1.

-frontEnd

-FileService

-SessionService

-SessionFunctionService

-APIGateway

-MemberService

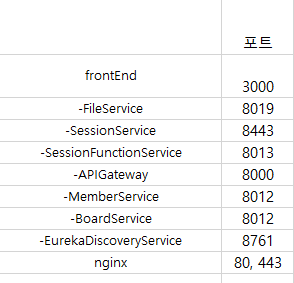
-BoardService

-EurekaDiscoveryService

상기한 브랜치를 클론하고, ec2위에 jenkins와 docker를 활용해 빌드.

빌드에 필요한 docekrfile은 exec 파일에 dockerfiles 폴더에 정리해놨음.

각 dockerfile을 /test/<서비스명>으로 만들어지는 ec2 디렉토리에 저장하고 빌드.



각 서비스별 포트는 다음과 같음.

Ec2에 docker 설치 후 Jenkins를 도커로 설치. 그 후 상기한 모든 서비스를 docker를 활용해 설치함. Nginx conf는 다음과 같음.

user nginx;

worker\_processes 1;

error\_log /var/log/nginx/error.log warn;

pid /var/run/nginx.pid;

events {

worker\_connections 1024;

}

http {

upstream FrontServer{

server 172.17.0.1:3000;

keepalive 32;

}

server {

listen 80;

listen [::]:80;

server\_name i8a703.p.ssafy.io;

location /{

proxy\_pass http://FrontServer/;

}

location ^~/.well-known/acme-challenge/ {

default\_type "text/plain";

root /var/www/cossafyco;

}

}

server {

listen 443 ssl;

server\_name ssafy.cossafyco.kro.kr;

#access\_log /var/log/nginx/proxy/access.log;

#error\_log /var/log/nginx/proxy/error.log;

ssl\_certificate fullchain.pem;

ssl\_certificate\_key privkey.pem;

location / {

proxy\_pass http://FrontServer/;

}

location /api{

rewrite ^/api(.\*)$ $1 break;

proxy\_pass http://172.17.0.1:8000;

}

location /session {

rewrite ^/session(.\*)$ $1 break;

proxy\_pass https://172.17.0.1:8443;

}

location /ab {

rewrite ^/ab(.\*)$ $1 break;

proxy\_pass http://172.17.0.1:8011;

}

}

include /etc/nginx/mime.types;

default\_type application/octet-stream;

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;

#tcp\_nopush on;

keepalive\_timeout 65;

#gzip on;

include /etc/nginx/conf.d/\*.conf;

#security

server\_tokens off;