

## **Module 3. CI/CD 변화**

3.1 CI/CD 요소의 변화

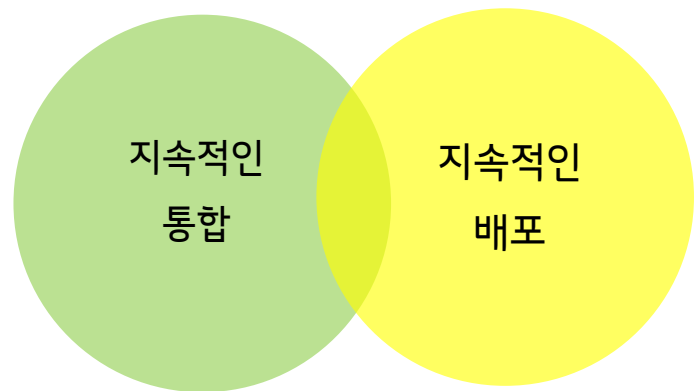
3.2 자동화 범위의 변화

3.3 인프라 관리 방식의 변화

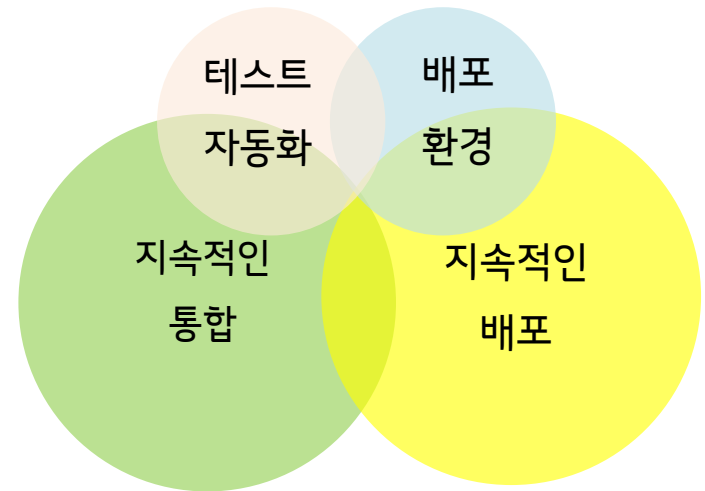
3.4 과정의 변화

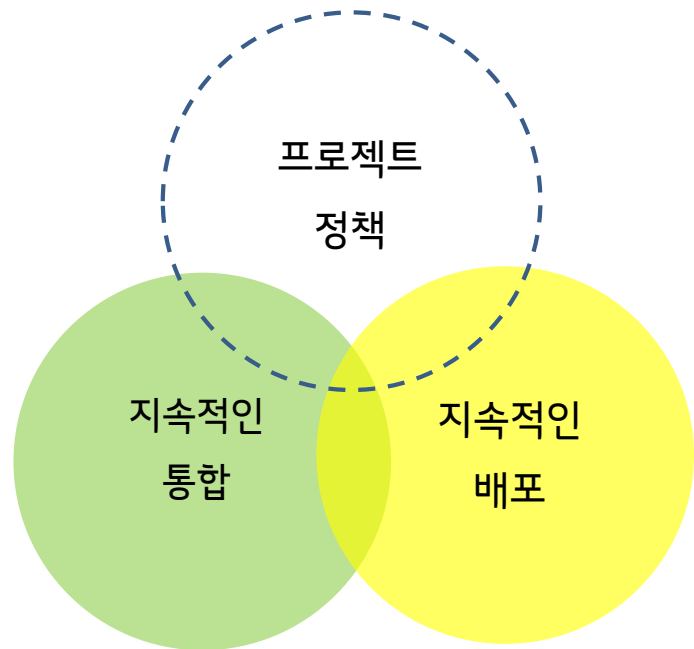
3.5 구성의 변화

Summary

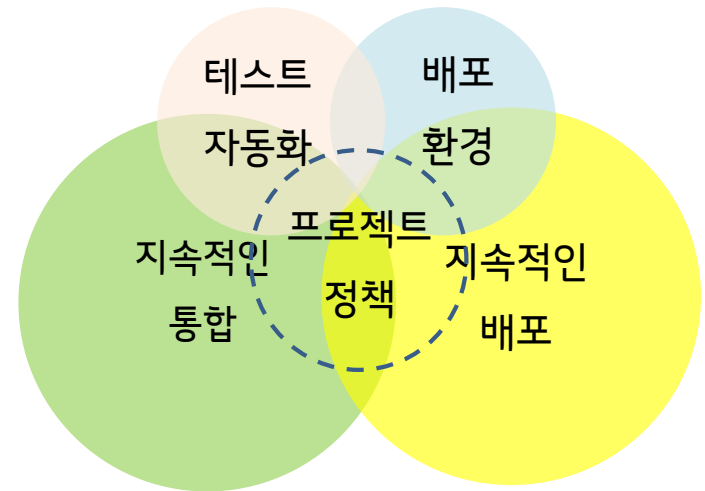


**VS**



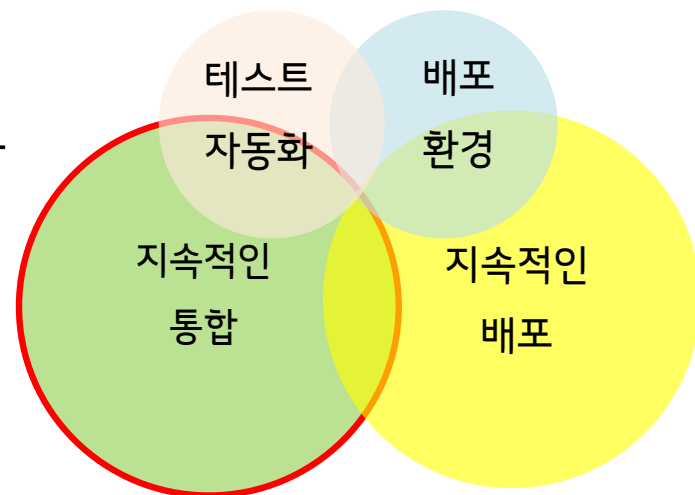


**VS**



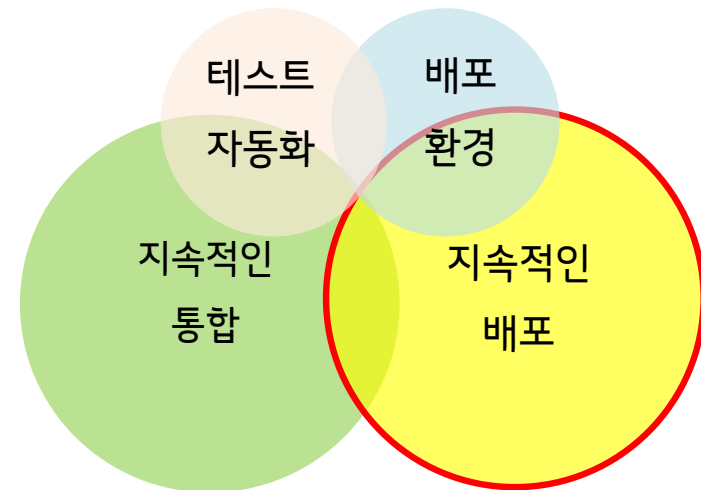
