



# CO:RE 포팅 매뉴얼



- ♥ 웹 서버
- ♥ 개발 환경
- ♥ 배포 환경 변수
- ♥ Config 파일 목록
- ♥ EC2 사용 포트
- ♥ 아키텍처 구조도
- ♥ DB 설정
- ♥ JIRA 정보 권한 설정

## ♥ 웹 서버

- AWS EC2
- AWS Lightsail Instance 4GB

## ♥ 개발 환경

### Backend

Language	Java Temurin 17.0.13
IDE	IntelliJ Ultimate
Database	MySQL Redis

#### • Framework

Spring Boot	JPA Redis
-------------	--------------

	Web WebFlux Security OAuth2 Actuator Validation
<b>Spring Cloud</b>	OpenFeign Config Bootstrap Eureka

## Frontend

<b>Language</b>	TypeScript
<b>IDE</b>	Visual Studio Code
<b>Framework</b>	React Vite
<b>Library</b>	<b>Zustand</b> <b>React-query</b> <b>Styled-component</b>  Fullcalendar Uuid Axios Recharts Slick-carousel Gitgraph Husky Date-fns

## DevOps

- Nginx
- Jenkins
- Docker

## Collaboration Tools

- GitLab
- Jira
- Notion
- Mattermost
- Discord

## 💙 배포 환경 변수

ID	Kind	Description
remote-server	SSH Username with private key	Remote Server IP & PEM Key
deployServerIP	Secret text	Remote Server IP
mattermost-url	Secret text	Mattermost Webhook URL
GITHUB_PASSWORD	Secret text	Config 저장 Repository Access Token
GITHUB_USERNAME	Secret text	Config 저장 Repository user ID
GITHUB_REPO	Secret text	Config 정보가 저장된 GitHub Repository URL

## 💙 Config 파일 목록



github config repository에 `/config` 경로로 해당 파일들이 존재해야합니다.

- `api.yml`

```
spring:
  datasource:
```

```

url: jdbc:mysql://mysql:3306/integration?serverTimezone=A
username: {MySQL 계정명}
password: {MySQL 비밀번호}
jpa:
  hibernate:
    ddl-auto: update
    show-sql: true
    generate-ddl: true
data:
  redis:
    host: redis
    port: 6379
error:
  printStackTrace: true

chatgpt:
  api:
    token: {ChatGPT API 토큰}

```

- `application.yml`

```
JWT_SECRET_KEY: {JWT secret key}
```

- `backend.properties`

```

MYSQL_URL=jdbc:mysql://mysql:3306/business?serverTimezone=Asia
MYSQL_USERNAME={MySQL 계정명}
MYSQL_PASSWORD={MySQL 비밀번호}

REDIS_HOST=redis
REDIS_PORT=6379
REDIS_PASSWORD={Redis 비밀번호}

GITHUB_CLIENT_ID={Github Client ID}
GITHUB_CLIENT_SECRET={Github Client Secret key}

```

```
GITHUB_REDIRECT_URI=http://localhost:8082/login/oauth2/code/g
```

```
JWT_SECRET_KEY={JWT Secret key}
```

```
JWT_ACCESS_TOKEN_EXPIRED=5000
```

```
JWT_ACCESS_HEADER=Access-Token
```

```
JWT_REFRESH_TOKEN_EXPIRED=5000
```

```
JWT_REFRESH_HEADER=Refresh-Token
```

```
SERVER_PORT=8082
```

```
JIRA_CLIENT={Jira Client Key}
```

```
JIRA_SECRET={Jira Client Secret key}
```

```
JIRA_CALLBACK=https://k11s106.p.ssafy.io/api/callback/oauth2/
```

```
FRONT_URL=https://k11s106.p.ssafy.io/dashboard
```

- `email.yml`

```
spring:
```

```
  autoconfigure:
```

```
    exclude: org.springframework.boot.autoconfigure.jdbc.Data
```

```
  mail:
```

```
    host: smtp.gmail.com
```

```
    port: 587
```

```
    username: {Google SMTP 계정 email}
```

```
    password: {Google App 비밀번호}
```

```
    properties:
```

```
      mail.smtp.debug: true
```

```
      mail.smtp.connectiontimeout: 1000
```

```
      mail.starttls.enable: true
```

```
      mail.smtp.auth: true
```

