aws re/start

DevOps 및 지속적 통합

Python 기본 사항

발표자 이름

날짜

DevOps 소개

학습 내용

강의 핵심 내용

학습 내용:

- DevOps를 정의합니다.
- DevOps의 목표를 파악합니다.
- DevOps로 해결할 수 있는 문제를 파악합니다.
- DevOps의 문화를 설명합니다.

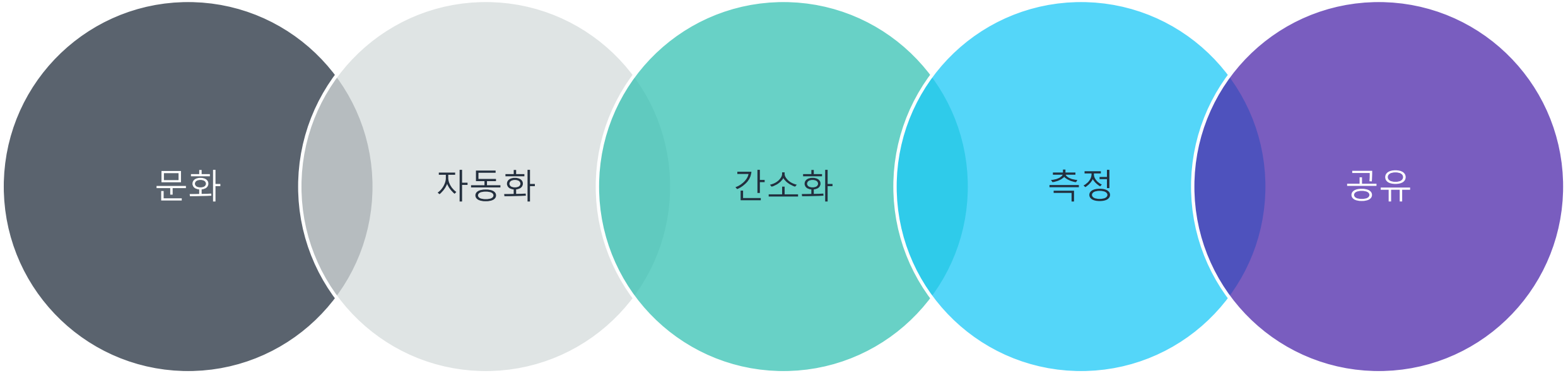
DevOps란?

DevOps란 소프트웨어 개발(Dev)과
소프트웨어 운영(Ops)의 통합을 목표로 하는
소프트웨어 엔지니어링 문화이자 실무
방식입니다.

DevOps로 이동하는 데 있어 가장 큰 특징은
소프트웨어 구축의 모든 단계에서 자동화와
모니터링을 강조하는 것입니다. 소프트웨어
구축 단계는 통합, 테스트, 배포 릴리스,
인프라 관리 등 다양합니다.

