

설치 안내서

AnyEIMS 1.0

TMAXSOFT

저작권 공지

Copyright 2024. TmaxSoft Co., Ltd. All Rights Reserved.

제한된 권리

이 소프트웨어(AnyEIMS™) 사용설명서의 내용과 프로그램은 저작권법과 국제 조약에 의해서 보호받고 있습니다. 사용설명서의 내용과 여기에 설명된 프로그램은 TmaxSoft Co., Ltd.와의 사용권 계약하에서만 사용이 가능하며, 사용설명서는 사용권 계약의 범위 내에서만 배포 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 TmaxSoft의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2차적 저작물작성 등의 행위를 하여서는 안 됩니다.

이 소프트웨어 사용설명서와 프로그램의 사용권 계약은 어떠한 경우에도 사용설명서 및 프로그램과 관련된 지적재산권(등록 여부를 불문)을 양도하는 것으로 해석되지 아니하며, 브랜드나 로고, 상표 등을 사용할 권한을 부여하지 않습니다. 사용설명서는 오로지 정보의 제공만을 목적으로 하고, 이로 인한 계약상의 직접적 또는 간접적 책임을 지지 아니하며, 사용설명서 상의 내용은 법적 또는 상업적인 특정한 조건을 만족시키는 것을 보장하지는 않습니다. 사용설명서의 내용은 제품의 업그레이드나 수정에 따라 그 내용이 예고 없이 변경될 수 있으며, 내용상의 오류가 없음을 보장하지 아니합니다.

상표 공지

AnyEIMS™는 TmaxSoft Co., Ltd.의 상표입니다. 본 사용설명서에 기재된 모든 제품과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용되며 반드시 상표 표시 (™, ®)를 하지는 않습니다

오픈소스 소프트웨어 공지

본 제품의 일부 파일 또는 모듈은 다음의 라이선스를 준수합니다. : Apache-2.0, BSD-2-Clause, CDDL-1.0, EPL-2.0, GPL2 w/ CPE, LGPL-2.1, MIT

안내서 이력

제품 버전	안내서 버전	발행일	비고
AnyEIMS 1.0	3.1.1	2024-05-24	-

목차

1. 설치 전 준비사항	3
1.1. 개요	3
1.2. 시스템 요구사항	3
2. 로컬 환경에서 설치	4
2.1. 설치 순서	4
2.2. 설치 파일 준비	4
2.3. 설치 파일 구성	5
2.3.1. application-extensions.yml 파일 설정	5
2.3.2. nginx.conf 파일 설정	6
2.4. 데이터베이스 설정	6
2.5. 환경 변수 설정	7
2.6. 바이너리 실행	7
2.6.1. 백엔드 서버	7
2.6.2. 프론트엔드 서버	7
3. 쿠버네티스 환경에서 설치	9
3.1. 설치 순서	9
3.2. 설치 파일 준비	9
3.3. YAML 파일 설정	10
3.3.1. namespace.yaml	10
3.3.2. redis.yaml	11
3.3.3. be-deployment.yaml	12
3.3.4. be-service.yaml	13
3.3.5. fe-deployment.yaml	13
3.3.6. fe-service.yaml	14
3.4. 데이터베이스 설정	14
3.5. 이미지 로드	15
3.6. 볼륨 설정	15
3.6.1. 백엔드 서버	15
3.6.2. 프론트엔드 서버	16
3.7. YAML 파일 실행	17

1. 설치 전 준비사항

1.1. 개요

AnyEIMS는 로컬 또는 쿠버네티스 환경에서 설치가 가능합니다.

다음은 AnyEIMS를 설치하기 앞서 준비해야 할 사항입니다.

구분	준비사항
공통	AnyEIMS 설치를 위한 충분한 시스템 공간 확보 JDK 1.8 설치 Oracle 12c R2 설치
로컬 환경	NGINX 1.24.0 설치 Redis 7.2.4 설치
쿠버네티스 환경	쿠버네티스 설치 및 환경 구성

1.2. 시스템 요구사항

AnyEIMS를 설치하기 위해 필요한 소프트웨어와 하드웨어는 다음과 같습니다. (최소 사양)

플랫폼	하드웨어	소프트웨어
linux (Ubuntu 22.04)	1.5GB 이상의 하드디스크 여유 공간	JDK 1.8 이상 Oracle(12c R2 이상) Nginx (1.24.0 이상) Redis (7.2.4)

2. 로컬 환경에서 설치

2.1. 설치 순서

로컬 환경에서 AnyEIMS 설치하는 다음과 같은 과정으로 진행됩니다.

