginx

EXPOSE 80 443



Dockerfile 🖰 295 bytes

```
FROM node:18.13.0 AS build
RUN mkdir -p /app
WORKDIR /app
COPY . .
RUN mkdir -p dist
RUN npm i
RUN npm run build
FROM nginx
COPY --from=build /app/dist /usr/share/nginx/html ) /app/dist → (컨테이너) /usr/share/nginx/html
RUN rm /etc/nginx/conf.d/default.conf
```

COPY nginx.conf /etc/nginx/conf.d

14 CMD ["nginx", "-g", "daemon off;"]

EXPOSE 80 443

RUN mkdir -p /app WORKDIR /app

RUN mkdir -p dist

RUN npm i

COPY . .

RUN npm run build

13 EXPOSE 80 443

FROM nginx

COPY --from=build /app/dist /usr/share/nginx/html

RUN rm /etc/nginx/conf.d/default.conf (컨테이너) nginx 기본설정 제거 COPY nginx.conf /etc/nginx/conf.d

FROM node:18.13.0 AS build

14 CMD ["nginx", "-g", "daemon off;"]



- RUN mkdir -p /app
- WORKDIR /app COPY . .
- RUN mkdir -p dist
- RUN npm i
- RUN npm run build
- FROM nginx
- COPY -- from = build /app/dist /usr/share/nginx/html
- RUN rm /etc/nginx/conf.d/default.conf
  - COPY nginx.conf /etc/nginx/conf.d (로컬) nginx설정  $\rightarrow$  (컨테이너) nginx설정
- EXPOSE 80 443

FROM node:18.13.0 AS build

14 CMD ["nginx", "-g", "daemon off;"]

- FROM node:18.13.0 AS build RUN mkdir -p /app WORKDIR /app
- COPY . . RUN mkdir -p dist
- RUN npm i
- RUN npm run build
- FROM nginx
- COPY --from=build /app/dist /usr/share/nginx/html RUN rm /etc/nginx/conf.d/default.conf
  - COPY nginx.conf /etc/nginx/conf.d
- EXPOSE 80 443 localhost: 80 localhost: 443 14 CMD ["nginx", "-g", "daemon off;"]

