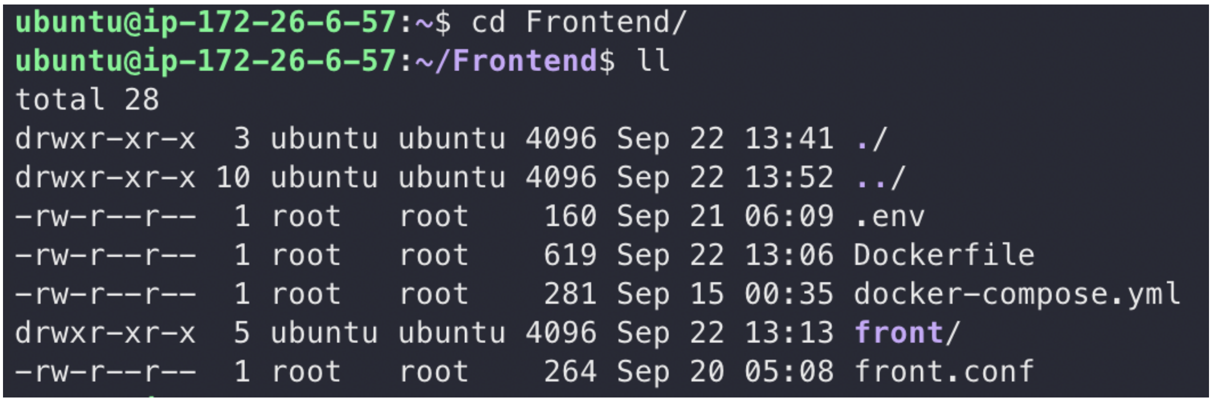
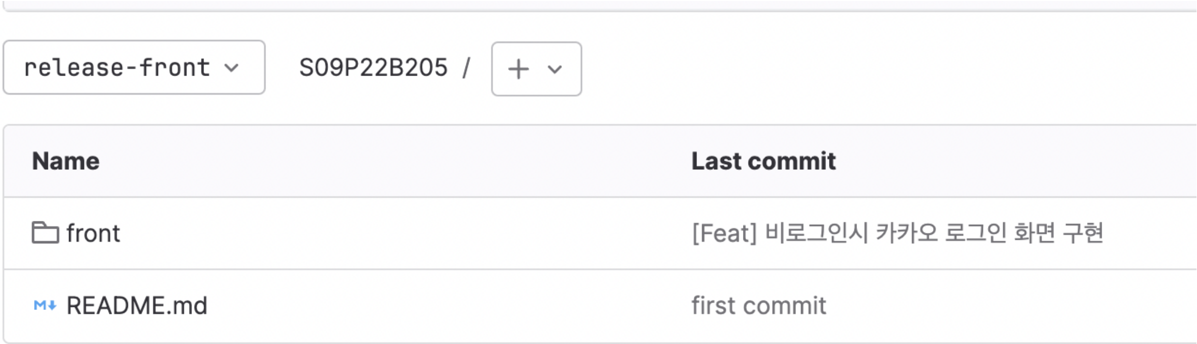
프론트엔드 - React

ec2 **디렉토리 구조**

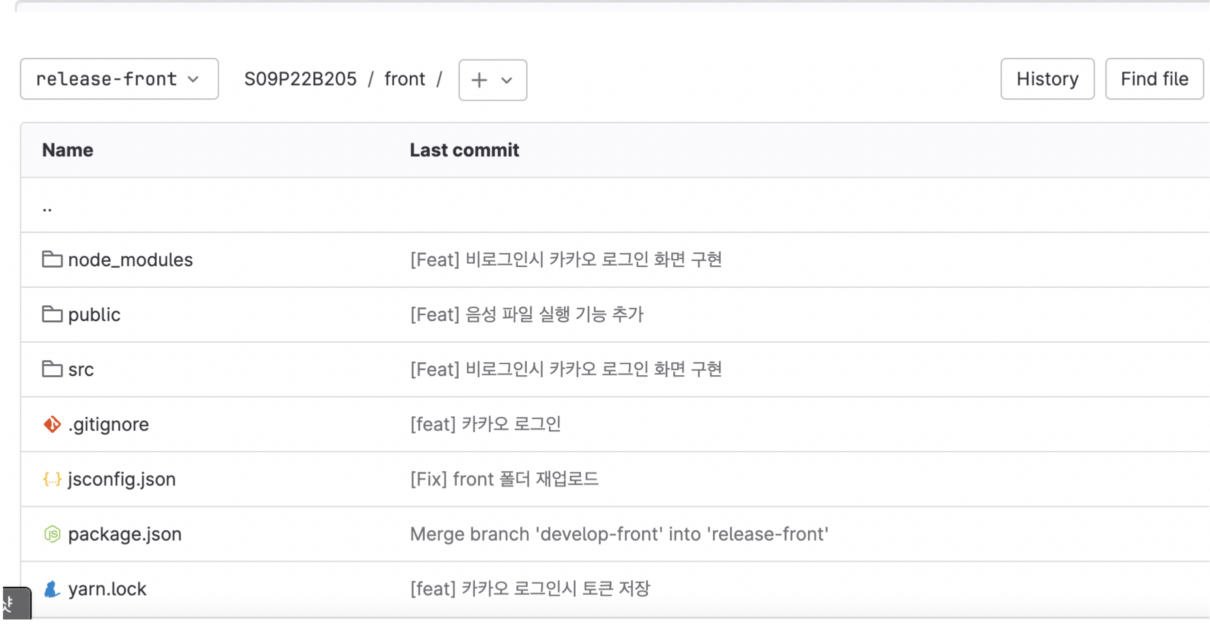


```
ubuntu@ip-172-26-6-57:~$ cd Frontend/
ubuntu@ip-172-26-6-57:~/Frontend$ ll
total 28
drwxr-xr-x  3 ubuntu ubuntu 4096 Sep 22 13:41 ./
drwxr-xr-x 10 ubuntu ubuntu 4096 Sep 22 13:52 ../
-rw-r--r--  1 root   root    160 Sep 21 06:09 .env
-rw-r--r--  1 root   root    619 Sep 22 13:06 Dockerfile
-rw-r--r--  1 root   root    281 Sep 15 00:35 docker-compose.yml
drwxr-xr-x  5 ubuntu ubuntu 4096 Sep 22 13:13 front/
-rw-r--r--  1 root   root    264 Sep 20 05:08 front.conf
```

