포팅 매뉴얼

I. 사용 프로그램 버전

1) 프로젝트 사용 도구

- 이슈 관리: Jira
- 형상 관리: Gitlab
- 커뮤니케이션: Notion, Mattermost
- 디자인: Figma

- UCC: Movavi video editor plus 2022
- CI/CD: Jenkins
- ERD: ERD Cloud
- Architecture diagram: Cloudcraft

2) 개발 도구

- Visual Studio Code: 1.85.1
- Intellij Ultimate: 2023.3.2
- WSL2
- Ubuntu 22.0.4
- Jira Webhook
- Terminus
- MySQL workbench
- MongoDB compass

3) 개발 환경

Frontend

Node.js	`21.6.1` `(includes npm 10.2.4)`
React	`18.2.0`

▼ package.json

```
{
  "name": "prolog",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warnings 0",
    "preview": "vite preview"
  },
  "dependencies": {
    "@ark-ui/react": "^2.0.1",
    "@chakra-ui/react": "^2.8.2",
    "@emotion/react": "^11.11.3",
    "@emotion/styled": "^11.11.0",
    "@reduxjs/toolkit": "^2.1.0",
    "@tomickigrzegorz/react-circular-progress-bar": "^1.1.2",
    "antd": "^5.13.3",
    "axios": "^1.6.7",
    "bootstrap": "^5.3.2",
    "classnames": "^2.5.1",
    "cors": "^2.8.5",
    "date-fns": "^3.3.1",
    "express": "^4.18.2",
    "framer-motion": "^11.0.3",
    "mysql": "^2.18.1",
    "node-sass": "^9.0.0",
    "react": "^18.2.0",
    "react-beautiful-dnd": "^13.1.1",
    "react-bootstrap": "^2.10.0",
    "react-datepicker": "^6.1.0",
    "react-dom": "^18.2.0",
    "react-icons": "^5.0.1",
    "react-modal": "^3.16.1",
    "react-redux": "^9.1.0",
    "react-router-dom": "^6.22.0",
    "recoil": "^0.7.7",
    "styled-components": "^6.1.8",
    "react-scripts": "4.0.3",
    "openvidu-browser": "2.29.0"
  },
  "devDependencies": {
    "@types/react": "^18.2.43",
    "@types/react-dom": "^18.2.17",
    "@vitejs/plugin-react-swc": "^3.5.0",
    "eslint": "^8.55.0",
    "eslint-plugin-react": "^7.33.2",
    "eslint-plugin-react-hooks": "^4.6.0",
    "eslint-plugin-react-refresh": "^0.4.5",
    "sass": "^1.70.0",
    "vite": "^5.0.8"
  }
}
```

Backend

Java	openJDK 17.0.9" 2023-10-17 LTS
SpringBoot	`3.2.2`
JPA + Hibernate	
Gradle	`8.5`

▼ build.gradle

```
plugins {
    id 'java'
    id 'org.springframework.boot' version '3.2.2'
    id 'io.spring.dependency-management' version '1.1.4'
}

group = 'com.b112'
version = '0.0.1-SNAPSHOT'

java {
    sourceCompatibility = '17'
}

configurations {
    compileOnly {
        extendsFrom annotationProcessor
    }
}

repositories {
    mavenCentral()
}

dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    implementation 'org.springframework.boot:spring-boot-starter-data-mongodb'
    implementation 'org.springframework.boot:spring-boot-starter-data-redis'
    implementation 'org.springframework.boot:spring-boot-starter-oauth2-authorization-server'
    implementation 'org.springframework.boot:spring-boot-starter-oauth2-client'
    implementation 'org.springframework.boot:spring-boot-starter-oauth2-resource-server'
    implementation 'org.springframework.boot:spring-boot-starter-security'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    implementation 'org.springframework.boot:spring-boot-starter-web-services'
    implementation 'org.springframework.boot:spring-boot-starter-websocket'
    implementation 'com.googlecode.json-simple:json-simple:1.1'


    implementation 'io.jsonwebtoken:jjwt-api:0.11.5'
    runtimeOnly 'io.jsonwebtoken:jjwt-impl:0.11.5'
    runtimeOnly 'io.jsonwebtoken:jjwt-jackson:0.11.5'

    compileOnly 'org.projectlombok:lombok'
    runtimeOnly 'com.mysql:mysql-connector-j'
    annotationProcessor 'org.projectlombok:lombok'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
    //testImplementation 'org.springframework.security:spring-security-test'
}

tasks.named('test') {
    useJUnitPlatform()
}
```

DB

MySQL	`8.0.35`
MongoDB	`7.0.5`
Redis	`7.2.4`

Server

AWS EC2(xlarge)	CPU : 4vCPUs RAM : 16GB SSD : 320GB S OS: Ubuntu 20.0.4

Service

docker	`25.0.1`
docker-compose	`2.24.5`
NginX	Openvidu 내장 nginx 사용
Jenkins	`2.426.3`
Openvidu	`2.29.0`

