Docker, EC2, Nginx

1. Docker-compose

version: "3.8"

services:

app:

image: "jc21/nginx-proxy-manager:latest"

restart: unless-stopped

ports:

- "80:80"

- "60081:81"

- "443:443"

environment:

# Mysql/Maria connection parameters:

DB\_MYSQL\_HOST: "db"

DB\_MYSQL\_PORT: 3306

DB\_MYSQL\_USER: "npm"

DB\_MYSQL\_PASSWORD: "npm"

DB\_MYSQL\_NAME: "npm"

# Uncomment this if IPv6 is not enabled on your host

# DISABLE\_IPV6: 'true'

volumes:

- ./data:/data

- ./letsencrypt:/etc/letsencrypt

depends\_on:

- db

db:

image: "jc21/mariadb-aria:latest"

restart: unless-stopped

environment:

MYSQL\_ROOT\_PASSWORD: "npm"

MYSQL\_DATABASE: "npm"

MYSQL\_USER: "npm"

MYSQL\_PASSWORD: "npm"

MARIADB\_AUTO\_UPGRADE: "1"

volumes:

- ./data/mysql:/var/lib/mysql

frontend:

image: mine0702/beevarium-frontend

restart: unless-stopped

container\_name: app\_frontend

expose:

- "5000"

stdin\_open: true

# mem\_limit: 1024m

backend:

image: mine0702/beevarium-backend

restart: unless-stopped

container\_name: app\_backend

stdin\_open: true

expose:

- "8080" # 백엔드 서비스가 사용하는 포트, 외부에 노출하지 않음

# mem\_limit: 1024m

tts:

image: mine0702/beevarium-tts

restart: unless-stopped

container\_name: app\_tts

stdin\_open: true

expose:

- "8999" # 백엔드 서비스가 사용하는 포트, 외부에 노출하지 않음

# networks:

# default:

# external:

# name: nginx-proxy

2. Front-end

2.1 Docker

FROM node:alpine as build-stage

WORKDIR /app

COPY package\*.json ./

RUN npm install

COPY . .

RUN npm run build

FROM nginx:alpine as production-stage

COPY ./nginx/default.conf /etc/nginx/conf.d/default.conf

COPY --from=build-stage /app/dist /usr/share/nginx/html

2.2 Nginx

server{

listen 5000;

location / {

root /usr/share/nginx/html;

index index.html index.htm;

try\_files $uri $uri/ /index.html;

}

}

3. Back-end

FROM openjdk:17  
  
ARG *JAR\_FILE*=build/libs/\*.jar  
  
COPY ${*JAR\_FILE*} app.jar  
  
ENTRYPOINT ["java","-jar","/app.jar"]

4. TTS

# Node.js 공식 이미지를 사용하는 베이스 이미지 설정

FROMnode:alpine

# 앱 디렉토리 생성

WORKDIR/usr/src/app

# 앱 의존성 설치

# package.json과 package-lock.json을 모두 복사

COPYpackage\*.json ./

RUNnpm install

# 프로덕션을 위한 코드인 경우

# RUN npm ci --only=production

# 앱 소스 추가

COPY. .

# # SSL/TLS 인증서 파일 복사

# COPY ./192.168.31.163.crt ./

# COPY ./192.168.31.163.key ./

# # 앱이 HTTPS 포트에서 실행됨을 알림

# EXPOSE 80

# 앱 실행

CMD[ "node", "index.js"]

5. AI

5.1 Docker

# Node.js 공식 이미지를 사용하는 베이스 이미지 설정

FROM node:alpine

# 앱 디렉토리 생성

WORKDIR /usr/src/app

# 앱 의존성 설치

# package.json과 package-lock.json을 모두 복사

COPY package\*.json ./

RUN npm install

# 프로덕션을 위한 코드인 경우

# RUN npm ci --only=production

# 앱 소스 추가

COPY . .

