설치 안내서

AnyEIMS 1.0

TMAXSOFT

저작권 공지

회사 정보

(주)티맥스소프트

주소 : 경기도 성남시 분당구 황새울로258번길 29, 티맥스수내타워 8-9층

기술 서비스 센터: 1544-8629

홈페이지: https://www.tmaxsoft.com

제한된 권리

상표 공지

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

유지 보수

구분	지원항목	서비스 내용
제품지원	패치 & 업그레이드	무상 패치 서비스 제공 메이저 버전 업그레이드 시 할인 혜택 웹 지원을 통한 패치 내역 제공

구분	지원항목	서비스 내용
기술 지원 - 기본 서비스	장애 지원	장애 발생 시 원인 분석 및 조치 Service Desk팀 → 기술팀 → R&D의 3단계 장애 분석 및 조치
	일상 지원(온라인 지원)	E-mail, 전화, 원격, 웹 사이트 등 온라인 자원을 통한 질의 응답 서비스
	고객 맞춤 지원(방문 지원)	고객의 요청으로 수행하는 방문 지원 서비스
기술 지원 - 옵션 서비스	예방 지원	정기 점검을 통한 시스템 운영현황 보고 및 장애 예방 ◦ 관리자 또는 운영자의 요구사항 수렴 ◦ 운영 현황(시스템, 엔진 운영) 보고서 제공 ◦ 필요 시 시스템 개선 권장 사항 보고
유지 보수 비용 및 기간	계약 시 별도 협의	계약 시 EOL/EOS 문서 제공

안내서 이력

제품 버전	안내서 버전	발행일	비고
AnyEIMS 1.0	3.1.2	2024-12-17	현행화(Ver 1.0.13)
AnyEIMS 1.0	3.1.1	2024-05-24	-

목차

1. 설치 전 준비사항

1.1. 개요

AnyEIMS 설치 전에 제품의 구성을 확인합니다.

구분	항목

설치 파일
(로컬 환경)
	<pre>|-- libs
	    |-- ojdbc8-12.2.0.1.jar
	    |-- AnyEIMS-stream-wrapper-<버전>.jar
	    |-- AnyEIMS-export-model-<버전>.jar
	    |-- AnyEIMS-custom-notification-<버전>.jar
	    |-- AnyEIMS-custom-action-<버전>.jar
	    |-- AnyEIMS-custom-deployment-notification-<버전>.jar
	    |-- AnyEIMS-encrypt-<버전>.jar
	    |-- AnyEIMS-adapter-lib-<버전>.jar
	|-- nginx-conf
	    |-- nginx.conf
	|-- properties
	    |-- application-extensions.yml
	|-- oracle
	    |-- create.sql
	    |-- insert.sql
	|-- anyeims_front-<버전>.zip
	    |-- favicon.ico
	    |-- index.html
	    +-- assets
	|-- anyeims-backend.<버전>.zip
	    |-- anyeims-backend.jar</pre>

설치 파일
(쿠버네티스 환경)
	<pre>|-- images
	    |-- anyeims_backend.<버전>.tar
	    |-- anyeims_frontend.<버전>.tar
	    |-- redis.latest.tar
	|-- yamls
	    |-- be-deployment.yaml
	    |-- be-service.yaml
	    |-- fe-deployment.yaml
	    |-- fe-service.yaml
	    |-- redis.yaml
	|-- sqls
	    |-- oracle
	        |-- create.sql
	        |-- insert.sql</pre>

매뉴얼	설치 안내서 (AnyEIMS_1.0_Installation-Guide.pdf) 사용자 안내서 (AnyEIMS_1.0_User-Guide.pdf)

1.2. 설치 전 준비사항

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

1.3. 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 설치는 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.

본 안내서에서 설명하는 설치 파일은 기본적으로 제공되는 파일입니다.

```
|-- libs
    |-- ojdbc8-12.2.0.1.jar  ....(1)
    |-- AnyEIMS-stream-wrapper-<버전>.jar  ....(2)
    |-- AnyEIMS-export-model-<버전>.jar  ....(3)
    |-- AnyEIMS-custom-notification-<버전>.jar  ....(4)
    |-- AnyEIMS-custom-action-<버전>.jar  ....(5)
    |-- AnyEIMS-custom-deployment-notification-<버전>.jar  ....(6)
    |-- AnyEIMS-encrypt-<버전>.jar  ....(7)
    |-- AnyEIMS-adapter-lib-<버전>.jar  ....(8)
|-- nginx-conf
    |-- nginx.conf  ....(9)
|-- properties
    |-- application-extensions.yml  ....(10)
|-- oracle
    |-- create.sql  ....(11)
    |-- insert.sql  ....(12)
|-- anyeims_front-<버전>.zip  ....(13)
    |-- favicon.ico
    |-- index.html
    +-- assets
|-- anyeims-backend.<버전>.zip  ....(14)
    |-- anyeims-backend.jar
```