4) 외부 서비스

- Kakao OAuth

  Kakao Developers

  카카오 API를 활용하여 다양한 어플리케이션을 개발해보세요. 카카오 로그인, 메시지 보내기, 친구 API, 인공지능 API 등을 제공합니다.

  https://developers.kakao.com/docs/latest/ko/kakaologin/rest-api

- Naver OAuth

  네이버 로그인 - INTRO

  네이버 로그인 NAVER Developers - 네이버 로그인 소개 별도의 아이디, 비밀번호없이 네이버 아이디로 간편하게 외부 서비스에 로그인 할 수 있도록 하는 서비스입니다. 이용자는 복잡하고 번거로운 회원 가입 절차 없이 편하게 서비스를 이용하고, 사업자는 회

  https://developers.naver.com/products/login/api/api.md

- 사람인 API

  Saramin API

  https://oapi.saramin.co.kr/

- MongoDB Atlas Database
  - 무료버전으로 DB 생성

II. 빌드

1) Git ignore된 핵심 파일들

- Backend - Springboot

  ▼ application.properties (/var/lib/docker/volumes/jenkins/_data/workspace/.properties/application.properties 에 위치)

```
# Host
host.server.base-url=https://i10b112.p.ssafy.io/api
host.server.domain=i10b112.p.ssafy.io
host.server.name=i10b112.p.ssafy.io
host.client.base-url=https://i10b112.p.ssafy.io
host.client.domain=i10b112.p.ssafy.io
host.client.name=i10b112.p.ssafy.io
host.serverwebrtc.base-url=https://i10b112.p.ssafy.io/webrtc
host.serverwebrtc.domain=i10b112.p.ssafy.io
host.serverwebrtc.name=i10b112.p.ssafy.io
# Spring MVC Configuration
spring.mvc.pathmatch.matching-strategy=ant_path_matcher
# Database (MySQL) Configurations
spring.jpa.database=mysql
spring.jpa.hibernate.ddl-auto=update
spring.jpa.generate-ddl=true
spring.jpa.show-sql=true
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
spring.jackson.serialization.fail-on-empty-beans=false
# Database Access Information
spring.datasource.url=jdbc:mysql://${host.server.name}:3306/prolog?characterEncoding=UTF-8
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.username="username"
spring.datasource.password="user password"
# Database (MySQL) Configurations
#spring.data.mongodb.uri=mongodb://${host.name}:27017/"projectname"
spring.data.mongodb.uri=mongodb+srv://"username":DzbAX6v72t32VS3E@"projectname".gtfn6wx.mongodb.net/"DBname"?retryWrites=true&w=majority
spring.main.allow-bean-definition-overriding=true
# Springdoc Swagger Configuration
springdoc.api-docs.path=/api-docs
springdoc.swagger-ui.disable-swagger-default-url=true
springdoc.swagger-ui.oauth2-redirect-url=/login/oauth2/code/{registrationId}
springdoc.packages-to-scan=com.b112.prolog
# Kakao OAuth2
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
spring.security.oauth2.client.provider.kakao.user-info-authentication-method=header
spring.security.oauth2.client.registration.kakao.client-id="발급받은 id"
spring.security.oauth2.client.registration.kakao.client-secret="발급받은 secret"
spring.security.oauth2.client.registration.kakao.client-authentication-method=client_secret_post
spring.security.oauth2.client.registration.kakao.redirect-uri=${host.server.base-url}/login/oauth2/code/kakao
spring.security.oauth2.client.registration.kakao.authorization-grant-type=authorization_code
spring.security.oauth2.client.registration.kakao.client-name=Kakao
spring.security.oauth2.client.registration.kakao.scope=profile_nickname, profile_image, account_email
# Naver OAuth2
spring.security.oauth2.client.provider.naver.authorization-uri=https://nid.naver.com/oauth2.0/authorize
spring.security.oauth2.client.provider.naver.token-uri=https://nid.naver.com/oauth2.0/token
spring.security.oauth2.client.provider.naver.user-info-uri=https://openapi.naver.com/v1/nid/me
spring.security.oauth2.client.provider.naver.user-name-attribute=response
spring.security.oauth2.client.provider.naver.user-info-authentication-method=header
spring.security.oauth2.client.registration.naver.client-id=WVyPet7jEUtFmMxGmHNN
spring.security.oauth2.client.registration.naver.client-secret=c3rCdF0PXO
spring.security.oauth2.client.registration.naver.client-authentication-method=client_secret_post
spring.security.oauth2.client.registration.naver.redirect-uri=${host.server.base-url}/login/oauth2/code/naver
spring.security.oauth2.client.registration.naver.authorization-grant-type=authorization_code
spring.security.oauth2.client.registration.naver.client-name=Naver
spring.security.oauth2.client.registration.naver.scope=name, email, profile_image
# JWT
jwt.secret=iDsS0vXHW9QhrgeCfPtZkkdzQ195yln5j7VrJOEIlfXsjDg54pHx5mrXyJTXqwodfijw2452845ualskdfzDFkj2d8fLdbf23nsmdnflk8difasdfkj23kasd9f8SDfksjdflwkerwlkLSfkjsldkfjsd
# Https
#server.ssl.enabled=true
#server.ssl.key-store=key.pem
# spring.data.redis.host=i10b112.p.ssafy.io
spring.data.redis.host=52.78.227.239
spring.data.redis.port=6379
# Saramin API
SARAMIN_API_ACCESS_KEY= "사람인에서 발급받은 Access Key"
SARAMIN_PARAM= ?access-key=${SARAMIN_API_ACCESS_KEY}&sr=directhire&job_type=1&edu_lv=&fields=posting-date+expiration-date+keyword-code&count=110&job_mid_cd=2&ind_cd=314+313+308+307+306+305+304+302+301&bbs_gb=1&keywords=&start=
SARMAIN_URL= https://oapi.saramin.co.kr/job-search${SARAMIN_PARAM}
```

