

## A Study on the Implementation of Issue Tracking System Based on DevOps

Seong-Hyun Park\*, Ji-Hoon Choi\*

\*Ph.D, Dept. of Computer Engineering, Kongju National University, Chungnam, Korea

\*Ph.D. Student, Dept. of Computer Engineering, Kongju National University, Chungnam, Korea

### [Abstract]

With the development of the cloud industry, various efforts are being made to improve resource efficiency and environmental continuity in software development and operation methods. However, in constructing a conventional structure of structural scheme, if the functionality is improved and added, it has a problem that the efficiency of full build arrangement is reduced for the changed source. In this paper, if a problem occurs in a solution service company, communication between the development department and the operation department can be facilitated and responded quickly, and the latest source updated during processing of the problem is the management system, provided is a system that can take advantage of CI tools and continuously integrate them. As communication through the proposed system is smooth, it is expected that unnecessary work can be reduced and the speed of development can be improved.

▶ **Key words:** DevOps, Issue, Issue Tracking System, CI, CD

### [요 약]

클라우드 산업의 발전으로 소프트웨어의 개발과 운영 방식에서 자원의 효율성 개선과 환경의 연속성을 위한 다양한 노력이 이루어지고 있다. 그러나 전통적인 구조 방식의 서비스 구축에서는 기능을 개선하고, 추가할 경우 수정된 소스에 대하여 전체 빌드로 인한 배포의 효율성이 저하되는 문제를 가지고 있다. 본 논문에서는 솔루션 서비스 기업에서 이슈가 발생할 경우 개발 부서와 운영 부서 사이의 의사소통을 원활하게 하여 신속하게 이슈에 대응할 수 있고, 이슈 처리 중 업데이트 되는 최신 소스들은 관리시스템과 CI 툴을 활용하여, 지속적으로 통합하여 배포할 수 있는 시스템을 제안한다. 제안 시스템을 통하여 의사소통이 원활해짐에 따라 불필요한 작업을 줄일 수 있고, 개발의 속도를 향상시킬 수 있을 것으로 기대된다.

▶ **주제어:** 개발운영, 이슈, 이슈트래킹시스템, 지속적 통합, 지속적 배포

- 
- First Author: Seong-Hyun Park, Corresponding Author: Ji-Hoon Choi
  - \*Seong-Hyun Park (a94270816@gmail.com), Dept. of Computer Engineering, Kongju National University
  - \*Ji-Hoon Choi (hunnx27@gmail.com), Dept. of Computer Engineering, Kongju National University
  - Received: 2021. 11. 17, Revised: 2021. 12. 20, Accepted: 2021. 12. 23.

## I. Introduction

최근 4차산업혁명과 함께 클라우드 산업이 발전하면서 소프트웨어의 개발과 운영 방식의 변화를 기반으로, 운영 환경의 연속성과 자원의 효율성 개선을 위한 많은 노력이 이루어지고 있다. 기존의 전통적인 구조의 서비스 구축에서는 기능을 개선하거나 추가하면, 전체를 다시 빌드하여 배포하기에 자원 활용과 유연성 측면에서 효율성이 저하되는 문제가 있었으나[1], 현재 클라우드의 환경과 고객 경험에 의한 사용자 중심의 서비스 플랫폼이 확대됨으로서, 오픈 API 기반의 기능들을 하나로 연결시키는 서비스로 합성이 가능한 MSA(Micro Service Architecture) 구조로 변화되고 있는 상황이다[2,3], 실제로 대부분의 IT 서비스들은 확장과 유지보수가 비교적 용이하기에 MSA를 기반으로 서비스 및 설계되고 있다[4]. 또한 클라우드로 제공되는 막대한 인프라와 리소스를 통하여 MSA와 무중단 서비스가 가능해졌고, 급격한 기술 발전을 기반으로 이를 뒷받침할 수 있는 소프트웨어 개발 속도를 단축시키는 것은 상당히 중요한 요소라 할 수 있다[5]. 그러나 개발 환경과 운영 환경의 많은 차이점과 개발 부서와 운영 부서가 분리되어 있는 대부분 조직에서는 의사소통에 대한 이슈가 계속 나타나고 있으며, 개발 시스템에서 운영 서비스로의 전환 또한 어려운 상황이라 할 수 있다[6]. 따라서 급속도로 발전하는 기술 발전에 개발 속도를 향상시키고, 개발 및 관련 부서간의 의사소통을 원활하게 할 수 있는 시스템에 대한 연구가 필요한 실정이다. 최근 소프트웨어의 서비스와 개발을 신속하게 제공할 수 있는 DevOps 방법론의 채택이 확산되고 있는 가운데, 소프트웨어 프로젝트를 수행할 경우에 수정 코드의 지속적인 통합에 의하여 테스트 및 빌드가 가능한 CI(Continuous Integration) 과정과 통합 과정을 통과한 코드에 대한 테스트 서버 및 사용자들이 사용할 수 있는 서비스 운영 서버에 지속적으로 배포하는 CD(Continuous Delivery, Deploy) 과정에 대한 자동화가 요구되고 있다. 본 논문에서는 소프트웨어 프로젝트 개발에 적용하여 개발 부서와 운영 부서간의 이슈를 신속하게 처리할 수 있는 DevOps방법을 적용한 이슈 트래킹 모델을 제안한다. 제안 모델을 통하여 이슈 관리의 편리성과 부서 간의 원활한 의사소통 능력이 향상될 것으로 기대된다. 본 논문의 구성은 다음과 같다. 제 2장에서는 관련연구인 DevOps와 CI/CD, MSA에 대하여 기술하고, 제 3장에서는 제안 시스템의 구성도와, 프로세스, 데이터베이스 구성 및 구현 화면에 대하여 살펴보고, 제 4장에서는 개발 및 서비스 프로젝트에 적용한 결과 및 기술하며, 제 5장에서는 결론과 향후 연구방향에 대하여 기술한다.

