포팅 매뉴얼

개발 환경

Front-End

- React 18.2.0
- Node.js LTS 18.13.0(includes npm 8.19.3)
- Visual Studio code 1.74.3

Back-End

- Intellij
  - Build 223.8214.52, built on December 20, 2022
- Java 11
  - zulu-11 java version “11.0.17”
- Springboot 2.7.6
- MySQL 8.0.31

ETC

- Nginx 1.18.0 (ubuntu)
- AWS EC2 Ubuntu 20.04 LTS
- AWS S3 2.2.6

설정 파일 및 환경 변수 정보

Nginx

- storyge.conf 파일

```
upstream storyge {
        server localhost:8080;
        keepalive 4;
    }
    server {
        location /{
            try_files $uri $uri/ /index.html =404;
        }

        location /api/ {
                proxy_pass http://localhost:8080/;
        }

        location /api/sub {
                proxy_http_version 1.1;
                proxy_pass http://localhost:8080/sub;
                proxy_set_header Connection '';
                proxy_cache off;
                chunked_transfer_encoding off;
                proxy_read_timeout 300;
                proxy_connect_timeout 300;
        }

    listen 443 ssl http2; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/storyge.xyz/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/storyge.xyz/privkey.pem; # managed by Certbot
    root /home/ubuntu/test/build;
    index index.html index.htm;

    http2_max_field_size 64k;
    http2_max_header_size 512k;

    }
    server {
    #    if ($host = storyge.xyz) {
        return 301 https://$host$request_uri;
    #    } # managed by Certbot

        listen 80;
        server_name storyge.xyz;
    return 404; # managed by Certbot
    }
```

React

- .env

```
REACT_APP_OPENAI_API_KEY = OPENAI API 키
REACT_APP_YOUTUBE_API_KEY = YOUTUBE API 키
```

Spring

- application.properties

```
#logging
logging.level.com.amazonaws.util.EC2MetadataUtils=ERROR
logging.level.com.hibernate.SQL=DEBUG
logging.level.com.hibernate.type.descriptor.sql.BasicBinder=TRACE

#db connection
spring.jpa.open-in-view=false
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://[서버도메인]:3306/storyge
spring.datasource.username= 유저네임
spring.datasource.password= 비밀번호

#springfox.documentation.swagger.use-model-v3=false
spring.mvc.pathmatch.matching-strategy=ant_path_matcher

#multifile upload 파일 용량 제한
spring.servlet.multipart.max-file-size=50MB
spring.servlet.multipart.max-request-size=50MB

#소셜로그인 & aws
spring.profiles.include=oauth, aws
```

- application-aws.properties

```
#aws 설정
cloud.aws.s3.bucket=storyge-project
cloud.aws.credentials.access-key= AWS S3 access-key
cloud.aws.credentials.secret-key= AWS S3 secret-key
cloud.aws.region.static=ap-northeast-2
cloud.aws.stack.auto=false
cloud.aws.region.auto=false
```

- appication-oauth.properties

```
#google 소셜로그인 설정
spring.security.oauth2.client.registration.google.client-id=구글 클라이언트 ID
spring.security.oauth2.client.registration.google.client-secret=구글 클라이언트 secret
spring.security.oauth2.client.registration.google.scope=https://www.googleapis.com/auth/userinfo.profile, https://www.googleapis.c
spring.security.oauth2.client.registration.google.redirect-uri=구글 redirect URI

#naver 소셜로그인 설정
spring.security.oauth2.client.registration.naver.client-id=네이버 클라이언트 ID
spring.security.oauth2.client.registration.naver.client-secret=네이버 클라이언트 secret
spring.security.oauth2.client.registration.naver.scope=name, email, profile_image
spring.security.oauth2.client.registration.naver.client-name=Naver
spring.security.oauth2.client.registration.naver.authorization-grant-type=authorization_code
spring.security.oauth2.client.registration.naver.redirect-uri=네이버 redirect URI

##naver provider 설정
spring.security.oauth2.client.provider.naver.authorization-uri=https://nid.naver.com/oauth2.0/authorize
spring.security.oauth2.client.provider.naver.token-uri=https://nid.naver.com/oauth2.0/token
spring.security.oauth2.client.provider.naver.user-info-uri=https://openapi.naver.com/v1/nid/me
spring.security.oauth2.client.provider.naver.user-name-attribute=response

#kakao 소셜로그인 설정
spring.security.oauth2.client.registration.kakao.client-id=카카오 클라이언트 ID
spring.security.oauth2.client.registration.kakao.client-secret=카카오 클라이언트 secret
spring.security.oauth2.client.registration.kakao.scope=profile_nickname, account_email, profile_image
spring.security.oauth2.client.registration.kakao.client-name=kakao
spring.security.oauth2.client.registration.kakao.authorization-grant-type=authorization_code
spring.security.oauth2.client.registration.kakao.redirect-uri=카카오 redirect URI
spring.security.oauth2.client.registration.kakao.client-authentication-method=POST

#kakao provider 설정
spring.security.oauth2.client.provider.kakao.authorization-uri=https://kauth.kakao.com/oauth/authorize
spring.security.oauth2.client.provider.kakao.token-uri=https://kauth.kakao.com/oauth/token
spring.security.oauth2.client.provider.kakao.user-info-uri=https://kapi.kakao.com/v2/user/me
spring.security.oauth2.client.provider.kakao.user-name-attribute=id
```

배포 과정

서버 설정

서버 타임존 설정

KST로 변경

`timedatectl set-timezone Asia/Seoul`

변경된 시간 확인

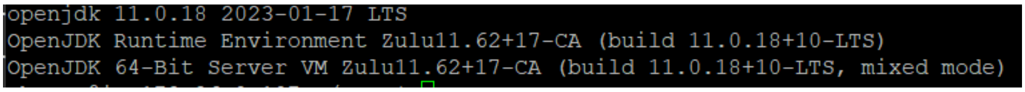
`date`

JDK 설치(zulu-11)

`sudo apt-get install zulu11.0.18-jdk`

버전 확인

`java –version`



```
openjdk 11.0.18 2023-01-17 LTS
OpenJDK Runtime Environment Zulu11.62+17-CA (build 11.0.18+10-LTS)
OpenJDK 64-Bit Server VM Zulu11.62+17-CA (build 11.0.18+10-LTS, mixed mode)
```

Node.js 설치

1. nvm 설치

`curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash`

2. nvm 활성화

`. ~/.nvm/nvm.sh`

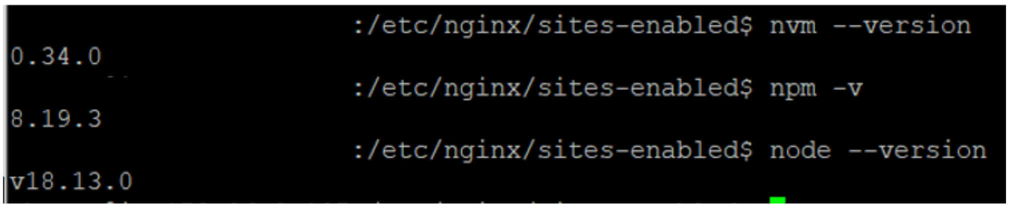
3. node 18.13.0 설치

`nvm install 18.13.0`

4. npm 설치

`sudo apt-get install npm`

5. 버전 확인



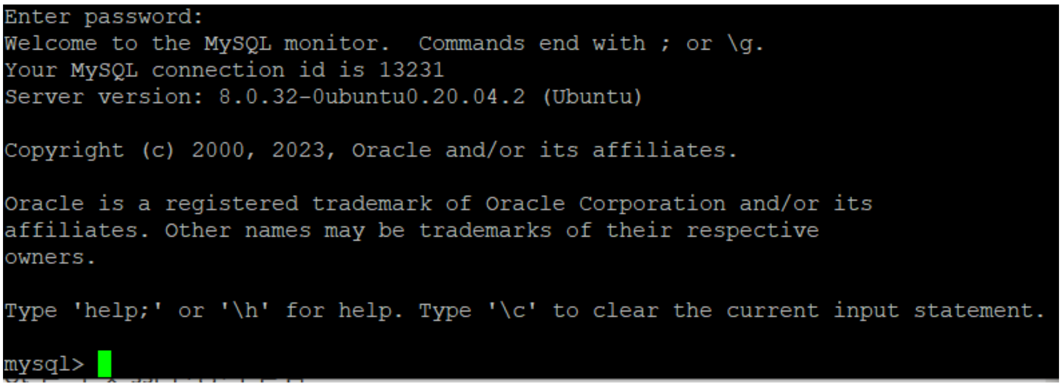
```
                      :/etc/nginx/sites-enabled$ nvm --version
0.34.0
                      :/etc/nginx/sites-enabled$ npm -v
8.19.3
                      :/etc/nginx/sites-enabled$ node --version
v18.13.0
```