- **백엔드 서버용 파일**

파일	설명
(1) ojdbc8-12.2.0.1.jar	데이터베이스에서 사용할 JDBC 드라이버 파일입니다.

파일	설명
(2) AnyEIMS-stream-wrapper-<버전>.jar	InputStream, OutputStream 조작을 위한 인터페이스를 제공하는 파일입니다.
(3) AnyEIMS-export-model-<버전>.jar	어댑터 및 사용자 정의 기능에 사용되는 모델 객체를 정의한 파일입니다.
(4) AnyEIMS-custom-notification-<버전>.jar	결재 관련 사용자 정의 알림 기능 추가를 위한 파일입니다.
(5) AnyEIMS-custom-action-<버전>.jar	VO 다운로드 및 사용자 정의 기능 추가를 위한 파일입니다.
(6) AnyEIMS-custom-deployment-notification-<버전>.jar	배포 관련 사용자 정의 알림 기능 추가를 위한 파일입니다.
(7) AnyEIMS-encrypt-<버전>.jar	YAML 값 암호화 입력 기능 추가를 위한 파일입니다.
(8) AnyEIMS-adapter-lib-<버전>.jar	어댑터 기능을 수행하는 파일입니다.
(10) application-extensions.yml	백엔드 바이너리의 확장 속성을 정의한 파일입니다.
(14) anyeims-backend.<버전>.zip	백엔드 바이너리 파일을 포함한 ZIP 파일입니다.

- **프론트엔드 서버용 파일**

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

- **데이터베이스 서버용 파일**

파일	설명
(11) create.sql	AnyEIMS에서 사용될 테이블을 생성하는 SQL 파일입니다.
(12) 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
    |-- AnyEIMS-stream-wrapper.jar
    |-- AnyEIMS-export-model.jar
    |-- AnyEIMS-custom-notification.jar
    |-- AnyEIMS-custom-action.jar
    |-- AnyEIMS-custom-deployment-notification.jar
```

```
    |-- AnyEIMS-encrypt.jar
    |-- AnyEIMS-adapter-lib.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
```

DB와 Redis의 비밀번호는 기본적으로 평문으로 저장됩니다. 보안을 위해 AES256 방식으로

암호화할 수 있습니다.

암호화 적용 절차는 다음과 같습니다.

1. AnyEIMS 설치 파일에 포함된 AnyEIMS-encrypt.jar 파일을 **libs** 폴더에 넣습니다.
2. aes-key.bin 파일을 생성하여 아래와 같이 비밀 키를 설정하고, 원하는 위치에 배치합니다.

```
rUj4irjV3WD9JYokcTQ/ag==12312311
```

3. application-extensions.yml 파일의 "password:" 부분을 https://anycript.com/crypto에서 암호화한 값으로 변경하고, 아래 내용을 추가합니다.

```
## 암호화 적용 값
spring:
  datasource:
    password: iRtb0kwvxdWy1w7bPQ+r1g==
  redis:
    password: "3KbQuLF9uSH8cCKe8xOg6Q=="

## AnyEIMS-encrypt.jar 및 aes-keys.bin 파일 적용
eims:
  encrypted-properties: spring.redis.password,spring.datasource.password
  encryption:
    decrypt:
      ## 파일이 위치한 절대 경로 입력
      key: "/home/eims/properties/aes-keys.bin"
      class-name: com.tmax.anyeims.encryption.sample.AES256Decrypt
```

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 아이디, 비밀번호는 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/ngin.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 설치는 다음과 같은 과정으로 진행됩니다.

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.

본 안내서에서 설명하는 설치 파일은 기본적으로 제공되는 파일입니다.

3.2.1. 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.

파일	설명
(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.
(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.

3.2.2. 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)
    |-- AnyEIMS-stream-wrapper-<버전>.jar  ....(2)
    |-- AnyEIMS-export-model-<버전>.jar  ....(3)
    |-- AnyEIMS-custom-notification-<버전>.jar  ....(4)
    |-- AnyEIMS-custom-action-<버전>.jar  ....(5)
    |-- AnyEIMS-custom-deployment-notification-<버전>.jar  ....(6)
    |-- AnyEIMS-encrypt-<버전>.jar  ....(7)
    |-- AnyEIMS-adapter-lib-<버전>.jar  ....(8)
|-- nginx-conf
    |-- nginx.conf  ....(9)
|-- properties
    |-- application-extensions.yml  ....(10)
|-- anyeims_front-<버전>.zip  ....(11)
    |-- favicon.ico
    |-- index.html
    +-- assets
|-- anyeims-backend.<버전>.zip  ....(12)
    |-- anyeims-backend.jar
```