## II. Preliminaries

### 1. DevOps

DevOps는 일반적으로 소스 코드 업데이트와 배포 사이에는 지연시간의 발생과, 개발자와 운영자 간의 충돌의 격차를 해결하기 위하여 등장한 것으로 배포 파이프라인을 DevOps 플랫폼이라고 하며[7], 지속적인 전달 및 배포와 관련된 다양한 연구가 이루어지고 있다[8,9]. 또한 DevOps는 정보기술 전문가와 소프트웨어 개발자 사이에서 의사소통, 통합, 협업을 강조하는 개발 환경이나 문화와 같은 개발 철학을 의미하며, 소프트웨어 운영 조직과 개발 조직 사이에서 서로 의존적 대응이라 할 수 있고, 각각의 조직이 소프트웨어 서비스와 제품을 빠른 시간에 개발하고, 이를 배포하는 것을 목적으로 하고 있다[10]. 운영과 개발 과정 동안에 지속적인 전달을 전제로 하여 클러스터링된 일련의 작업 단계별로 구성되어 있고, 특정한 기능을 수행하기 위한 관련된 툴 등을 포함하고 있다[6]. 다음의 [그림 1]은 파이프라인을 통한 DevOps 단계를 나타낸 것으로, 이러한 단계가 자동화되면 신속하고 안정적인 소프트웨어를 제공할 수 있다[11].

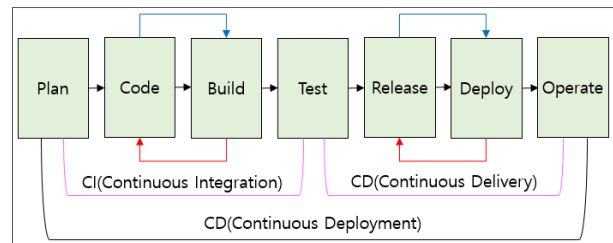


Fig. 1. DevOps Phases

### 2. CI/CD

CI/CD는 소프트웨어 개발 단계부터 운영 배포까지의 애플리케이션 생명주기를 자동화하여 수동 방식으로 진행됐을 때보다 상당한 시간 효율을 주는 개발 방법론이다[14]. CI는 지속적인 통합을 의미하고 개발 생명주기에서 빌드, 패키징 및 테스트 과정이다. CD는 지속적인 서비스 제공 혹은 지속적인 배포를 의미하고 CI 과정에서 패키징된 소스를 운영환경으로 자동으로 배포하는 과정으로, 다음의 [그림 2]는 CD/CD 의 기능적 구조를 나타낸 것이다[13].

