ℹ️

# 1. 빌드 및 배포 정보

## 1-1. 개발 환경

### OS

- Windows10
- Ubuntu 20.04 LTS

### FE

- **IDE**
  - `VSCode`
- **Framework(Library)**
  - `React 18.2.0`
  - `Vite 4.4.5`
  - `Firebase 10.6.0`
- **Language**
  - `TypeScript 5.0.2`
- **Style**
  - `MUI 5.14.14`
  - `Tailwind CSS 3.3.3`
- **State Management**
  - `jotai 2.4.3`
- **Data Prefetch**
  - `SWR 2.2.4`
  - `axios 1.5.1`
- **Package Management**
  - pnpm 8.7.4

### BE

- **IDE**
  - `Intellij IDEA`
  - `Pycharm`
- **Framework**
  - `Spring Boot 2.7.16`
  - `gradle 8.3`
  - `Flask 2.3.2`
  - **dependencies**
    - Spring Boot DevTools
    - Lombok
    - Spring Data JPA
    - QueryDSL
    - Spring Web
    - Spring Security
    - Spring Cloud Gateway
    - Redis
    - Swagger 3.0.0
    - JWT 0.11.1
    - Java Mail Sender 1.6.2
- **Language**
  - `Java JDK`
    - zulu-11
- **DB**
  - `MariaDB`

### Infra

- AWS EC2
- Docker
- Nginx

### CI/CD

- Jenkins

## 1-2. 빌드 시 사용되는 환경 변수

## 소스코드 실행 방법

### FrontEnd

1. Node.js 사이트 접속해서 LTS 설치
2. 레포지토리에서 소스코드 다운로드

```
>> git clone https://lab.ssafy.com/s09-final/S09P31A402.git
```

3. frontend/tori-story 경로에 아래 파일 붙여 넣기

▼ .env.properties

```
VITE_APP_KAKAO_MAP_API = 85b60f5420531bd71dfe9dd4ee5508a1
SENTRY_AUTH_TOKEN=sntrys_eyJpYXQiOjE2OTg4MTQ5MzEuMjAzNDQ1LCJ1cmwiOiJodHRwczovL3NlbnRyeS5pbyIsInJlZ2lvbl91cmwiOiJodHRwczovL3VzL
# firebase
VITE_APP_FIREBASE_API_KEY=AIzaSyDH8Pr9fhrYLmMwQFtv9ADfk8eyDax_WMw
VITE_APP_FIREABASE_AUTH_DOMAIN=tori-story.firebaseapp.com
VITE_APP_PROJECT_ID=tori-story
VITE_APP_FIREBASE_STORAGE_BUCKET=tori-story.appspot.com
VITE_APP_FIREBASE_MESSAGING_SENDER_ID=620178363335
VITE_APP_FIREBASE_APP_ID=1:620178363335:web:a6cb34f7e2becb909a7fe2
VITE_APP_FIREBASE_MEASUREMENT_ID=G-BXHR5CREZJ
VITE_APP_FIREBASE_VAPKEY=BMZZuCYvRhxy_keXRA2e9ViPWbdKgJFB0IDPNTtgVSRFJNLbf2-fNBR0vktpVQZ-0eLJMBsLBypKFa0iZfa3czU
```

4. 프론트엔드 폴더로 이동 및 Node.js와 npm 버전 확인

```
>> cd S09P31A402/frontend/tori-story
>> node -v
v18.17.1
>> pnpm -v
8.7.4
```