\$ docker build -t test .

```
#1 [internal] load build definition from Dockerfile
#1 sha256:2835a0b9c9ecf2c22df14cc10656de9c6428794ce01251bea626c59c91cb7755
#1 transferring dockerfile: 347B 0.0s done
#1 DONE 0.1s
#2 [internal] load .dockerignore
#2 sha256:7bb46f0b59fa423e170018cda458e70e2f357b9bbd623c5521ae0371f85ba53f
#2 transferring context: 2B done
#2 DONE 0.1s
#3 [internal] load metadata for docker.io/library/nginx:latest
#3 sha256:06c466a4eb6821b81bd3e48610e5f38dab858b1e9acb01d6e2f6b11c8fabe6bc
#3 DONE 2.7s
#4 [internal] load metadata for docker.io/library/node:18.13.0
#4 sha256:db4f3ba1569e7a23d6941036c0b091c78d80a57dcbb6924e814f79b69a2bba60
#4 DONE 2.8s
#9 [internal] load build context
#9 sha256:f19d72fb925f8ceb4d33062964f12752170a8160368accf29d4db19f18c6e22a
#9 transferring context: 25.67MB 1.0s done
#9 DONE 1.0s
#6 [build 1/7] FROM docker.io/library/node:18.13.0@sha256:d871edd5b68105ebcbfcde3fe8c79d24cbdbb30430d9bd6251c57c56c7bd7646
#6 sha256:42aa2ea5e17a04b3cf61e55078cf44e18bd660068507bb3531e4f9df7c2f72b1
#6 resolve docker.io/library/node:18.13.0@sha256:d871edd5b68105ebcbfcde3fe8c79d24cbdbb30430d9bd6251c57c56c7bd7646 0.1s done
#6 sha256:d871edd5b68105ebcbfcde3fe8c79d24cbdbb30430d9bd6251c57c56c7bd7646 1.21kB / 1.21kB done
#6 sha256:d9061fd0205c20cd47f70bdc879a7a84fb472b822d3ad3158aeef40698d2ce36 2.21kB / 2.21kB done
#6 sha256:b68a472583efc83662785c5c6eb325c13b6662a3b421e7d78943e210efe3b254 7.52kB / 7.52kB done
#6 sha256:bbeef03cda1f5d6c9e20c310c1c91382a6b0a1a2501c3436b28152f13896f082 6.29MB / 55.03MB 0.9s
#6 sha256:f049f75f014ee8fec2d4728b203c9cbee0502ce142aec030f874aa28359e25f1 5.16MB / 5.16MB 0.6s done
#6 sha256:56261d0e6b05ece42650b14830960db5b42a9f23479d868256f91d96869ac0c2 2.10MB / 10.88MB 0.9s
#6 sha256:9bd150679dbdb02d9d4df4457d54211d6ee719ca7bc77747a7be4cd99ae03988 0B / 54.58MB 0.9s
#6 sha256:56261d0e6b05ece42650b14830960db5b42a9f23479d868256f91d96869ac0c2 4.19MB / 10.88MB 1.0s
#6 sha256:bbeef03cda1f5d6c9e20c310c1c91382a6b0a1a2501c3436b28152f13896f082 15.73MB / 55.03MB 1.2s
#6 sha256:56261d0e6b05ece42650b14830960db5b42a9f23479d868256f91d96869ac0c2 9.44MB / 10.88MB 1.2s
#6 sha256:bbeef03cda1f5d6c9e20c310c1c91382a6b0a1a2501c3436b28152f13896f082 21.31MB / 55.03MB 1.3s
#6 sha256:5b282ee9da04b94adc14866f2da98baed3cdf897c27f7490b8d3b1ae1522ddc7 0B / 196.88MB 1.3s
#6 sha256:bbeef03cda1f5d6c9e20c310c1c91382a6b0a1a2501c3436b28152f13896f082 25.17MB / 55.03MB 1.4s
#6 sha256:56261d0e6b05ece42650b14830960db5b42a9f23479d868256f91d96869ac0c2 10.88MB / 10.88MB 1.2s done
#6 sha256:bbeef03cda1f5d6c9e20c310c1c91382a6b0a1a2501c3436b28152f13896f082 31.46MB / 55.03MB 1.6s
#6 sha256:bbeef03cda1f5d6c9e20c310c1c91382a6b0a1a2501c3436b28152f13896f082 37.75MB / 55.03MB 1.7s
#6 sha256:9bd150679dbdb02d9d4df4457d54211d6ee719ca7bc77747a7be4cd99ae03988 3.15MB / 54.58MB 1.7s
#6 sha256:bbeef03cda1f5d6c9e20c310c1c91382a6b0a1a2501c3436b28152f13896f082 45.09MB / 55.03MB 2.0s
#6 sha256:9bd150679dbdb02d9d4df4457d54211d6ee719ca7bc77747a7be4cd99ae03988 10.49MB / 54.58MB 2.0s
#6 sha256:bbeef03cda1f5d6c9e20c310c1c91382a6b0a1a2501c3436b28152f13896f082 49.28MB / 55.03MB 2.2s
#6 sha256:9bd150679dbdb02d9d4df4457d54211d6ee719ca7bc77747a7be4cd99ae03988 14.68MB / 54.58MB 2.2s
#6 sha256:9bd150679dbdb02d9d4df4457d54211d6ee719ca7bc77747a7be4cd99ae03988 17.83MB / 54.58MB 2.4s
#6 sha256:bbeef03cda1f5d6c9e20c310c1c91382a6b0a1a2501c3436b28152f13896f082 53.48MB / 55.03MB 2.5s
```

