DevOps

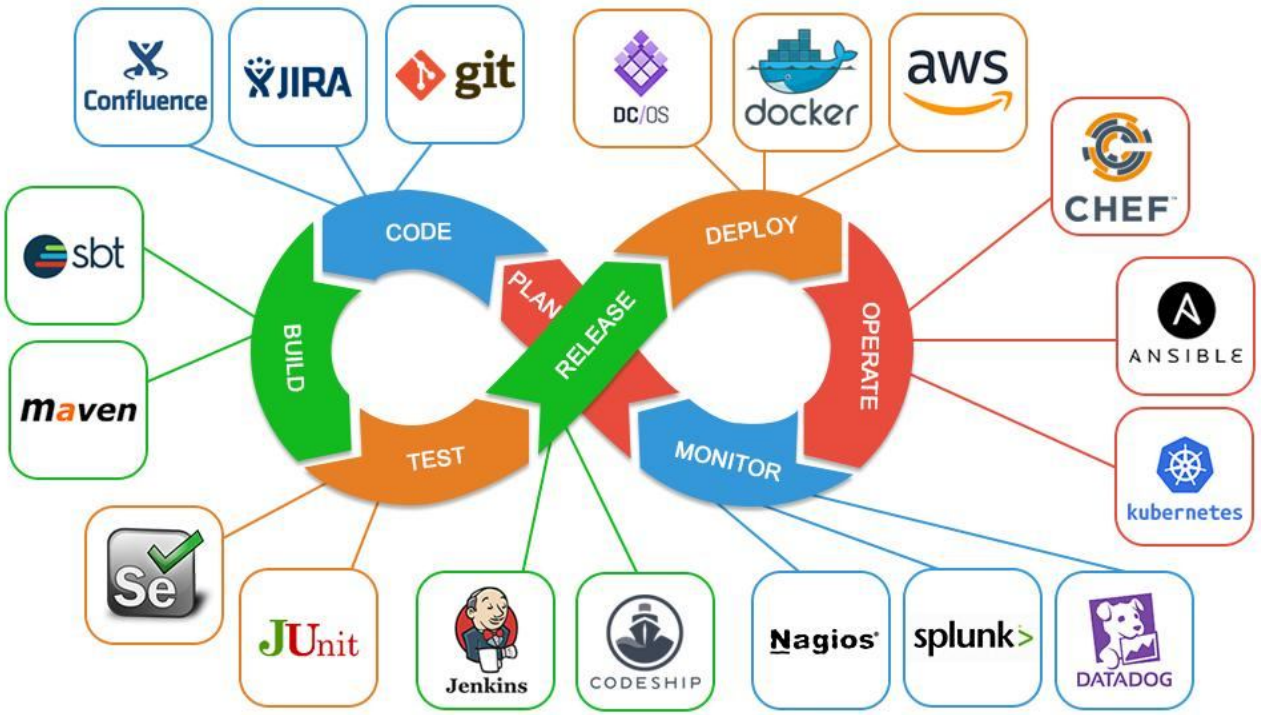
- 프로세스 자동화를 실현하기 위한 대표적 실천 프레임워크(또는 접근 방식)
- 개발, 통합, 테스트, 배포, 모니터링, 피드백 및 운영과 같은 자동화를 통해 "모든 것이 연속적으로" 진행되는 프로세스에서 발생
- DevOps 라이프 사이클에서 제공하는 즉각적인 피드백 은 팀이 더 민첩하게 작업할 수 있으며, 외부 문제로부터 팀을 보호할 수 있음.