- Webrtc - Openvidu
  - ▼ .env 파일 ($/opt/openvidu)

```
# OpenVidu configuration
# ----------------------
# Documentation: https://docs.openvidu.io/en/stable/reference-docs/openvidu-config/

# NOTE: This file doesn't need to quote assignment values, like most shells do.
# All values are stored as-is, even if they contain spaces, so don't quote them.

# Domain name. If you do not have one, the public IP of the machine.
# For example: 198.51.100.1, or openvidu.example.com
DOMAIN_OR_PUBLIC_IP="서비스 도메인 IP"

# OpenVidu SECRET used for apps to connect to OpenVidu server and users to access to OpenVidu Dashboard
```

```
OPENVIDU_SECRET="OPENVIDU 비밀번호 defalut:MY_SECRET"

# Certificate type:
# - selfsigned:  Self signed certificate. Not recommended for production use.
#                Users will see an ERROR when connected to web page.
# - owncert:     Valid certificate purchased in a Internet services company.
#                Please put the certificates files inside folder ./owncert
#                with names certificate.key and certificate.cert
# - letsencrypt: Generate a new certificate using letsencrypt. Please set the
#                required contact email for Let's Encrypt in LETSENCRYPT_EMAIL
#                variable.
CERTIFICATE_TYPE="Certificate type"

# If CERTIFICATE_TYPE=letsencrypt, you need to configure a valid email for notifications
LETSENCRYPT_EMAIL="알림을 받을 email 설정"

# Proxy configuration
# If you want to change the ports on which openvidu listens, uncomment the following lines

# Allows any request to http://DOMAIN_OR_PUBLIC_IP:HTTP_PORT/ to be automatically
# redirected to https://DOMAIN_OR_PUBLIC_IP:HTTPS_PORT/.
# WARNING: the default port 80 cannot be changed during the first boot
# if you have chosen to deploy with the option CERTIFICATE_TYPE=letsencrypt
# HTTP_PORT=80

# Changes the port of all services exposed by OpenVidu.
# SDKs, REST clients and browsers will have to connect to this port
# HTTPS_PORT=443

# Old paths are considered now deprecated, but still supporte
```

```
d by default.
# OpenVidu Server will log a WARN message every time a deprecated path is called, indicating
# the new path that should be used instead. You can set property SUPPORT_DEPRECATED_API=false
# to stop allowing the use of old paths.
# Default value is true
# SUPPORT_DEPRECATED_API=false

# If true request to with www will be redirected to non-www requests
# Default value is false
# REDIRECT_WWW=false

# How many workers to configure in nginx proxy.
# The more workers, the more requests will be handled
# Default value is 10240
# WORKER_CONNECTIONS=10240

# Access restrictions
# In this section you will be able to restrict the IPs from which you can access to
# Openvidu API and the Administration Panel
# WARNING! If you touch this configuration you can lose access to the platform from some IPs.
# Use it carefully.

# This section limits access to the /dashboard (OpenVidu CE) and /inspector (OpenVidu Pro) pages.
# The form for a single IP or an IP range is:
# ALLOWED_ACCESS_TO_DASHBOARD=198.51.100.1 and ALLOWED_ACCESS_TO_DASHBOARD=198.51.100.0/24
# To limit multiple IPs or IP ranges, separate by commas like this:
# ALLOWED_ACCESS_TO_DASHBOARD=198.51.100.1, 198.51.100.0/24
# ALLOWED_ACCESS_TO_DASHBOARD=

# This section limits access to the Openvidu REST API.
# The form for a single IP or an IP range is:
# ALLOWED_ACCESS_TO_RESTAPI=198.51.100.1 and ALLOWED_ACCESS_TO_RESTAPI=198.51.100.0/24
# To limit multiple IPs or or IP ranges, separate by commas like this:
```

```
# ALLOWED_ACCESS_TO_RESTAPI=198.51.100.1, 198.51.100.0/24
# ALLOWED_ACCESS_TO_RESTAPI=

# Whether to enable recording module or not
OPENVIDU_RECORDING=false

# Use recording module with debug mode.
OPENVIDU_RECORDING_DEBUG=false

# Openvidu Folder Record used for save the openvidu recording videos. Change it
# with the folder you want to use from your host.
OPENVIDU_RECORDING_PATH=/opt/openvidu/recordings

# System path where OpenVidu Server should look for custom recording layouts
OPENVIDU_RECORDING_CUSTOM_LAYOUT=/opt/openvidu/custom-layout

# if true any client can connect to
# https://OPENVIDU_SERVER_IP:OPENVIDU_PORT/recordings/any_session_file.mp4
# and access any recorded video file. If false this path will be secured with
# OPENVIDU_SECRET param just as OpenVidu Server dashboard at
# https://OPENVIDU_SERVER_IP:OPENVIDU_PORT
# Values: true | false
OPENVIDU_RECORDING_PUBLIC_ACCESS=false