\$ docker build -t test .

```
#6 extracting sha256:5201db2cd6c6e6a1cb611884f264dbbcbbfdc49091d891e28ae332e8ade
                                     8674e
                                     #6 extracting sha256:5201db2cd6c6e6a1cb611884f264dbbcbbfdc49091d891e28ae332e8ade
                                     8674e 1.5s done
                                     #6 extracting sha256:2a7091f85153a0d40843e4009e0455af83c62c499f0c2754abe4e3a3b3f
                                     37cc0
                                     #6 extracting sha256:2a7091f85153a0d40843e4009e0455af83c62c499f0c2754abe4e3a3b3f
                                     37cc0 0.1s done
FROM node: 18.13.0 AS build
                                     #6 extracting sha256:2bcce6ea6105ab5bcfc37e6303d9064e3054735beaff5fe6ab0cc5b34fb
RUN mkdir -p /app
                                     61694 done
                                     #6 DONE 19.2s
WORKDIR /app
COPY . .
                                     #7 [build 2/7] RUN mkdir -p /app
                                     #7 sha256:ef3ecb4a81575ffc9a990288f5ce3367a4189cf7a1469693f8adf3f0d260ff25
RUN mkdir -p dist
                                     #7 DONE 2.9s
RUN npm i
                                     #8 [build 3/7] WORKDIR /app
                                     #8 sha256:ffdc1a302955597d3762139f20801939a871683dbbf8574324788f7e88ac128d
                                     #8 DONE 0.1s
                                     #10 [build 4/7] COPY . .
                                     #10 sha256:ba5d54d002f55e7ecea80faa5352d8686f5f296a951d33290eac6825dccab849
                                     #10 DONE 0.2s
                                     #11 [build 5/7] RUN mkdir -p dist
                                     #11 sha256:30f557590bf3cd7dd50a8156541a5c825ac4058dcb8e39351b4b03d6536b3558
                                     #11 DONE 0.4s
                                     #12 [build 6/7] RUN npm i
                                     #12 sha256:3b5acb1ed6d0d080d5a5b6e220292f945cbf088253df6566afb3a07be7eb7327
```

2ddc7 4.5s done

f8ddf done

#6 [build 1/7] FROM docker.io/library/node:18.13.0@sha256:d871edd5b68105ebcbfcde

#6 extracting sha256:8bc43c905b24c76f05269590f2a9971756515cf<u>54cd1c59e87ed6f1f197</u>

#6 sha256:42aa2ea5e17a04b3cf61e55078cf44e18bd660068507bb3531e4f9df7c2f72b1 #6 extracting sha256:5b282ee9da04b94adc14866f2da98baed3cdf897c27f7490b8d3b1ae152

3fe8c79d24cbdbb30430d9bd6251c57c56c7bd7646

#### Dockerfile 👸 295 bytes

```
2 RUN mkdir -p /app
```

- WORKDIR /app
- 5 RUN mkdir -p dist
- 6 RUN npm i
- 6 KON ripm 1
- 7 RUN npm run build
- 9 FROM nginx
- 10 COPY --from=build /app/dist /usr/share/nginx/html
- 11 RUN rm /etc/nginx/conf.d/default.conf
- 12 COPY nginx.conf /etc/nginx/conf.d

FROM node:18.13.0 AS build

- 13 EXPOSE 80 443
- 14 CMD ["nginx", "-g", "daemon off;"]

```
de19d34 [FIX] docker compose yaml
7a670b2 Merge branch 'fe' into 'fe-build'
54bc905 Merge branch 'fe-login' into 'fe'
37d10b7 [FEAT] naver dynamic map
508f090 [FEAT] user location label
1919aad [FEAT] message history
54f0814 [FEAT] my message history
```



#### Dockerfile 👸 295 bytes

1	FROM node:18.13.0 AS build
	RUN mkdir -p /app
	WORKDIR /app
	COPY
	RUN mkdir -p dist
	RUN npm i
	RUN npm run build
	FROM nginx
10	COPYfrom=build /app/dist /usr/share/nginx/html
11	RUN rm /etc/nginx/conf.d/default.conf
12	COPY nginx.conf /etc/nginx/conf.d
13	EXPOSE 80 443

14 CMD ["nginx", "-g", "daemon off;"]

