



Main webserver 설정



작업 디렉토리: /home/ubuntu/webserver

1. docker network 생성

```
docker network create -d bridge chaekbang-webserver-net
```

2. docker-compose.yml 파일 생성

```
version: '3.8'

services:
  nginx:
    container_name: nginx-main
    image: nginx:latest
    restart: always
    #외부와 소통하는 경우에는 ports로 포트 설정.
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - "./proxy/nginx.conf:/etc/nginx/nginx.conf"
      - "./data/certbot/conf:/etc/letsencrypt:ro"
      - "./data/certbot/www:/var/www/certbot:ro"
      - "/var/lib/docker/volumes/jenkins_jenkins-data/_data/workspace/chaekbang_front/Frontend/build:/var/chaekbang_front"
    environment:
      - TZ=Asia/Seoul
    command: "/bin/sh -c 'while ;; do sleep 6h & wait $$!}; nginx -s reload; done & nginx -g \"daemon off;\""
    networks:
      - chaekbang-webserver-net
```

```

- chaekbang-jenkins-net

certbot:
  container_name: webserver-certbot
  image: certbot/certbot
  restart: unless-stopped
  volumes:
    - "./data/certbot/conf:/etc/letsencrypt:rw"
    - "./data/certbot/www:/var/www/certbot:rw"
  entrypoint: "/bin/sh -c 'trap exit TERM; while :; do certbot renew; sleep 12h & wait
  $$!}; done;'"

networks:
  chaekbang-webserver-net:
    external: true
    name: "chaekbang-webserver-net"
  chaekbang-jenkins-net:
    external: true
    name: "chaekbang-jenkins-net"

```

3. 하위 proxy 디렉토리에 nginx.conf 파일 생성

```

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;
    client_max_body_size 20M;

    server {
        listen 80;
        listen [::]:80;
        server_name www.chaekbang.xyz prod.chaekbang.xyz dev.chaekbang.xyz jenkins.chaekba
ng.xyz;
        server_tokens off;

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

```

```

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

    server {
        listen 443 ssl;
        server_name www.chaekbang.xyz;
        server_tokens off;

        ssl_certificate /etc/letsencrypt/live/www.chaekbang.xyz/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/www.chaekbang.xyz/privkey.pem;
        include /etc/letsencrypt/options-ssl-nginx.conf;
        ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

        location / {
            root /var/chaekbang_front;
            index index.html index.htm;
            try_files $uri $uri/ /index.html;
        }
    }

    server {
        listen 443 ssl;
        server_name dev.chaekbang.xyz;
        server_tokens off;

        ssl_certificate /etc/letsencrypt/live/dev.chaekbang.xyz/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/dev.chaekbang.xyz/privkey.pem;
        include /etc/letsencrypt/options-ssl-nginx.conf;
        ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

        location / {
            proxy_set_header Host $http_host;
            proxy_http_version 1.1;
            proxy_pass http://nginx-dev:80;
            proxy_set_header X-Real-IP $remote_addr;
        }

        location /ws {
            proxy_pass http://nginx-dev:80/ws;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";
            proxy_set_header Host $host;
        }
    }
}

```

```

server {
    listen 443 ssl;
    server_name prod.chaekbang.xyz;
    server_tokens off;

    ssl_certificate /etc/letsencrypt/live/prod.chaekbang.xyz/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/prod.chaekbang.xyz/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    location / {
        proxy_set_header Host $http_host;
        proxy_http_version 1.1;
        proxy_pass http://nginx-prod:80;
        proxy_set_header X-Real-IP $remote_addr;
    }

    location /ws {
        proxy_pass http://nginx-prod:80/ws;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
    }
}

server {
    listen 443 ssl;
    server_name jenkins.chaekbang.xyz;
    server_tokens off;

    ssl_certificate /etc/letsencrypt/live/jenkins.chaekbang.xyz/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/jenkins.chaekbang.xyz/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    location / {
        proxy_set_header Host $http_host;
        proxy_http_version 1.1;
        proxy_pass http://jenkins:8080;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                '$status $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for"';

access_log /var/log/nginx/access.log main;

```

```
    sendfile          on;  
    keepalive_timeout 65;  
    include /etc/nginx/conf.d/*.conf;  
}
```

4. nginx-dev, nginx-prod, jenkins container가 다 떠있는지 확인 후 docker compose 실행

```
docker compose -p main-webserver up -d
```