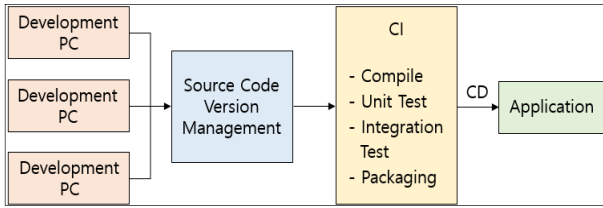


Fig. 2. CI/CD Functional Structure

3. MSA

한 개의 애플리케이션을 독립적으로 결합된 모듈로 분할하여 서비스가 가능한 MSA는 기능의 개선과 확장이 비교적 용이하며, 단일 모듈의 장애에 영향을 크게 받지 않는다. 즉 마이크로서비스는 독립적으로 배포되고, 확장되며, 테스트될 수 있고, 단일 책임을 가지고 있는 소규모 애플리케이션으로서[14], 마이크로서비스 아키텍처라는 용어는 시스템이 여러 개의 작은 협력 구성요소로 분해되는 개발 스타일을 의미한다[15]. 구성요소는 일반적으로 http와 같이 직접 지점 간의 인터페이스를 통하여 상호 작용하거나 이벤트 버스 기술 등이며, 자체 프로세스에서 실행되고 HTTP 리소스 API인 경량 매커니즘과 통신하는 소규모 서비스 모음으로서, 단일 애플리케이션을 개발하는 접근 방식이라 할 수 있다. 또한 이러한 서비스는 비즈니스 기능을 중심으로 구축되며, 완전히 자동화된 배포를 통하여 독립적으로 배포할 수 있다[16].

III. The Proposed Scheme

1. System Configuration

다음의 [그림 3]은 제안하는 시스템의 구성도이다.

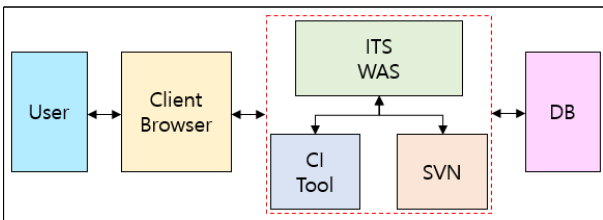


Fig. 3. System Configuration

제안 시스템은 사용자, 클라이언트 브라우저, 이슈 트래킹 시스템(ITS WAS), CI와 CD를 수행하는 CI Tool, 소스의 형상관리가 가능한 SVN(SubVersion) 으로 구성되어 있다. 사용자는 이슈 트래킹 시스템을 사용하는 운영 담당자, 검토자, 이슈를 조치하는 개발자이며, 소프트웨어 개발과 서비스에서 발생할 수 있는 이슈를 관리하여 운영자

와 개발자 간의 의사소통을 신속하고 원활하게 지원할 수 있다. 백 그라운드에서는 CI Tool과 SVN 사이의 인터페이스 역할을 하며, CI Tool은 CI/CD 수행을 하는 배치 스크립트를 미리 작성하고 배치 스크립트를 실행시켜주는 JOB을 관리할 수 있다. 또한 SVN 소스 버전 관리와 변경된 소스를 저장하고 다운받을 수 기능을 보유하고 있다.

2. System Process

다음의 [그림 4]는 이슈 트래킹 시스템의 관리 프로세스로서 여섯 단계로 진행된다. 첫째, 이슈 등록 단계로서 ITS에서 운영 담당자가 이슈를 등록한다. 둘째, 개발자에게 새로운 이슈를 할당하는 단계로서, 등록된 이슈를 검토자가 검토하고 처리해야 될 이슈인지 확인하여 개발자에게 이슈를 할당한다. 셋째, 알림 메시지 전송 단계로서 이슈가 할당이 되면 배정된 개발자에게 알림 메시지가 전송된다. 넷째, 개발 소스 수정 단계로서 개발자가 배정된 이슈 내용을 확인하고 소스를 수정한다. 다섯째, 커밋 단계로서 조치한 소스를 SVN에 커밋한다. 여섯째, 저장 단계로서 SVN은 커밋된 소스와 이력을 저장한다.