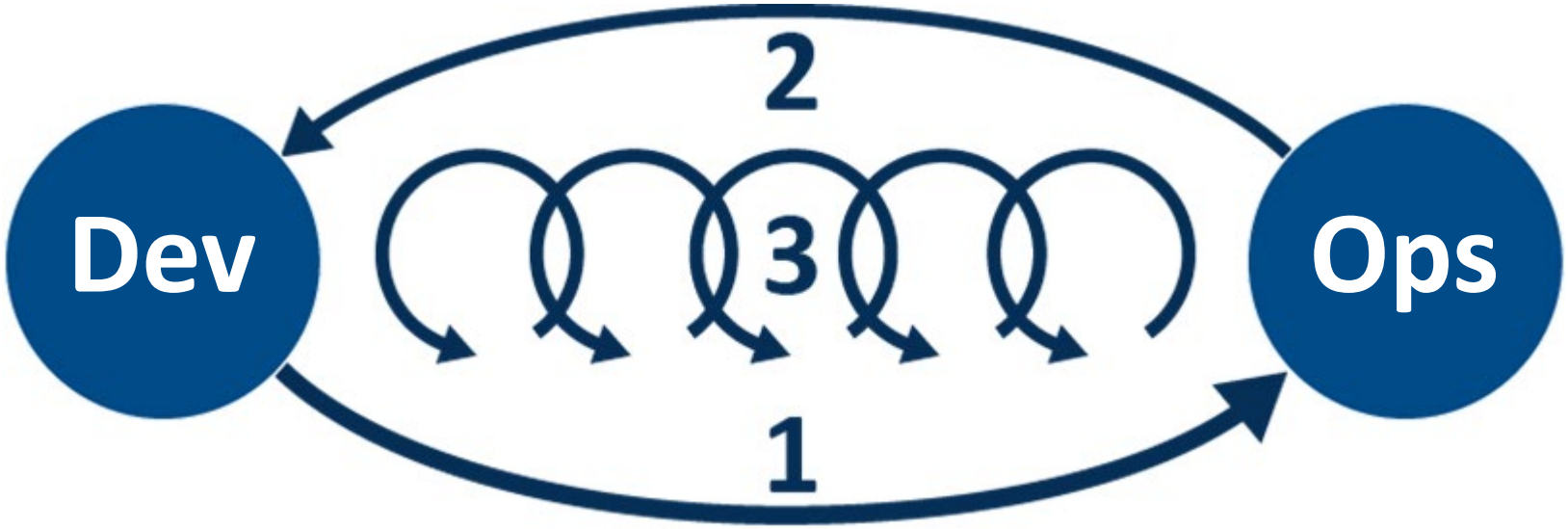
aws re/start

DevOps란(계속)?

