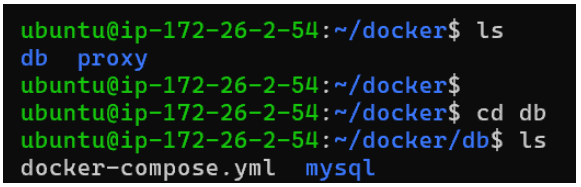
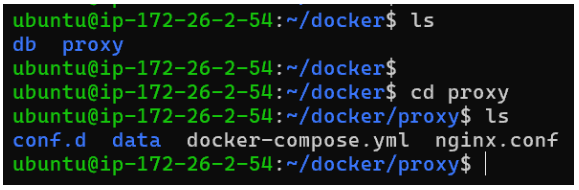
DB 및 Nginx (certbot) 웹 서버 컨테이너 구성

EC2 내부 DB 관련 파일 경로



EC2 내부 Nginx 관련 파일 경로




```
sudo mkdir -p ~/docker/db
sudo chown -R ubuntu:ubuntu ~/docker
 cd ~/docker
 ls
 cd db
nano docker-compose.yml → 파일 생성
```

6. PostgreSQL, Redis 컨테이너 기동

6.1. docker-compose.yml

```
# EC2 내부 경로 : /home/ubuntu/docker/db/docker-compose.yml

services:
  postgres:
    image: postgres:15
    container_name: postgres
    restart: always
    environment:
      POSTGRES_USER: ${POSTGRES_USER}
      POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
      POSTGRES_DB: ${POSTGRES_DB}
    ports:
      - "5432:5432"
```

```
    volumes:
      - postgres_data:/var/lib/postgresql/data
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U ${POSTGRES_USER}"]
      interval: 10s
      timeout: 5s
      retries: 5
    networks:
      - app-network

  redis:
    image: redis:7
    container_name: redis
    ports:
      - "${REDIS_PORT}:6379"
    volumes:
      - redis_data:/data
    command: redis-server --requirepass '${REDIS_PASSWORD}' --appendonl
    networks:
      - app-network

volumes:
  postgres_data:
  redis_data:

networks:
  app-network:
    external: true
```

7.2 docker 명령어 실행

```
#해당 폴더 위치로 이동
cd home/ubuntu/docker/db/docker-compose.yml

# 현재 디렉토리의 docker-compose.yml 파일 기반으로 서비스 시작
docker compose up -d

# 특정 docker-compose.yml 파일 지정하여 서비스 시작
```

```
docker compose -f docker-compose.yml up -d

# 현재 디렉토리의 docker-compose.yml 파일 기반으로 서비스 중지 및 삭제
docker compose down
```

7.3 DB init 설정하기

```
# PostgreSQL 컨테이너 접속
docker exec -it ptsd bash

# PostgreSQL에 root 유저로 접속 (Postgres는 보통 'postgres' 사용자로 접속해)
psql -U ptsd

# 데이터베이스 사용
\c ptsd

# 권한 부여
GRANT ALL PRIVILEGES ON DATABASE ptsd TO ptsd;

# 확인하기
\du
\l

# 나가기
\q

컨테이너 다시띄우기

docker compose down
docker compose up -d
```

7. NGINX 컨테이너 기동

7.1. docker-compose.yml

```
# EC2 내부 경로 : home/ubuntu/docker/proxy/docker-compose.yml
services:
```

```
    nginx:
      image: nginx:latest
      container_name: nginx
      ports:          # 포트 매핑 (호스트:컨테이너)
        - "80:80"    # HTTP 트래픽용 포트
        - "443:443"  # HTTPS 트래픽용 포트
      volumes:       # 볼륨 마운트 (호스트 경로:컨테이너 경로)
     - ./nginx.conf:/etc/nginx/nginx.conf
     - ./conf.d:/etc/nginx/conf.d
     - ./data/certbot/conf:/etc/letsencrypt    # SSL 인증서 저장 위치
     - ./data/certbot/www:/var/www/certbot     # Let's Encrypt 인증 파일 위치
        - /home/ubuntu/inquiry/images:/var/www/inquiry/images  # 호스트 디
        - ./frontend/dist:/usr/share/nginx/html   # 프론트 빌드 결과물 연결
    command: "/bin/sh -c 'while :; do sleep 6h & wait $${!}; nginx -s reload; d
      networks:
        - app-network
      restart: always

    certbot:
      image: certbot/certbot
      container_name: certbot
      volumes:
     - ./data/certbot/conf:/etc/letsencrypt
      - ./data/certbot/www:/var/www/certbot
      networks:
        - app-network
    entrypoint: "/bin/sh -c 'trap exit TERM; while :; do certbot renew; sleep 12

networks:
  app-network:       # app-network에 연결 (백엔드와 통신하기 위함)
    external: true    # 컨테이너가 종료되면 항상 재시작
```