### ○ 지속적인 통합

- 빌드 트리거는 정책 결정  
(자동-스케줄 빌드, webhook / 수동)
- Commit되면 Integration 바로 시작(webhook)
- webhook 사용 시, 스케줄 빌드보다 빠른 피드백 가능
- Pull Request를 통한 merge 승인
- 빠른 버전 출시를 위해 테스트자동화의 중요도 증가
- Integration 완료되면 신뢰성 있는 신규 릴리즈 버전 보유
- 빌드 실패 시, 빠른 피드백(slack, SMTP 등)



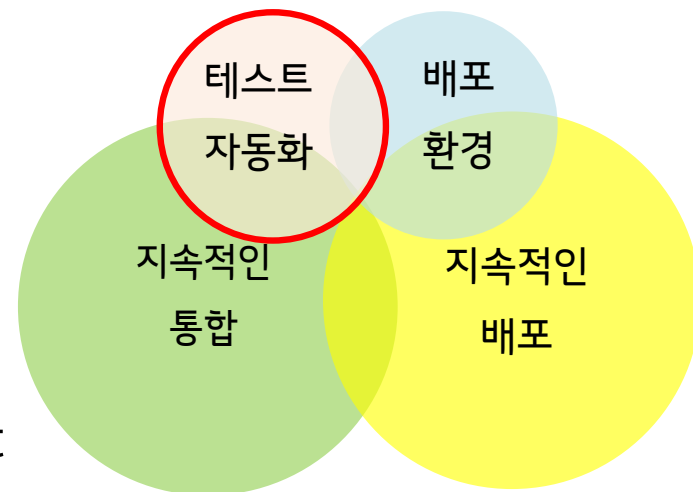
### ○ 지속적인 배포

- 배포환경으로의 어플리케이션 배포
- 배포 승인
- 배포 롤백
- 다양한 배포 기법(롤링업데이트, 블루/그린, 카나리)



