

포팅 메뉴얼

[사용 프로그램 버전](#)

[배포 환경 설정](#)

[서버 구성](#)

[서버1](#)

[서버2](#)

[서버3](#)

[서버4](#)

[서버5](#)

[Common Server](#)

[Chat Server](#)

[Judge Server](#)

[Discovery Server](#)

[Gateway Server](#)

[Open Vidu Server](#)

[Open Vidu 설치 메뉴얼](#)

[Front Server](#)

[배포 시 특이사항](#)

[외부 서비스 정보](#)

1. Gitlab 소스 클론 이후 빌드 및 배포할 수 있도록 정리한 문서

- 1) 사용한 JVM, 웹서버, WAS 제품 등의 종류와 설정 값, 버전(IDE버전 포함) 기재
- 2) 빌드 시 사용되는 환경 변수 등의 내용 상세 기재
- 3) 배포 시 특이사항 기재
- 4) DB 접속 정보 등 프로젝트(ERD)에 활용되는 주요 계정 및 프로퍼티가 정의된 파일 목록

2. 프로젝트에서 사용하는 외부 서비스 정보를 정리한 문서

: 소셜 인증, 포톤 클라우드, 코드 컴파일 등에 활용된 '외부 서비스'가입 및 활용에 필요한 정보

3. DB 덤프 파일 최신본

4. 시연 시나리오

: 시연 순서에 따른 site 화면별, 실행별(클릭 위치 등) 상세 설명

사용 프로그램 버전

- JVM : 17

- Spring Boot : 3.2.1
- gradle : 8.5
- IntelliJ IDEA Ultimate : 2023.3.2
- Jenkins : 2.426.2
- Visual Studio Code : **1.86**
- React : 18.2.0
- npm: 10.4.0
- node: 18.17.1

배포 환경 설정



docker run으로 설명이 되어있는 경우, 프로젝트 안에 Docker 파일이 포함되어 있습니다.

서버 구성

서버1

- Front Server 항목의 도커 파일을 실행합니다. 자세한 명령어와 환경변수는 밑의 항목을 참고하시기 바랍니다.
- 도메인과 HTTPS 적용이 되어있습니다.
 - 클라우드 플레이어에서 직접 연결하여 사용하기 때문에 nginx에서는 80포트로 포워딩 되어있습니다.
- NGINX 설정은 도커 파일에 같이 있습니다.

서버2

- GatewayServer, DiscoveryServer 항목의 도커파일을 실행합니다. 자세한 명령어와 환경변수는 밑의 항목을 참고하시기 바랍니다.
- 도메인과 HTTPS 적용이 되어있습니다.

- 클라우드 플레어에서 직접 연결하여 사용하기 때문에 nginx에서는 80포트로 포워딩 되어있습니다.
- nginx 설정은 다음과 같습니다.

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http{
    server {
        listen 80;
        listen [::]:80;
        server_name 3.39.190.22;

        location / {
            proxy_pass http://127.0.0.1:8888;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
        }
    }
}
```

서버3

- Backend/chatting 디렉토리내 도커 컴포즈 파일을 실행합니다

```
docker-compose up -d
```

- CommonServer 항목의 도커파일을 실행합니다.
- 도메인과 HTTPS 적용이 되어있습니다.
 - 클라우드 플레어에서 직접 연결하여 사용하기 때문에 nginx에서는 80포트로 포워딩 되어있습니다.
- nginx 설정은 다음과 같습니다.

```

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dyn
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format main '$remote_addr - $remote_user [$time_
                    '$status $body_bytes_sent "$http_ref
                    '"$http_user_agent" "$http_x_forward

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

    sendfile            on;
    tcp_nopush          on;
    keepalive_timeout   65;
    types_hash_max_size 4096;

    include              /etc/nginx/mime.types;
    default_type         application/octet-stream;
    # Load modular configuration files from the /etc/nginx
    # See http://nginx.org/en/docs/nginx_core_module.html#in
    # for more information.
    include /etc/nginx/conf.d/*.conf;
    server {

```

```

listen 80;
server_name example.com;

location / {
    proxy_pass http://localhost:8888;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "Upgrade";
}

location /ws {
    proxy_pass http://localhost:8888/ws;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "Upgrade";
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_for;
}
}
}

```

서버4

- CommonServer 항목의 도커파일을 실행합니다. 자세한 명령어와 환경변수는 밑의 항목을 참고하시기 바랍니다.
- 엔진엑스 설정은 다음과 같습니다.