1. 설치 파일 준비
2. 설치 파일 구성
3. 데이터베이스 설정
4. 환경 변수 설정
5. 바이너리 실행

2.2. 설치 파일 준비

AnyEIMS 설치에 필요한 파일을 준비합니다.

```
|-- libs
  |-- ojdbc8-12.2.0.1.jar ....(1)
|-- nginx-conf
  |-- nginx.conf ....(2)
|-- properties
  |-- application-extensions.yml ....(3)
|-- oracle
  |-- create.sql ....(4)
  |-- insert.sql ....(5)
|-- anyeims_front-<버전>.zip ....(6)
  |-- favicon.ico
  |-- index.html
  +-- assets
|-- anyeims-backend.<버전>.zip ....(7)
  |-- anyeims-backend.jar
```

• 백엔드 서버용 파일

파일	설명
(1) ojdbc8-12.2.0.1.jar	사용할 데이터베이스에 맞는 JDBC 파일입니다.
(3) application-extensions.yml	백엔드 바이너리의 확장 속성을 정의하는 파일입니다.
(7) anyeims-backend.<버전>.zip	백엔드 바이너리 파일을 포함한 ZIP 파일입니다.

• 프론트엔드 서버용 파일

파일	설명
(2) nginx.conf	Nginx 설정을 변경할 수 있는 파일입니다.
(6) anyeims_front-<버전>.zip	프론트엔드 파일입니다.

- 데이터베이스 서버용 파일

파일	설명
(4) create.sql	AnyEIMS에서 사용될 테이블을 생성하는 SQL 파일입니다.
(5) insert.sql	AnyEIMS에서 기본적으로 사용될 값을 넣는 SQL 파일입니다.

2.3. 설치 파일 구성

준비한 설치 파일을 각 서버에 다음과 같이 구성합니다.

- 백엔드 서버

```
/home/eims/  
|-- applications  
|   |-- anyeims-backend.jar  
|-- libs  
|   |-- ojdbc8-12.2.0.1.jar  
+-- local-storage  
|-- properties  
|   |-- application-extensions.yml
```

- 프론트엔드 서버

```
/home/eims/  
|-- nginx  
|   |-- html  
|       |-- anyeims_front-<버전>.zip  
|   |-- conf.d  
|       |-- nginx.conf
```

2.3.1. application-extensions.yml 파일 설정

application-extensions.yml 파일 설정을 통해 백엔드 바이너리의 확장 속성을 정의할 수 있습니다. 만약 최초 실행 시 미리 값을 설정하려면 **sample:init**을 **true**로 설정하고, 데이터를 입력합니다. 그 이후에는 **false**로 설정 변경이 필요합니다.

```
sample:  
  init: false # UI나 API로 설정할 수 없는 값을 넣으려면 true로 변경  
  
spring:  
  datasource:  
    driver-class-name: oracle.jdbc.driver.OracleDriver  
    url: jdbc:oracle:thin:@<DB의 IP 주소>:<포트 번호>/<DB 이름>  
    username: <DB 사용자 이름>  
    password: <DB 사용자 비밀번호>  
  jpa:  
    defer-datasource-initialization: true  
    show-sql: true
```

```

database-platform: org.hibernate.dialect.Oracle12cDialect
hibernate:
  ddl-auto: none
redis:
  host: localhost
  port: 6379

# 로그 설정
logging:
  file:
    name: /home/eims/log/eims.log # 로그 파일의 이름 및 경로
    max-size: 500MB
    max-history: 10
  level:
    org : info

```

2.3.2. nginx.conf 파일 설정

nginx.conf 파일 설정을 통해 Nginx의 속성을 정의할 수 있습니다.

```

server {
  listen 80; # 프론트 포트 설정 가능
  client_max_body_size 20M;

  location /v1 {
    proxy_pass http://<백엔드 접근 IP 주소>:<백엔드 접근 포트 번호>;
    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;
  }

  location / {
    root /var/www/html; # 연결할 Root 경로
    index index.html index.htm;
    try_files $uri $uri/ /index.html;
  }

  error_page 500 502 503 504 /50x.html;
}

```

2.4. 데이터베이스 설정

데이터베이스 서버에서 AnyEIMS이 사용할 데이터베이스를 설정하는 SQL문을 실행합니다.