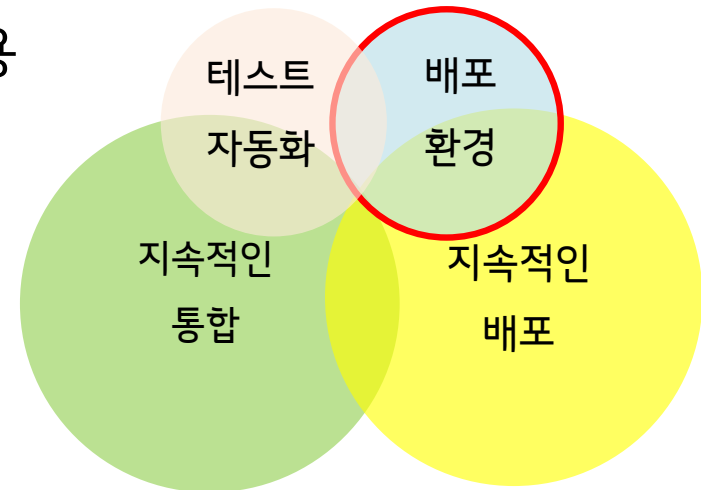
### ○ 테스트자동화

- 기존에도 테스트자동화 가능했음
- 개발부터 배포 중 가장 오래걸리는 과정 중 하나
- 사람의 손을 거치지 않을수록 배포 기간 단축
- 테스트자동화의 수준에 따라 배포 속도 좌우
- 단위테스트, E2E테스트, API테스트 등
- 테스트자동화를 통해 신뢰성 있는 어플리케이션 확보 시, 승인과정 생략 가능 → Continuous Deployment
- 테스트코드 작성 등 현실적인 어려움 여전히 존재



### ○ 배포환경

- Lambda 및 정적파일 등 배포자원 형태 및 배포환경 다양, 컨테이너 환경 ↑
- Cloud에 배포환경 구성 시, 다양한 Cloud 서비스 사용 가능
- AWS EC2, ECS, EKS, S3, Lambda, Fargate 등
- GCP VM, GKE, Cloud Storage 등
- Public Cloud 사용 시, IaC를 통한 인프라 구성 가능