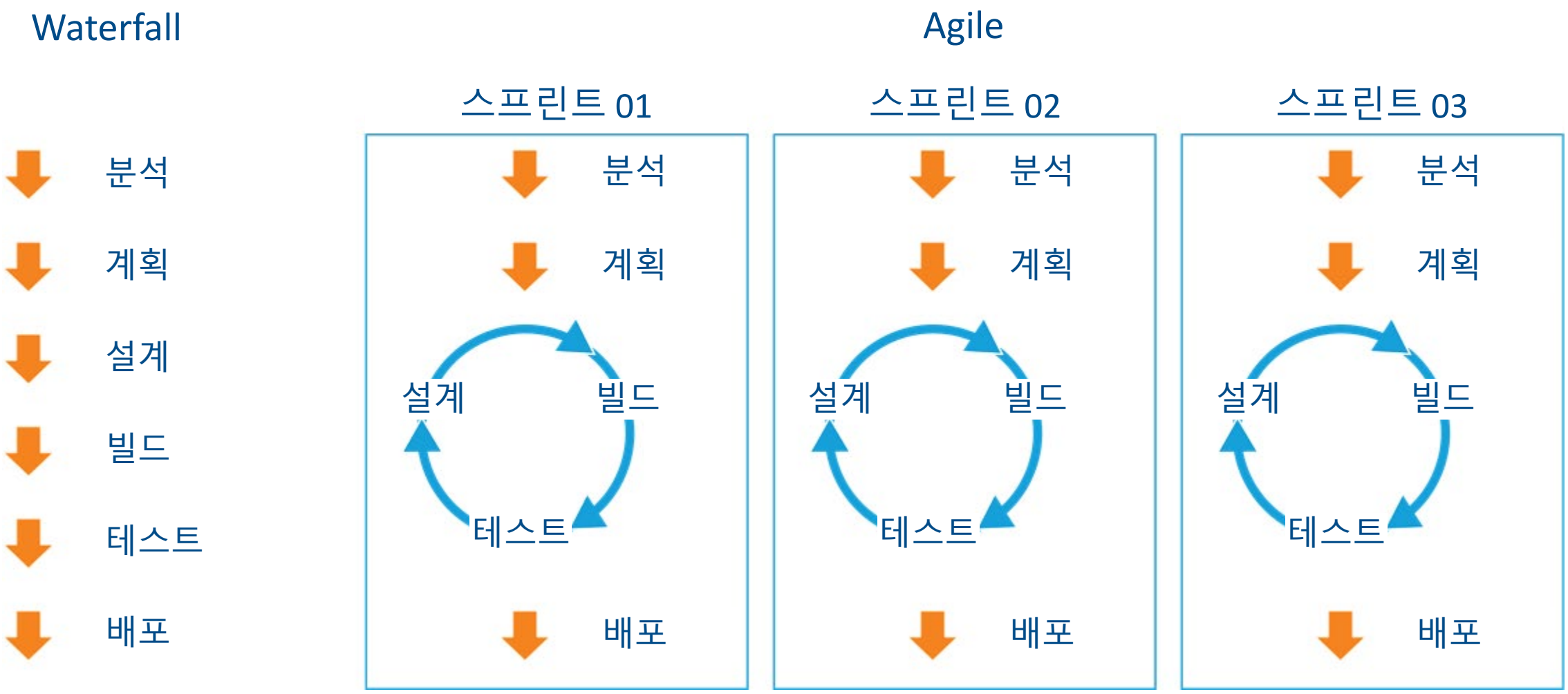


DevOps 문화

1. **시스템 전체 - 흐름**의 성능 고려
2. 오른쪽에서 왼쪽 방향으로 **피드백** 루프 생성 및 증폭
3. 다음과 같은 환경을 육성하는 문화 조성:
    a. 지속적인 실험, 위험 감수, **학습**
    b. **반복과 연습** - 숙련도의 사전 조건



