🏢

빌드 및 배포

기술 스택

- IDE
  - Visual Studio Code 1.64.2
  - IntelliJ IDEA Community Edition 2021.3.1
- FrontEnd
  - React 17.0.2
  - HTML5
  - CSS
  - TypeScript 4.6.2
  - JavaScript (ES6)
  - Recoil 0.5.1
  - styled-components 5.3.3
  - framer-motion 4.1.17
  - draft-js 0.11.7
  - axios 0.26.1
  - web3 1.7.1
- Backend
  - Java (Open-JDK 1.8.0_312)
  - Spring Boot 2.4.5
  - Spring Data JPA
  - Hibernate 5.4.30
  - Lombok 1.18.20
  - Spring Boot Gradle 7.4.1
  - Swagger 2 3.0.0
- Blockchain
  - Solidity 0.8.4+
  - Truffle 5.4.24
  - Hardhat: 2.9.1
  - openzeppelin/contracts 4.5.0

- DB
  - MySQL 8.0.26
- 운영체제, 서버
  - Window10
  - Ubuntu 20.04.2 LTS
  - Jenkins 2.3
  - Nginx 1.18.0
  - Docker 20.10.7
  - Certbot 0.40.0
  - AWS EC2 (Ubuntu 20.04 LTS)
  - AWS S3
- 형상 관리
  - GitLab
- 이슈 관리
  - Jira
- 커뮤니케이션
  - Mattermost, Notion, Webex

배포

Nginx 설치

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt install nginx
$ sudo service start nginx
```

Certbot 설치 및 설정

```
$ sudo apt update
$ sudo apt install certbot

$ sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
$ sudo mkdir -p /var/lib/letsencrypt/.well-known
$ sudo chgrp www-data /var/lib/letsencrypt
$ sudo chmod g+s /var/lib/letsencrypt


$ sudo vi /etc/nginx/snippets/letsencrypt.conf
# /etc/nginx/snippets/letsencrypt.conf
location ^~ /.well-known/acme-challenge/ {
  allow all;
  root /var/lib/letsencrypt/;
  default_type "text/plain";
  try_files $uri =404;
}


$ sudo nano /etc/nginx/snippets/ssl.conf
# /etc/nginx/snippets/ssl.conf
ssl_dhparam /etc/ssl/certs/dhparam.pem;

ssl_session_timeout 1d;
ssl_session_cache shared:SSL:50m;
ssl_session_tickets off;

ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers 'ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE
ssl_prefer_server_ciphers on;
```

```
ssl_stapling on;
ssl_stapling_verify on;
resolver 8.8.8.8 8.8.4.4 valid=300s;
resolver_timeout 30s;

add_header Strict-Transport-Security "max-age=15768000; includeSubdomains; preload";
add_header X-Frame-Options SAMEORIGIN;
add_header X-Content-Type-Options nosniff;

$ sudo systemctl restart nginx
```

SSL 인증서 생성

```
$ sudo certbot certonly --agree-tos --email admin@example.com --webroot -w /var/lib/letsencrypt/ -d j6a407.p.ssafy.io

인증서를 성공적으로 얻으면 다음과 같이 출력되는 것을 확인할 수 있다.
IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/example.com/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/example.com/privkey.pem
   Your cert will expire on 2018-07-28. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot
   again. To non-interactively renew *all* of your certificates, run
   "certbot renew"
 - Your account credentials have been saved in your Certbot
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Certbot so
   making regular backups of this folder is ideal.
 - 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
```

Nginx Configuration 설정

```
$ sudo vi /etc/nginx/sites-enabled/default

# Default server configuration
server {

  ...

  root /home/ubuntu/dotore/frontend/build;

  server_name j6a407.p.ssafy.io;

  location /api {
        proxy_pass http://j6a407.p.ssafy.io:8080;
        proxy_http_version 1.1;
        proxy_set_header Connection "";

        include /etc/nginx/proxy_params;
   }
   include snippets/letsencrypt.conf;
   return 301 https://$host$request_uri;

  ...

}

server {
    listen 443 ssl http2;
    server_name j6a407.p.ssafy.io;

    ssl_certificate /etc/letsencrypt/live/j6a407.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/j6a407.p.ssafy.io/privkey.pem;
    ssl_trusted_certificate /etc/letsencrypt/live/j6a407.p.ssafy.io/chain.pem;
    include snippets/ssl.conf;
    include snippets/letsencrypt.conf;

    location / {
        root /home/ubuntu/dotore/frontend/build;
        index index.html index.htm;

        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ /index.html =404;
    }

    location /api {
        proxy_pass http://j6a407.p.ssafy.io:8080;
        proxy_http_version 1.1;
        proxy_set_header Connection "";

        include /etc/nginx/proxy_params;
    }
  }
}
```

Frontend

frontend 디렉토리로 이동 후 빌드

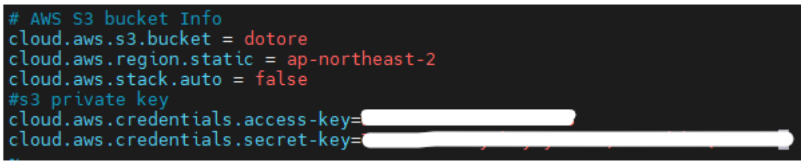
```
$ cd ./dotore/frontend
$ npm i
$ npm build
```

Backend

기존 실행중인 프로세스를 종료하고 backend 디렉토리로 이동 후 빌드

```
$ sudo kill $(sudo lsof -t -i:8080)
$ cd ./dotore/backend
$ sudo chmod +x gradlew
$ ./gradlew build
$ nohup java -jar build/libs/dotore-backend.jar &
$ sudo systemctl reload nginx;
```

backend/src/main/resources/s3.properties 파일 설정



```
# AWS S3 bucket Info
cloud.aws.s3.bucket = dotore
cloud.aws.region.static = ap-northeast-2
cloud.aws.stack.auto = false
#s3 private key
cloud.aws.credentials.access-key=[redacted]
cloud.aws.credentials.secret-key=[redacted]
```