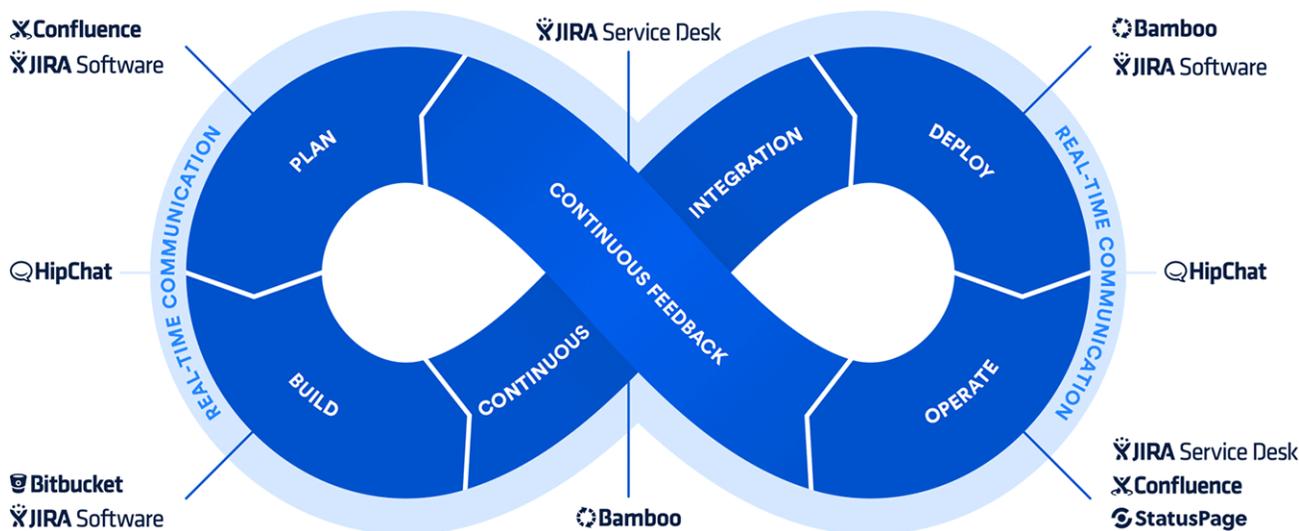


How to choose the right DevOps tools

세상 어떤 도구도 마술처럼 DevOps를 만들 수 없습니다. DevOps는 개발 팀과 운영 팀 간의 협업과 의사 소통을 통해 이뤄지는 거라 도구보다는 문화적인 변화가 필요합니다.

그러나 DevOps를 위해 자동화와 협업을 지원하는 기술과 도구가 있습니다. 사람들은 Atlassian에서 DevOps를 위해 일하는 방법을 지원하는 도구에 대해서 물어봅니다. 그래서 이 블로그에는 DevOps 도구를 선택하기 위한 가이드를 작성했습니다.

DevOps 도구들에 관하여 생각할때, 단계를 나눠서 생각하는 것이 도움이 됩니다. 이것은 계획(Plan), 빌드(Build), 지속적인 통합(Continuous integration), 배포(Deploy), 운영(Operate), 지속적인 피드백(Continuous feedback)과 같이 나눠집니다.



01 Plan

Collaborate on vision and design

스프린트 계획을 지원하는 도구가 필요합니다. 애자일 핸드북에 따라 반복 계획을 세울 수 있는 도구를 권장합니다. 사용자들은 빠르게 학습하고 피드백을 통해 제품을 최적화할 수 있습니다.

"비동기 브레인스토밍"을 지원하는 도구가 필요합니다. 모든 사람들은 아이디어, 전략, 목표, 요구사항, 로드맵 및 문서 등 무엇이든 공유하고 의견을 말할 수 있어야 합니다.

마지막으로 통합을 잊으면 안됩니다. 기능이나 프로젝트의 범위를 결정할때 개발 백로그에서 사용자 스토리로 변환해야 합니다.

- 사용도구 : [Confluence](#), [Hipchat](#), [Jira Software](#)

02 Build

Staging environments for development

Puppet과 Chef는 운영에 도움이되지만 개발자는 Docker와 같은 도구를 사용하여 개별 개발 환경을 제공하기를 원합니다. 코딩을 위해 제품에 대한 가상 환경은 더 많은 작업을 수행하는 데 도움이 됩니다.