# Which users should receive the recording events in the client side
# (recordingStarted, recordingStopped). Can be all (every user connected to
# the session), publisher_moderator (users with role 'PUBLISHER' or
# 'MODERATOR'), moderator (only users with role 'MODERATOR') or none
# (no user will receive these events)
OPENVIDU_RECORDING_NOTIFICATION=publisher_moderator

# Timeout in seconds for recordings to automatically stop (and the session involved to be closed)
# when conditions are met: a session recording is started but no user is publishing to it or a session
# is being recorded and last user disconnects. If a user publishes within the timeout in either case,
# the automatic stop of the recording is cancelled
# 0 means no timeout
OPENVIDU_RECORDING_AUTOSTOP_TIMEOUT=120

# Maximum video bandwidth sent from clients to OpenVidu Server, in kbps.
# 0 means unconstrained
OPENVIDU_STREAMS_VIDEO_MAX_RECV_BANDWIDTH=1000

# Minimum video bandwidth sent from clients to OpenVidu Server, in kbps.
# 0 means unconstrained
OPENVIDU_STREAMS_VIDEO_MIN_RECV_BANDWIDTH=300

# Maximum video bandwidth sent from OpenVidu Server to clients, in kbps.
# 0 means unconstrained
OPENVIDU_STREAMS_VIDEO_MAX_SEND_BANDWIDTH=1000

# Minimum video bandwidth sent from OpenVidu Server to clients, in kbps.
# 0 means unconstrained
OPENVIDU_STREAMS_VIDEO_MIN_SEND_BANDWIDTH=300

# All sessions of OpenVidu will try to force this codec. If OPENVIDU_STREAMS_ALLOW_TRANSCODING=true
# when a codec can not be forced, transcoding will be allowed
# Values: MEDIA_SERVER_PREFERRED, NONE, VP8, VP9, H264
# Default value is MEDIA_SERVER_PREFERRED
# OPENVIDU_STREAMS_FORCED_VIDEO_CODEC=MEDIA_SERVER_PREFERRED

# Allow transcoding if codec specified in OPENVIDU_STREAMS_FORCED_VIDEO_CODEC can not be applied
# Values: true | false
# Default value is false
# OPENVIDU_STREAMS_ALLOW_TRANSCODING=false

# true to enable OpenVidu Webhook service. false' otherwise
# Values: true | false
OPENVIDU_WEBHOOK=false
```

```
# HTTP endpoint where OpenVidu Server will send Webhook HTTP POST messages
# Must be a valid URL: http(s)://ENDPOINT
#OPENVIDU_WEBHOOK_ENDPOINT=

# List of headers that OpenVidu Webhook service will attach to HTTP POST messages
#OPENVIDU_WEBHOOK_HEADERS=

# List of events that will be sent by OpenVidu Webhook service
# Default value is all available events
OPENVIDU_WEBHOOK_EVENTS=[sessionCreated,sessionDestroyed,participantJoined,participantLeft,webrtcConnectionCreated,webrtcConnectionDestroyed,recordingStatusChanged,filterEventDispatched,mediaNodeStatusChanged,nodeCrashed,nodeRecovered,broadcastStarted,broadcastStopped]

# How often the garbage collector of non active sessions runs.
# This helps cleaning up sessions that have been initialized through
# REST API (and maybe tokens have been created for them) but have had no users connected.
# Default to 900s (15 mins). 0 to disable non active sessions garbage collector
OPENVIDU_SESSIONS_GARBAGE_INTERVAL=900

# Minimum time in seconds that a non active session must have been in existence
# for the garbage collector of non active sessions to remove it. Default to 3600s (1 hour).
# If non active sessions garbage collector is disabled
# (property 'OPENVIDU_SESSIONS_GARBAGE_INTERVAL' to 0) this property is ignored
OPENVIDU_SESSIONS_GARBAGE_THRESHOLD=3600

# Call Detail Record enabled
# Whether to enable Call Detail Record or not
# Values: true | false
OPENVIDU_CDR=false

# Path where the cdr log files are hosted
```

```
OPENVIDU_CDR_PATH=/opt/openvidu/cdr

# Kurento Media Server image
# --------------------------
# Docker hub kurento media server: https://hub.docker.com/r/kurento/kurento-media-server
# Uncomment the next line and define this variable with KMS image that you want use
# KMS_IMAGE=kurento/kurento-media-server:7.0.1

# Kurento Media Server Level logs
# -------------------------------
# Uncomment the next line and define this variable to change
# the verbosity level of the logs of KMS
# Documentation: https://doc-kurento.readthedocs.io/en/stable/features/logging.html
# KMS_DOCKER_ENV_GST_DEBUG=

# Openvidu Server Level logs
# --------------------------
# Uncomment the next line and define this variable to change
# the verbosity level of the logs of Openvidu Service
# RECOMENDED VALUES: INFO for normal logs DEBUG for more verbose logs
# OV_CE_DEBUG_LEVEL=INFO

# Java Options
# --------------------------
# Uncomment the next line and define this to add
# options to java command
# Documentation: https://docs.oracle.com/cd/E37116_01/install.111210/e23737/configuring_jvm.htm#OUDIG00058
# JAVA_OPTIONS=-Xms2048m -Xmx4096m -Duser.timezone=UTC
```