DevOps Life Cycle

- 1단계 : 지속적인 개발 및 제공
- 2단계 : 지속적인 통합
- 3단계 : 지속적인 테스트
- 4단계 : 지속적인 모니터링
- 5단계 : 지속적인 피드백
- 6단계 : 지속적인 배포
- 7단계 : 지속적인 운영

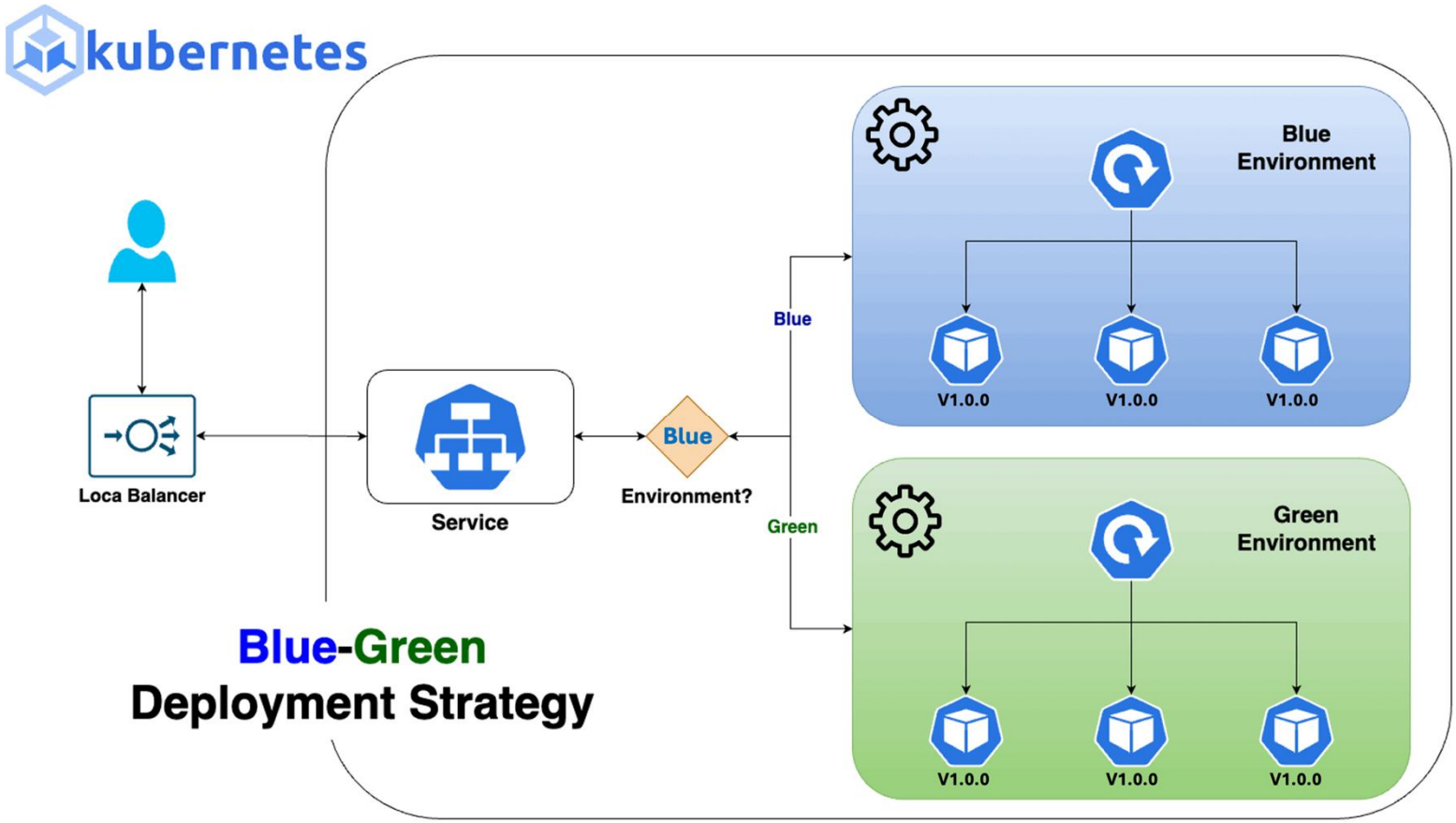
DevOps Tool Chain

소프트웨어를 개발, 배포, 운영, 개선 하는 전 과정을 자동화 도구로 연결한 하나의 연속된 파이프라인(또는 일련의 과정 및 시스템)