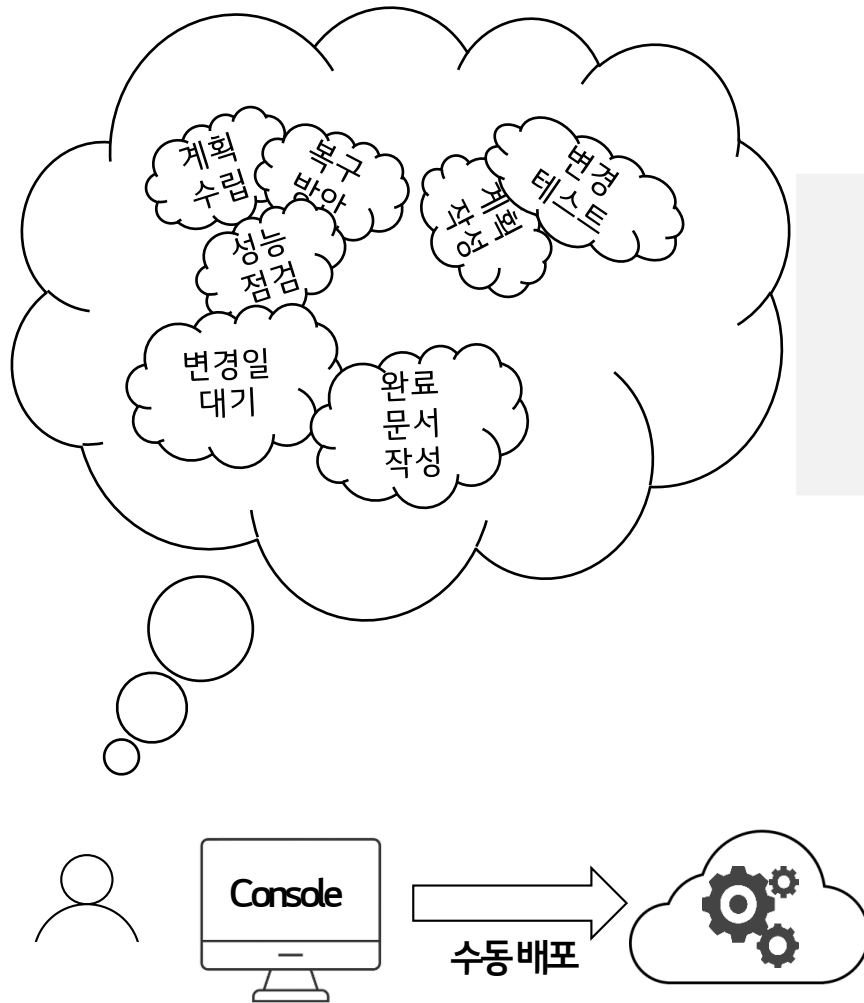




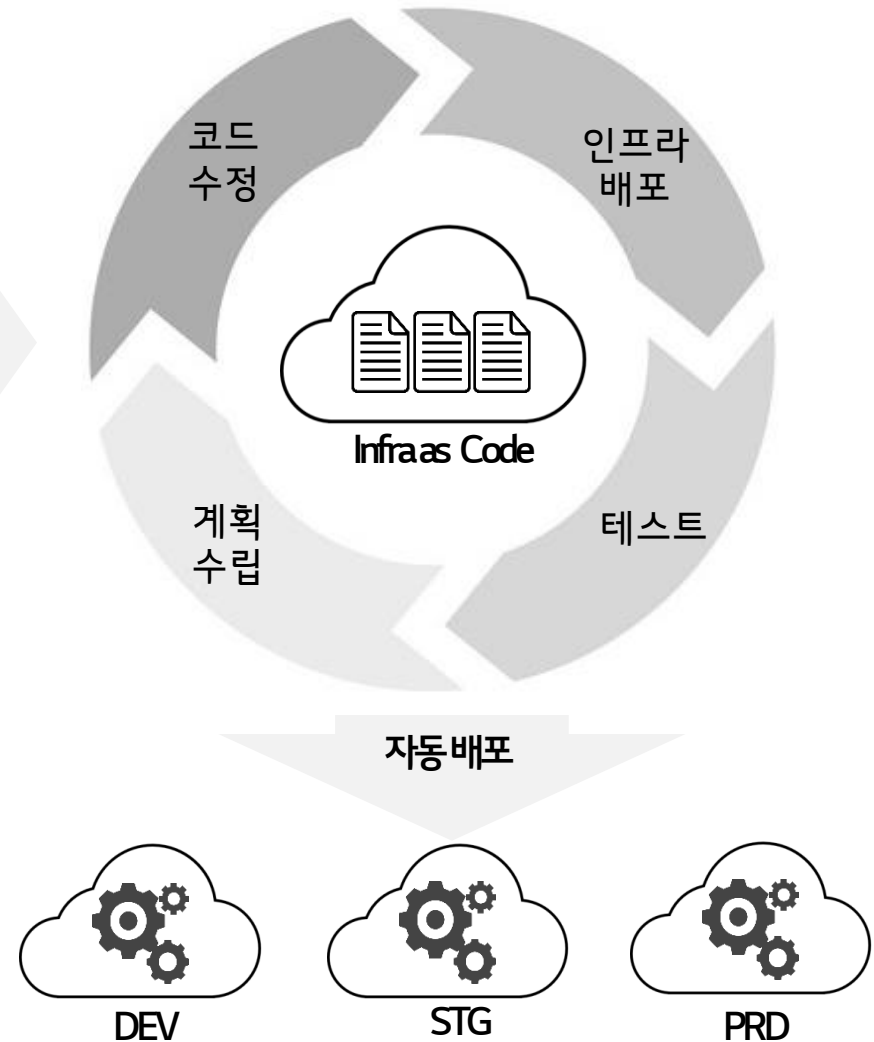
- 모놀리식은 인프라와 런타임의 변경이 어려우며 변경에 대한 제약사항이 다수 존재하여 어플리케이션 영역이 빌드/배포 자동화 대상이었음
- 컨테이너 등장으로 런타임 영역의 배포 자동화가 가능해짐 (Dockerfile)
- 클라우드 사용 시, IaC를 통한 인프라 구성이 가능해졌으며 자동화도 가능해짐
- 어플리케이션 자동화와 인프라 구성 자동화를 하나의 파이프라인으로 실행하여 서비스를 위해 필요한 리소스를 one step으로 구성 가능해짐
- 인프라 구성 자동화를 통해 인프라 구축 시간을 대폭 줄일 수 있음
- 인프라 구성 시간에 대한 waiting 타임을 줄일 수 있음
- 인프라 추가/변경 요구사항에 대한 대처 가능