▼ custom-nginx.conf ($/opt/openvidu)

```
# Your App
upstream yourapp {
    server localhost:5442;
}

upstream openviduserver {
    server localhost:5443;
```

```
}

upstream jenkins {
    server localhost:9090;
}

upstream frontend {
    server localhost:5173;
}

upstream backend {
    server localhost:8080;
}

upstream webrtc {
    server localhost:5000;
}

# upstream redis {
#    server localhost:6379;
# }

# server {
#    listen 6379 ssl;
#    listen [::]:6379 ssl;
#
#    ssl_certificate         /etc/letsencrypt/live/i10b112.p.ssafy.io/fullchain.pem;
#    ssl_certificate_key     /etc/letsencrypt/live/i10b112.p.ssafy.io/privkey.pem;
#    ssl_trusted_certificate /etc/letsencrypt/live/i10b112.p.ssafy.io/fullchain.pem;
#
#    location / {
#        proxy_pass http://localhost:6379;
#    }
#}

server {

    listen 9091 ssl;
    listen [::]:9091 ssl;
    server_name jenkins;
    # access_log /home/ubuntu/nginx/jenkins/access.log
    # error_log /home/ubuntu/nginx/jenkins/error.log

    # SSL Config
    ssl_certificate         /etc/letsencrypt/live/i10b112.p.ssafy.io/fullchain.pem;
    ssl_certificate_key     /etc/letsencrypt/live/i10b112.p.ssafy.io/privkey.pem;
    ssl_trusted_certificate /etc/letsencrypt/live/i10b112.p.ssafy.io/fullchain.pem;

    ssl_session_cache shared:SSL:50m;
    ssl_session_timeout 5m;
    ssl_stapling on;
    ssl_stapling_verify on;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers "ssl cipher";
    ssl_prefer_server_ciphers off;

    add_header Strict-Transport-Security "max-age=63072000" always;

    # 나머지 설정은 그대로 두고, 필요에 따라 수정
    location / {
        allow "접속을 허용할 IP"
        deny all;
        proxy_pass http://jenkins;
        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;
    }
}

server {
    listen 80;
    listen [::]:80;
    server_name i10b112.p.ssafy.io;
    return 301 https://i10b112.p.ssafy.io$request_uri;
    # Redirect to https
    location / {
        rewrite ^(.*) https://i10b112.p.ssafy.io:443$1 permanent;
    }

    # location / {
    #    return 301 https://$server_name$request_uri;
    # }
    # letsencrypt
    location /.well-known/acme-challenge/ {
        root /var/www/certbot;
    }

    location /nginx_status {
        stub_status;
        allow 127.0.0.1;        #only allow requests from localhost
        deny all;               #deny all other hosts
    }
}


server {
    listen 443 ssl;
    listen [::]:443 ssl;
#    root /var/lib/docker/volumes/jenkins/_data/workspace/Prolog/client/prolog/dist;
#    index index.html index.htm ;

    server_name i10b112.p.ssafy.io;
    # error_page 497 https://$server_name$request_uri;

#    location / {
#        root   /home/user/myapp/build;
#       root /var/lib/docker/volumes/jenkins/_data/workspace/Prolog/client/prolog/dist;
#       index index.html;
#        try_files $uri/ /index.html =404;

#    }
#    location / {
#       # try_files $uri $uri/ /index.html;
#       try_files $uri $uri/ =404;
#    }

    # SSL Config
    ssl_certificate         /etc/letsencrypt/live/i10b112.p.ssafy.io/fullchain.pem;
    ssl_certificate_key     /etc/letsencrypt/live/i10b112.p.ssafy.io/privkey.pem;
    ssl_trusted_certificate /etc/letsencrypt/live/i10b112.p.ssafy.io/fullchain.pem;

    ssl_session_cache shared:SSL:50m;
    ssl_session_timeout 5m;
    ssl_stapling on;
    ssl_stapling_verify on;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers "ssl cipher";
    ssl_prefer_server_ciphers off;

    add_header Strict-Transport-Security "max-age=63072000" always;

    # Proxy
    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;
    proxy_set_header X-Forwarded-Proto https;
    proxy_headers_hash_bucket_size 512;
    proxy_redirect off;

    # Websockets
     proxy_http_version 1.1;
     proxy_set_header Upgrade $http_upgrade;
     proxy_set_header Connection "upgrade";

     location / {
        proxy_pass http://frontend/;
    }
```

```
   # location / {
#       root /var/lib/docker/volumes/jenkins/_data/workspace/Prolog/client/prolog/dist/;
#       index index.html;
#       try_files $uri $uri/ /index.html;
#    }

    # Your App
    location /api {
        # rewrite ^/api(/.*)$ $1 break;
        proxy_pass http://backend;
        proxy_set_header Host $http_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;
    }

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

    location /webrtc {
        proxy_pass http://webrtc;
        proxy_set_header Host $http_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;
    }

    ########################
    # OpenVidu Locations   #
    ########################
    ################################
    # Common rules CE              #
    #################################
    # Dashboard rule
    location /dashboard {
        allow all;
        deny all;
        proxy_pass http://openviduserver;

    }

    # Websocket rule
    location ~ /openvidu$ {
        proxy_pass http://openviduserver;
    }


    ##################################
    # New API                        #
    ##################################
    location /openvidu/layouts {
        rewrite ^/openvidu/layouts/(.*)$ /custom-layout/$1 break;
        root /opt/openvidu;
    }

    location /openvidu/recordings {
        proxy_pass http://openviduserver;
    }

    location /openvidu/api {
        allow all;
        deny all;
        proxy_pass http://openviduserver;
    }

    location /openvidu/info {
        allow all;
        deny all;
        proxy_pass http://openviduserver;
    }

    location /openvidu/accept-certificate {
        proxy_pass http://openviduserver;
    }

    location /openvidu/cdr {
        allow all;
        deny all;
        proxy_pass http://openviduserver;
    }

    ##################################
    # LetsEncrypt                    #
    ##################################
    location /.well-known/acme-challenge {
        root /var/www/certbot;
        try_files $uri $uri/ =404;
    }

}
```