프로젝트 디렉토리 구조



Name	Last commit
front	[Feat] 비로그인시 카카오 로그인 화면 구현
README.md	first commit



Name	Last commit
..	
node_modules	[Feat] 비로그인시 카카오 로그인 화면 구현
public	[Feat] 음성 파일 실행 기능 추가
src	[Feat] 비로그인시 카카오 로그인 화면 구현
.gitignore	[feat] 카카오 로그인
jsconfig.json	[Fix] front 폴더 재업로드
package.json	Merge branch 'develop-front' into 'release-front'
yarn.lock	[feat] 카카오 로그인시 토큰 저장

젠킨스 React용 파이프라인

```
pipeline {

    agent any

    stages {
        stage('github clone') {
            steps {
                git branch: 'release-front', credentialsId: 'jenkins_token', url: 'https://lab.ssafy.com/s09-ai-speech-sub2/S09P22B205
            }
        }

        stage('Build') {
            steps {
                dir("./front") {
                    nodejs(nodeJSInstallationName: 'NodeJS 16.14.0') {
                        sh 'CI=false npm install && CI=sudo chmod +x node_modules/.bin/react-scripts'
                        sh 'CI=false npm run build'
                    }
                }
            }
        }

        stage('Compression') {
            steps {
                dir("./front") {
                    sh '''
                    rm -rf node_modules
                    tar -cvf front_0.1.0.tar .
                    '''
                }
            }
        }

        stage('Deploy') {
            steps {
                sshagent(credentials: ['ssh_key']) {
                    sh '''
                        ssh -o StrictHostKeyChecking=no ubuntu@43.202.44.111
                        scp /var/jenkins_home/workspace/front/front/front_0.1.0.tar ubuntu@43.202.44.111:/home/ubuntu/Frontend
                        ssh -t ubuntu@43.202.44.111 ./deploy_fe.sh
                    '''
                }
                timeout(time: 30, unit: 'SECONDS') {
                }
            }
        }
    }
}
```

/home/ubuntu/Frontend/.env

```
REACT_APP_CLIENT_ID=
REACT_APP_REDIRECT_URI=
```

/home/ubuntu/Frontend/front.conf

```
server {
  listen 3000;
  server_name j9b205.p.ssafy.io;

  location / {
    root /usr/share/nginx/html;
    index index.html index.htm;
    try_files $uri /index.html;
  }

  error_page 500 502 503 504 /50x.html;

  location = /50x.html {
    root /usr/share/nginx/html;
  }
}
```

/home/ubuntu/deploy_fe.sh

```
# 기존의 소스코드들 삭제
rm -rf Frontend/front
cd Frontend
mkdir front
# 새로운 소스코드파일 복붙
cp ./front_0.1.0.tar ./front/front_0.1.0.tar
cd front
tar -xvf front_0.1.0.tar
rm -rf ../front_0.1.0.tar
rm -rf front_0.1.0.tar
# 설정파일 복붙
cp ../docker-compose.yml ./docker-compose.yml
cp ../Dockerfile ./Dockerfile
cp ../.env ./.env
cp ../front.conf ./front.conf
sudo docker compose down
sudo docker compose up -d --build
```

/home/ubuntu/Frontend/Dockerfile

```
# Node.js image get
FROM node:16 as build

# Create directory for tar file
RUN mkdir -p /Frontend/front

WORKDIR /Frontend/front

COPY ./package.json ./package-lock.json ./

# install dependency for application
RUN npm install --force

COPY . .

# build React application
RUN npm run build

# set Runtime image
FROM nginx:latest

# copy build react app file to nginx static file directory
COPY --from=build /Frontend/front/build /usr/share/nginx/html

# 도커 상의 엔진엑스에 컨피그 파일 입히기
COPY front.conf /etc/nginx/conf.d/default.conf

# RUN React Application
CMD ["nginx", "-g", "daemon off;"]
```