# # SSL/TLS 인증서 파일 복사

# COPY ./192.168.31.163.crt ./

# COPY ./192.168.31.163.key ./

# # 앱이 HTTPS 포트에서 실행됨을 알림

# EXPOSE 80

# 앱 실행

CMD [ "node", "index.js" ]

5.2 Docker-compose

version: "3.8"

services:

app:

image: "jc21/nginx-proxy-manager:latest"

restart: unless-stopped

ports:

- "80:80"

- "81:81"

- "443:443"

environment:

# Mysql/Maria connection parameters:

DB\_MYSQL\_HOST: "db"

DB\_MYSQL\_PORT: 3306

DB\_MYSQL\_USER: "npm"

DB\_MYSQL\_PASSWORD: "npm"

DB\_MYSQL\_NAME: "npm"

# Uncomment this if IPv6 is not enabled on your host

# DISABLE\_IPV6: 'true'

volumes:

- ./data:/data

- ./letsencrypt:/etc/letsencrypt

depends\_on:

- db

db:

image: "jc21/mariadb-aria:latest"

restart: unless-stopped

environment:

MYSQL\_ROOT\_PASSWORD: "npm"

MYSQL\_DATABASE: "npm"

MYSQL\_USER: "npm"

MYSQL\_PASSWORD: "npm"

MARIADB\_AUTO\_UPGRADE: "1"

volumes:

- ./data/mysql:/var/lib/mysql

backend:

image: mine0702/beevarium-ai

restart: unless-stopped

container\_name: app\_backend

stdin\_open: true

ports:

- 8080:8080

expose:

- "8080" # 백엔드 서비스가 사용하는 포트, 외부에 노출하지 않음

# mem\_limit: 1024m

# networks:

# default:

# external:

# name: nginx-proxy

6. Media-node

# ------------------------------------------------------------------------------

#

# DO NOT MODIFY THIS FILE !!!

#

# Configuration properties should be specified in .env file

#

# This docker-compose file coordinates all services of OpenVidu CE Platform.

#

# Openvidu Version: 2.29.0

#

# Installation Mode: On Premises

#

# ------------------------------------------------------------------------------

version: '3.1'

services:

media-node-controller:

image: openvidu/media-node-controller:2.29.0

restart: always

ulimits:

core: -1

entrypoint: ['/bin/sh', '-c', '/beats/copy\_config\_files.sh && /usr/local/bin/entrypoint.sh']

environment:

- KMS\_IMAGE=kurento/kurento-media-server:7.0.1

- MEDIASOUP\_IMAGE=openvidu/mediasoup-controller:2.29.0

- METRICBEAT\_IMAGE=docker.elastic.co/beats/metricbeat-oss:7.8.0

- FILEBEAT\_IMAGE=docker.elastic.co/beats/filebeat-oss:7.8.0

- OPENVIDU\_RECORDING\_IMAGE=openvidu/openvidu-recording:2.29.0

- COTURN\_IMAGE=openvidu/openvidu-coturn:2.29.0

- SPEECH\_TO\_TEXT\_IMAGE=openvidu/speech-to-text-service:2.29.0

- NO\_COLOR=true

ports:

- 3000:3000

volumes:

- /opt/openvidu/recordings:/opt/openvidu/recordings

- /opt/openvidu/beats:/opt/openvidu/beats

- /var/run/docker.sock:/var/run/docker.sock

- /opt/openvidu/kurento-logs:/opt/openvidu/kurento-logs

- ./beats:/beats

logging:

options:

max-size: "100M"

7. Main-node

7.1 Docker-compose

# ------------------------------------------------------------------------------

#

# DO NOT MODIFY THIS FILE !!!

#

# Configuration properties should be specified in .env file

#

# Application based on OpenVidu should be specified in

# docker-compose.override.yml file

#

# This docker-compose file coordinates all services of OpenVidu Pro Platform

#

# This file will be overridden when update OpenVidu Platform

#

# Openvidu Version: 2.29.0

#

# Installation Mode: On Premises

#

# ------------------------------------------------------------------------------

version: '3.1'

services:

openvidu-server:

image: openvidu/openvidu-server-pro:2.29.0

restart: on-failure

network\_mode: host

entrypoint: ['/usr/local/bin/entrypoint.sh']

volumes:

- ./coturn:/run/secrets/coturn

- /var/run/docker.sock:/var/run/docker.sock

- ${OPENVIDU\_RECORDING\_PATH}:${OPENVIDU\_RECORDING\_PATH}

- ${OPENVIDU\_RECORDING\_CUSTOM\_LAYOUT}:${OPENVIDU\_RECORDING\_CUSTOM\_LAYOUT}

- ${OPENVIDU\_CDR\_PATH}:${OPENVIDU\_CDR\_PATH}

- ./cluster:/opt/openvidu/cluster

- .env:${PWD}/.env

env\_file:

- .env

environment:

- SERVER\_SSL\_ENABLED=false

- SERVER\_PORT=5443

- KMS\_URIS=[]