2) build 전 사전작업

1. Git clone

```
git credentialsId: 'personal_token', url: 'https://lab.ssafy.com/s10-webmobile1-sub3/S10P13B112.git'
```

2. 실행중인 컨테이너, 생성된 도커 이미지가 있다면 제거

```
docker stop springboot || true && docker rm springboot || true && docker rmi b112/springboot || true
docker stop react || true && docker rm react || true && docker rmi b112/react || true
```

2) Front build

1. 프로젝트 폴더로 이동 후 빌드파일 생성

```
cd S10P13B112/client/prolog
npm install && npm run build
```

2. 미리 만들어둔 Dockerfile로 어플리케이션을 dockerize

```
FROM node:20.11.0

# 빌드파일을 도커 컨테이너 내부로 복사
COPY <빌드 파일 경로> .
```

```
ENTRYPOINT ["npx", "http-server", "--port", "5173", "-P", "<Domain
```

3. Dockerfile이 존재하는 경로에서 도커 이미지 생성

```
docker build . -t b112/react
```

4. 도커 이미지로 컨테이너 생성

```
docker run --name react -d -p 5173:5173 b112/react
```

3) Backend build

1. 프로젝트 폴더로 이동 후 빌드파일 생성

```
# WINDOWS
gradlew.bat build

# UNIX
chmod +x gradlew && ./gradlew build
chmod +x gradlew && ./gradlew clean --info build # with build log
```