mysql 설치

`sudo apt-get install mysql-server-8.0`

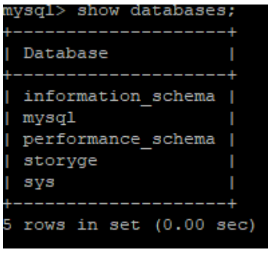
1. storyge.sql 파일을 서버에 업로드 한다.
2. `sudo mysql -u root -p` 명령어로 root 계정으로 접근한다

초기 접근 시에 비밀번호를 설정하고 이후 접근 시에는 설정된 비밀번호를 통해 접근할 수 있다.



```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13231
Server version: 8.0.32-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

3. `source /소스파일 경로/소스파일.sql` 명령어로 덤프파일을 실행시킨다.
4. `show databases;` 명령어로 데이터베이스 목록을 확인한다.



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| storyge            |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

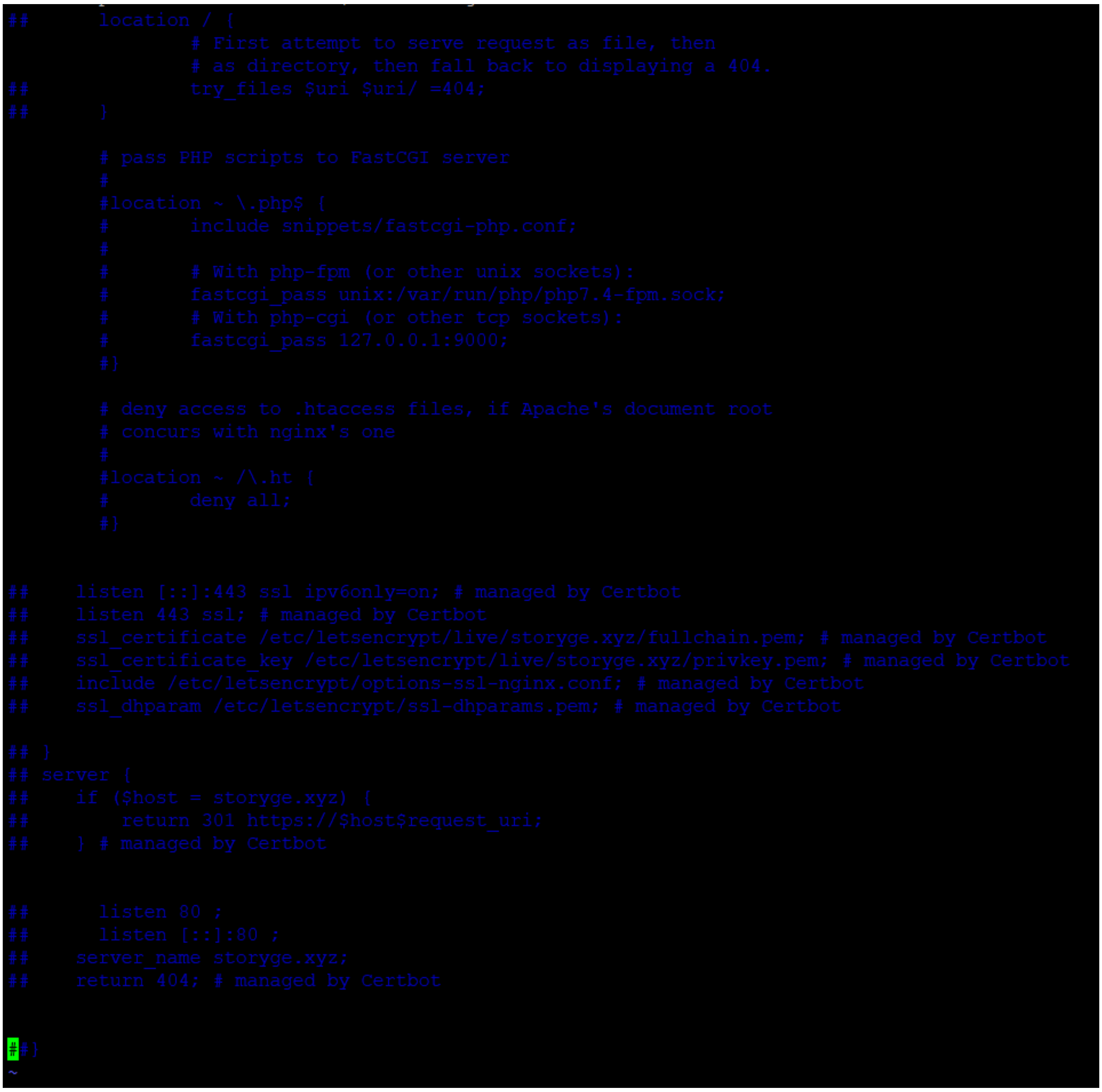
5. `use storyge;` 명령어로 데이터베이스를 선택하고 `show tables;` 로 테이블을 확인할 수 있다.

Nginx 설치

`sudo apt-get install nginx`

nginx의 기본 설치 경로는 /etc/nginx 디렉토리로 설정된다.

1. /etc/nginx/sites-available 디렉토리에 default 파일의 내용을 모두 주석 처리 해주고,



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

        # pass PHP scripts to FastCGI server
        #
        #location ~ \.php$ {
        #       include snippets/fastcgi-php.conf;
        #
        #       # With php-fpm (or other unix sockets):
        #       fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
        #       # With php-cgi (or other tcp sockets):
        #       fastcgi_pass 127.0.0.1:9000;
        #}

        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        #       deny all;
        #}


##    listen [::]:443 ssl ipv6only=on; # managed by Certbot
##    listen 443 ssl; # managed by Certbot
##    ssl_certificate /etc/letsencrypt/live/storyge.xyz/fullchain.pem; # managed by Certbot
##    ssl_certificate_key /etc/letsencrypt/live/storyge.xyz/privkey.pem; # managed by Certbot
##    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
##    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

## }
## server {
##    if ($host = storyge.xyz) {
##        return 301 https://$host$request_uri;
##    } # managed by Certbot


##      listen 80 ;
##      listen [::]:80 ;
##    server_name storyge.xyz;
##    return 404; # managed by Certbot


##}
~
```

2. 위 설정 파일 항목에 있는 **storyge.conf** 파일을 작성한다.

3. sites-enable 디렉토리로 symbolic link 설정을 해준다.