5. 필요한 라이브러리 설치

```
>> pnpm i
```

6. 실행

```
>> pnpm start
```

### Backend

1. 버전에 맞는 자바(JDK 11) 다운로드
2. backend/auth/src/main/resources 경로에 아래 파일 붙여 넣기

▼ application-develop.yml

```
spring:
  config:
    activate:
      on-profile: develop
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    hikari:
      username: tori
      password: toristory
    url: jdbc:mariadb://k9a402.p.ssafy.io:3324/authdb?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&zero
DateTimeBehavior=convertToNull&rewriteBatchedStatements=true
  redis:
    host: k9a402.p.ssafy.io
    port: 6379
    password: 'ssafy'
  mail:
    host: smtp.gmail.com
    port: 587
```

```
    username: toooristooory
    password: fgkoewgrbjedrumc
    properties:
      mail:
        debug: true
        smtp:
          ssl:
            enable: false
          auth: true
          starttls:
            enable: true
            required: true
          connectiontimeout: 1800000
          timeout: 1800000
          writetimeout: 1800000
    auth-code-expiration-millis: 30000000  # 500분
    pw-code-expiration-millis: 300000  # 5분
jwt:
  secret: e61ef2b9454dd45a86d7235aade23a932205f12cc8bf4720778be1bd5c28847f99c2c9dc173d7f2af90b9efc0de5ffb05ca8f6ab774de3d98
0e4db8048b8d221
  accessTokenValidity: 1800000 # 30분
  refreshTokenValidity: 1.21e+9 # 14일
  cookieName: refreshToken
refreshToken:
  path: /api/member/refresh
profile:
  defaultImgUrl: https://tori-bucket.s3.ap-northeast-2.amazonaws.com/%ED%8E%AD%EA%B7%84/%EA%B8%B0%EB%B3%B8%EB%8B%A4%EB%9E%8
C%EC%A5%90.png
  defaultId: 1
```

3. backend/challenge/src/main/resources 경로에 아래 파일 붙여 넣기

▼ application-develop.yml

```
spring:
  config:
    activate:
      on-profile: develop
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    hikari:
      username: tori
      password: toristory
    url: jdbc:mariadb://k9a402.p.ssafy.io:3324/businessdb?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&
zeroDateTimeBehavior=convertToNull&rewriteBatchedStatements=true
  redis:
    host: k9a402.p.ssafy.io
    port: 6379
    password: 'ssafy'
cloud:
  aws:
    s3:
      bucket: tori-bucket
    region:
      static: ap-northeast-2
      auto: false
    stack:
      auto: false
    credentials:
      access-key: AKIAZKNGOTGT5UHWISHL
      secret-key: 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
flask:
  server:
    url: http://tori-story.com:8204
```

4. backend/tori/src/main/resources 경로에 아래 파일 붙여 넣기

▼ application-develop.yml

```
spring:
  config:
    activate:
      on-profile: develop
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    hikari:
      username: tori
      password: toristory
    url: jdbc:mariadb://k9a402.p.ssafy.io:3324/businessdb?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&
```

```
zeroDateTimeBehavior=convertToNull&rewriteBatchedStatements=true
  redis:
    host: k9a402.p.ssafy.io
    port: 6379
    password: 'ssafy'
```

5. backend/notification/src/main/resources 경로에 아래 파일 붙여 넣기

▼ application-develop.yml

```
spring:
  config:
    activate:
      on-profile: develop
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    hikari:
      username: tori
      password: toristory
    url: jdbc:mariadb://k9a402.p.ssafy.io:3324/businessdb?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&
zeroDateTimeBehavior=convertToNull&rewriteBatchedStatements=true

  redis:
    host: k9a402.p.ssafy.io
    port: 6379
    password: 'ssafy'

fcm:
  project-id: tori-story
  api-url: https://fcm.googleapis.com/v1/projects/tori-story/messages:send
  firebase-config-path: firebase/tori-story-firebase-adminsdk.json
  api-scope: https://www.googleapis.com/auth/cloud-platform
```

6. backend/notification/src/main/resources/firebase 경로에 아래 파일 붙여 넣기

▼ tori-story-firebase-adminsdk.json

```
{
  "type": "service_account",
  "project_id": "tori-story",
  "private_key_id": "485c745329c96d5237e2b081b145b8894f2fb8cb",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQDQD5bTtmaJuZ21\nuDvWBjMitQHRX
  "client_email": "firebase-adminsdk-cwmr0@tori-story.iam.gserviceaccount.com",
  "client_id": "108578454146952732537",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-cwmr0%40tori-story.iam.gservice
  "universe_domain": "googleapis.com"
}
```

7. backend/thank/src/main/resources 경로에 아래 파일 붙여 넣기

▼ application-develop.yml

```
spring:
  config:
    activate:
      on-profile: develop
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    hikari:
      username: tori
      password: toristory
    url: jdbc:mariadb://k9a402.p.ssafy.io:3324/businessdb?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&
zeroDateTimeBehavior=convertToNull&rewriteBatchedStatements=true
```