Infrastructure as code를 통해 빠르고 안정적인 개발 환경을 제공할 수 있습니다. 이는 또한 개발 환경을 빠르게 변화할 수 있습니다. 전체 팀이 동일하게 프로비저닝 된 환경에서 작업할 때, 정말 정상적으로 동작하는 것을 알 수 있습니다.

- 사용도구 : [Docker](#)

Infrastructure as code

개발자는 안정적이고 유지보수 하기 좋기 때문에 어플리케이션을 모듈화하여 만듭니다. 이러한 생각을 IT 인프라로 확장하는 것이 어떻겠습니까?

IT 인프라를 모듈화하는 것은 인프라가 항상 변화하기 때문에 시스템 적용이 어렵습니다. 하지만 프로비저닝을 위한 코드를 사용하여 이를 해결할 수 있습니다. 프로비저닝 코드는 베어 메탈에 적용할 수 있으며 재적용하여 서버를 베이스라인으로 돌릴 수 있습니다.

이것은 버전 제어에 저장될 수 있습니다. 이것은 테스트될 수 있습니다. CI와 통합되고 Peer-reviewed될 수 있습니다.

- 사용도구 : [Bamboo](#), [Bitbucket](#), [Chef](#), [Docker](#), [Puppet](#)

Collaborative coding

프로덕션에 배포하기전에 변경 승인을 기다리는 대신, Pull request를 통한 Peer Review로 코드 품질과 처리량을 향상시킬 수 있습니다.

Pull request가 무엇인가 당신을 물어볼 수 있습니다. Pull request는 개발 브랜치에 코드 변경을 푸시하는 것을 말합니다. 당신의 팀은 메인 코드 라인에 통합 전 제안된 변경을 리뷰하고 수정사항을 토론할 수 있습니다.

03 Continuous integration

Continuous Integration

지속적인 통합은 하루에 여러번 코드를 공유 저장소에 체크인하고 테스트하는 프랙티스입니다. 이를 통해 문제를 조기에 발견하고 문제를 쉽게 해결할 수 있으며, 가능한 빨리 사용자에게 새 기능을 제공합니다. CI 도구는 브랜치-앤-머지 워크플로우와 멀티 브랜치 환경을 지원할 수 있어야 합니다. 개발 브랜치에서 테스트를 자동으로 수행하고 빌드가 성공되면 마스터로 푸시할 수 있는 도구가 필요합니다. 그리고 이 결과를 채팅 도구에서 실시간으로 알림을 받을 수 있으면 좋습니다.

- 사용도구 : [Bamboo](#), [Hipchat](#)

Automated Testing

자동화된 테스트는 장기적으로 개발 및 테스트 사이클을 가속화하여 시간이 지남에 따라 보장됩니다. 그리고 또 다른 이유로 DevOps 환경에서 자동화된 테스트는 중요합니다.

수동 테스트와 달리, 자동화된 테스트는 매번 동일하게 엄격함으로 충실히 수행됩니다. 또한 위험한 영역을 식별하는데 도움이되는 보고서 및 추세 그래프를 생성합니다.

위험 요소는 소프트웨어의 실제 요소이지만 예상할 수 없는 요소는 완화할 수 없습니다.

- 사용도구 : [Bamboo](#), [Bitbucket](#)

04 Deploy

Release dashboards

배포되는 소프트웨어의 가장 스트레스가 많은 부분 중 하나는 릴리스의 모든 변경, 테스트, 배포 정보를 한 곳으로 가져오는 것입니다.

코드 저장소와 배포 도구와 통합된 단일 대시보드를 가진 도구를 찾으세요. 하나의 장소에서 브랜치, 빌드, Pull Request, 배포 경고를 전체 가시화할 수 있는 도구를 찾으세요.

- 사용도구 : [Jira Software](#)

Automated deployment

모든 어플리케이션과 IT 환경에서 사용할 수 있는 배포 자동화를 위한 매직 레시피는 없습니다.