`sudo ln -s /etc/nginx/sites-available/storyge.conf /etc/nginx/sites-enabled/storyge.conf`

4. `sudo service nginx start` 명령어로 nginx를 실행한다

certbot 설치 및 ssl 인증서 발급

1. snapd 설치

`sudo snap install core;`

`sudo snap refresh core`

2. 설치된 certbot이 있다면 삭제

`sudo apt-get remove certbot`

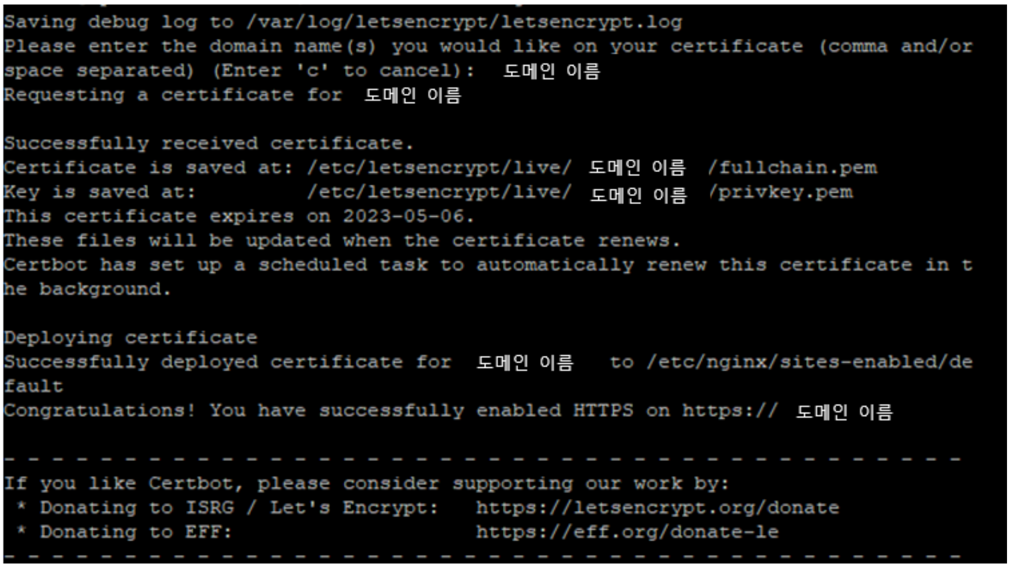
3. Certbot 설치

`sudo snap install --classic certbot`

`sudo ln -s /snap/bin/certbot /usr/bin/certbot`

4. Nginx에서 사용할 SSL 인증서 발급

`sudo certbot --nginx`



```
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Please enter the domain name(s) you would like on your certificate (comma and/or
space separated) (Enter 'c' to cancel):  도메인 이름
Requesting a certificate for  도메인 이름

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/ 도메인 이름  /fullchain.pem
Key is saved at:         /etc/letsencrypt/live/ 도메인 이름  /privkey.pem
This certificate expires on 2023-05-06.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in t
he background.

Deploying certificate
Successfully deployed certificate for  도메인 이름   to /etc/nginx/sites-enabled/de
fault
Congratulations! You have successfully enabled HTTPS on https://  도메인 이름

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
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
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

5. 도메인 이름을 입력하면 SSL 인증서 발급을 할 수 있다.

   경로는

   /etc/letsencrypt/live/도메인 이름/fullchain.pem,

   /etc/letsencrypt/live/도메인 이름/privkey.pem

   이 된다.

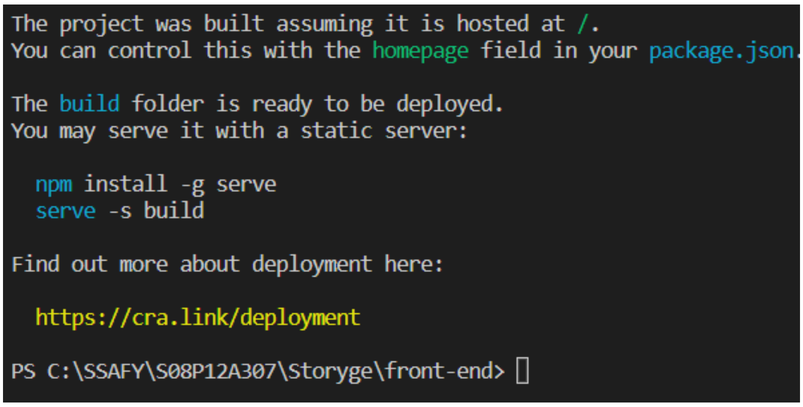
배포

IDE를 이용한 배포

원하는 경로에 git clone을 한다.

Front-End 빌드

1. VSCode에서 S08P12A307/Storyge/front-end 디렉토리 열기
2. 상단의 Terminal → new Terminal을 선택하여 Terminal을 생성한다.
3. `npm i --force` 로 node_modules 생성
4. 위 설정 파일 및 환경변수 항목을 참고하여 .env 파일을 front-end 디렉토리 최상위 경로(/front-end)에 추가한다.
5. `npm run build` 를 입력하여 빌드를 한다.

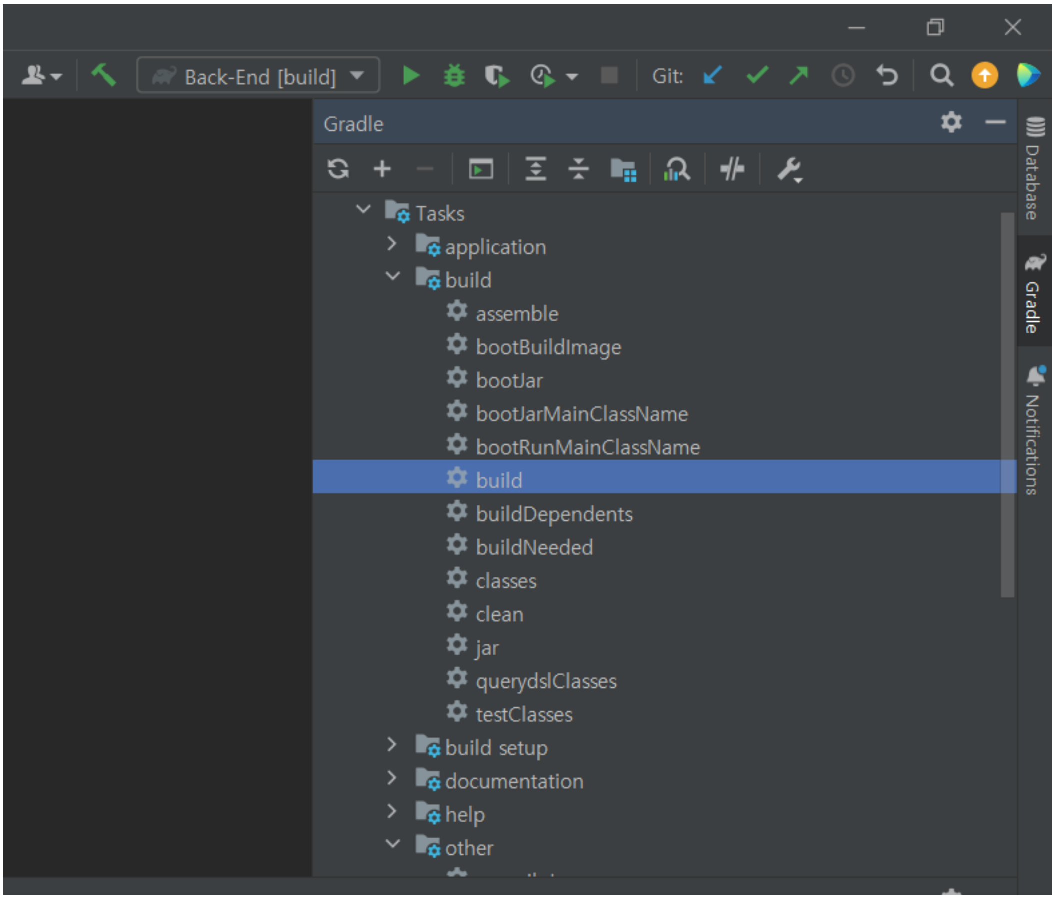