이때 데이터베이스의 ID, PW는 application-extension.yml 파일에 작성되어 있는 값과 일치해야 합니다.

- AnyEIMS에서 사용될 테이블을 생성하는 SQL

```
$ create.sql
```

- AnyEIMS에서 기본적으로 사용될 값을 넣는 SQL

```
$ insert.sql
```

2.5. 환경 변수 설정

백엔드 서버에서 터미널 실행 후 환경 변수를 통해 Redis 접근 정보를 설정합니다.

```
ANYEIMS_REDIS_HOST: <Redis에 접근 가능한 호스트명 또는 IP 주소>  
ANYEIMS_REDIS_PORT: <Redis 접근 포트 번호>
```



Redis 접근 정보는 application-extensions.yml 파일에서도 설정이 가능합니다.

2.6. 바이너리 실행

2.6.1. 백엔드 서버

백엔드 서버에서 아래의 명령을 실행합니다.

- 사용법

```
$ java -Dspring.profiles.active=${ANYEIMS_SPRING_PROFILE:prod}  
-Dspring.config.import=file:<application-extensions.yml 파일 위치>/application-extensions.yml -cp  
<anyeims-backend.jar 파일 위치>/anyeims-backend.jar:<ojdbc 파일 위치>/  
org.springframework.boot.loader.JarLauncher --server.port=<설정할 백엔드 포트>
```

- 예시

```
$ java -Dspring.profiles.active=${ANYEIMS_SPRING_PROFILE:prod}  
-Dspring.config.import=file:/home/eims/properties/application-extensions.yml -cp  
/home/eims/applications/anyeims-backend.jar:/home/eims/libs/  
org.springframework.boot.loader.JarLauncher --server.port=8081
```

2.6.2. 프론트엔드 서버

프론트엔드 서버에서 anyeims_front-<버전>.zip 파일을 /var/www/html 하위에 압축 해제합니다. 압축 해제 후 **ls** 명령을 실행하여 아래와 같은 결과가 나오는지 확인합니다.

```
$ ls /var/www/html  
assets favicon.ico index.html
```

또한 /etc/nginx/nginx.conf 파일 설정을 통해 /home/eims/nginx/conf.d 디렉터리를 참조하도록 설정해야 합니다.

...


```
http {  
    ...  
    include /home/eims/nginx/conf.d/*.conf; # 참조할 위치 설정  
}
```

설정이 완료되면 Nginx를 실행합니다.

```
$ nginx -g "daemon off;"
```

3. 쿠버네티스 환경에서 설치

3.1. 설치 순서

쿠버네티스 환경에서 AnyEIMS 설치에 필요한 과정은 다음과 같습니다.

1. 설치 파일 준비
2. YAML 파일 설정
3. 데이터베이스 설정
4. 이미지 로드
5. 볼륨 설정
6. YAML 파일 실행

3.2. 설치 파일 준비

AnyEIMS 설치에 필요한 파일을 준비합니다.

- 필수 파일

기동을 위한 필수 파일입니다.

```
|-- images
  |-- anyeims_backend.<버전>.tar ....(1)
  |-- anyeims_frontend.<버전>.tar ....(2)
  |-- redis.latest.tar
|-- yamls
  |-- be-deployment.yaml ....(3)
  |-- be-service.yaml ....(4)
  |-- fe-deployment.yaml ....(5)
  |-- fe-service.yaml ....(6)
  |-- redis.yaml ....(7)
|-- sqls
  |-- oracle
    |-- create.sql ....(8)
    |-- insert.sql ....(9)
```

파일	설명
(1) anyeims_backend.<버전>.tar	쿠버네티스 환경 위에서 컨테이너를 실행하기 위한 이미지 파일입니다.
(2) anyeims_frontend.<버전>.tar	쿠버네티스 환경 위에서 컨테이너를 실행하기 위한 이미지 파일입니다.
(3) be-deployment.yaml	백엔드의 디플로이먼트를 생성하기 위한 명령문이 작성된 파일입니다.
(4) be-service.yaml	백엔드의 서비스를 생성하기 위한 명령문이 작성된 파일입니다.
(5) fe-deployment.yaml	프론트엔드 바이너리, Nginx 설정 파일이 담긴 PV를 설정할 수 있는 파일입니다.