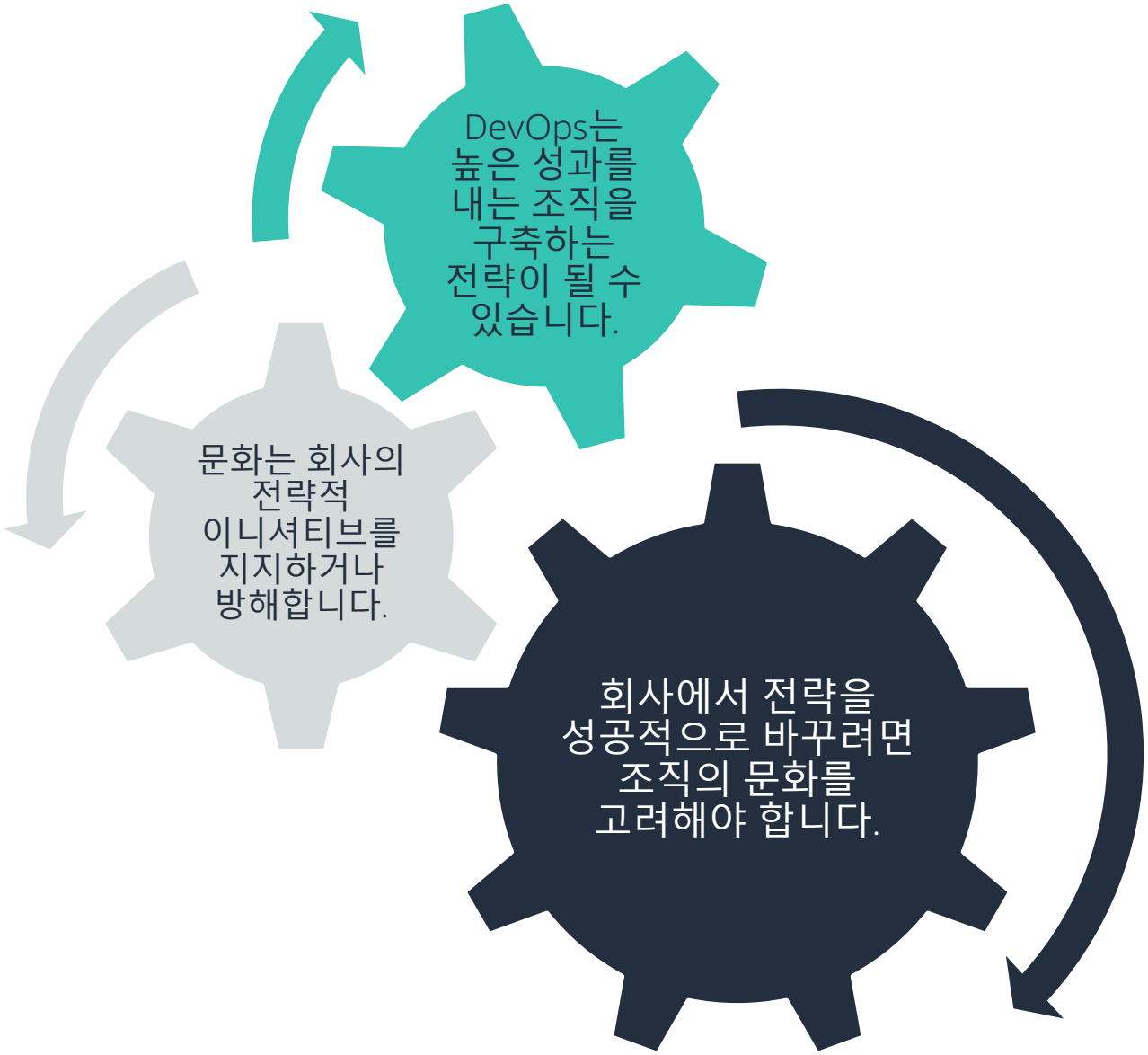
Waterfall과 Agile의 비교



DevOps의 목표

- 전통적인 IT, 소프트웨어 개발, 품질 보장(QA)의 간극을 좁히는 것이 DevOps의 목표입니다.
  - 초보자에게 가장 어려운 영역이 QA 부분입니다. 코드 모양이 중요합니다.
- 더 신속하고 유동적으로 일할 수 있도록 하는 것이 DevOps의 목표입니다.
  - 어려운 점은 QA와 보안을 더 빠르게 더 짧은 주기로 보다 잘 통합하는 일입니다 (Waterfall과 Agile의 비교 다이어그램 참조).
- 전문적인 개별 업무의 필요성을 줄이거나 관련 업무를 연결하는 것이 DevOps의 목표입니다.
  - 개발을 시작하면 담당 업무에 몰입하기 쉽다고 느낄 것입니다. 팀에서 개발 업무를 보다 수월하게 진행하는 것이 DevOps의 목표입니다.

이해 관계자: 조직 문화


DevOps는 높은 성과를 내는 조직을 구축하는 전략이 될 수 있습니다.
문화는 회사의 전략적 이니셔티브를 지지하거나 방해합니다.
회사에서 전략을 성공적으로 바꾸려면 조직의 문화를 고려해야 합니다.

이해 관계자: 조직 문화 유형

협업 문화는 협업을 중심으로 합니다.

애드호크라시 문화는 기업가 정신, 혁신과 관련 있습니다.

합리 문화는 목표 달성, 경쟁에서의 승리, 시장 지분 또는 투자 수익(ROI)과 같은 측정 가능한 성과를 증진시키는 데 주력합니다.

위계 문화는 예측 가능성과 타임라인, 효율성을 중시합니다.

지속적 통합 및 지속적 전달(CI/CD)

학습 내용

강의 핵심 내용

학습 내용:

- 자동화가 필요한 이유를 설명합니다.
- CI/CD 파이프라인의 상태를 파악합니다.
- 지속적 통합을 설명합니다.
- 지속적 전달을 설명합니다.
- CI/CD 도구에서 중요하게 살펴봐야 할 기능을 파악합니다.

자동화

- 소프트웨어를 개발할 때 같은 태스크를 반복적으로 수행하기에 지루하고 효율성이 떨어질 수 있습니다. 자동화로 이 문제를 해결할 수 있습니다.
- 자동화의 목표는 창의적인 효율성입니다. 그러나 자동화에는 이 목표를 달성하는 데 방해가 되는 몇 가지 위험 요소가 있습니다.

- 이 개념은 다음 슬라이드에서 설명합니다.

자동화: 위험 요소

과도한 자동화

- 과도한 자동화는 개발 프로세스의 단계를 자동화하여 창의성이 저하되는 경우에 발생합니다. 특정 단계를 수행할 때마다 다른 방식으로 생각하고 고려해야 한다면, 그 단계는 자동화하지 않는 것이 좋습니다. 예를 들면 분석, 계획, 설계 업무가 여기에 속합니다.

부족한 자동화

- 부족한 자동화는 모든 단계가 제대로 처리되었는지 확인하기 위해 자동화를 회피하는 경우 또는 코드 작동이 중단되는 정확한 지점을 찾는 데 도움이 된다는 이유로 발생합니다. 빌드, 테스트, 배포 프로세스는 자동화해도 좋습니다.

잘못된 자동화

- 제대로 작동하지 않는 프로세스를 자동화하면 자동화를 잘못한 것입니다. 개발의 계획 단계를 다시 수행하여 잘못된 자동화를 수정할 수 있습니다.