2. 미리 만들어둔 Dockerfile로 어플리케이션을 dockerize

```
FROM openjdk:17-oracle

# 빌드파일을 도커 컨테이너 내부로 복사
COPY <jar 파일> prolog-springboot.jar

ENTRYPOINT ["java", "-jar", "prolog-springboot.jar"]
```

3. Dockerfile이 존재하는 경로에서 도커 이미지 생성

```
docker build . -t b112/springboot
```

4. 도커 이미지로 컨테이너 생성

```
docker run --name springboot -d -p 8080:8080 b112/springboot
```

4) webrtc Backend build - 수동

1. mvn clean package로 SNAPSHOT(.jar) 파일 생성

2. nohup java -jar webrtc-0.0.1-SNAPSHOT.jar & 로 ssh 연결 여부에 관계 없이 작동되도록 실행

III. 배포하기

1) 서버 구성

포트번호

- Http `80:80`
- Https `443:443`
- backend `8080:8080`
- frontend `5173:5173`
- openviduserver `5443:5443`

> 💡 **openvidu를 사용하기 위해 열어야하는 포트들**
>
> - **22 TCP**: to connect using SSH to admin OpenVidu.
> - **80 TCP**: if you select Let's Encrypt to generate an SSL certificate this port is used by the generation process.
> - **443 TCP**: OpenVidu server and application are published by default in standard https port.
> - **3478 TCP+UDP**: used by STUN/TURN server to resolve clients IPs.
> - **40000 - 57000 TCP+UDP**: used by Kurento Media Server to establish media connections.
> - **57001 - 65535 TCP+UDP**: used by TURN server to establish relayed media connections.
>
> 80, 443, 3478, 5442, 5443, 6379 and 8888.

- openvidu-api `5000:5000`
- redis `6379:6379`
- jenkins `9091:9091`
- kurento `8888:8888`

※배포는 다음의 순서를 꼭 따라하세요
Openvidu가 포트 80과 443을 사용하기 때문에 먼저 배정 후 커스텀 해야합니다.

1. Openvidu-server On Premises 배포

사전준비

- EC2에 Docker와 Docker compose가 설치되어 있어야 합니다.
- 아래 명령어를 통해 포트를 열어줍니다.

```
ufw allow ssh
ufw allow 80/tcp
ufw allow 443/tcp
ufw allow 3478/tcp
ufw allow 3478/udp
ufw allow 40000:57000/tcp
ufw allow 40000:57000/udp
ufw allow 57001:65535/tcp
ufw allow 57001:65535/udp
ufw enable
```

단계

1. .env 파일을 작성합니다.
2. 다음 명령어를 통해 openvidu-server를 배포합니다.

```
./openvidu start
```

3. 그러면 docker-compose 파일이 build 됩니다

```
Creating openvidu-docker-compose_coturn_1          ... done
Creating openvidu-docker-compose_app_1             ... done
Creating openvidu-docker-compose_kms_1             ... done
Creating openvidu-docker-compose_nginx_1           ... done
Creating openvidu-docker-compose_redis_1           ... done
Creating openvidu-docker-compose_openvidu-server_1 ... done
```

4. 성공적으로 배포가 되면 해당 메시지가 나오게 됩니다. /dashboard 링크로 접속하여 연결을 확인하세요. `user: OPENVIDUAPP pass: OPENVIDU_SECRET in .env file 값`

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

```
    OpenVidu Platform is ready!
    ----------------------------

    * OpenVidu Server: https://DOMAIN_OR_PUBLIC_IP/

    * OpenVidu Dashboard: https://DOMAIN_OR_PUBLIC_IP/dashboard/

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

6. 더 자세한 내용은 해당 링크를 통해 반드시 확인하세요.

On premises - OpenVidu Docs

OpenVidu is deployed in production as a set of Docker containers managed with Docker Compose. You can deploy OpenVidu in any modern Linux distribution.

https://docs.openvidu.io/en/2.29.0/deployment/ce/on-premises/#2-deployment

2. Openvidu 內 nginx custom

openvidu-server를 배포하게 되면 자체적으로 nginx를 가지고 있게 됩니다.

사용자는 custom-nginx.conf 파일을 생성하여 nginx 설정을 override 합니다.

자세한 내용은 해당 링크를 통해 반드시 확인하세요

Troubleshooting / FAQ - OpenVidu Docs

You have an app that uses OpenVidu to stream some video user-to-user, and the process looks perfectly okey. No errors on the console and all the OpenVidu events you are subscribed to are correctly triggered. So what's happening?

https://docs.openvidu.io/en/2.29.0/troubleshooting/#16-how-can-i-customize-deployed-nginx

0. 사용자는 먼저 openvidu-server가 배포되어 있어야 합니다.
1. Nginx가 실행 중이면 다음을 실행하여 nginx 컨테이너에서 생성된 구성을 가져옵니다.

