

한화 시스템 BEYOND SW 15기

데브옵스 프로젝트 평가

□ 개요

- 프로젝트 명 : Goody
- 평가 산출물 : 시스템 아키텍처, 빌드 및 배포 문서, Jenkins CI/CD 테스트 결과 화면
- 깃허브 주소 : <https://github.com/beyond-sw-camp/be15-4th-DevEagles-Goody>
- 평가 일시 : 2025.05.27
- 평가 대상 : 김경민, 김소원, 김태환, 박양하, 부재녕, 이채은
- 평가 강사 : 강사 우별림

□ 평가 기준

산출물	평가 요소	배점	점수
○ 시스템 아키텍처	- 시스템 아키텍처는 실제 구현한 아키텍처에 대한 적절한 이해를 기반으로 작성 되었는가? - 아키텍처 구현에 불필요하거나 누락 된 지점은 없는가?	20	16
○ 빌드 및 배포 문서	- 이미지 생성을 위한 Dockerfile이 적절하게 작성 되었는가? - Kubernetes Cluster 구성을 위한 매니페스트 파일이 적절하게 작성 되었는가? - Jenkins Pipeline을 위한 스크립트가 적절하게 작성 되었는가?	60	60
○ CI/CD 테스트 결과	- Github의 Webhook 을 활용한 Jenkins 이벤트 트리거 설정이 잘 되어 있는가? - Jenkins 파이프라인이 자동화 되어 정상적으로 동작하는가?	20	20
		100	96

□ 평가 의견

1. 시스템 아키텍처

- 시스템 아키텍처는 실제 구현한 아키텍처에 대한 적절한 이해를 기반으로 작성 되었는가?
- 아키텍처 구현에 불필요하거나 누락 된 지점은 없는가?

Gemini, Hugging Face 및 SMTP 시스템 연동 등을 통해 실무 수준의 아키텍처를 구현함. 또한 Kafka와 WebSocket 통합, Elasticsearch 기반 검색, Prometheus + Grafana 기반 모니터링, ELK(Logstash + Kibana) 기반 로깅 등 실시간 처리 및 관측성 확보까지 통합적으로 구성된 점이 돋보임.

아키텍처 다이어그램에서 각 컴포넌트 간 관계 및 흐름이 명확하게 드러나며, 전체적인 구조 이해에 도움이 됨.

다만 API Gateway를 포함한 MSA 구조로 표현 되어 있으나 실제 백엔드 개발에는 해당 사항이 적용 되지 않은 것으로 보이므로 수정이 필요하며 k8s 운영 환경에 대한 설명도 추가 되면 좋을 듯 함.

2. 빌드 및 배포 문서

- 이미지 생성을 위한 Dockerfile이 적절하게 작성 되었는가?
- kubernetes Cluster 구성을 위한 매니페스트 파일이 적절하게 작성 되었는가?
- Jenkins Pipeline을 위한 스크립트가 적절하게 작성 되었는가?

Backend Dockerfile은 Gradle 기반 멀티 스테이지 빌드로 코드 포맷 검사와 빌드를 별도로 분리하여 이미지 용량 최적화 및 코드 품질을 함께 고려함.

Jenkins Pipeline은 백엔드와 프론트엔드 소스를 분리된 디렉토리 구조로 관리하면서 개별적으로 빌드하고 Docker Hub에 각각의 이미지를 push하는 방식으로 구성되며 Pipeline 성공/실패 여부를 Discord로 알림 전송하여 운영 관점의 가시성을 확보함.

Kubernetes 매니페스트는 각 서비스의 Deployment, Service, PVC(PersistentVolumeClaim), Ingress 등을 명확히 나누어 정의하고 있으며, Redis, MariaDB에 대해 독립적인 PVC 구성도 반영되어 있음.

Spring Boot 환경 변수 설정은 ConfigMap/Secret으로 분리하여 보안 및 유지보수

측면에서 우수한 구성임. 특히 외부 API(Gemini, Hugging Face), SMTP, Kafka, S3, MongoDB, Redis 등 다양한 연계 항목이 명확하게 분리됨.

3. CI/CD 테스트 결과

- Github의 Webhook을 활용한 Jenkins 이벤트 트리거 설정이 잘 되어 있는가?
- Jenkins Pipeline이 자동화 되어 정상적으로 동작하는가?

GitHub 소스 변경을 기반으로 Jenkins에서 Pipeline이 트리거되며, 빌드 → Docker 이미지 생성 → 이미지 Push까지 자동화됨.

CI/CD 파이프라인은 Unix/Windows 환경을 모두 고려하여 OS별 분기 로직을 포함하고 있어 범용성이 뛰어남.

Prometheus 설정은 Spring Boot Actuator 기반 메트릭 엔드포인트를 통해 수집되며, Grafana와 연동하여 시각화 가능한 구조임.

Logstash를 통한 JSON 포맷 로그 수집 및 Elasticsearch 저장, Kibana를 통한 로그 조회도 ConfigMap 기반 설정으로 잘 구성되어 있음.

Kafka, WebSocket 등 실시간 처리를 위한 구성도 환경 변수 및 외부 연결 설정을 통해 실제 운영 배포를 전제로 한 구성임.