DevOps 도구: 자동화

자동화에는 많은 도구가 있습니다.

빌드 자동화

- 빌드 자동화는 코드를 수정한 후에 코드를 자동 컴파일하는 방식입니다.

테스트 자동화

- 논리적 테스트는 코드 변경 후 의도대로 실행되는지 확인하기 위해 로직을 자동으로 테스트합니다.

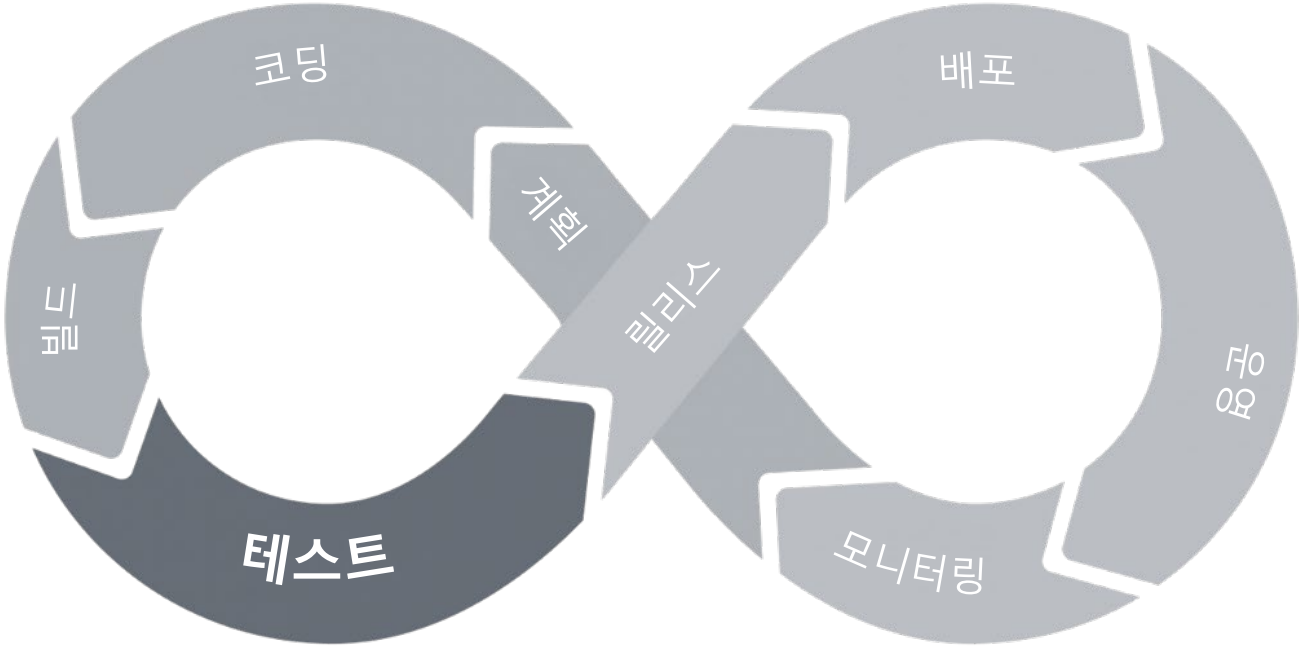
배포 자동화

- 배포 자동화는 테스트 또는 사용을 위해 사용 가능한 형식으로 코드를 가져오는 방식입니다.

자동화에는 다양한 방식과 도구가 포함되지만 다음 세 가지 도구에 집중하는 것이 좋습니다.

CI/CD 파이프라인

- CI/CD 파이프라인은 자동화의 또 다른 도구입니다. 각 부분은 다음을 의미합니다.
  - 지속적 통합(CI)
  - 지속적 전달(CD)
- 다음 두 장의 슬라이드에서 CI와 CD를 자세히 살펴보겠습니다.