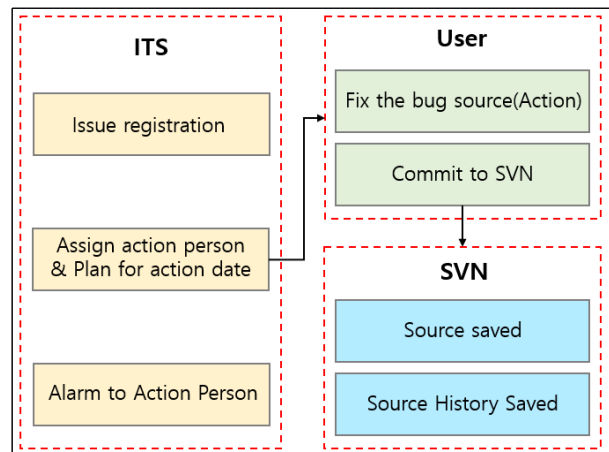


Fig. 4. Issue Management Process

다음의 [그림 5]는 DevOps 환경에서의 CI/CD 프로세스이다. 첫째, 소스가 커밋이 되면 SVN은 CI Tool에게 CI JOB을 수행하라는 API 호출을 한다. 둘째, CI Tool은 등록된 CI 수행을 통해 수정된 소스에 대하여 빌드 및 패키징을 수행한다. 셋째, ITS에서 SVN의 커밋 내역을 조회하여 이슈 내용과 일치하는 커밋 메시지를 찾아서 저장한다. 이 경우에는 이슈 제목과 커밋 제목이 일치하는 것을 연결하게 된다. 넷째, 검토자는 ITS를 통하여 CI의 성공 여부, 오류 내용, 테스트 결과와 같은 이슈의 처리 상태를 파악한다. 또한 필요에 따라서 SVN 저장소에 커밋된 소스 이

력을 웹을 통해 확인할 수 있어 수정된 소스 내용을 직접 확인하고 판단할 수 있다. 이 과정에서 별다른 문제가 파악되지 않을 경우 검토자는 해당 이슈에 대해 승인 요청을 하게 된다. 다섯째, CI Tool에 CD JOB을 수행하는 API 호출을 통해 운영환경에 소스를 배포한다.

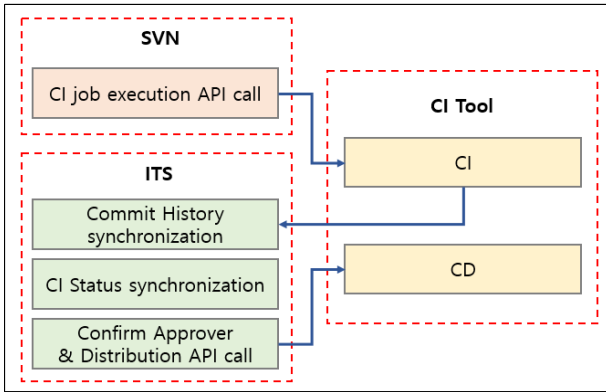


Fig. 5. DevOps CI/CD Process

### 3. CI/CD Process

다음의 [그림 6]은 CI와 CD를 상세화한 프로세스이다.

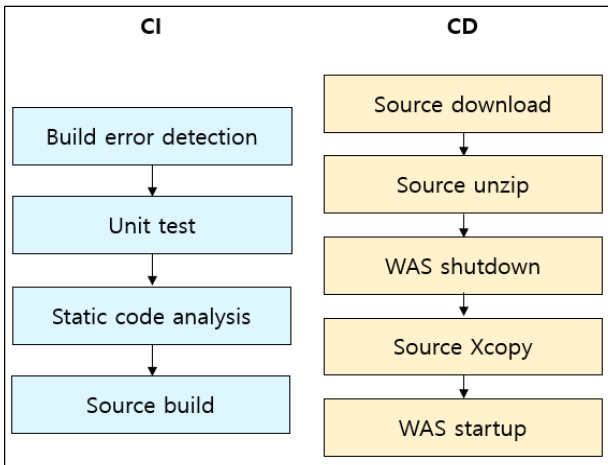


Fig. 6. CI/CD Process

CI에서는 수정된 소스를 빌드할 때 오류가 발생하는지 검출하며, 단위 테스트를 통하여 각 모듈 사이에 오류가 있는지 확인한다. 또한 정적 코드 분석을 통해 소스 단위의 오류를 검출하며, 3단계의 검사 과정을 통하여 이상이 없는 경우 수정된 소스를 빌드하고 패키징한다. CD에서는 패키징 되어 있는 소스를 운영환경으로 다운로드하는 역할을 수행하며, 압축되어 있는 소스의 압축을 해제하고, 실행되고 있는 WAS를 종료한다. 이후 압축 해제한 소스를 배포 위치에 복사하며, 모든 단계가 정상적으로 수행될 때 WAS를 재시작 한다.