Code → Build → Test → Release → Deploy → Operate → Monitor

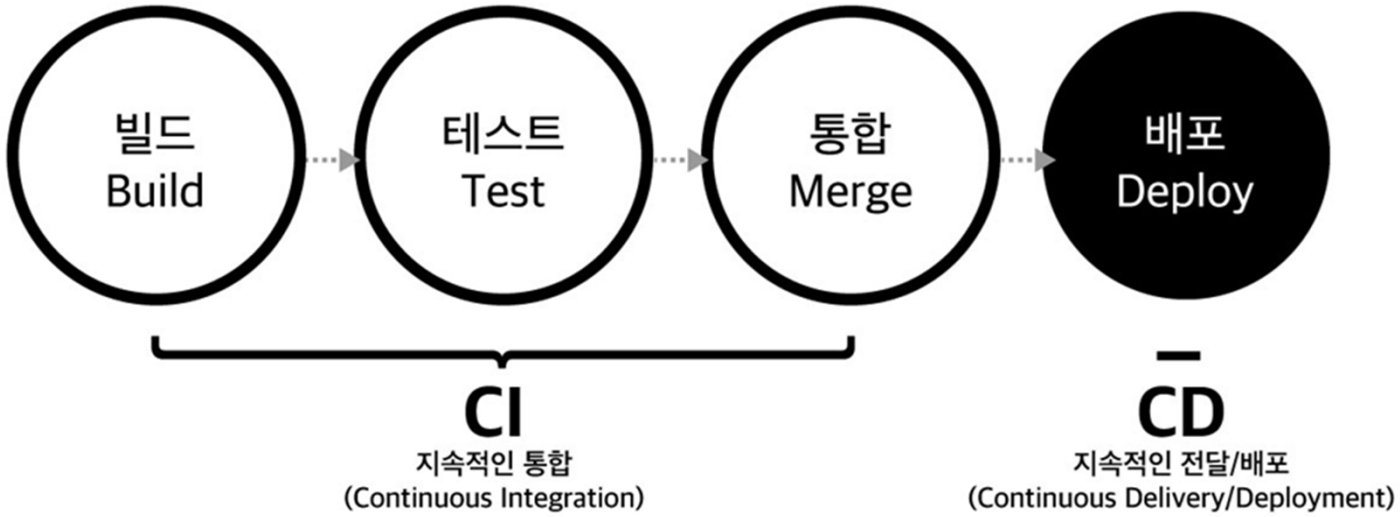
Plan	요구사항과 목표를 정의하고 개발·운영 전략을 수립하는 단계
Code	개발자가 애플리케이션 소스 코드를 작성하고 관리하는 단계, 협업, 변경 이력 관리, 코드 품질 유지
Build	소스 코드를 실행 가능한 형태로 변환하는 과정 3가지 빌드 타입 컴파일 : 소스 코드를 기계가 실행 가능한 코드로 변환 패키징 : 컴파일 결과물, 설정, 라이브러리를 하나의 배포 단위로 묶음 (배포 및 전달 용이성) 컨테이너 이미지 생성 : 애플리케이션, 라이브러리, 실행 환경을 컨테이너로 실행 가능하도록 이미지화
Test	빌드된 코드가 정상 동작하는지 자동 검증
Release	검증된 빌드 결과물을 배포 가능한 상태로 승인
Deploy	애플리케이션을 실제 실행 환경에 배포(서비스 제공, 무중단 배포) 배포방식 - Rolling update : 순차교차 - Blue/Green : 환경 2개 운영 - Canary : 일부 사용자 대상
Operate	배포된 시스템을 안정적으로 유지·관리 목적: 가용성 유지, 장애 대응, 리소스 최적화 주요 운영 작업 : 오토 스케일링, 장애 복구(Self-healing), 설정 변경, 보안 패치
Monitor	시스템 상태를 지속적으로 관찰·분석 장애 조기 탐지, 성능 최적화, 피드백 제공