```
The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Find out more about deployment here:

  https://cra.link/deployment

PS C:\SSAFY\S08P12A307\Storyge\front-end> 
```

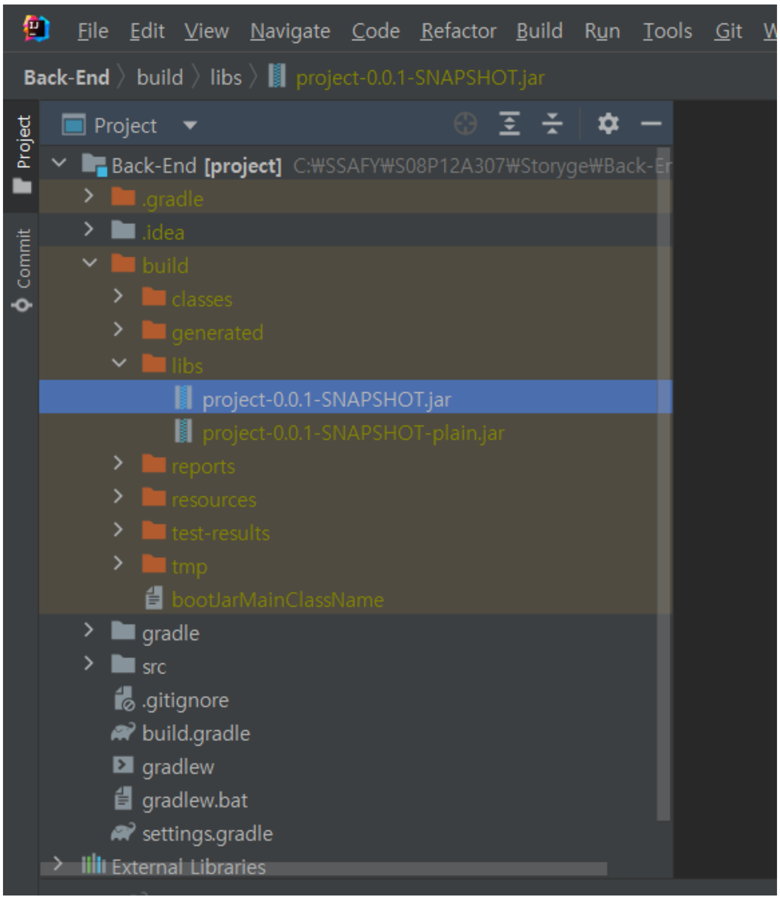
6. /front-end 경로에 생성된 build 디렉토리 확인

Back-End 빌드

1. Intellij에서 S08P12A307/Storyge/Back-End 디렉토리 열기
2. 우측의 Gradle 탭을 선택하여 Tasks → build → build 실행

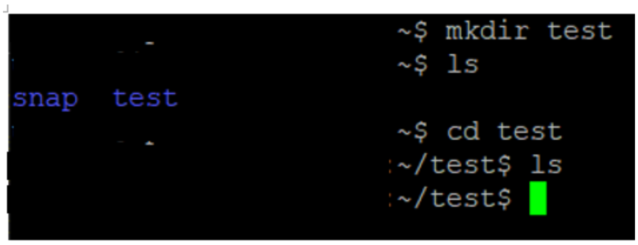


3. /Back-End에 생성된 build/libs/project-0.0.1-SNAPSHOT.jar 파일 확인


File Edit View Navigate Code Refactor Build Run Tools Git W
Back-End 〉 build 〉 libs 〉 project-0.0.1-SNAPSHOT.jar
Project
Commit
Project
Back-End [project] C:₩SSAFY₩S08P12A307₩Storyge₩Back-En
.gradle
.idea
build
classes
generated
libs
project-0.0.1-SNAPSHOT.jar
project-0.0.1-SNAPSHOT-plain.jar
reports
resources
test-results
tmp
bootJarMainClassName
gradle
src
.gitignore
build.gradle
gradlew
gradlew.bat
settings.gradle
External Libraries

배포

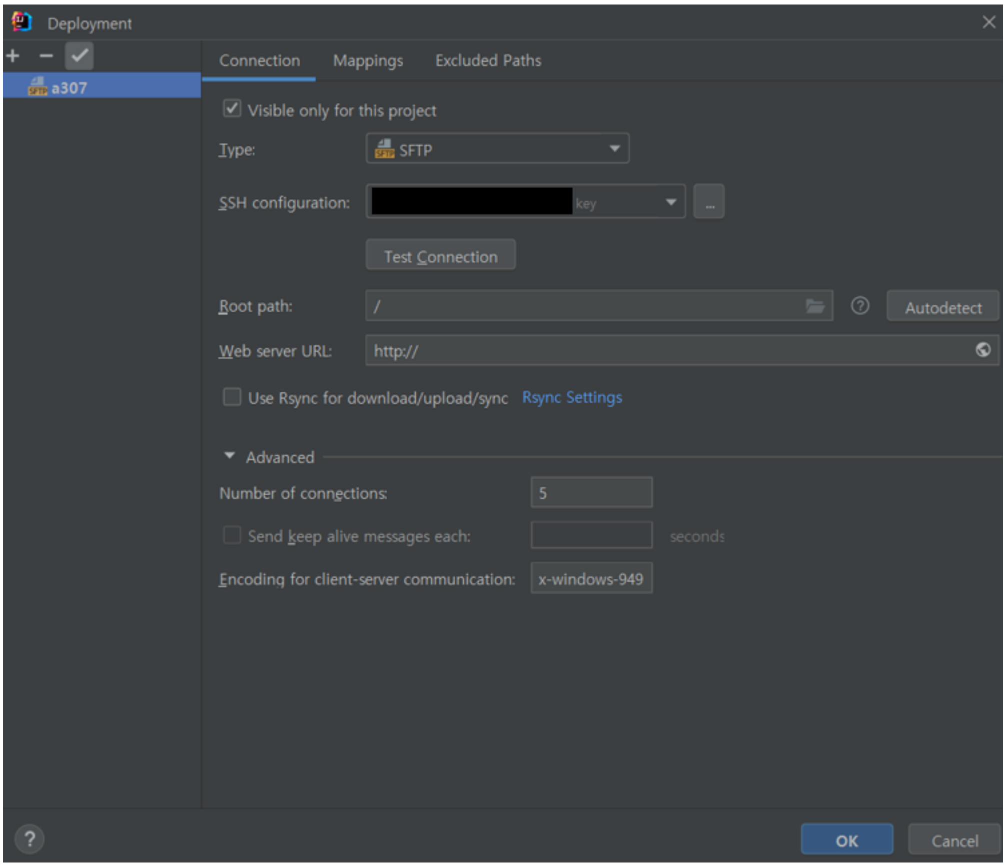
1. 서버 콘솔창에서 `cd /home/ubuntu` 명령어로 해당 경로로 이동
2. `mkdir test` 명령어로 test 디렉토리 생성