### 4. Database Configuration

다음의 [표 1]은 이슈를 관리하고 CI/CD를 매핑하는 테이블이다. 커밋 UID를 외래키로 커밋 테이블과 연결한다.

Table 1. Issue List Table

Field Name	Type	Description
ID	BIGINT	ID
ISSUE_ID	VARCHAR(20)	Issue ID
DEVOPS_YN	CHAR(1)	DEVOPS Use
WF_STATE	VARCHAR(3)	WorkFlow State
COMMIT_UID	BIGINT	Commit UID
ACT_USER_ID	VARCHAR(100)	Action User ID
ACTION_DATE	VARCHAR(8)	Due Date

추가적으로, SVN 설정 테이블, 커밋 이력 테이블, JOB 상태 관리 테이블로 구성된다. SVN 설정 테이블은 SVN과 API 통신을 위해 필요한 기본적인 프로토콜, URL, 포트, Context 경로 인증정보를 저장한다. 커밋 이력 테이블은 브랜치와 커밋 ID, 커밋 생성 일시, 커밋 제목, 커밋 메시지, 커밋 정보, 커밋한 소스를 볼 수 있는 리모트 저장소, URL을 저장하고 수행한 CI JOB ID를 매핑한다. CI JOB의 상태를 관리하는 테이블은 JOB의 이름, 수행 상태, 생성일시, 시작일시, 종료일시가 저장되도록 구성하였다.

### 5. ITS DevOps Management Screen

다음의 [그림 7]은 제안 시스템인 DevOps 기반의 ITS 관리 화면이다.

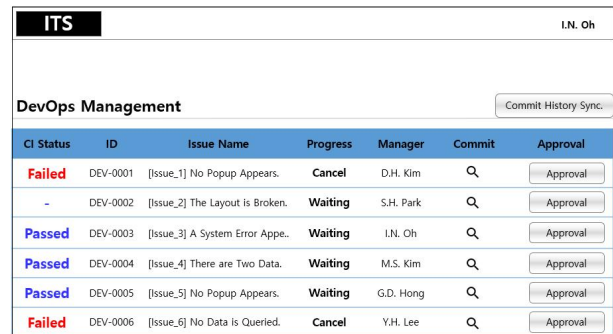


Fig. 7. ITS DevOps Management Screen

DevOps 관리 화면에서는 이슈를 관리하고 CI의 상태를 확인할 수 있으며, 진행 상태를 통해 개발자의 개발완료 여부와 CI의 빌드 수행 결과를 통해 상태를 표시하고, 빌드 수행이 성공적일 때는 승인대기 상태로 표시된다. 커밋과 JOB의 상태가 동기화 되지 않았을 때에는 커밋 내역 동기화 버튼을 눌러 최신 상태의 커밋 이력과 JOB 수행 상태

를 갱신하고, 5분마다 한번 씩 배치의 수행으로 최신 상태를 동기화한다. 빌드 수행까지 정상일 때 관리자는 커밋 컬럼의 돋보기 버튼을 통하여 커밋의 내역을 확인하고, 승인배포 컬럼의 승인 버튼을 통하여 운영환경에 빌드된 소스를 반영할 수 있다. 검토자의 승인이 늦어질 경우를 대비하여 운영 책임자, 검토자, 개발자의 팀장에게 권한을 부여하여 승인 지연을 방지할 수 있다. 다음의 [그림 8]은 CI/CD를 위한 JOB 스크립트의 내용을 나타낸 것이다.

```
@rem # GIT Clone(Source Download)
rmdir /S /Q %SOURCE_PATH%
git clone -b ^
%BRANCH% -q %CLONE_URL% ^
%SOURCE_PATH%
@rem # Maven Build(Source Build/Test)
mvn.cmd -f ^
%SOURCE_PATH%\pom.xml ^
clean install
@rem # Kill(WAS End)
taskkill /F /I ^
"WINDOWTITLE eq %PROJECT_NAME%"
@rem # (Operating Source Delete)
rmdir /S /Q ^
%TARGET_PATH%\%WAR_FILE%
@rem # xcopy(Copy & Move Build Files)
xcopy %SOURCE_PATH%\target\%
%WAR_FILE% ^
%TARGET_PATH%\%WAR_FILE% /y /f
@rem # (WAS Run)
call %LAUNCH_BAT%
```