동일한 서비스 환경을 두 세트(Blue, Green)로 동시에 유지하면서 트래픽만 선택적으로 전환하는 방식

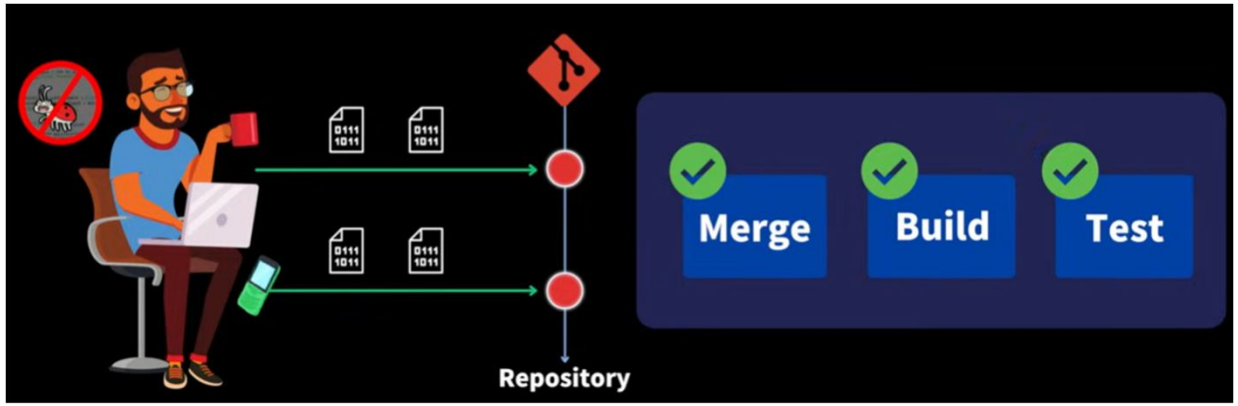
Blue	현재 사용자에게 서비스 중인 운영 환경
Green	새 버전을 배포·검증하는 대기 환경

CI/CD 파이프라인



- 애플리케이션 개발단계에서 배포까지 모든 단계들을 자동화를 수행
- 좀 더 효율적으로 빠르게 사용자에게 빈번이 배포할 수 있게 만드는 것

CI (Continuous Integration, 지속적 통합)



개발 여러 명의 개발자가 개발

통합 소스 코드를 하나로 통합

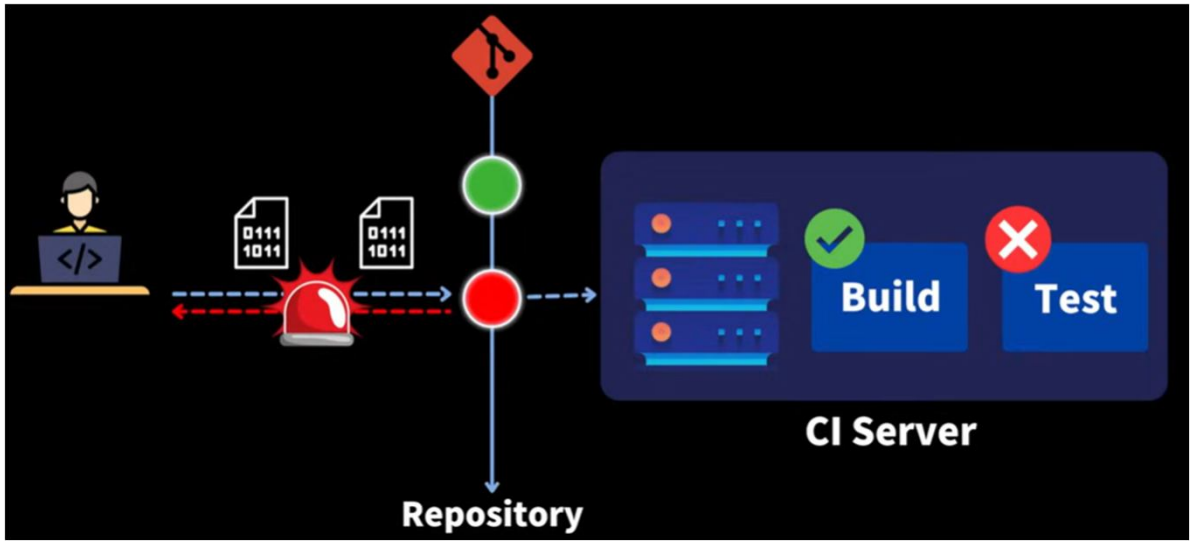
저장 '소스 저장소'에 저장
(GitHub, GitLab, Bitbucket 등)