```
[redacted]~$ mkdir test
[redacted]~$ ls
snap  test
[redacted]~$ cd test
[redacted]~/test$ ls
[redacted]~/test$
```

3. Intellij에서 S08P12A307/Storyge 디렉토리 열기

4. Remote Host 등록



Deployment

a307

Connection | Mappings | Excluded Paths

☑ Visible only for this project

Type: SFTP

SSH configuration: [redacted] key

Test Connection

Root path: /

Autodetect

Web server URL: http://

□ Use Rsync for download/upload/sync Rsync Settings

Advanced

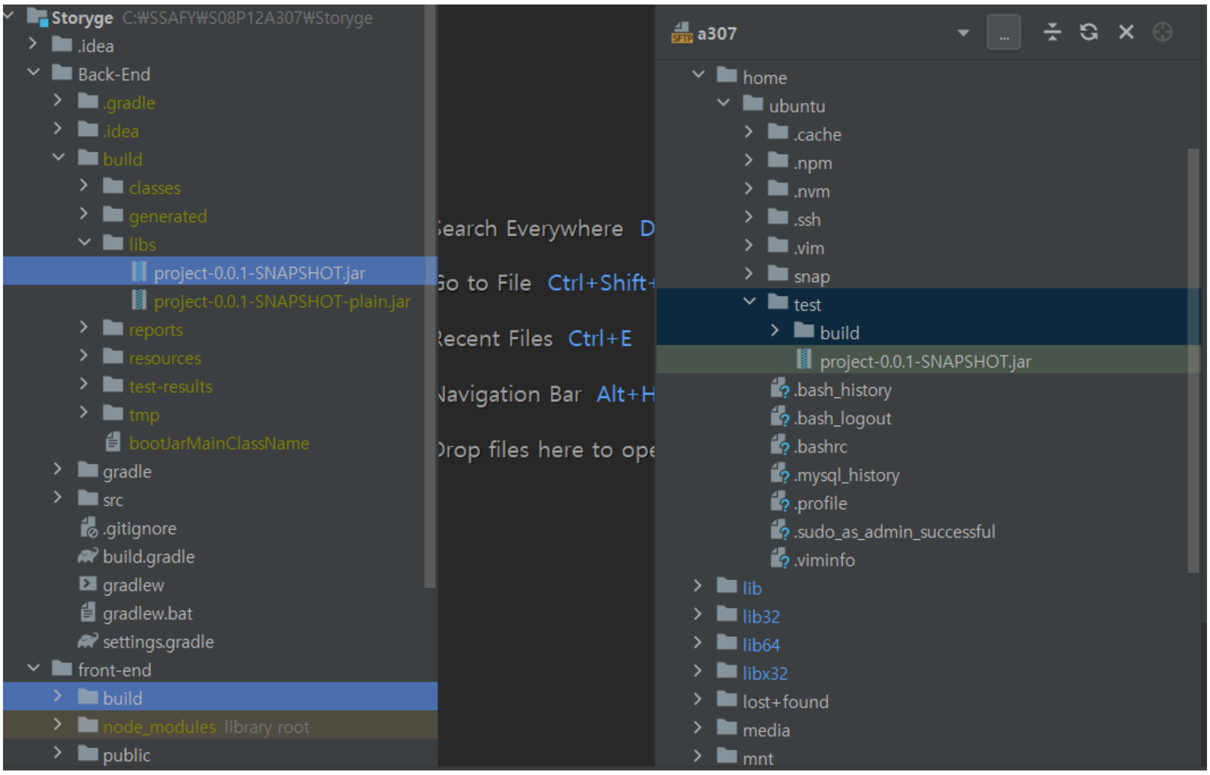
Number of connections: 5

□ Send keep alive messages each: seconds

Encoding for client-server communication: x-windows-949

OK Cancel

5. 빌드된 front-end의 build 디렉토리와 Back-End의 project-0.0.1-SNAPSHOT.jar 파일을 /home/ubuntu/test 경로에 업로드



```
Storyge C:₩SSAFY₩S08P12A307₩Storyge
  .idea
  Back-End
    .gradle
    .idea
    build
      classes
      generated
      libs
        project-0.0.1-SNAPSHOT.jar
        project-0.0.1-SNAPSHOT-plain.jar
      reports
      resources
      test-results
      tmp
      bootJarMainClassName
    gradle
    src
    .gitignore
    build.gradle
    gradlew
    gradlew.bat
    settings.gradle
  front-end
    build
    node_modules library root
    public
```

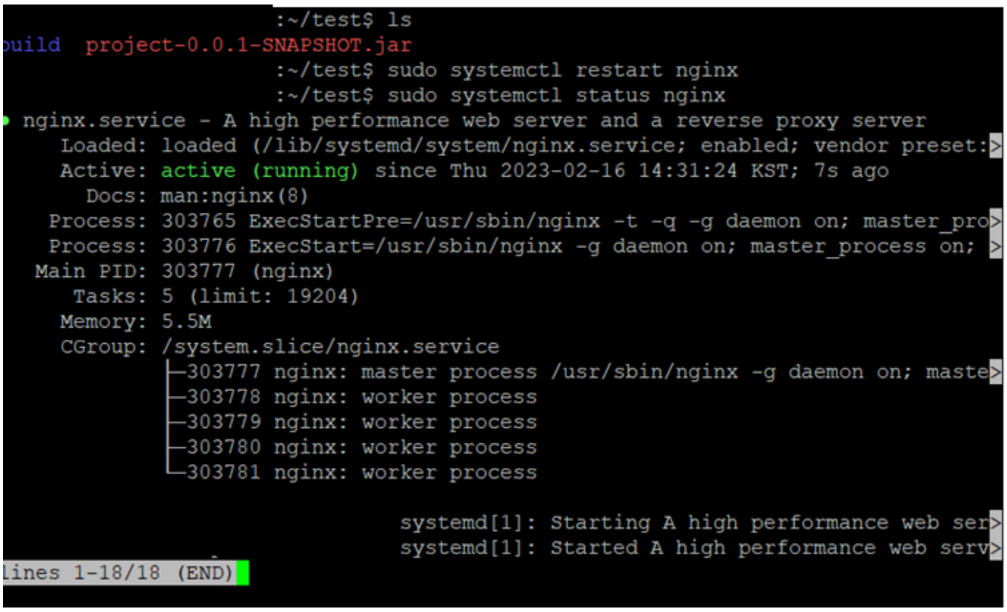
```
a307
  home
    ubuntu
      .cache
      .npm
      .nvm
      .ssh
      .vim
      snap
      test
        build
        project-0.0.1-SNAPSHOT.jar
      .bash_history
      .bash_logout
      .bashrc
      .mysql_history
      .profile
      .sudo_as_admin_successful
      .viminfo
  lib
  lib32
  lib64
  libx32
  lost+found
  media
  mnt
```

6. 서버 콘솔창에서 test 디렉토리로 이동 후 ls 명령어로 디렉토리, 파일 목록 조회

`sudo systemctl restart nginx`

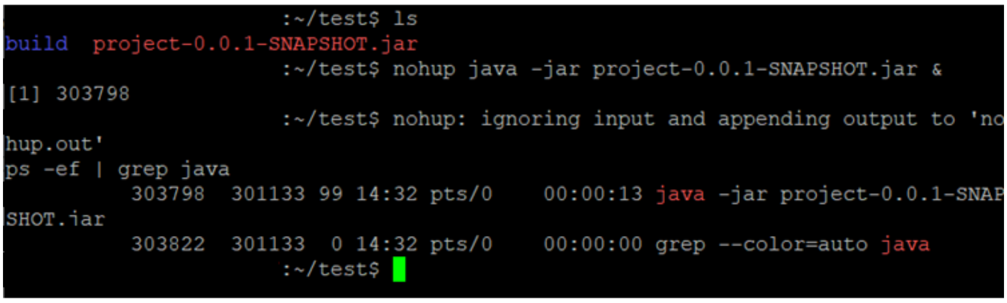
`sudo systemctl status nginx`

명령어로 nginx 재시작 후 상태 확인



```
                      :~/test$ ls
build  project-0.0.1-SNAPSHOT.jar
                      :~/test$ sudo systemctl restart nginx
                      :~/test$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset:>
     Active: active (running) since Thu 2023-02-16 14:31:24 KST; 7s ago
       Docs: man:nginx(8)
    Process: 303765 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_pro>
    Process: 303776 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; >
   Main PID: 303777 (nginx)
      Tasks: 5 (limit: 19204)
     Memory: 5.5M
     CGroup: /system.slice/nginx.service
             ├─303777 nginx: master process /usr/sbin/nginx -g daemon on; maste>
             ├─303778 nginx: worker process
             ├─303779 nginx: worker process
             ├─303780 nginx: worker process
             └─303781 nginx: worker process