파일	설명
(6) fe-service.yaml	프론트엔드의 서비스를 생성하기 위한 명령문이 작성된 파일입니다.
(7) redis.yaml	Redis의 디플로이먼트와 서비스를 생성하기 위한 파일입니다.
(8) create.sql	AnyEIMS에서 사용될 테이블을 생성하는 SQL 파일입니다.
(9) insert.sql	AnyEIMS에서 기본적으로 사용될 값을 넣는 SQL 파일입니다.

• 추가 설정 파일

쿠버네티스 환경에 올려놓은 프로그램에서 추가 설정으로 인해 파드를 재기동하여도 그대로 유지되어야 하는 파일들을 외부 디렉터리에 따로 관리합니다. 해당 디렉터리를 Persistent Volume(PV)이라고 합니다.

```
|-- libs
|   |-- ojdbc8-12.2.0.1.jar ....(1)
|-- nginx-conf
|   |-- nginx.conf ....(2)
|-- properties
|   |-- application-extensions.yml ....(3)
|-- anyeims_front-<버전>.zip ....(4)
|   |-- favicon.ico
|   |-- index.html
|   +-- assets
|-- anyeims-backend.<버전>.zip ....(5)
|   |-- anyeims-backend.jar
```

◦ 백엔드 서버용 파일

파일	설명
(1) ojdbc8-12.2.0.1.jar	사용할 데이터베이스에 맞는 JDBC 파일입니다.
(3) application-extensions.yml	백엔드 바이너리의 확장 속성을 정의하는 파일입니다.
(5) anyeims-backend.<버전>.zip	백엔드 바이너리 파일을 포함한 ZIP 파일입니다.

◦ 프론트엔드 서버용 파일

파일	설명
(2) nginx.conf	Nginx 설정을 변경할 수 있는 파일입니다.
(4) anyeims_front-<버전>.zip	프론트엔드 파일입니다.

3.3. YAML 파일 설정

쿠버네티스 명령어를 실행하기 위한 파일입니다. 네임스페이스 생성, 서비스 생성, 컨테이너 생성을 하기 위해 만든 명령어 파일입니다.

3.3.1. namespace.yaml

작업할 네임스페이스를 생성하는 yaml 파일입니다.

```
apiVersion: v1
kind: Namespace
metadata:
  name: eims
```

3.3.2. redis.yaml

Redis의 디플로이먼트와 서비스를 생성하기 위한 파일입니다.

• 서비스

Redis의 서비스를 생성하기 위한 명령문이 작성되어 있습니다. 설치할 네임스페이스, 개방할 포트, 이용할 외부 IP를 설정할 수 있습니다.

```
apiVersion: v1
kind: Service
metadata:
  labels:
    name: anyeims-redis
  name: anyeims-redis
  namespace: <작업할 네임스페이스>
spec:
  type: ClusterIP
  ports:
    - name: "http"
      port: <통신할 포트 번호>
      targetPort: 6379
  externalIPs:
    - <통신할 IP 주소> # 외부에 노출되는 IP
  selector:
    name: anyeims-redis
```

• 디플로이먼트

Redis의 디플로이먼트를 생성하기 위한 명령문이 작성되어 있습니다. 설치할 네임스페이스, 생성에 이용할 이미지를 설정할 수 있습니다. 이때 이미지의 경우 로드한 이미지 이름에 맞춰 변경합니다.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: anyeims-redis
  namespace: <작업할 네임스페이스>
  labels:
    name: anyeims-redis
spec:
  replicas: 1
  selector:
    matchLabels:
      name: anyeims-redis
  strategy:
    type: Recreate
  template:
    metadata:
```

```

labels:
  name: anyeims-redis
spec:
  containers:
    - image: any-eims:5000/anyeims_redis:7.2.4 # 사용할 이미지
      name: anyeims-reids
      ports:
        - containerPort: 6379
          protocol: TCP
      restartPolicy: Always

```

3.3.3. be-deployment.yaml

백엔드 바이너리, 설정 파일이 담긴 PV를 설정할 수 있습니다. 내부에서 외부와 연동하고 싶은 mountPath를 정의하고, 해당 부분을 volumes에서 로컬 머신의 물리 디렉터리와 연동합니다.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: anyeims-backend
  namespace: <작업할 네임스페이스>
  labels:
    name: anyeims-backend
spec:
  replicas: 1
  selector:
    matchLabels:
      name: anyeims-backend
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        name: anyeims-backend
    spec:
      containers:
        - env:
            - name: ANYEIMS_REDIS_HOST
              value: <Redis IP 주소> # redis.yaml에서 설정한 IP
            - name: ANYEIMS_REDIS_PORT
              value: <Redis 포트 번호> # redis.yaml에서 설정한 포트
          image: any-eims:5000/anyeims_backend:1.0.0-r1 # 사용할 이미지
          name: anyeims-backend
          ports:
            - containerPort: 8080
              protocol: TCP
          volumeMounts:
            - mountPath: /home/eims
              name: anyeims-backend

      restartPolicy: Always
      volumes:
        - name: anyeims-backend
          # 아래 내용을 알맞은 PVC로 변경 필요
          hostPath:
            type: Directory