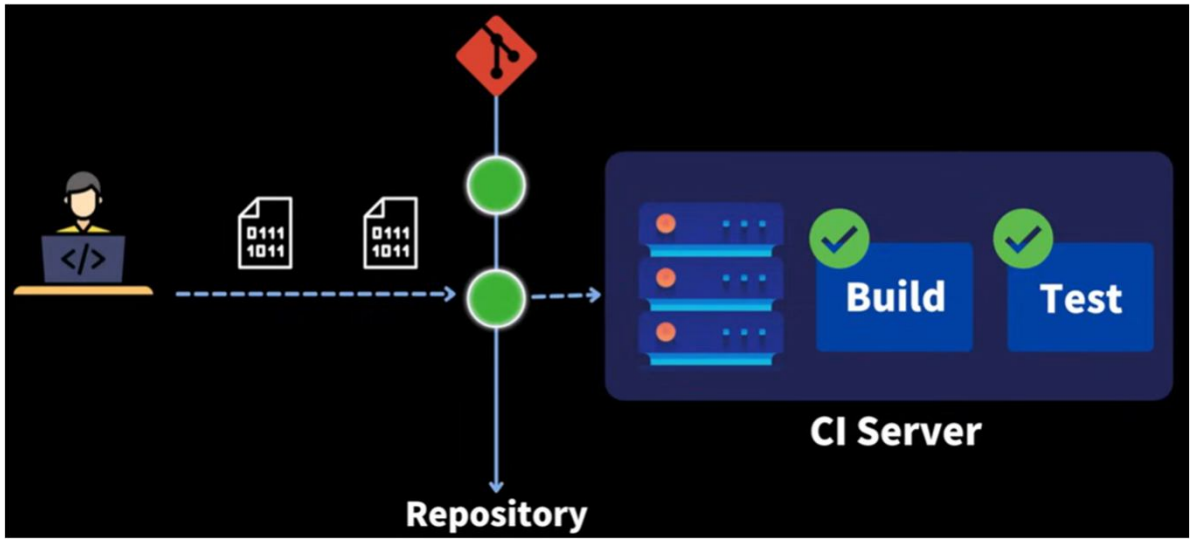
빌드 실행 파일(Artifact) 생성

테스트 실행 파일의 동작 여부 점검

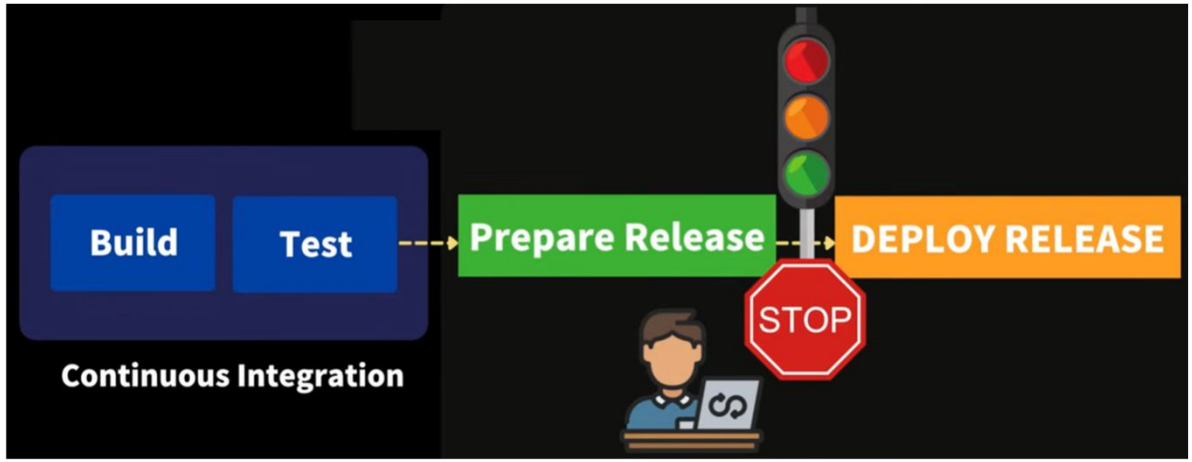
CI(지속적 통합) 목표

- 버그를 신속하게 해결