                                systemd[1]: Starting A high performance web ser>
                                systemd[1]: Started A high performance web serv>
lines 1-18/18 (END)
```

7. `nohup java -jar .jar파일명 &` 명령어로 백그라운드로 .jar파일 실행함과 동시에 nohup.out 로그 파일 생성

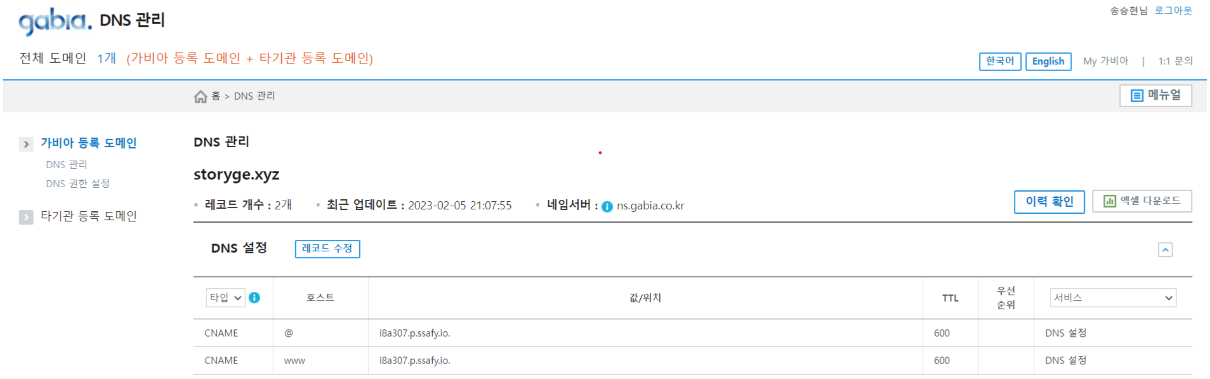