#### ○ 기존 클라우드 관리 방식

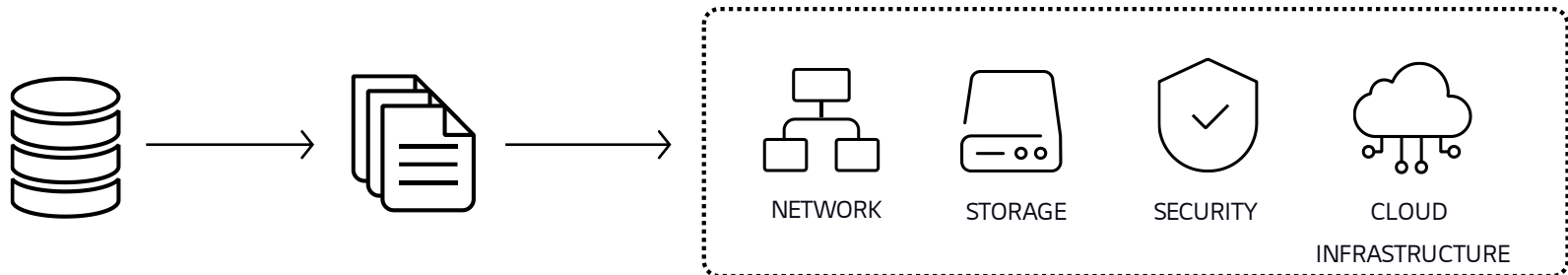


#### ○ 새로운 클라우드 관리 방식

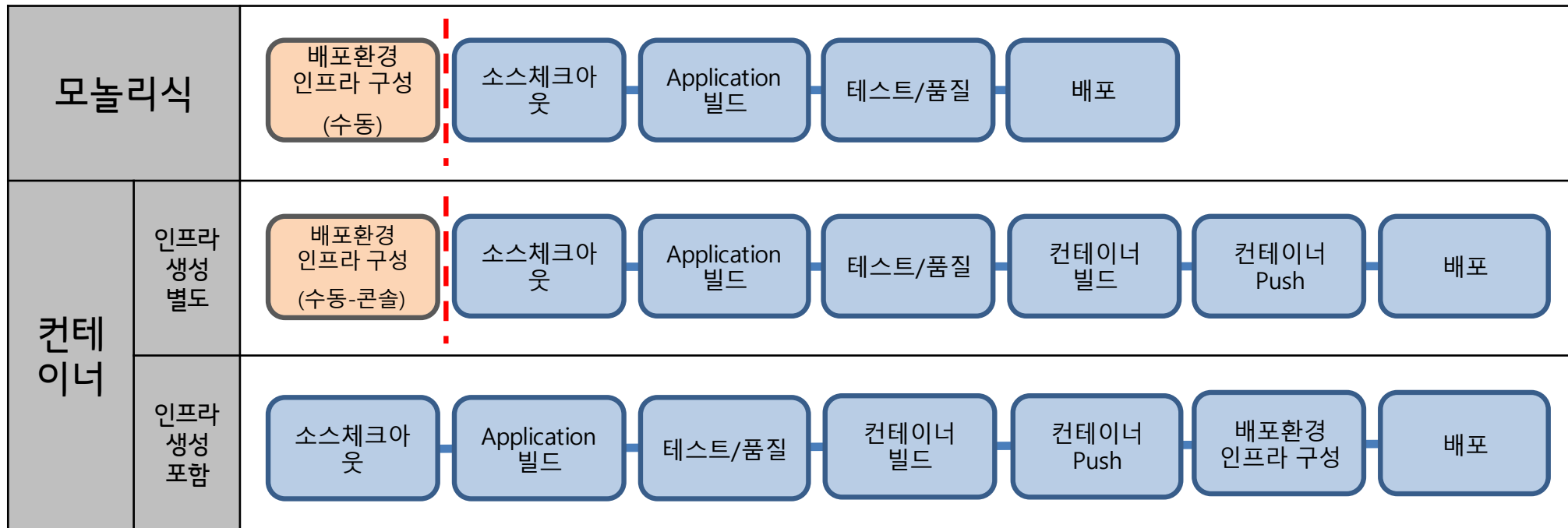


### ○ Infra as Code(IaC)

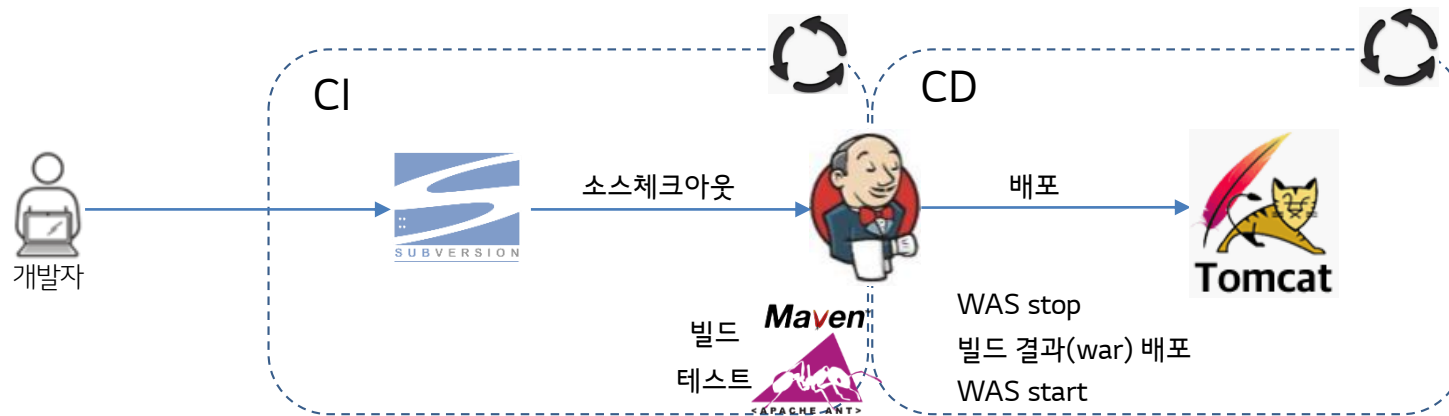
- 인프라에 대한 형상관리 가능
- 빠른 인프라 구축 및 제거(apply, destroy) (콘솔을 통한 수작업X)
- 인프라 변경에 유연함
- 코드의 재사용 가능
- 구성 관리를 위한 문서 작업 등의 시간 절약
- 휴먼에러 방지
- CI/CD 파이프라인에 포함하여 배포환경 인프라 구성 및 Application 배포까지 One Step으로 제공 가능(시너지 ↑)
- Cloud 리소스, 유지보수 비용 고려 필요



- 컨테이너 환경이 아닌 모놀리식 배포의 경우. 배포환경 구성 후 어플리케이션만 배포한다.
- 컨테이너 환경인 경우 인프라 생성을 별도로 한 후 컨테이너(어플리케이션)를 배포 하거나 인프라 생성을 CI/CD 파이프라인에 포함하여 진행할 수 있다.
- 인프라 생성 과정 포함여부는 클라우드 사용여부 및 파이프라인 구성 전략에 따라 선택 가능하다.
- 컨테이너 환경인 경우 컨테이너를 배포한다.



- 소스형상관리 : SVN 사용
- CI/CD 도구 : Jenkins
- Application 빌드 : Maven, Ant, gradle
- Deploy : Ant, shell
- 배포환경 : WEB/WAS



- 소스형상관리 : git 사용
- CI/CD 도구 : Jenkins 등 다양
- Application 빌드 : Maven, Ant, gradle
- Container Registry : 컨테이너 이미지 저장소
- Deploy : jenkins에서 Deploy 또는 Spinnaker 등 CD 도구의 파이프라인 별도 구성 가능
- 배포환경 : 컨테이너 환경(docker, k8s, ECS, EKS, GKE 등), S3, Lambda 다양화

