

# Dive Into ISV With MegazoneCloud

2024-09-03 | 비즈니스

## Dive Into ISV With MegazoneCloud

ISV는 "Independent Software Vendor"의 약자로, 독립 소프트웨어 개발사로 Independent Software Vendor(ISV)는 특정 하드웨어나 플랫폼에 의존하지 않고, 다양한 환경에서 작동할 수 있는 소프트웨어를 개발하고 판매하는 업체를 의미합니다. 이걸 무인 ISV라고 부르고 있습니다.

### Dev, Ops 그리고 Sec

2023년 기준으로 미국에서만 18,387개의 ISV 회사가 활발히 솔루션 개발 및 운영하고 있습니다. 백 개, 천 개도 아니고 만 개가 넘는 솔루션이 존재한다고 생각하면 복잡함이 느껴질 정도입니다. ISV는 개발하는 소프트웨어의 목적과 기능에 따라 여러 카테고리 분류됩니다. 비즈니스 애플리케이션부터 보안, 데이터 관리 및 분석, 클라우드 솔루션, 개발 도구 및 플랫폼, 커뮤티케이션 및 협업, 산업별 솔루션, 이커머스 및 디지털 마케팅, 엔터프라이즈용 서비스 및 게임 개발도구와 멀티 미디어 및 그래픽 소프트웨어로 나뉘져 있습니다.

개발의 뿌리라고도 할 수 있는데요, 이번 달에는 ISV 카테고리중 하나인 DevSecOps를 소개해보겠습니다.

DevOps는 많이 들어보셨을 거라 생각이 됩니다. 아래 이미지를 많이 보셨을 것 같지만, DevOps라는 말은 Development Operations의 약어로, 2009년 패트릭 드부외가 처음으로 언급해 사용된 용어입니다.

기존에는 개발팀과 IT운영팀이 분리되어 각자의 역할을 수행했지만 고객 만족과 더 빠른 가치 제공이라는 가치를 위해 개발팀과 운영팀의 작업을 결합하고 자동화하여 소프트웨어를 제공하는 프로세스입니다. 개발과 운영 사이를 안에서 동시에 가동하면서 변화에 빠르게 대응하고, 결국 개발에서 배포까지, 수정에서 재배포까지 유기적으로 연결되는 고리를 통해 릴리스 시간을 줄이는데 목적을 둡니다.



<reference: <https://velog.io/@dongminlee94/MEIT-Data-Standardization>>

근데 어느 순간부터 DevOps에 중요 요소로 보안도 추가되어 DevOps에서 업그레이드된 DevSecOps를 ISV들이 선보이기 시작했습니다

### DevOps는 알겠는데 DevSecOps는 무엇인가요?

DevSecOps는 개발(Development), 보안(Security), 운영(Operations)의 약자로, 기존의 개발과 운영을 하나의 사이클로 묶은 DevOps(데브옵스)에 보안을 합친 개념입니다.

최근의 개발 주기는 몇일 또는 몇 주의 빠른 개발 주기 환경으로 기존의 보안 방식과 정적인 효율성을 떨어뜨리는 원인으로 인식되어 최종 단계에서 보안 점검을 형식적인 행위로 처리해 버리는 경우가 발생하고 있습니다. 이를 개선하기 위해서는 DevSecOps를 통해 개발 Pipeline의 과정에서 보안 정책 및 기술을 자동화하는 것이 필요합니다.



<reference: <https://bigstepstech.com/blog/devsecops-what-is-it-why-is-it-important/>>

### GitLab과 DevSecOps

DevOps와 함께 자주 언급되는 ISV 솔루션이 있는데요, 이 솔루션은 바로 바로, GitLab입니다. GitLab은 단일 애플리케이션에서 DevSecOps의 모든 기능을 제공하는 혁신적인 도구로, 개발자들에게 체계적이고 통합된 DevOps 플랫폼을 제공합니다. 이를 통해 협업과 소프트웨어 개발 과정을 혁신적으로 향상시킬 수 있습니다.

GitLab은 DevOps 혁신을 위한 완벽한 도구로, 효율적인 개발과 지속적인 협업을 위한 모든 기능을 갖추고 있습니다. GitLab을 활용하면 개발 업무의 효율성을 크게 높일 수 있습니다.

### 왜 GitLab인가요?

GitLab이 DevSecOps와 함께 자주 언급되는 이유는 그 기능과 통합된 플랫폼이 DevSecOps의 요구사항을 모두 충족하기 때문입니다. DevSecOps는 개발, 보안, 운영을 모두 포함하는 접근 방식으로, 이를 효율적으로 구현하려면 다양한 기능을 통합적으로 제공하는 도구가 필요합니다. GitLab은 이러한 요구를 완벽하게 소화하기 때문에 DevSecOps 들로 적합하다고 할 수 있습니다.