```

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

```

```

# Load dynamic modules. See /usr/share/doc/nginx/README.d
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format main '$remote_addr - $remote_user [$time_
                    '$status $body_bytes_sent "$http_ref
                    '"$http_user_agent" "$http_x_forward

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

    sendfile            on;
    tcp_nopush          on;
    keepalive_timeout   65;
    types_hash_max_size 4096;

    include              /etc/nginx/mime.types;
    default_type         application/octet-stream;
    # Load modular configuration files from the /etc/nginx
    # See http://nginx.org/en/docs/nginx_core_module.html#in
    # for more information.
    include /etc/nginx/conf.d/*.conf;
    server {
        listen 80;
        server_name example.com;

        location / {
            proxy_pass http://localhost:8080;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_fo
            proxy_set_header X-Forwarded-Proto $scheme;
        }
        location /log {
            proxy_pass http://localhost:9000;

```

```

        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

#mail {
#    # See sample authentication script at:
#    # http://wiki.nginx.org/ImapAuthenticateWithApache
#
#    # auth_http localhost/auth.php;
#    # pop3_capabilities "TOP" "USER";
#    # imap_capabilities "IMAP4rev1" "UIDPLUS";
#
#    server {
#        listen     localhost:110;
#        protocol   pop3;
#        proxy      on;
#    }
#
#    server {
#        listen     localhost:143;
#        protocol   imap;
#        proxy      on;
#    }
#}

```

서버5

- 실제 환경에서는 AWS lambda로 이관되어있습니다.
- 로컬에서 실행할 경우 /BackEnd/judge 디렉토리 내 도커 파일을 실행하시기 바랍니다.
- 실행 명령어는 아래 항목을 참고하시기 바랍니다.

Common Server

- Port

- NginX
- Spring Boot

```
sudo docker run -d -p 8080:8080 --name Common -e DB_URL="j
```

Chat Server

- Spring Boot

```
sudo docker run -d -p 8888:8888 --name chatting -e CHATTI
```

- 실제 배포환경에서는 환경변수 하드코딩이 아닌 젠킨스 스크립트와 젠킨스 내부 환경변수를 이용하여 진행합니다.
- 실행의 용이함을 위해 환경변수를 직접 입력하여 드립니다.

Judge Server

- Spring Boot

```
sudo docker run -d -p 8080:8080 --name judge -e DISCOVERY_
```

- 실제 배포환경에서는 환경변수 하드코딩이 아닌 젠킨스 스크립트와 젠킨스 내부 환경변수를 이용하여 진행합니다.
- 실행의 용이함을 위해 환경변수를 직접 입력하여 드립니다.

Discovery Server

- Spring Boot

```
sudo docker run -d -p 8761:8761 --name Discovery_Server --
```

- 실제 배포환경에서는 환경변수 하드코딩이 아닌 젠킨스 스크립트와 젠킨스 내부 환경변수를 이용하여 진행합니다.
- 실행의 용이함을 위해 환경변수를 직접 입력하여 드립니다.

Gateway Server

- Spring Boot

```
sudo docker run -d -p 8888:8888 --name Gateway --network f
```

- 실제 배포환경에서는 환경변수 하드코딩이 아닌 젠킨스 스크립트와 젠킨스 내부 환경변수를 이용하여 진행합니다.
- 실행의 용이함을 위해 환경변수를 직접 입력하여 드립니다.

Open Vidu Server

Open Vidu 설치 메뉴얼

1. 도커 및 도커 컴포즈 설치
2. 도메인으로 SSL 발급
3. 해당 포트 오픈

```
22 TCP: to connect using SSH to admin OpenVidu.  
80 TCP: if you select Let's Encrypt to generate an SSL certifi  
443 TCP: OpenVidu server and application are published by def  
3478 TCP+UDP: used by STUN/TURN server to resolve clients IPS  
40000 - 57000 TCP+UDP: used by Kurento Media Server to establ  
57001 - 65535 TCP+UDP: used by TURN server to establish relay  
5442, 5443, 6379, 8888
```

4. OpenVidu 설치

```
sudo su  
  
cd /opt  
  
curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/insta
```

5. OpenVidu 설정

```
cd openvidu

nano .env

# env 파일
DOMAIN_OR_PUBLIC_IP=도메인 혹은 Public IP 입력

OPENVIDU_SECRET=비밀번호입력

CERTIFICATE_TYPE=letsencrypt

LETSENCRYPT_EMAIL=이메일입력
```

6. OpenVidu 실행

```
./openvidu start
```

Front Server

```
sudo docker run -d -p 80:80 --name ReactApp 1w2k/code-odysse
```

배포 시 특이사항

- 해당 프로젝트는 각 서버가 같은 인스턴스가 아닌 서로 다른 인스턴스에 배포되어 진행되었습니다.

외부 서비스 정보



외부 서비스를 사용하기 위한 키들은 환경설정란에 포함되어있습니다

- 백엔드 서버에서 각 서비스를 사용하기 위한 환경변수는 도커 실행명령어에 포함되어 있습니다.
- 소셜 로그인
 - 카카오
 - 네이버
 - 구글
- Papago API
- Google Cloud Storage

```
.env
//FireBase, Youtube API 공용키
REACT_APP_API_KEY=AIzaSyC8iwm-6CPh9SqaPycyGwZuIBuBfI2g0_Y
REACT_APP_AUTH_DOMAIN=react-code-odyssey.firebaseio.com
REACT_APP_PROJECT_ID=react-code-odyssey
REACT_APP_STORAGE_BUCKET=react-code-odyssey.appspot.com
REACT_APP_MESSAGING_SENDER_ID=1073177536985
REACT_APP_APP_ID=1:1073177536985:web:86f66826d85fb04f4cd1c8
```