8. backend/certFlask 경로에 아래 파일 붙여 넣기

▼ requirements.txt

```
# YOLOv5 requirements
# Usage: pip install -r requirements.txt


Flask==2.3.2
Flask-Cors==4.0.0

# Base ----------------------------------------------------------------------
gitpython>=3.1.30
matplotlib>=3.3
numpy>=1.22.2
opencv-python>=4.1.1
Pillow>=7.1.2
psutil  # system resources
PyYAML>=5.3.1
requests>=2.23.0
scipy>=1.4.1
thop>=0.1.1  # FLOPs computation
torch>=1.8.0  # see https://pytorch.org/get-started/locally (recommended)
torchvision>=0.9.0
tqdm>=4.64.0
ultralytics>=8.0.147
# protobuf<=3.20.1  # https://github.com/ultralytics/yolov5/issues/8012

# Logging -------------------------------------------------------------------
# tensorboard>=2.4.1
# clearml>=1.2.0
# comet

# Plotting ------------------------------------------------------------------
pandas>=1.1.4
seaborn>=0.11.0

# Export --------------------------------------------------------------------
# coremltools>=6.0  # CoreML export
# onnx>=1.10.0  # ONNX export
# onnx-simplifier>=0.4.1  # ONNX simplifier
# nvidia-pyindex  # TensorRT export
# nvidia-tensorrt  # TensorRT export
# scikit-learn<=1.1.2  # CoreML quantization
# tensorflow>=2.4.0  # TF exports (-cpu, -aarch64, -macos)
# tensorflowjs>=3.9.0  # TF.js export
# openvino-dev>=2023.0  # OpenVINO export

# Deploy --------------------------------------------------------------------
setuptools>=65.5.1 # Snyk vulnerability fix
# tritonclient[all]~=2.24.0

# Extras --------------------------------------------------------------------
# ipython  # interactive notebook
# mss  # screenshots
# albumentations>=1.0.3
# pycocotools>=2.0.6  # COCO mAP
```

9. backend/{각 도메인}/src/main/java/com/{각 도메인}/{각 도메인}Application.java 파일의 {각 도메인}Application 실행

## 1-3. 배포 시 특이사항

- letsencrypt를 통한 ssl 설정
- nginx를 통한 리버스 프록시 설정 (ec2 /etc/nginx/conf.d/default.conf)

```
# HTTP 서버 설정
server {
 # 80 포트에서 들어오는 HTTP 요청을 수신
 listen 80;
 # 요청을 처리할 도메인 이름
 server_name 13.124.54.55 k9a402.p.ssafy.io;
 # 서버 버전 정보 숨기기 (보안상의 이유)2
 server_tokens off;
 # 모든 HTTP 요청을 HTTPS로 리다이렉트
 location / {
  return 301 https://tori-story.com$request_uri;
 }
}

# HTTPS 서버 설정
```

```
server {
 # 443 포트에서 들어오는 HTTPS 요청을 수신
 listen 443 ssl;
 server_name tori-story.com;
 server_tokens off;
 # 액세스 로그 기록 비활성화
 access_log off;
 # Let's Encrypt로부터 받은 SSL 인증서와 키 파일 경로
 ssl_certificate /etc/letsencrypt/live/tori-story.com/fullchain.pem;
 ssl_certificate_key /etc/letsencrypt/live/tori-story.com/privkey.pem;
 include /etc/letsencrypt/options-ssl-nginx.conf; # SSL 설정 포함
 ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # DH 파라미터 경로

 client_max_body_size 10M;

 # 기본 요청을 특정 도메인의 3126 포트로 프록시
 location / {
  proxy_pass http://k9a402.p.ssafy.io:3126/;
  proxy_set_header Host $host;
  proxy_set_header X-Forwarded-Host $server_name;
  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 Upgrade $http_upgrade;
  proxy_set_header Connection "upgrade";
  proxy_redirect off;
 }

 # /api/로 시작하는 요청을 특정 도메인의 8200 포트로 프록시
 location /api/ {
  proxy_pass http://k9a402.p.ssafy.io:8200/;
  proxy_set_header Host $host;
  proxy_set_header X-Forwarded-Host $server_name;
  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 Upgrade $http_upgrade;
  proxy_set_header Connection "upgrade";
  proxy_redirect off;
 }
}
```

- docker-compose를 통한 컨테이너 다중 실행
- Jenkins pipeline을 통한 CI/CD
- ufw-docker를 통한 방화벽 설정

## 1-4. 계정 및 프로퍼티 목록

### Frontend

- .env 있음.

### Backend

- 각 도메인 별 application-develop.yml 파일
  - AWS EC2
  - AWS S3
  - mariaDB
  - redis
  - SMTP
  - FCM
- DB
  - public IP: 13.124.54.55
  - id: tori
  - password: toristory