```
sudo su
cd /opt/openvidu
docker-compose exec nginx cat /etc/nginx/conf.d/default.conf > cus
docker-compose exec nginx cat /etc/nginx/nginx.conf > nginx.conf
```

2. 두가지 파일을 사용할 것 입니다.

```
/opt/openvidu/custom-nginx.conf: openvidu 모든 설정이 들어있는 nginx ㅍ
/opt/openvidu/nginx.conf: nginx의 기본 설정 파일
```

3. custom-nginx.conf 파일에 이전에 생성된 기본 구성을 필요에 맞게 수정합니다.

4. nginx.conf 파일을 수정할 수도 있지만 다음 두줄을 삭제하게 되면 서버가 block 되므로 건드리지 마십시오

```
include /etc/nginx/conf.d/*.conf;
include /etc/nginx/vhost.d/*.conf;
```

5. nginx 서비스를 docker-compose.yml 파일에 추가하세요

```
nginx:
        ...
        volumes:
            ...
            - ./custom-nginx.conf:/custom-nginx/custom-nginx.conf
            - ./nginx.conf:/etc/nginx/nginx.conf
```

6. 기본 nginx 구성이 재정의 될 것입니다.

7. openvidu를 시작하거나 재시작 하십시오.

```
sudo su
cd /opt/openvidu
./openvidu start

or

sudo su
cd /opt/openvidu
./openvidu restart
```

※ 절대 custom-nginx.conf 이름을 임의로 수정하거나 오타를 내지 마세요

IV. DB Connection

1. MySQL

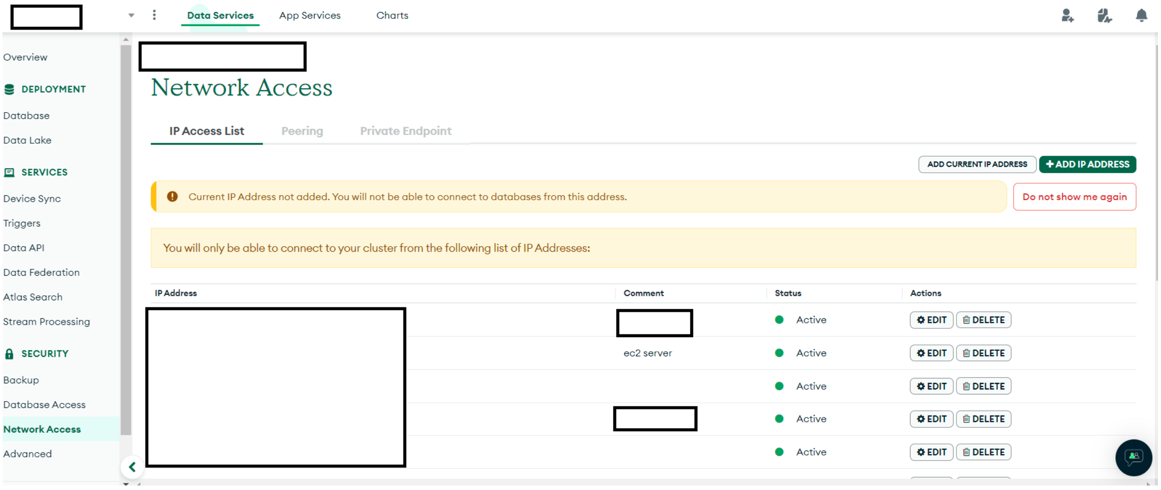
1. EC2 내 MySQL 다운로드
2. 데이터베이스명은 prolog
3. Springboot는 application.properties에 명령어 추가

2. MongoDB

1. Atlas MongoDB를 사용해 MongoDB를 클라우드환경에서 사용

2. 메뉴얼 따라 DB 생성

3. EC2 ip를 등록해서 EC2에서 접근, 개인 ip 등록해서 화이트리스트로 DB 접근



Network Access

Current IP Address not added. You will not be able to connect to databases from this address.

You will only be able to connect to your cluster from the following list of IP Addresses:

IP Address	Comment	Status	Actions
[redacted]	[redacted]	● Active	EDIT DELETE
[redacted]	ec2 server	● Active	EDIT DELETE
[redacted]		● Active	EDIT DELETE
[redacted]	[redacted]	● Active	EDIT DELETE
[redacted]		● Active	EDIT DELETE

4. Springboot application.properties 명령어 추가

V. 배포 시 특이사항

- 원활한 배포를 위해 openvidu-server먼저 배포하고 진행해주세요.