- COTURN\_IP=${COTURN\_IP:-auto-ipv4}

- COTURN\_PORT=${COTURN\_PORT:-3478}

- OPENVIDU\_PRO\_CLUSTER=true

- OPENVIDU\_PRO\_ELASTICSEARCH=${OPENVIDU\_PRO\_ELASTICSEARCH:-true}

- OPENVIDU\_PRO\_KIBANA\_HOST=${OPENVIDU\_PRO\_KIBANA\_HOST:-http://127.0.0.1/kibana}

- OPENVIDU\_PRO\_ELASTICSEARCH\_HOST=${OPENVIDU\_PRO\_ELASTICSEARCH\_HOST:-http://127.0.0.1:9200}

- WAIT\_KIBANA\_URL=${OPENVIDU\_PRO\_KIBANA\_HOST:-http://127.0.0.1/kibana}

- DOTENV\_PATH=${PWD}

- OPENVIDU\_PRO\_COTURN\_IN\_MEDIA\_NODES=${OPENVIDU\_PRO\_COTURN\_IN\_MEDIA\_NODES:-false}

- OPENVIDU\_PRO\_COTURN\_PORT\_MEDIA\_NODES=${OPENVIDU\_PRO\_COTURN\_PORT\_MEDIA\_NODES:-443}

- OPENVIDU\_PRO\_MEDIA\_NODE\_PUBLIC\_IP\_AUTODISCOVER=${OPENVIDU\_PRO\_MEDIA\_NODE\_PUBLIC\_IP\_AUTODISCOVER:-auto-ipv4}

logging:

options:

max-size: "${DOCKER\_LOGS\_MAX\_SIZE:-100M}"

coturn:

image: openvidu/openvidu-coturn:2.29.0

restart: on-failure

network\_mode: host

env\_file:

- .env

volumes:

- ./coturn:/run/secrets/coturn

command:

- --log-file=stdout

- --external-ip=$$(detect-external-ip)

- --listening-port=${COTURN\_PORT:-3478}

- --fingerprint

- --min-port=${COTURN\_MIN\_PORT:-40000}

- --max-port=${COTURN\_MAX\_PORT:-65535}

- --realm=openvidu

- --verbose

- --use-auth-secret

- --static-auth-secret=$${COTURN\_SHARED\_SECRET\_KEY}

logging:

options:

max-size: "${DOCKER\_LOGS\_MAX\_SIZE:-100M}"

nginx:

image: openvidu/openvidu-proxy:2.29.0

restart: always

network\_mode: host

volumes:

- ./certificates:/etc/letsencrypt

- ./owncert:/owncert

- ./custom-nginx-vhosts:/etc/nginx/vhost.d/

- ./custom-nginx-locations:/custom-nginx-locations

- ${OPENVIDU\_RECORDING\_CUSTOM\_LAYOUT}:/opt/openvidu/custom-layout

environment:

- DOMAIN\_OR\_PUBLIC\_IP=${DOMAIN\_OR\_PUBLIC\_IP}

- CERTIFICATE\_TYPE=${CERTIFICATE\_TYPE}

- LETSENCRYPT\_EMAIL=${LETSENCRYPT\_EMAIL}

- PROXY\_HTTP\_PORT=${HTTP\_PORT:-}

- PROXY\_HTTPS\_PORT=${HTTPS\_PORT:-}

- PROXY\_HTTPS\_PROTOCOLS=${HTTPS\_PROTOCOLS:-}

- PROXY\_HTTPS\_CIPHERS=${HTTPS\_CIPHERS:-}

- PROXY\_HTTPS\_HSTS=${HTTPS\_HSTS:-}

- ALLOWED\_ACCESS\_TO\_DASHBOARD=${ALLOWED\_ACCESS\_TO\_DASHBOARD:-}

- ALLOWED\_ACCESS\_TO\_RESTAPI=${ALLOWED\_ACCESS\_TO\_RESTAPI:-}

- PROXY\_MODE=PRO

- WITH\_APP=true

- SUPPORT\_DEPRECATED\_API=${SUPPORT\_DEPRECATED\_API:-false}

- REDIRECT\_WWW=${REDIRECT\_WWW:-false}

- WORKER\_CONNECTIONS=${WORKER\_CONNECTIONS:-10240}

- PUBLIC\_IP=${PROXY\_PUBLIC\_IP:-auto-ipv4}

logging:

options:

max-size: "${DOCKER\_LOGS\_MAX\_SIZE:-100M}"

elasticsearch:

image: openvidu/openvidu-elasticsearch:7.8.0

restart: always

environment:

- discovery.type=single-node

- xpack.security.enabled=true

- "ES\_JAVA\_OPTS=${ES\_JAVA\_OPTS:--Xms2g -Xmx2g}"

ports:

- 9200:9200

volumes:

- ./elasticsearch:/usr/share/elasticsearch/data

command: >

/bin/bash -c "elasticsearch-users useradd ${ELASTICSEARCH\_USERNAME}

-p ${ELASTICSEARCH\_PASSWORD} -r superuser;

elasticsearch-users passwd ${ELASTICSEARCH\_USERNAME} -p ${ELASTICSEARCH\_PASSWORD};

docker-entrypoint.sh"

logging:

options:

max-size: "${DOCKER\_LOGS\_MAX\_SIZE:-100M}"

kibana:

image: docker.elastic.co/kibana/kibana:7.8.0

restart: always

environment:

- SERVER\_BASEPATH="/kibana"

- xpack.security.enabled=true

- ELASTICSEARCH\_USERNAME=${ELASTICSEARCH\_USERNAME}

- ELASTICSEARCH\_PASSWORD=${ELASTICSEARCH\_PASSWORD}

ports:

- 5601:5601

logging:

options:

max-size: "${DOCKER\_LOGS\_MAX\_SIZE:-100M}"

metricbeat:

image: docker.elastic.co/beats/metricbeat-oss:7.8.0

network\_mode: host

restart: always

user: root

env\_file:

- .env

environment:

- OPENVIDU\_PRO\_ELASTICSEARCH\_HOST=${OPENVIDU\_PRO\_ELASTICSEARCH\_HOST:-http://127.0.0.1:9200}

- OPENVIDU\_PRO\_STATS\_MONITORING\_INTERVAL=${OPENVIDU\_PRO\_STATS\_MONITORING\_INTERVAL:-10}

volumes:

- /var/run/docker.sock:/var/run/docker.sock

- ./beats/metricbeat.yml:/usr/share/metricbeat/metricbeat.yml:ro

- /proc:/hostfs/proc:ro

- /sys/fs/cgroup:/hostfs/sys/fs/cgroup:ro

- /:/hostfs:ro

entrypoint: >

/bin/bash -c 'if [[ $${OPENVIDU\_PRO\_ELASTICSEARCH\_HOST} =~ ^(https?://)?([^:/]+)(:([0-9]+))?(/.\*)?$$ ]]; then

ES\_PROTO=$${BASH\_REMATCH[1]}

ES\_HOST=$${BASH\_REMATCH[2]}

ES\_PORT=$${BASH\_REMATCH[4]}

ES\_PATH=$${BASH\_REMATCH[5]}

if [[ $$ES\_PROTO == "https://" ]] && [[ -z $$ES\_PORT ]]; then

ES\_PORT=443

elif [[ $$ES\_PROTO == "http://" ]] && [[ -z $$ES\_PORT ]]; then

ES\_PORT=80

fi

export OPENVIDU\_PRO\_ELASTICSEARCH\_HOST=$${ES\_PROTO}$${ES\_HOST}:$${ES\_PORT}$${ES\_PATH}

else

exit 1

fi;

exec /bin/bash -c "$$@"'

command: >

/bin/bash -c "metricbeat -e -strict.perms=false -e -system.hostfs=/hostfs

`if [ ! -z $ELASTICSEARCH\_USERNAME ]; then echo '-E output.elasticsearch.username=$ELASTICSEARCH\_USERNAME'; fi`

`if [ ! -z $ELASTICSEARCH\_PASSWORD ]; then echo '-E output.elasticsearch.password=$ELASTICSEARCH\_PASSWORD'; fi`"

logging:

options:

max-size: "${DOCKER\_LOGS\_MAX\_SIZE:-100M}"

filebeat:

image: docker.elastic.co/beats/filebeat-oss:7.8.0

network\_mode: host

restart: always

user: root

env\_file:

- .env

environment:

- OPENVIDU\_PRO\_ELASTICSEARCH\_HOST=${OPENVIDU\_PRO\_ELASTICSEARCH\_HOST:-http://127.0.0.1:9200}

volumes:

- ./beats/filebeat.yml:/usr/share/filebeat/filebeat.yml:ro

- /var/lib/docker:/var/lib/docker:ro

- /var/run/docker.sock:/var/run/docker.sock

entrypoint: >

/bin/bash -c 'if [[ $${OPENVIDU\_PRO\_ELASTICSEARCH\_HOST} =~ ^(https?://)?([^:/]+)(:([0-9]+))?(/.\*)?$$ ]]; then