- 소프트웨어 품질을 높일 수 있음
- 새로운 소프트웨어 업데이트를 검증 및 릴리스 하는데 시간이 적게 걸림





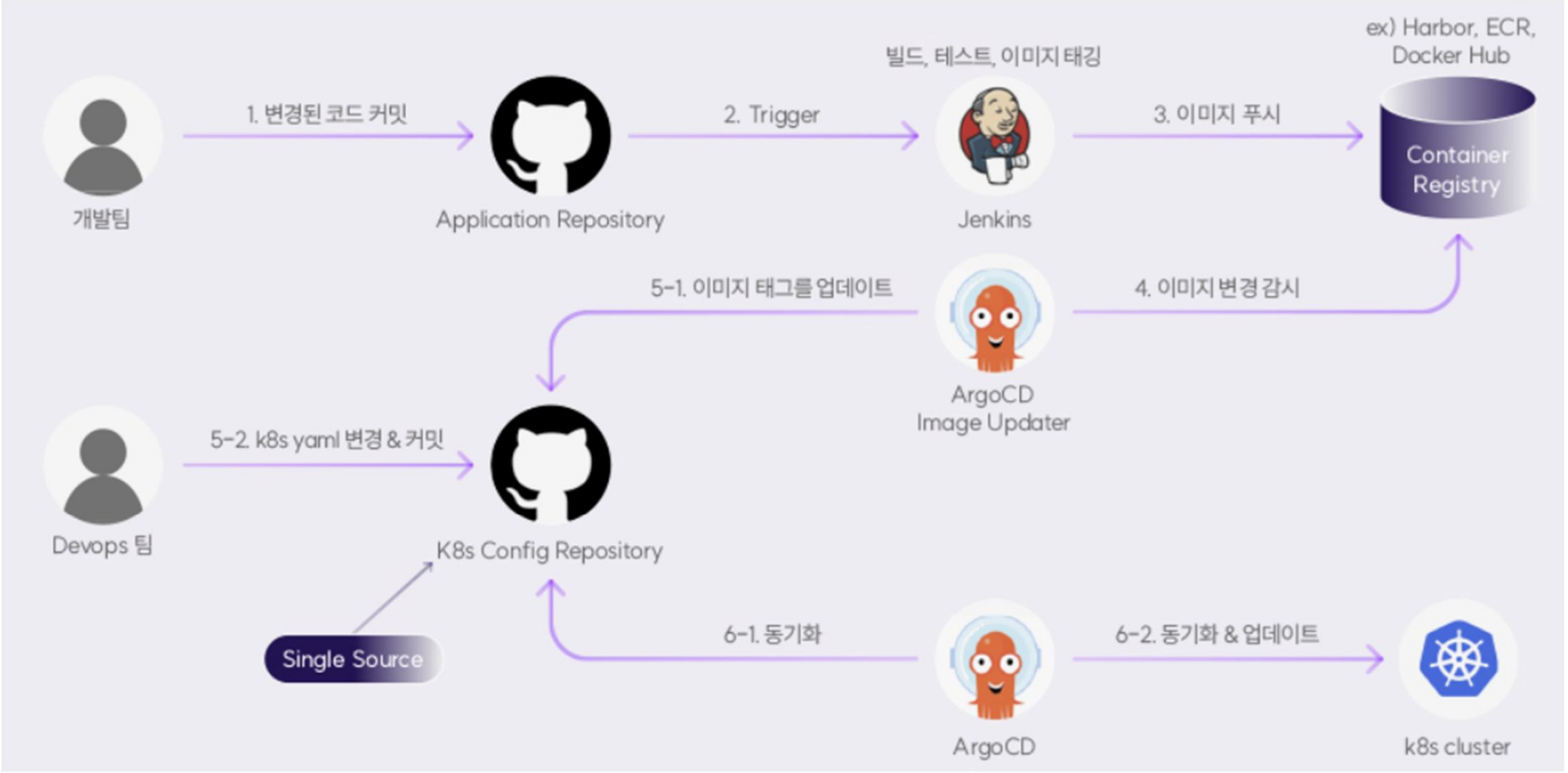
CD(Continuous Delivery/Deployment, 지속적 배포)



