

제5장 Github Actions



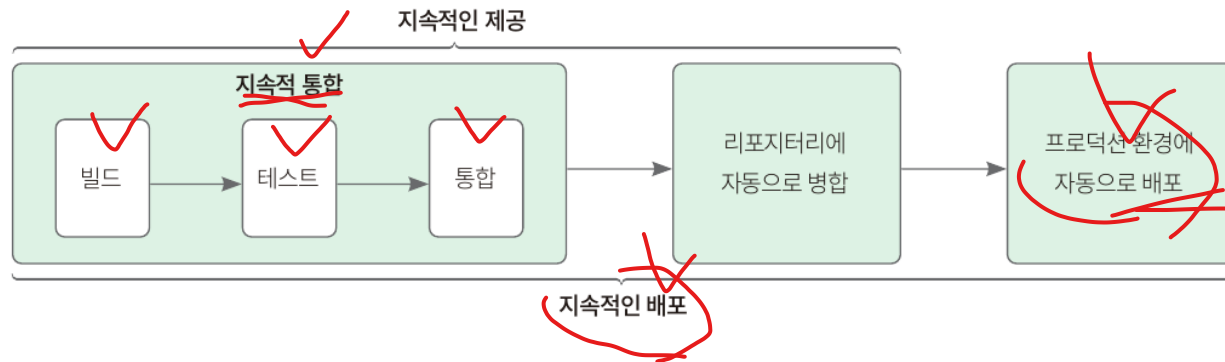
목차

- ✓ 1. CI/CD 개요
- ✓ 2. Github Actions 개요
- ✓ 3. CI/CD 환경 구성

1. CI/CD 개요

프로세스, 방법론, 배포

- DevOps는 개발에서 운영까지 서비스를 빠르고 안정적으로 제공할 수 있게 조직의 역량을 향상시키는 개발 문화 및 방법론
- CI Continuous Integration 는 자동 빌드 수행하고 CD Continuous Deployment 는 자동 배포를 수행하는 DevOps의 핵심 개발 인프라
- CI/CD 도구로 Jenkins, Circle CI, Travis CI, Github Actions 등



2. Github Actions 개요

- Github Actions는 github에서 제공하는 CI/CD 도구
- Build, Test, Deploy 등 필요한 Workflow를 등록하고 Gihtub의 특정 이벤트가 발생했을 때 해당 워크 플로우를 수행
- Github Actions는 별도의 도구를 설치하지 않아도 Github Repository에서 바로 사용할 수 있는 장점

구성	설명
✓ <u>Workflow</u>	<ul style="list-style-type: none">• 자동화된 전체 프로세스• .github/workflows 디렉터리에 작성되는 YAML 파일
✓ <u>Event</u>	<ul style="list-style-type: none">• Workflow 실행을 Trigger 하는 특정 활동이나 규칙
✓ <u>Job</u>	<ul style="list-style-type: none">• Job은 여러 Step으로 구성되고, 가상 환경의 인스턴스에서 실행• Job의 순서로 인한 의존관계 발생
✓ <u>Step</u>	<ul style="list-style-type: none">• Job에서 순차적으로 실행되는 프로세스 단위• Step에서 명령을 내리거나, action을 실행
✓ <u>Action</u>	<ul style="list-style-type: none">• Job을 구성하기 위한 Step들의 조합으로 구성된 독립적인 명령• 재사용이 가능한 컴포넌트

3. AWS 환경 구성

- AWS EC2는 실제 서비스 운영을 위한 배포 서버
- AWS S3 는 배포 코드를 배포하기 위해 보관하는 스토리지 서버
- AWS CodeDeploy는 S3에 보관된 배포 코드를 실제 프로덕션 환경으로 배포하는 배포 시스템