ES\_PROTO=$${BASH\_REMATCH[1]}

ES\_HOST=$${BASH\_REMATCH[2]}

ES\_PORT=$${BASH\_REMATCH[4]}

ES\_PATH=$${BASH\_REMATCH[5]}

if [[ $$ES\_PROTO == "https://" ]] && [[ -z $$ES\_PORT ]]; then

ES\_PORT=443

elif [[ $$ES\_PROTO == "http://" ]] && [[ -z $$ES\_PORT ]]; then

ES\_PORT=80

fi

export OPENVIDU\_PRO\_ELASTICSEARCH\_HOST=$${ES\_PROTO}$${ES\_HOST}:$${ES\_PORT}$${ES\_PATH}

else

exit 1

fi;

exec /bin/bash -c "$$@"'

command: >

/bin/bash -c "filebeat -e -strict.perms=false

`if [ ! -z $ELASTICSEARCH\_USERNAME ]; then echo '-E output.elasticsearch.username=$ELASTICSEARCH\_USERNAME'; fi`

`if [ ! -z $ELASTICSEARCH\_PASSWORD ]; then echo '-E output.elasticsearch.password=$ELASTICSEARCH\_PASSWORD'; fi`"

logging:

options:

max-size: "${DOCKER\_LOGS\_MAX\_SIZE:-100M}"

7.2 Docker-compose override

version: '3.1'

services:

# --------------------------------------------------------------

#

# Change this if your want use your own application.

# It's very important expose your application in port 5442

# and use the http protocol.

#

# Default Application

#

# Openvidu-Call Version: 2.29.0

#

# --------------------------------------------------------------

app:

image: openvidu/openvidu-call:2.29.0

restart: on-failure

network\_mode: host

environment:

- SERVER\_PORT=5442

- OPENVIDU\_URL=http://localhost:5443

- OPENVIDU\_SECRET=${OPENVIDU\_SECRET}

- CALL\_OPENVIDU\_CERTTYPE=${CERTIFICATE\_TYPE}

- CALL\_PRIVATE\_ACCESS=${CALL\_PRIVATE\_ACCESS:-}

- CALL\_USER=${CALL\_USER:-}

- CALL\_SECRET=${CALL\_SECRET:-}

- CALL\_ADMIN\_SECRET=${CALL\_ADMIN\_SECRET:-}

- CALL\_RECORDING=${CALL\_RECORDING:-}

logging:

options:

max-size: "${DOCKER\_LOGS\_MAX\_SIZE:-100M}"

8. Main-node EC2 UFW

• **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 Inspector is served by default in standard https port.

• **3478 TCP+UDP**: used by STUN/TURN server to resolve clients IPs

• **5044 TCP**: Necessary for Media Nodes instances to send metrics to OpenVidu. **WARNING!!** This port must be closed to the Internet and **must only be accessible for your Media Nodes**, or anyone could send metrics information to OpenVidu.

• **9200 TCP** Necessary for Media Nodes Instances to send metrics and logs to ElasticSearch. **WARNING!!** This port must be closed to the Internet and **must only be accessible for your Media Nodes**, or anyone could do http requests to your ElasticSearch.

• **40000 - 65535 TCP+UDP**: used by TURN server to establish relayed media connections.

9. Media-node UFW

• **22 TCP**: to connect using SSH to admin OpenVidu.

• **443 TCP+UDP**: used by STUN/TURN server to resolve clients IPs when OPENVIDU\_PRO\_COTURN\_IN\_MEDIA\_NODES=true.

• **40000 - 65535 TCP+UDP**: used by Kurento Media Server to establish media connections.

• **8888 TCP**: Kurento Media Server handler listens on port 8888. **WARNING!!** This port must be closed to the Internet and **must only be accessible for your Master Node**, or anyone could spy your sessions.

• **3000 TCP**: All *Media Nodes* offer a REST API endpoint to provision their services. **WARNING!!** This port must be closed to the Internet and **must only be accessible for your Master Node**, or anyone could spy your sessions.

• **4000 TCP (Optional)** : This port is only needed if you want to use the [Speech to text](https://docs.openvidu.io/advanced-features/speech-to-text/) functionality. **WARNING!!** This port must be closed to the Internet and **must only be accessible for your Master Node**, or anyone could spy your sessions.

10. Nginx-manager