CD(Continuous Delivery) : 개발자 또는 관리자가 release를 수동으로 검증 후 수동적 배포

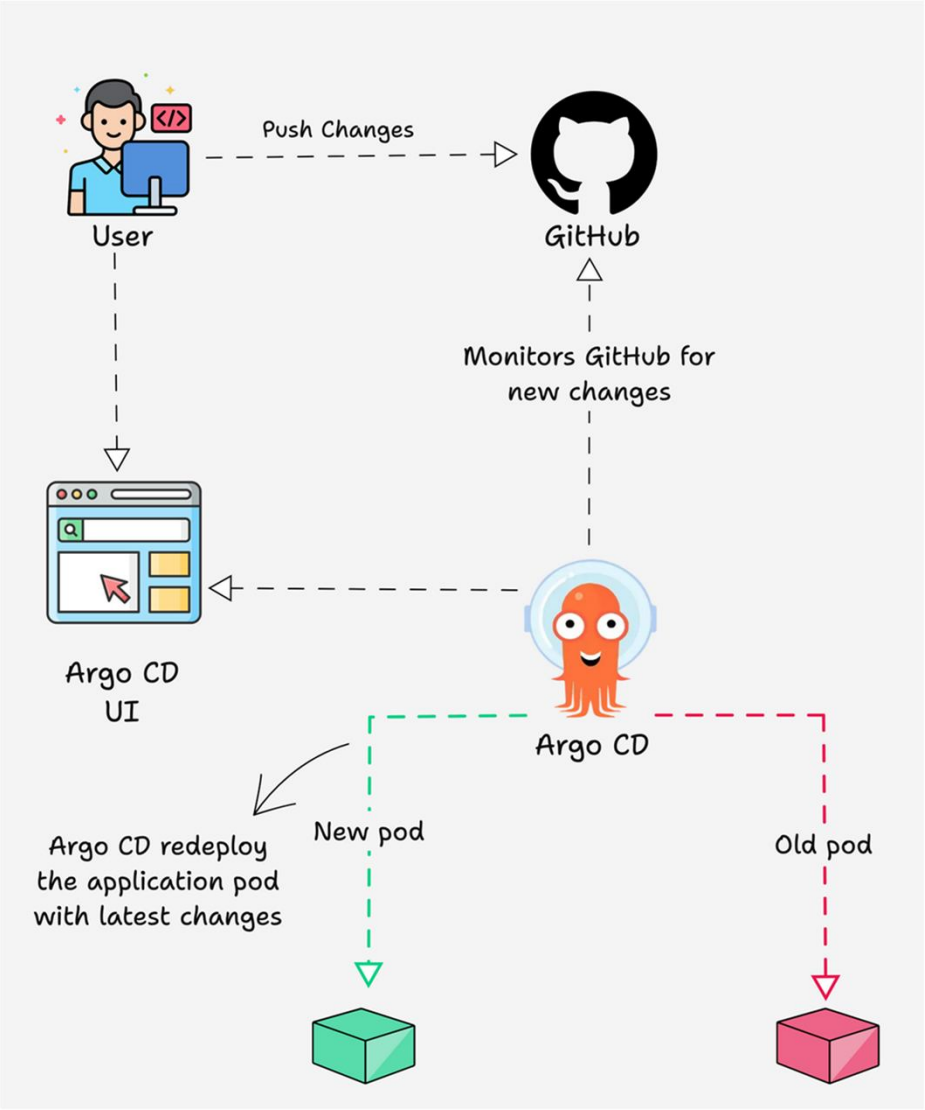
CD(Continuous Deployment) : release 준비 후 사용자에게 자동으로 배포