```
                      :~/test$ ls
build  project-0.0.1-SNAPSHOT.jar
                      :~/test$ nohup java -jar project-0.0.1-SNAPSHOT.jar &
[1] 303798
                      :~/test$ nohup: ignoring input and appending output to 'no
hup.out'
ps -ef | grep java
          303798  301133 99 14:32 pts/0    00:00:13 java -jar project-0.0.1-SNAP
SHOT.jar
          303822  301133  0 14:32 pts/0    00:00:00 grep --color=auto java
                      :~/test$
```

8. `ps -ef | grep java` 명령어로 실행중인 java 서비스 확인

외부 서비스 가입 및 활동에 필요한 정보

가비아 도메인



전체 도메인 1개 (가비아 등록 도메인 + 타기관 등록 도메인)

DNS 관리

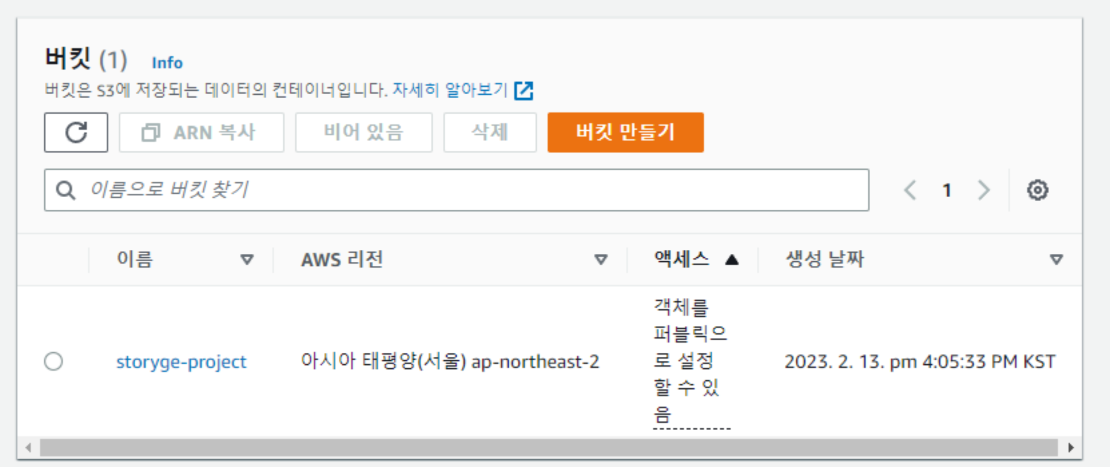
storyge.xyz

- 레코드 개수 : 2개
- 최근 업데이트 : 2023-02-05 21:07:55
- 네임서버 : ns.gabia.co.kr

DNS 설정

타입	호스트	값/위치	TTL	우선 순위	서비스
CNAME	@	i8a307.p.ssafy.io.	600		DNS 설정
CNAME	www	i8a307.p.ssafy.io.	600		DNS 설정

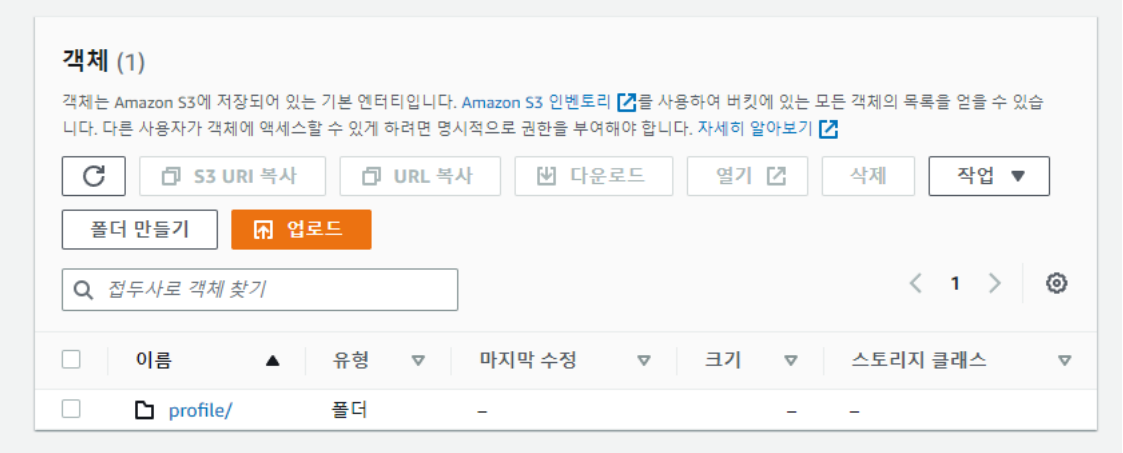
Amazon S3



버킷 (1) Info

버킷은 S3에 저장되는 데이터의 컨테이너입니다. 자세히 알아보기

	이름	AWS 리전	액세스	생성 날짜
○	storyge-project	아시아 태평양(서울) ap-northeast-2	객체를 퍼블릭으로 설정할 수 있음	2023. 2. 13. pm 4:05:33 PM KST



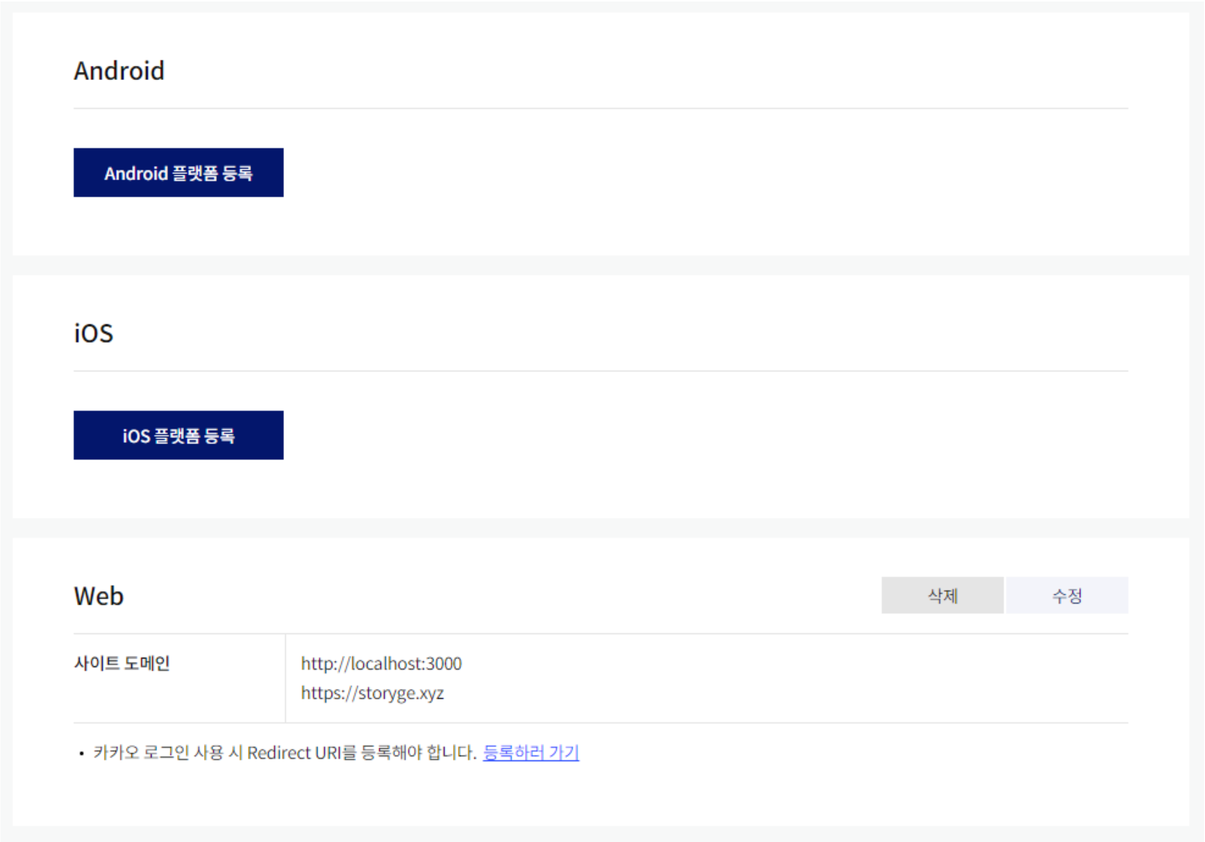
객체 (1)

객체는 Amazon S3에 저장되어 있는 기본 엔터티입니다. Amazon S3 인벤토리를 사용하여 버킷에 있는 모든 객체의 목록을 얻을 수 있습니다. 다른 사용자가 객체에 액세스할 수 있게 하려면 명시적으로 권한을 부여해야 합니다. 자세히 알아보기

□	이름	유형	마지막 수정	크기	스토리지 클래스
□	profile/	폴더	-	-	-

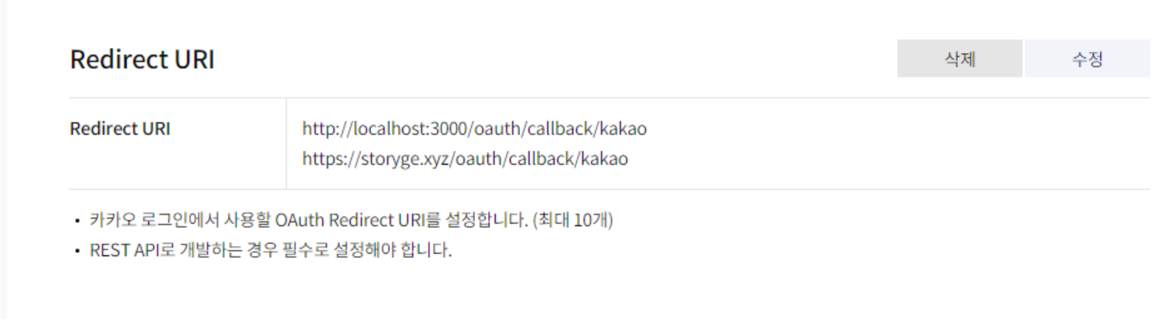
- profile 폴더 안에 사진들이 저장

Kakao login API



사이트 도메인	http://localhost:3000 https://storyge.xyz

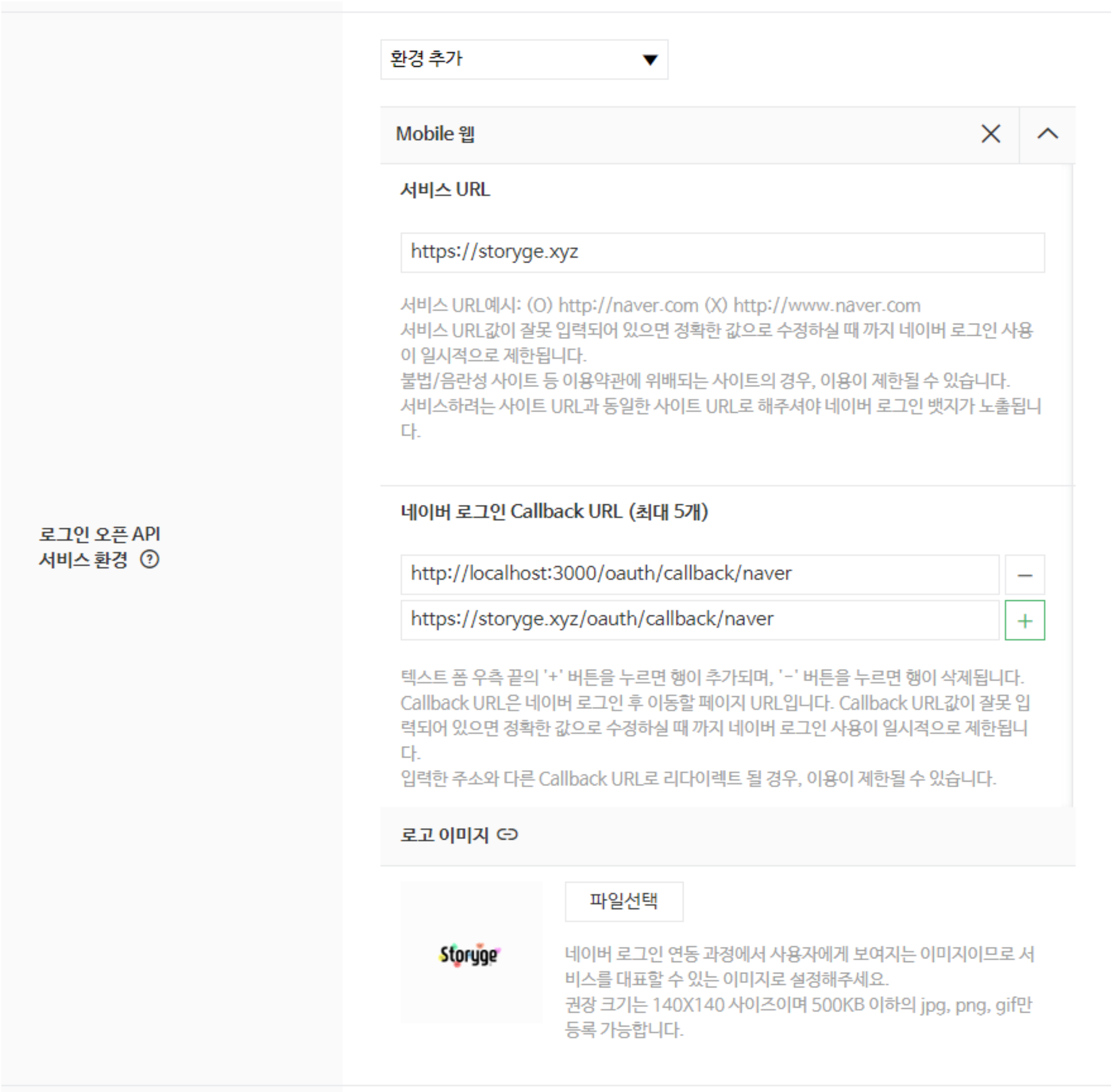
- 카카오 로그인 사용 시 Redirect URI를 등록해야 합니다. 등록하러 가기



Redirect URI	http://localhost:3000/oauth/callback/kakao https://storyge.xyz/oauth/callback/kakao