8. CertBot Https 인증서 발급/적용

CertBot 전용 발급 nginx.conf 작성 → 발급 후 완전한 nginx.conf 변경

8.1. CertBot 전용 발급 nginx.conf

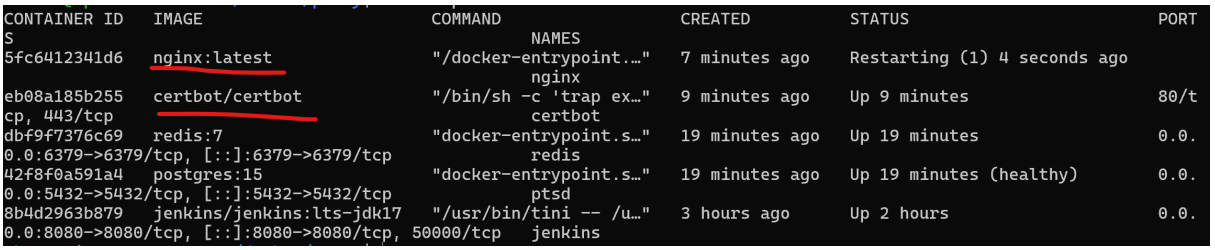
```
# EC2 내부 경로 : home/ubuntu/docker/proxy/nginx.conf

events {}

http {
    server {
        listen 80;
        server_name k12d101.p.ssafy.io;

        location /.well-known/acme-challenge/ {
            root /var/www/certbot;
            allow all;
        }

        location / {
            return 404;
        }
    }
}
}
```

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    upstream backend {
        server ptsd-app:8000;
```

```
    }

    server {
        listen 80;
        listen [::]:80;
        server_name k12d101.p.ssafy.io;

        location /.well-known/acme-challenge/ {
            root /var/www/certbot;
        }

        location / {
            return 301 https://$server_name$request_uri;
        }
    }

    server {
        listen 443 ssl;                    # SSL 포트 리스닝
        listen [::]:443 ssl;               # IPv6 지원
        server_name k12d101.p.ssafy.io;      # 도메인 이름 설정
        server_tokens off;                   # 서버 버전 정보 숨김

        # SSL 인증서 설정
        ssl_certificate /etc/letsencrypt/live/k12d101.p.ssafy.io/fullchain.pem;     #
        ssl_certificate_key /etc/letsencrypt/live/k12d101.p.ssafy.io/privkey.pem;
        include /etc/letsencrypt/options-ssl-nginx.conf;                            # SSL 옵
        ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

        # FastAPI
        location / {
            proxy_pass http://backend;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
        }

        # 🔥 Jenkins 추가
```

```
        location /jenkins {
            proxy_pass http://jenkins:8080/jenkins;
            proxy_http_version 1.1;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
        }
    }
}
```

docker compose restart nginx → 컨테이너 재 시



```
CONTAINER ID   IMAGE                          COMMAND                  CREATED          STATUS                          PORT
S                                                                      NAMES
5fc6412341d6   nginx:latest                   "/docker-entrypoint.…"   7 minutes ago    Restarting (1) 4 seconds ago
                                                                       nginx
eb08a185b255   certbot/certbot                "/bin/sh -c 'trap ex…"   9 minutes ago    Up 9 minutes                    80/t
cp, 443/tcp                                                            certbot
dbf9f7376c69   redis:7                        "docker-entrypoint.s…"   19 minutes ago   Up 19 minutes                   0.0.
0.0:6379->6379/tcp, [::]:6379->6379/tcp                                redis
42f8f0a591a4   postgres:15                    "docker-entrypoint.s…"   19 minutes ago   Up 19 minutes (healthy)         0.0.
0.0:5432->5432/tcp, [::]:5432->5432/tcp                                ptsd
8b4d2963b879   jenkins/jenkins:lts-jdk17      "/usr/bin/tini -- /u…"   3 hours ago      Up 2 hours                      0.0.
0.0:8080->8080/tcp, [::]:8080->8080/tcp, 50000/tcp   jenkins
```

8.2. 폴더 생성 및 권한 설정