- 통합 개발 환경(IDE) 제공**  
GitLab은 코드 작성, 빌드, 테스트, 배포 등의 모든 개발 프로세스를 하나의 플랫폼에서 관리할 수 있습니다.
- 보안 기능 강화**  
코드 스캔, 취약점 분석, 보안 정책 적용 등의 보안 기능을 제공하여 개발 단계부터 보안을 고려할 수 있습니다. 보안 위협을 빠르게 감지하고 대응할 수 있습니다.
- 협업 및 버전 관리**  
팀 단위로 협업하여 코드 변경 내역을 추적할 수 있습니다. 브랜치 관리, 병합 요청, 코드 리뷰 등의 기능으로 효율적인 개발 프로세스를 지원합니다.
- 자동화 및 CI/CD 기능**  
빌드, 테스트, 배포 등의 프로세스를 자동화하여 개발 속도와 품질을 높일 수 있습니다. 다양한 CI/CD 파이프라인 구축을 지원합니다.
- 오픈소스 및 확장성**  
GitLab은 오픈소스 기반으로 개발되어 커스터마이징이 용이합니다. 다양한 플러그인과 통합 기능을 제공하여 조직의 요구사항에 맞게 구축할 수 있습니다.

예를 들어, 새로운 기능을 개발할 때, SaaS 기반 CRM 도구에 새로운 코드를 커밋하면 GitLab은 자동으로 CI/CD 파이프라인을 트리거하여 애플리케이션을 빌드, 테스트 및 배포합니다. 이 과정에서 GitLab의 도구는 지속적으로 코드를 스캔하여 취약점을 찾아내어 별도의 보안 도구 없이 잠재적인 문제를 조기에 감지할 수 있습니다.

만약 프로젝트가 일반 데이터 보호 규정(GDPR)을 준수해야 한다면, GitLab은 데이터 처리 관행이 필요한 기준을 충족하는지 자동으로 검증하여 추가적인 수작업 없이 컴플라이언스 보고서를 생성할 수 있습니다. 또한, GitLab의 병합 요청 기능을 통해 팀원들은 코드 변경 사항을 쉽게 리뷰하고, 논의하며, 신속하게 승인할 수 있어 지연을 줄이고 코드 품질을 향상시킵니다.

배포 후에는 GitLab이 애플리케이션을 종합적으로 모니터링하여 문제가 발생할 경우 신속하게 해결할 수 있도록 도와주어 애플리케이션의 안정성과 보안을 유지할 수 있습니다. 이 모든 작업을 GitLab Enterprise를 사용하여 손쉽게 수행할 수 있습니다. GitLab과 함께라면 개발, 보안, 컴플라이언스를 한 번에 해결할 수 있습니다. GitLab Enterprise를 사용하게 되면 GitLab의 기능을 통해 더욱 효율적이고 안전한 개발 환경을 경험해 볼 수 있습니다./

### 포테토, 포트아, GitLab, GitHub, 같은거 아닌가요?

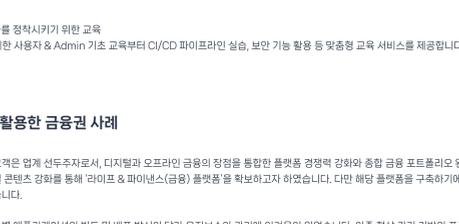
GitLab과 GitHub는 Git을 기술적 기반으로 하는 소스 코드 저장소라는 점에서 공통점이 있습니다. 이 둘의 차이는 DevOps 요소에 있습니다. GitLab은 CI/CD와 DevOps 워크플로우를 내장하고 있지만, GitHub는 사용자가 원하는 CI/CD 도구를 직접 통합해야 합니다. 예를 들어, Jenkins, CircleCI, TravisCI와 같은 3rd Party 프로그램을 사용합니다.

**GitLab (기업 운영용 설계) vs GitHub (개발자 프로젝트 공유용 설계)**  
GitLab은 Community Edition(무료버전)과 Enterprise Edition(유료버전) 2가지로 제공하고 있습니다. GitLab을 유료로 사용해야 하는 이유는 유료 버전과 비교했을 때, 유료 버전이 제공하는 추가 기능과 혜택 때문입니다.

**GitLab을 사용하고 싶은데 무료버전을 두고 유료버전을 써야하나요?**  
GitHub는 확장 사용시 3rd party 유료 서비스가 필요하며, GitHub가 Azure에 종속되어 있어 멀티클라우드 전략을 실행하는데 있어 다소 어려움이 있습니다. 또한, GitHub는 대규모 모노레포(monorepos)의 확장 및 배포에 어려움을 겪고 있어 배포 시간이 길어지고, 무중단 업그레이드(zero-downtime upgrades)를 제공하지 않습니다. 이는 특히 많은 사용자를 보유한 조직의 DevOps 팀의 작업을 지연시키는 요인이 됩니다. GitLab과 달리, GitLab은 거의 무중단 업그레이드를 제공하여 이러한 문제를 해결합니다.