- `gateway.yml`

```

spring:
  cloud:
    gateway:
      discovery:
        locator:
          enabled: true
          lower-case-service-id: true
      routes:
        - id: api-service-route
          uri: lb://API
          predicates:
            - Path=/github/**
          filters:
            - RewritePath=/github/(?<remaining>.*), /${remain.
        - id: backend-service-route
          uri: lb://BACKEND
          predicates:
            - Path=/**
          metadata:
            response-timeout: 3600000
            connect-timeout: 3600000

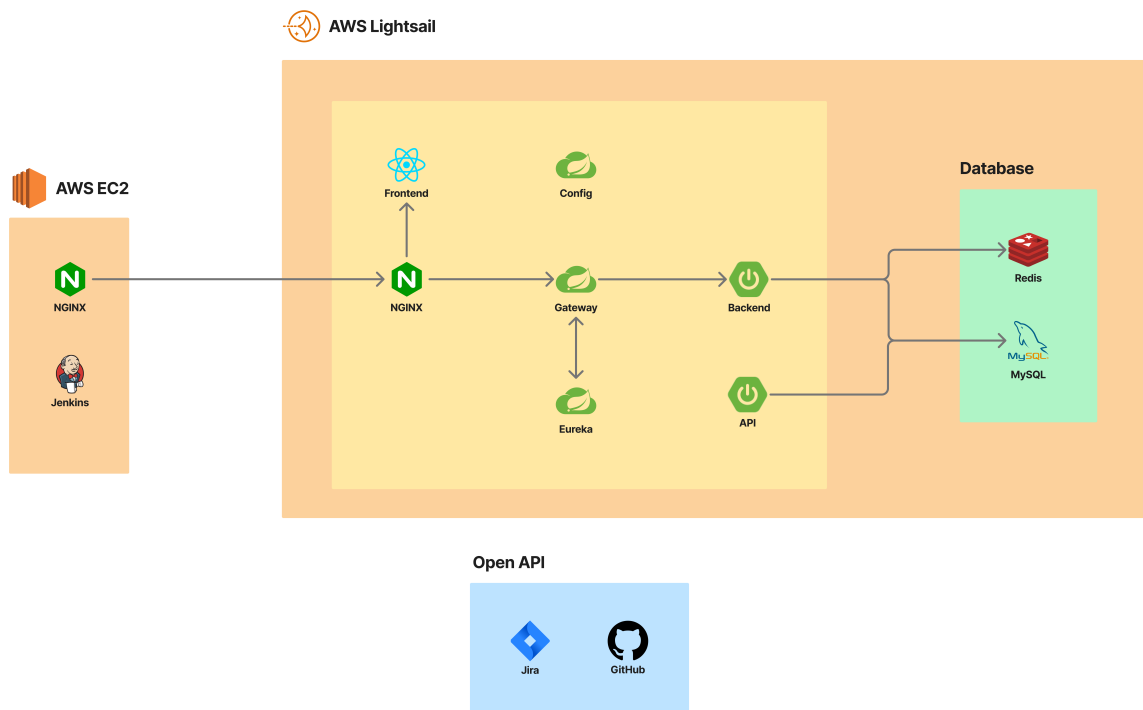
```

## 💙 EC2 사용 포트

80	HTTP 포트
443	HTTPS 포트
8080	Jenkins Docker Container
9092	kafka Docker Container
8082	Backend Service Docker Container
8081	API Service Service Docker Container
3306	MySQL Docker Container
8000	Gateway Docker Container
8083	Email Service Docker Container
8888	Config Server Docker Container

8761	Eureka Server Docker Container
6379	Redis Container

## ♥ 아키텍처 구조도



## ♥ DB 설정

- 하나의 MySQL 컨테이너에서 데이터베이스를 2개 사용합니다.

```
# backend server에서 사용하는 데이터베이스
CREATE DATABASE business;
# api server에서 사용하는 데이터베이스
CREATE DATABASE interation;
```

## ♥ JIRA 정보 권한 설정

- 서비스에서 JIRA의 데이터를 가져오기 위한 권한 설정이 필요합니다.
- [JIRA 내 프로젝트] → [프로젝트 설정] → [사용자]

- Administrators, Administrators (migrated), Developers 권한 설정