```

```
path: /home/eims/
```

3.3.4. be-service.yaml

백엔드의 서비스를 생성하기 위한 명령문이 작성되어 있습니다. 설치할 네임스페이스, 개방할 포트, 이용할 외부 IP를 설정할 수 있습니다.

```
apiVersion: v1
kind: Service
metadata:
  labels:
    name: anyeims-backend
  name: anyeims-backend
  namespace: <작업할 네임스페이스>
spec:
  type: ClusterIP
  ports:
    - name: "http"
      port: <통신할 포트 번호>
      targetPort: 8080
  externalIPs:
    - <통신할 IP 주소> # 외부에 노출되는 IP
  selector:
    name: anyeims-backend
```

3.3.5. fe-deployment.yaml

프론트엔드 바이너리, Nginx 설정 파일이 담긴 PV를 설정할 수 있습니다. 내부에서 외부와 연동하고 싶은 mountPath를 정의하고, 해당 부분을 volumes에서 로컬 머신의 물리 디렉터리와 연동합니다.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    name: anyeims-frontend
  name: anyeims-frontend
  namespace: <작업할 네임스페이스>
spec:
  replicas: 1
  selector:
    matchLabels:
      name: anyeims-frontend
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        name: anyeims-frontend
    spec:
      containers:
        - image: any-eims:5000/anyeims_frontend:1.0.0-r1 # 사용할 이미지
          name: anyeims-frontend
          ports:
```

```

- containerPort: 80
  protocol: TCP
volumeMounts:
- mountPath: /home/eims/nginx
  name: anyeims-frontend
restartPolicy: Always
volumes:
- name: anyeims-frontend
  # 아래 내용을 알맞은 PVC로 변경 필요
  hostPath:
    type: Directory
    path: /home/eims/nginx

```

3.3.6. fe-service.yaml

프론트엔드의 서비스를 생성하기 위한 명령문이 작성되어 있습니다. 설치할 네임스페이스, 개방할 포트, 이용할 외부 IP를 설정할 수 있습니다.

```

apiVersion: v1
kind: Service
metadata:
  labels:
    name: anyeims-frontend
  name: anyeims-frontend
  namespace: <작업할 네임스페이스>
spec:
  type: ClusterIP
  ports:
    - name: "http"
      port: <통신할 포트 번호>
      targetPort: 80
  externalIPs:
    - <통신할 IP 주소> # 외부에 노출되는 IP
  selector:
    name: anyeims-frontend

```

3.4. 데이터베이스 설정

데이터베이스 서버에서 AnyEIMS이 사용할 데이터베이스를 설정하는 SQL문을 실행합니다.

이때 데이터베이스의 ID, PW는 application-extension.yml 파일에 작성되어 있는 값과 일치해야 합니다.

- AnyEIMS에서 사용될 테이블을 생성하는 SQL

```
$ create.sql
```

- AnyEIMS에서 기본적으로 사용될 값을 넣는 SQL

```
$ insert.sql
```

3.5. 이미지 로드

워크 노드의 로컬 리포지터리에 이미지를 로드합니다.

- 컨테이너 런타임이 DOCKER인 경우

```
$ docker load -i <파일명>.tar
```

- 컨테이너 런타임이 CRI-O인 경우

```
$ podman load -u <파일명>.tar
```



워크 노드의 리포지터리에서 이미지를 로드하는 이유는 파드가 스케줄링된 노드의 리포지터리에서 이미지를 불러와서 사용하기 때문입니다.

3.6. 볼륨 설정

워크 노드의 디렉터리에 파드의 볼륨을 마운트합니다.



본 안내서에서는 호스트의 특정 디렉터리를 스토리지로 사용하는 방식인 hostPath 방식을 사용합니다.

3.6.1. 백엔드 서버

백엔드 이미지 파일(anyeims_backend.<버전>.tar)을 사용하여 컨테이너 실행 시 컨테이너 디렉터리 구조는 아래와 같습니다.