지속적 통합(CI)

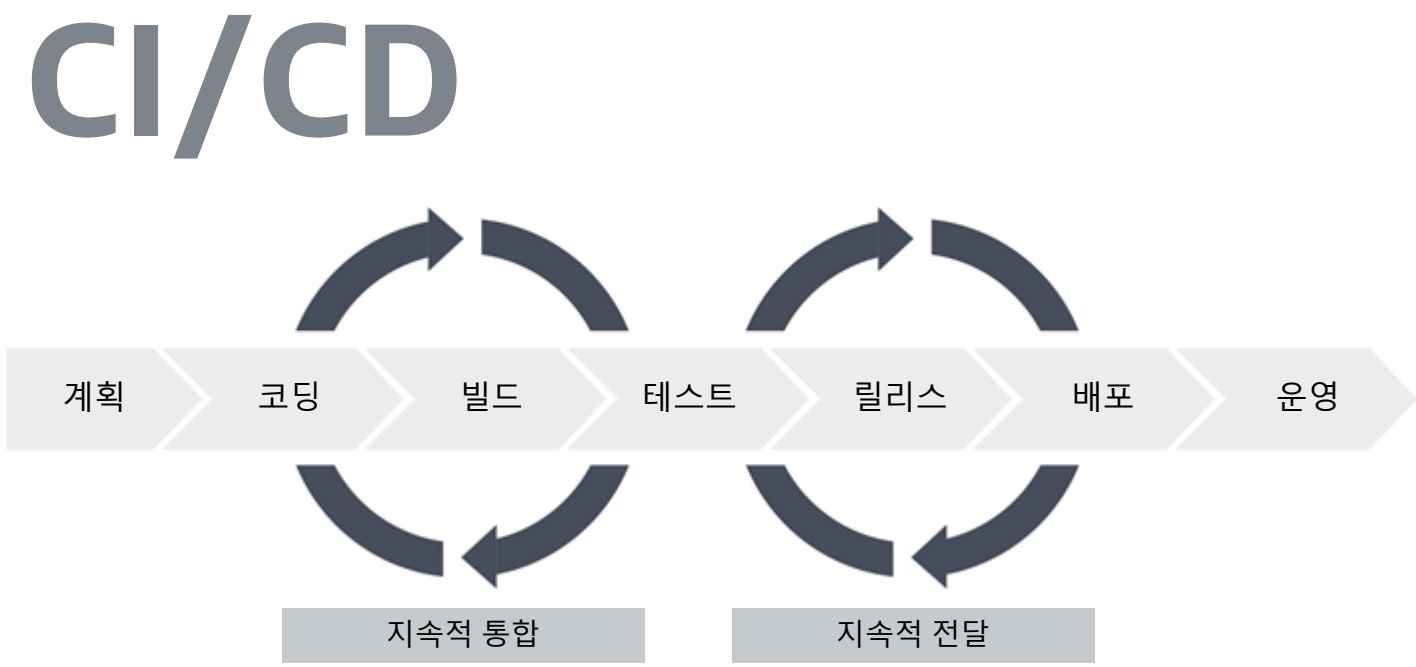
- CI는 팀원이 코드를 사용할 수 있도록 하는 자동화입니다.
- 일반적으로 앞서 설명한 빌드 자동화와 품질 보장 자동화가 포함됩니다.
- CI의 주요 목적은 두 가지입니다.

이미 끝낸 작업과 코드가
잘 작동하는지 확인

이후에 작업할 사람들이
코드를 읽을 수 있도록 함

지속적 전달(CD)

- CD는 CI가 확장된 개념입니다.
- CD에는 제출된 모든 코드의 테스트 자동화가 포함됩니다. 목적은 코드가 다음과 같이 작동하도록 하는 데 있습니다.
  - 의도한 목적대로 작동
  - 논리적으로 작동
- CD는 개발 프로세스의 어느 지점에서든 작업 중인 코드 버전을 즉시 생산할 수 있도록 합니다.
  - 이 부분은 배포 자동화입니다.



학습 내용 확인

DevOps는 소프트웨어 개발의 어떤 영역에서 간극을 좁혀 줍니까?

자동화의 세 가지 위험 요소는 무엇입니까?

요점

- DevOps는 소프트웨어 개발과 IT 운영이 결합된 업무 방식입니다.
- DevOps는 업무 방식이자 문화입니다.
- CI/CD를 도입하면 개발 팀이 메인 브랜치의 코드를 변경하면서도 다른 개발자가 적용한 변경 사항에는 영향을 주지 않을 수 있습니다.

aws re/start