Fig. 8. CI/CD Batch Script

관리자가 ITS에서 승인 배포를 수행할 경우 다음의 스크립트와 같은 과정을 통하여 운영되고 있는 기존 소스에 배포된다. 분산 버전 관리 시스템인 GIT를 이용하여 최신 소스를 복사하고, 복사한 소스를 Maven을 이용하여 빌드 및 테스트한 후, 빌드가 정상적으로 수행이 되면 운영에 실행하고 있는 WAS를 종료한다. 이후 운영소스를 삭제하고, 빌드한 소스를 운영 배포 위치에 복사하고, WAS를 재실행한다.

#### IV. System Evaluation

제안 시스템을 평가하기 위하여 실제 프로젝트를 수행하고 솔루션을 서비스하고 있는 기업 3개소를 선정하여 베타 테스트를 진행하였으며, 개발자와 운영 관리자 87명을 대상으로 설문 조사를 시행하였다. 설문 내용은 사용자 만족도와 시스템 사용 효율성, 효과성으로 구분하였으며, 특히 개발자와 각 부서간의 의사소통이 잘 이루어지고 있는지에 대한 설문을 실시하였다. 다음의 [표 2]는 조사 결과를 종합하여 정리한 것이다.

Table 2. Survey Results

Classification	User satisfaction	System Efficiency	System Effectiveness
Very good	43	47	51
Good	28	27	25
Moderate	10	8	8
Dissatisfied	6	5	3
Very Dissatisfied	0	0	0

제안 시스템의 사용 테스트를 실시한 결과를 종합하면, ‘좋다’ 이상의 응답자가 84.6%로 조사되었으며, 시스템을 사용한 만족도는 기대 이상으로 평가 되었다. 제안 시스템을 처음 사용하는 사용자들이 접근에 대한 어려움이 있었으나, 시스템에 빠르게 적응하여 개발 부서와 운영 부서간의 의사소통이 원활하게 진행되었다는 의견이 많았으며, 프로젝트 관리시스템에 적용 가능한 방안을 모색할 필요가 있다는 의견이 도출되었다.

#### V. Conclusions

클라우드 산업이 발전하면서 소프트웨어의 개발과 운영 방식이 변화됨에 따라 많은 서비스 기업들이 편리한 유지 관리와 안정적인 장점을 가지고 있는 클라우드 전환을 고려하고 있다. 클라우드로 전환하려면 DevOps 환경으로 운영 환경을 구성하는 것은 필수적이다. 클라우드로 제공되는 막대한 인프라와 리소스를 통하여 소프트웨어 개발 속도를 단축시키는 것은 상당히 중요한 요소라 할 수 있으며, 이를 통하여 다양한 서비스가 가능하다. 그러나 개발 환경과 운영 환경의 많은 차이점과 개발 부서와 운영 부서가 분리되어 있는 대부분 조직에서는 의사소통에 대한 이슈가 계속 나타나고 있으며, 개발 시스템에서 운영 서비스로의 전환 또한 어려운 상황이라 할 수 있다. 본 논문에서는 DevOps 환경을 기반으로 프로젝트 및 솔루션 운영에서 이슈 관리 시스템과 연동함으로써 CI/CD의 수행을 검토자가 확인하고 승인 프로세스를 진행할 수 있는 시스템을 제안한다. 제안 시스템을 통하여 개발환경과 운영환경의 격차를 해소할 수 있고, 각 부서 간에 원활한 의사소통이 가능하며, CI/CD 자동화 프로세스를 통해 불필요한 작업을 줄이고 개발 속도를 높일 수 있을 것으로 기대된다.