```
/home/eims/  
|-- applications  
    |-- anyeims_backend.jar  
|-- libs  
    |-- ojdbc8-12.2.0.1.jar  
|-- local-storage  
|-- properties  
    |-- application-extensions.yml
```

파드의 볼륨을 마운트 하기 위해 be_deployment.yaml 파일에 설정한 호스트 경로(hostPath)로 워크 노드에 관련 파일 및 디렉터리를 구성합니다.

- **anyeims_backend.jar**

{호스트 경로}/applications' 하위에 위치시킵니다.

- **ojdbc8-12.2.0.1.jar**

{호스트 경로}/libs' 하위에 위치시킵니다.

- **local-storage**

{호스트 경로} 하위에 디렉터리만 생성합니다. 해당 디렉터리에는 사용자가 결제 시 올리는 파일 같은 데이터가 저장됩니다.

- **application-extensions.yml**

{호스트 경로}/properties 하위에 위치시킵니다.

application-extensions.yml 파일 설정을 통해 백엔드 바이너리의 확장 속성을 정의할 수 있습니다. 만약 최초 실행 시 미리 값을 설정하려면 **sample:init**을 **true**로 설정하고, 데이터를 입력합니다. 그 이후에는 **false**로 설정 변경이 필요합니다.

```
sample:
  init: false # UI나 API로 설정할 수 없는 값을 넣으려면 true로 변경

spring:
  datasource:
    driver-class-name: oracle.jdbc.driver.OracleDriver
    url: jdbc:oracle:thin:@<DB의 IP 주소>:<포트 번호>/<DB 이름>
    username: <DB 사용자 이름>
    password: <DB 사용자 비밀번호>
  jpa:
    defer-datasource-initialization: true
    show-sql: true
    database-platform: org.hibernate.dialect.Oracle12cDialect
    hibernate:
      ddl-auto: none

# 로그 설정
logging:
  file:
    name: /home/eims/log/eims.log # 로그 파일의 이름 및 경로
    max-size: 500MB
    max-history: 10
  level:
    org : info
```

3.6.2. 프론트엔드 서버

프론트엔드 이미지 파일(anyeims_frontend.<버전>.tar)을 사용하여 컨테이너 실행 시 컨테이너 디렉터리 구조는 아래와 같습니다.

```
/home/eims/
|-- nginx
  |-- html
    |-- assets
    |-- favicon.ico
    |-- index.html
  |-- conf.d
    |-- nginx.conf
```

파드의 볼륨을 마운트 하기 위해 fe-deployment.yaml 파일에 설정한 호스트 경로(hostPath)로 워크 노드에 관련 파일 및 디렉터리를 구성합니다.

- **anyeims_front-<버전명>.zip**

{호스트 경로}/html' 하위에 해당 파일의 압축을 해제합니다.

- **nginx.conf**

{호스트 경로}/conf.d' 하위에 위치시킵니다.

nginx.conf 파일 설정을 통해 Nginx의 속성을 정의할 수 있습니다. 이때 proxy_pass 값을 be-deployment.yaml에서 설정한 'externalIP:port'로 설정합니다. 백엔드의 IP/Port가 변경될 때 수정해야 합니다.

```
server {
  listen 80; # 프론트 포트 설정 가능
  client_max_body_size 20M;

  location /v1 {
    proxy_pass http://<백엔드 접근 IP 주소>:<백엔드 접근 포트 번호>;
    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;
  }

  location / {
    root /home/eims/nginx/html; # 연결할 Root 경로
    index index.html index.htm;
    try_files $uri $uri/ /index.html;
  }

  error_page 500 502 503 504 /50x.html;
}
```

3.7. YAML 파일 실행

설정된 YAML 파일을 차례대로 실행합니다.

1. 네임스페이스 생성

```
$ kubectl apply -f namespace.yaml
```

2. Redis 생성

```
$ kubectl apply -f redis.yaml -n eims
```

3. 백엔드 디플로이먼트 생성

```
$ kubectl apply -f be-deployment.yaml -n eims
```

4. 백엔드 서비스 생성

```
$ kubectl apply -f be-service.yaml -n eims
```

5. 프론트엔드 디플로이먼트 생성

```
$ kubectl apply -f fe-deployment.yaml -n eims
```

6. 프론트엔드 서비스 생성

```
$ kubectl apply -f fe-service.yaml -n eims
```