- **백엔드 서버용 파일**

파일	설명
(1) ojdbc8-12.2.0.1.jar	데이터베이스에서 사용할 JDBC 드라이버 파일입니다.

파일	설명
(2) AnyEIMS-stream-wrapper-<버전>.jar	InputStream, OutputStream 조작을 위한 인터페이스를 제공하는 파일입니다.
(3) AnyEIMS-export-model-<버전>.jar	어댑터 및 사용자 정의 기능에 사용되는 모델 객체를 정의한 파일입니다.
(4) AnyEIMS-custom-notification-<버전>.jar	결재 관련 사용자 정의 알림 기능 추가를 위한 파일입니다.
(5) AnyEIMS-custom-action-<버전>.jar	VO 다운로드 및 사용자 정의 기능 추가를 위한 파일입니다.
(6) AnyEIMS-custom-deployment-notification-<버전>.jar	배포 관련 사용자 정의 알림 기능 추가를 위한 파일입니다.
(7) AnyEIMS-encrypt-<버전>.jar	YAML 값 암호화 입력 기능 추가를 위한 파일입니다.
(8) AnyEIMS-adapter-lib-<버전>.jar	어댑터 기능을 수행하는 파일입니다.
(10) application-extensions.yml	백엔드 바이너리의 확장 속성을 정의한 파일입니다.
(12) anyeims-backend.<버전>.zip	백엔드 바이너리 파일을 포함한 ZIP 파일입니다.

- **프론트엔드 서버용 파일**

파일	설명
(9) nginx.conf	Nginx 설정을 변경할 수 있는 파일입니다.
(11) 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 아이디, 비밀번호는 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

워커 노드의 로컬 리포지터리에 이미지를 로드합니다.

- 컨테이너 런타임이 Docker인 경우

```
$ docker load -i <파일명>.tar
```

- 컨테이너 런타임이 CRI-O인 경우

```
$ podman load -u <파일명>.tar
```

워커 노드의 리포지터리에서 이미지를 로드하는 이유는 파드가 스케쥴링된 노드의 리포지터리에서 이미지를 불러와서 사용하기 때문입니다.

3.6. 볼륨 설정

워커 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 결재 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
```

DB와 Redis의 비밀번호는 기본적으로 평문으로 저장됩니다. 보안을 위해 AES256 방식으로 암호화할 수 있습니다.

암호화 적용 절차는 다음과 같습니다.

1. AnyEIMS 설치 파일에 포함된 AnyEIMS-encrypt.jar 파일을 **libs** 폴더에 넣습니다.
2. aes-key.bin 파일을 생성하여 아래와 같이 비밀 키를 설정하고, 원하는 위치에 배치합니다.

```
rUj4irjV3WD9JYokcTQ/ag==12312311
```

3. application-extensions.yml 파일의 "password:" 부분을 https://anycript.com/crypto에서 암호화한 값으로 변경하고, 아래 내용을 추가합니다.

```
## 암호화 적용 값
spring:
  datasource:
    password: iRtb0kwvxdWy1w7bPQ+r1g==
  redis:
    password: "3KbQuLF9uSH8cCKe8xOg6Q=="

## AnyEIMS-encrypt.jar 및 aes-keys.bin 파일 적용
eims:
  encrypted-properties: spring.redis.password,spring.datasource.password
  encryption:
    decrypt:
      ## 파일이 위치한 절대 경로 입력
      key: "/home/eims/properties/aes-keys.bin"
      class-name: com.tmax.anyeims.encryption.sample.AES256Decrypt
```

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 구성힙니다.

- **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

설정한 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
```