## REFERENCES

- [1] J. B. Kim, J. I. Kim, "A Study of Application Development Method for Improving Productivity on Cloud Native Environment," *Journal of Korea Multimedia Society*, Vol. 23, No. 2, pp. 328-342, 2020. DOI: 10.9717/kmms.2020.23.2.328
- [2] J. Han, S. Park, and J. Kim, "Conceptual Design and Verification of Over Cloud Approach for Realizing Dynamic IoT-Cloud Services," *Korean Institute of Information Scientists and Engineers Transactions on Computing Practices*, Vol. 25, No. 1, pp. 9, 2019. DOI : 10.5626/KTCP.2019.25.1.9
- [3] N. Alshuqayran, N. Ali, and R. Evans, "A Systematic Mapping Study in Microservice Architecture," *Proceeding of IEEE 9th International Conference on Service-oriented Computing and Application*, pp. 44-51, 2016. DOI: 10.1109/SOCA.2016.15
- [4] K. J. Kim, J. S. Moon, M. H. Kim, J. Y. Lee, S. Y. Noh, "A Build Cache Optimization Strategy in Docker based CI/CD Environments," *Proceeding of the Korean Institute of Information Scientists and Engineers*, pp. 481-483, 2021.
- [5] C. E. G. Gallardo, J. Hernantes and N. Serrano, "DevOps," *IEEE Software*, Vol. 33, No. 3, pp. 94-100, 2016. DOI: 10.1109/MS.2016.68
- [6] I. S. Jeon, "Integrated management (DevOps) of development and operation organization in the non stop environment considering security," *Review of Korea Institute Of Information Security And Cryptology*, Vol. 5, No. 1, pp. 47-52, 2015.
- [7] Hasan Yasar and Kiriakos Kontostathis, "Where to Integrate Security Practices on DevOps Platform," *International Journal of Secure Software Engineering (IJSSE)*, Vol. 7, No. 4, pp. 39-50, 2016. DOI: 10.4018/IJSSE.2016100103
- [8] Mojtaba Shahin, Muhammad Ali Babar, and Liming Zhu, "The Intersection of Continuous Deployment and Architecting Process: Practitioners' Perspectives," In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '16)*. ACM, pp. 1-10, 2016. DOI: 10.1145/2961111.2962587
- [9] Johannes Wettinger, Vasilios Andrikopoulos, and Frank Leymann. 2015. "Enabling DevOps Collaboration and Continuous Delivery Using Diverse Application Environments," In *On the Move to Meaningful Internet Systems (OTM 2015 Conferences)*. Springer International Publishing, pp. 348-358. 2015.
- [10] K. H. Chung, J. S. Park, and G. Lee, "A Study on A Web-Based DevOps Platform Using Linux Container," *Journal of the Korea Convergence Society*. Vol. 10, No. 12, pp. 71-80, 2019. DOI: 10.15207/JKCS.2019.10.12.071
- [11] Purohit, K, "Executing DevOps & CI/CD, reduce in manual dependency," *International Journal of Scientific Development and Research*, Vol. 5, NO. 6, pp. 511-515, 2020.
- [12] M. J. Sim, S. Y. Oh, "Ansible-based automated deployment implementation for customized user environment," *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 820-821, 2018
- [13] H. K. Lee, D. K. Noh, "Design of the Efficient CI/CD (Continuous Integration and Continuous Deploy) Automation Model," *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 372-373, 2021.
- [14] P. Clarke, P. Elger and R. V. O'Connor. "Technology enabled continuous software development," *Proceedings of the International Workshop on Continuous Software Evolution and Delivery*, pp. 48-48, 2016. DOI: 10.1145/2896941.2896943
- [15] J. Thönes, "Microservices," *IEEE Software*, vol. 32, no. 1, pp. 116-118, 2015. DOI: 10.1109/MS.2015.11
- [16] O'Connor, R. V., Elger, P., & Clarke, P. M., "Continuous software engineering-A microservices architecture perspective," *Journal of Software: Evolution and Process*, Vol. 29, No. 11, 2017. DOI: 10.1002/smr.1866

## Authors



Seong-Hyun Park received the B. S. degree in College of Arts and Music from Chungnam National University, Korea in 2017. The M. S, and Ph. D. degrees in Computer Engineering from Kongju National

University, Korea, in 2017, 2020. He is currently a teaching in the Department of Computer Science & Engineering. Kongju National University. He is interested in computer music, convergence Education. and Real Time System and Management and Clout computing and Communication.



Ji-Hoon Choi completed his masters degree program in computer science at Kongju National University, Korea in 2020. Currently, the Doctoral program in Computer Science and Engineering at Gongju National

University, Korea, has been in progress since March 2021. He started programming as a web-based program in 2014. Currently, he is developing programs related to project management solutions and automated testing tools as a manager at PMS Plus. He is interested project management systems, automated testing, algorithms, machine learning, and automated systems.