**GitLab 유료버전이 제공하는 주요 기능**

- 대규모 프로젝트 관리를 위해 여러 개의 케이스/프로젝트 운영
- Merge Request 승인/인트를 가능, Merge History, Audit 기능
- 유저별 대시보드 커스텀, 전체 그룹 파이프라인 현황 관리, 컴플라이언스 준수/규제사항 관리



<reference: GitLab 대시보드>

### 메가존클라우드를 통해 GitLab을 사용하면 좋은 점

**고객이 편하게 사용할 수 있는 GitLab**

- DevOps 간소화  
체계적인 Assessment와 컨설팅을 수행하기 위한 요구사항을 파악 -> As-Is 분석 -> 적합성 평가 -> 파이프라인 & 아키텍처 준비 -> 보안 & 규정 사항 준수 등 고객사 환경에 맞춤형 컨설팅 전략을 함께 마련합니다.
- Gitlab 기반 DevOps Pipeline 구축  
여러 DevOps 도구와의 연동, Cloud Native 환경에 최적화된 DevOps Pipeline 구성 등 다양한 사례를 기반으로 고객이 지향하는 DevOps 환경 구성을 제공합니다.
- 안정적인 DevOps 서비스 운영을 위한 GitLab CarePack  
GitLab 전문 엔지니어가 배치되어 신속한 기술 문의 제공을 통해 GitLab의 운영과 유지보수에 대한 지원을 제공하여 시스템 안정성과 장애 해결에 도움을 드립니다.
- DevOps 문화를 정착시키기 위한 교육  
Gitlab 활용을 위한 사용자 & Admin 기초 교육부터 CI/CD 파이프라인 실습, 보안 기능 활용 등 맞춤형 교육 서비스를 제공합니다.

### GitLab을 활용한 금융권 사례

국내의 한 금융고객은 업계 선두주자로서, 디지털과 오프라인 금융의 장점을 통합한 플랫폼 경쟁력 강화와 종합 금융 포트폴리오 완성, 비금융 플랫폼 및 라이프 스타일 콘텐츠 강화를 통해 '라이프 & 피어낸스(금융) 플랫폼'을 확보하고자 하였습니다. 다만 해당 플랫폼을 구축하기에는 다량과 같은 어려움을 겪고 있었습니다.

고객은 운영환경 별 애플리케이션의 빌드 및 배포 방식이 달라 유지보수와 관리에 어려움이 있었습니다. 이중 행사 관리 기반의 프로젝트 관리로 인해 고객사 ITSM\* 프로세스 연동이 제한적인 이슈가 있었습니다.  
\*ITSM이란 IT Service Management로 IT서비스와 고객사 과제를 관리하기 위한 시스템입니다.

고객은 Jenkins, Spinnaker, SVN, GitLab, 타사 행사 관리 솔루션 등 다수의 플랫폼인 사용하여 관리와 비용 등의 이슈가 생겼습니다. 이에 고객은 서비스를 개발하는데 있어 개발 과정을 간소화하여 비용 절감 및 시간 절약을 통해 업무용 개선해야 했습니다. 오늘날 GitLab은 DevOps뿐만 아니라 보안 영역에 서비스도 제공하고 있으며, 위의 모든 플랫폼을 한번에 활용하고 고객의 이슈를 해결할 수 있는 GitLab을 선택하였습니다. 고객이 GitLab을 선택한 사유는 다음과 같습니다.

첫째로 고객은 DevOps 환경에서 Jenkins, Spinnaker, SVN, GitLab, 타사 행사 관리 솔루션 등 다수의 플랫폼인 사용으로 인해 솔루션 간 연동 및 관리에 복잡함을 느꼈고 GitLab이 통합 플랫폼 역할을 하면서 관리 간소화 및 업무 효율화를 이뤘습니다.  
둘째로 고객사 Legacy 시스템(ITSM) - 타사 행사 관리 솔루션\* 기반에서 GitLab 도입 시 이중 행사 관리 솔루션 적용으로 개발 소스 중복 관리와 이로 인한 장애 재현성 증가에 따라 ITSM - GitLab 연동으로 코드 관리나 버전 관리 같은 행사 관리 환경을 단일화하였습니다.

셋째로 고객사 Legacy 시스템(ITSM) - 타사 행사 관리 솔루션\* 환경에서 Legacy 컴플라이언스 체계의 확장성 한계에 부딪혔고, GitLab에서 제공하는 컨테이너 보안, 오픈소스 의존성(라이센스) 정책 및 취약성 검토 기반의 DevOps 관리 환경을 고도화하였습니다. 이로써 고객은 GitLab을 활용하여 개발 플랫폼 환경을 통합하고 DevOps 관리를 단일화할 수 있었습니다.

