🏝️

포팅 메뉴얼

사용한 언어 및 제품 등의 종류와 설정값 및 버전

BackEnd

- Java 11
- SpringBoot 2.7.10
- Websocket
    - Stomp 2.3.4
    - SockJS 1.5.1
- Kafka
- Redis
- test

FronEnd

- Node.js 18.15.0
- React 18.2.0
- Next.js 13.3.0
- Recoil 0.7.7
- styled-components 5.3.9
- sockjs-client 1.6.1
- stompjs 2.3.3
- storybook 7.0.3

Server

- EC2
- Docker, Docker-Compose
- Jenkins
- Nginx
- 가비아 도메인

빌드 시 사용되는 환경 변수 등의 주요 내용 상세 기재

- FrontEnd 빌드 시 환경 변수 설정

  .env.local 파일

  ```
  NEXT_PUBLIC_SERVER_URL = { 도메인 }
  ```

배포 시 특이사항 기재

- **Nginx 설정** (도메인을 여러개 쓰는 경우)

  ```
  server {
          listen 443 ssl;
          server_name {도메인 1};

          ssl_certificate /etc/letsencrypt/live/{도메인 1}/fullchain.pem;
          ssl_certificate_key /etc/letsencrypt/live/{도메인 1}/privkey.pem;
  ```

```
        location / {
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                proxy_set_header Host $host;
                proxy_pass http://localhost:3000;
        }

        location /api {
                proxy_pass http://localhost:8888/api;
        }

        location /yut {
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                proxy_set_header Host $host;
                proxy_pass http://localhost:8888/yut;
        }
}

server {
        listen 443 ssl;
        server_name {도메인 2};

        ssl_certificate /etc/letsencrypt/live/{도메인 2}/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/{도메인 2}/privkey.pem;

        location / {
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                proxy_set_header Host $host;
                proxy_pass http://localhost:3000;
        }

        location /api {
                proxy_pass http://localhost:8888/api;
        }

        location /yut {
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                proxy_set_header Host $host;
                proxy_pass http://localhost:8888/yut;
        }
}

server {
        listen 80;
        server_name {도메인 1} {도메인 2};
        return 301 https://$server_name$request_uri;
}
```

- **Docker-Compose Yml**

```
version: '3.7'
services:
  redis:
    image: redis:latest
    command: redis-server --requirepass ssafyYutjjak! --port 6379
    container_name: redis
    hostname: redis6379
    labels:
      - "name=redis"
      - "mode=standalone"
    ports:
      - 6379:6379
    volumes:
      - ./data/:/data
      - ./conf/redis.conf:/usr/local/conf/redis.conf
  jenkins:
    build:
      context: .
    container_name: jenkins
    user: root
    privileged: true
    ports:
      - 8080:8080
      - 50000:50000
    volumes:
      - ./jenkins_home:/var/jenkins_home
      - /var/run/docker.sock:/var/run/docker.sock
  zookeeper:
    image: confluentinc/cp-zookeeper:7.3.2
    container_name: zookeeper
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
      ZOOKEEPER_TICK_TIME: 2000
  broker:
    image: confluentinc/cp-kafka:7.3.2
    container_name: broker
    ports:
      - "9092:9092"
    depends_on:
      - zookeeper
    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_ZOOKEEPER_CONNECT: 'zookeeper:2181'
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: PLAINTEXT:PLAINTEXT,PLAINTEXT_INTERNAL:PLAINTEXT
      KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://{도메인 1}:9092,PLAINTEXT_INTERNAL://broker:29092
      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
      KAFKA_TRANSACTION_STATE_LOG_MIN_ISR: 1
      KAFKA_TRANSACTION_STATE_LOG_REPLICATION_FACTOR: 1
```

외부 서비스 정보

- Gabia Domain