K8S 기반의 CI/CD

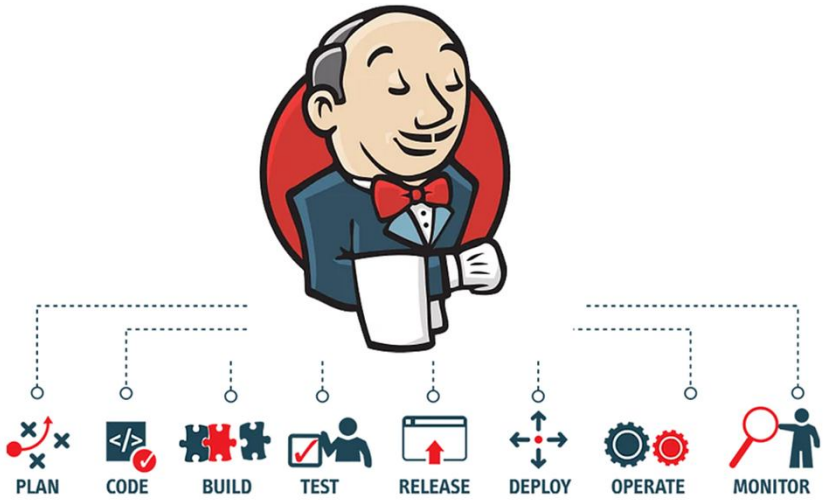


Jenkins와 ArgoCD를 이용한 CI/CD 파이프라인 예시

ArgoCD를 이용한 CD 파이프라인 예시



Jenkins



- 소프트웨어 개발 시 CI(지속적 통합) 서비스를 제공하는 툴

ArgoCD

- GitOps 기반의 CD(Continuous delivery) 도구
  *Git에 선언된 상태를 기준으로 시스템을 자동 배포·운영하는 방식

CI/CD를 위한 K8S

[개발자] → [CI/CD] → [컨테이너 이미지] → [K8s 클러스터] → [서비스]

구분	Jenkins	Argo CD	Kubernetes
주 역할	**CI 자동화**	**CD 배포(GitOps)**	**실행/운영 플랫폼**
핵심 기능	빌드·테스트·파이프라인	Git 기준 배포·동기화	컨테이너 실행·확장
파이프라인 위치	CI 단계	CD 단계	배포 대상
배포 방식	Push 기반	**Pull 기반**	N/A
UI	있음	있음	제한적