- 카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다. (최대 10개)
- REST API로 개발하는 경우 필수로 설정해야 합니다.

Naver login API



로그인 오픈 API 서비스 환경 ⓘ

환경 추가 ▼

Mobile 웹 ✕ ︿

서비스 URL

https://storyge.xyz

서비스 URL예시: (O) http://naver.com (X) http://www.naver.com
서비스 URL값이 잘못 입력되어 있으면 정확한 값으로 수정하실 때 까지 네이버 로그인 사용이 일시적으로 제한됩니다.
불법/음란성 사이트 등 이용약관에 위배되는 사이트의 경우, 이용이 제한될 수 있습니다.
서비스하려는 사이트 URL과 동일한 사이트 URL로 해주셔야 네이버 로그인 뱃지가 노출됩니다.

네이버 로그인 Callback URL (최대 5개)

http://localhost:3000/oauth/callback/naver —
https://storyge.xyz/oauth/callback/naver +

텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다.
Callback URL은 네이버 로그인 후 이동할 페이지 URL입니다. Callback URL값이 잘못 입력되어 있으면 정확한 값으로 수정하실 때 까지 네이버 로그인 사용이 일시적으로 제한됩니다.
입력한 주소와 다른 Callback URL로 리다이렉트 될 경우, 이용이 제한될 수 있습니다.

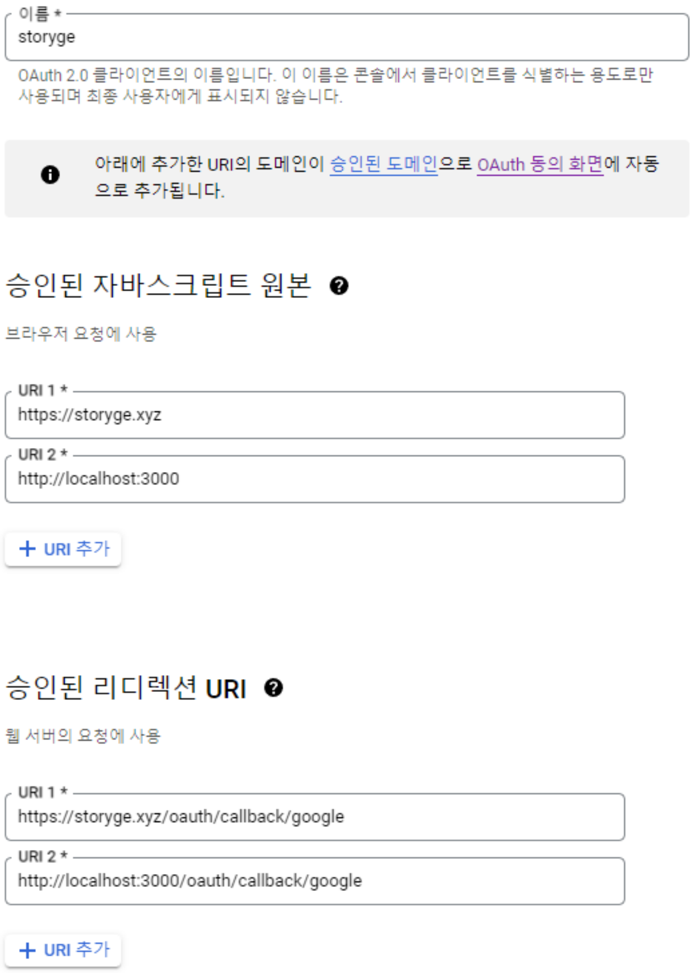
로고 이미지 ⊂⊃

Storyge

파일선택

네이버 로그인 연동 과정에서 사용자에게 보여지는 이미지이므로 서비스를 대표할 수 있는 이미지로 설정해주세요.
권장 크기는 140X140 사이즈이며 500KB 이하의 jpg, png, gif만 등록 가능합니다.

Google login API



storyge

OAuth 2.0 클라이언트의 이름입니다. 이 이름은 콘솔에서 클라이언트를 식별하는 용도로만 사용되며 최종 사용자에게 표시되지 않습니다.

아래에 추가한 URI의 도메인이 승인된 도메인으로 OAuth 동의 화면에 자동으로 추가됩니다.

승인된 자바스크립트 원본

브라우저 요청에 사용

URI 1 *: https://storyge.xyz

URI 2 *: http://localhost:3000

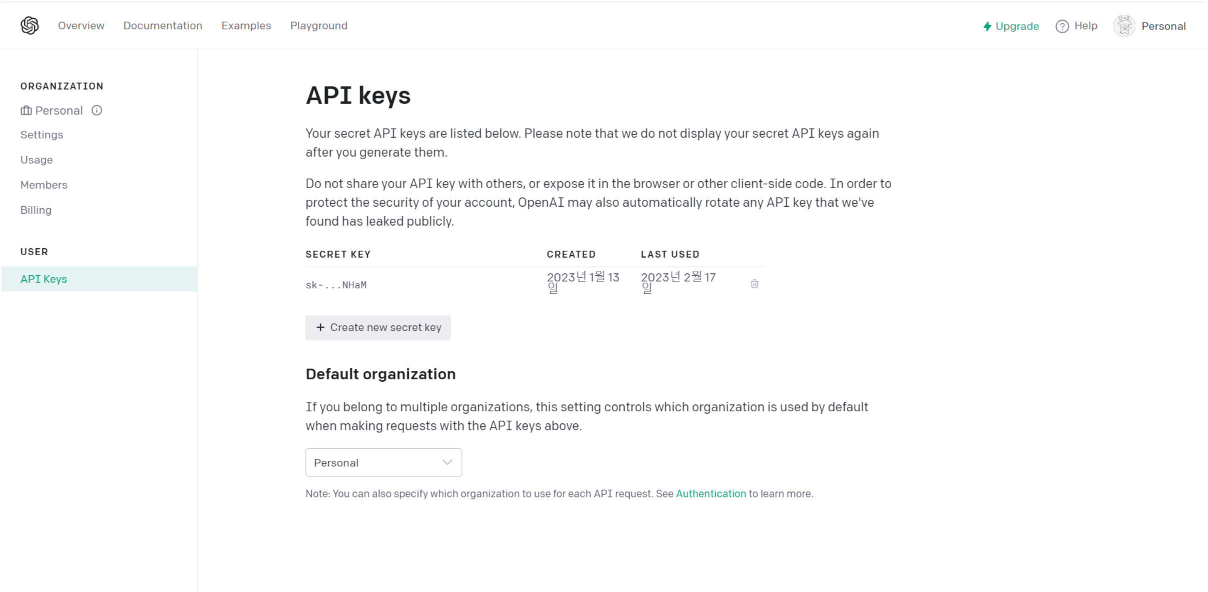
승인된 리디렉션 URI

웹 서버의 요청에 사용

URI 1 *: https://storyge.xyz/oauth/callback/google

URI 2 *: http://localhost:3000/oauth/callback/google

OpenAI API



API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

SECRET KEY	CREATED	LAST USED
sk-...NHaM	2023년 1월 13일	2023년 2월 17일

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

Personal

Note: You can also specify which organization to use for each API request. See Authentication to learn more.