Puppet 및 Chef와 같은 프로비저닝 도구는 표준화 환경의 고통을 줄여줍니다. 그리고 배포 자동화를 돕는 많은 도구가 있습니다. Atlassian의 Bamboo는 복잡한 배치를 단계별로 조정하고 각 환경의 히스토리에 대한 가시성을 제공합니다.

- 사용도구 : [AWS](#), [Bamboo](#), [Hipchat](#), [Puppet](#)

05 Operate

Application and server performance monitoring

자동화되어야하는 모니터링에는 서버 모니터링과 어플리케이션 성능 모니터링의 두 가지 유형이 있습니다.

어플리케이션의 전체 건강과 트렌드를 이해하기 위해서, 당신은 24/7 동안 데이터를 보고 레코딩할 필요가 있습니다.

이를 지원하는 여러 도구들이 있습니다. 이러한 도구들은 그룹 채팅 도구와 통합하여 팀의 룬에 알람을 줄 수 있어야 합니다.

- 사용도구 : [BigPanda](#), [Hipchat](#), [HostedGraphite](#), [Nagios](#), [New Relic](#), [Pager Duty](#), [Pingdom](#), [Splunk](#)

Communication and swarming

팀 간의 의사 소통은 문화적 변화를 향한 첫 걸음이며, 채팅 도구는 실시간으로 이를 용이하게 합니다. 채팅 도구를 통해 알람을 받고 전문가들은 빠르게 문제를 해결할 수 있습니다.

또한 가동 시간을 극대화함으로써 최신 정보를 유지하는 것이 중요합니다. 확장 가능한 채팅 도구를 찾고 모니터링 도구와 통합하여 중요한 서비스 저하 경고를 놓치지 마세요.

- 사용도구 : [BigPanda](#), [DataDog](#), [Hipchat](#), [New Relic](#), [Pager Duty](#), [Statuspage](#)

Incident, change and problem tracking

팀들간의 협업을 연락킹하기 위한 키는

인시던트들이 보고되면 무슨일이 일어날까? 소프트웨어 문제를 위한 링크되고 추적할 수 있을까요? 변경이 만들어질 때, 릴리즈와 연결될수 있는가요?

인시던트, 변경, 문제점 및 소프트웨어 프로젝트를 하나의 플랫폼에서 관리하여 문제를 보다 빠르게 식별하고 수정할 수 있는 도구를 찾으세요.

- 사용도구 : [Jira Service Desk](#), [Jira Software](#)

06 Continuous feedback

Better products through user feedback

고객은 제품에 대한 말하고 있습니다. 여기에는 NPS 데이터, 고객 설문 조사, 버그 보고서, 지원 티켓 및 트윗이 포함됩니다.

DevOps 문화에서 팀의 모두가 사용자 코멘트에 접근할 수 있습니다. 릴리즈 계획부터 탐색적인 테스트 세션까지 모든 것을 가이드하는 것을 도울 수 있기 때문입니다.

NPS 스타일 피드백을 지원하는 서베이 플랫폼과 채팅 도구와 통합되는 어플리케이션을 찾아라. Twitter, Facebook은 또한 실시간 피드백을 위한 채팅과 통합될 수 있다.

소셜미디어에서 오는 피드백에 대해 자세히 살펴보면 히스토리 데이터를 사용하여 보고서를 가져올 수 있는 소셜미디어 관리 플랫폼에 투자하는 것이 좋습니다.

분석 및 통합 피드백은 단기간에 개발 속도를 늦추는 것처럼 느낄 수 있지만, 아무도 바라지 않는 새로운 기능을 출시하는 것보다 장기적으로 더 효율적입니다.

- 사용도구 : [GetFeedback](#), [Hipchat](#), [Jira Service Desk](#), [Pendo](#), [SurveyMonkey](#), [Hootsuite](#)

Perfacting your DevOps tools collection

Atlassian은 개발 라이프 사이클의 각 단계에서 팀 간 공동 작업을 지원하는 도구를 만듭니다.

이 블로그 게시물은 Atlassian Sarah Zorah의 블로그 포스트를 저 마음대로 의역하였습니다. 원본은 아래 링크를 통해 확인하세요.

<https://www.atlassian.com/blog/devops/how-to-choose-devops-tools>