```
# Certbot 관련 디렉토리 생성
mkdir -p data/certbot/conf  # 인증서가 저장될 경로

mkdir -p data/certbot/conf    # 인증 챌린지 파일이 저장될 경로

# 디렉토리 소유권 및 권한 설정
sudo chown -R ubuntu:ubuntu data/certbot  # ubuntu 사용자로 소유권 변경
sudo chmod -R 755 data/certbot            # 읽기/쓰기 권한 설정

# 인증서 발급 명령어 실행
# --webroot: 웹 서버 루트 경로 방식으로 인증
# -w: 웹 루트 디렉토리 지정
# -d: 인증서를 발급받을 도메인
# --force-renewal: 기존 인증서가 있어도 강제로 갱신
ubuntu@ip-172-26-2-57:~/docker/proxy$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED          STA
```

```
30e7707f1a55   ptsd-app              "uvicorn PTSD.main:a..."   38 minutes ago
8bac4a3fd569   eclipse-mosquitto:2   "/docker-entrypoint...."   38 minutes ag
fe11f9a76c30   nginx:latest          "/docker-entrypoint...."   About an hour ago
b3594632205f   jenkins/jenkins:2.501   "/usr/bin/tini -- /u..."   5 days ago
ded084ae3684   postgres:15           "docker-entrypoint.s..."   12 days ago
2166e91afb5c   redis:7               "docker-entrypoint.s..."   12 days ago        Up
```

8.3. Certbot 인증서 발급 완료

```
ubuntu@ip-172-26-2-54:~/docker/proxy$ docker compose exec certbot certb
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for j12d109.p.ssafy.io

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/j12d109.p.ssafy.io/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/j12d109.p.ssafy.io/privkey.pem
This certificate expires on 2025-06-18.
These files will be updated when the certificate renews.

NEXT STEPS:
- The certificate will need to be renewed before it expires. Certbot can automa

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

8.4. 인증서 발급이 성공되면 SSL pem 파일 작성

```
# Diffie-Hellman 파라미터 생성 (SSL 보안 강화)
# 2048비트 키를 사용하여 생성
ubuntu@ip-172-26-2-57:~/docker/proxy$
명령어 ->  openssl dhparam -out data/certbot/conf/ssl-dhparams.pem 2048
```

8.5. Nginx 작성 (최종본)

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    # ✅ FastAPI 백엔드 (ptsd 서비스)
    upstream ptsd-app {
        server ptsd-app:8000;
    }

    # ✅ React 프론트엔드 (react 서비스)
    upstream react {
        server react:80;
    }

    # ✅ HTTP (Let's Encrypt 인증서 발급용)
    server {
        listen 80;
        listen [::]:80;
        server_name k12d101.p.ssafy.io;

        # certbot 인증 파일 경로
        location /.well-known/acme-challenge/ {
            root /var/www/certbot;
            allow all;
        }

        # 나머지 HTTP 요청은 HTTPS로 리디렉션
        location / {
            return 301 https://$server_name$request_uri;
```

```
    }
}

# ✅ HTTPS 서비스 블록 (실제 서비스용)
server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name k12d101.p.ssafy.io;
    server_tokens off;

    # SSL 인증서 설정
    ssl_certificate /etc/letsencrypt/live/k12d101.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/k12d101.p.ssafy.io/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    # 보안 헤더 추가
    add_header X-Content-Type-Options "nosniff" always;
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-XSS-Protection "1; mode=block" always;
    add_header Referrer-Policy "no-referrer" always;

    # ✅ React 프론트엔드
    location / {
        proxy_pass http://react;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    # ✅ FastAPI 백엔드
    location /api/ {
        proxy_pass http://ptsd-app;
        proxy_http_version 1.1;
```

```
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
    # ✅ WebSocket 경로 추가

    location /ws/ {
        proxy_pass http://ptsd-app/ws/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    # ✅ Jenkins
    location /jenkins {
        proxy_pass http://jenkins-v2501:8080/jenkins;
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    # ✅ 인증서 갱신용 (자동 renew)
    location /.well-known/acme-challenge/ {
        root /var/www/certbot;
        allow all;
    }
  }
} //
```