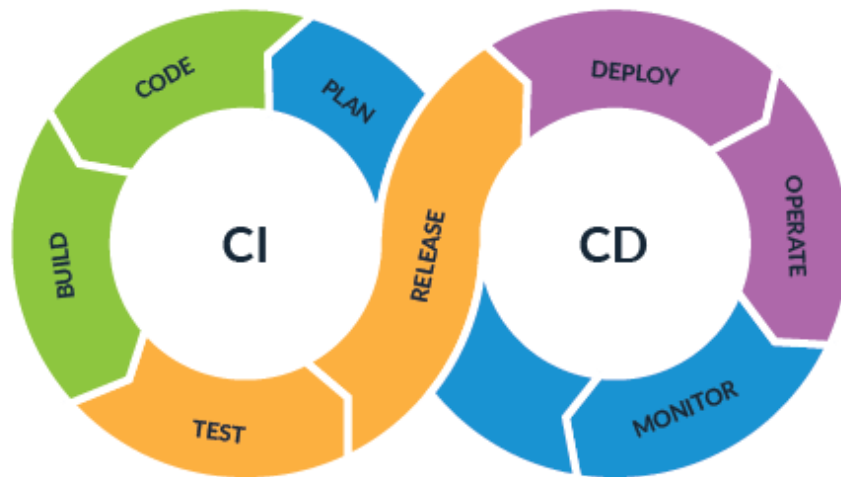
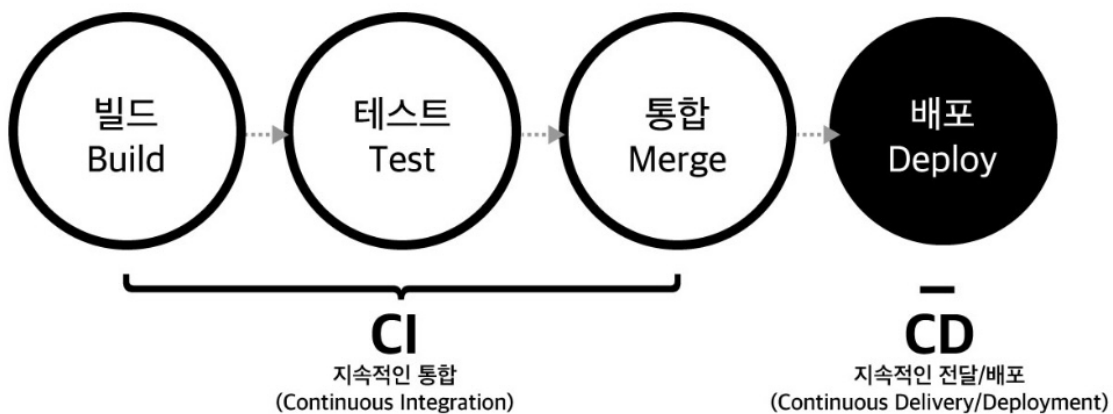


# 25주차 CI/CD

## CI/CD 란?



CI/CD (Continuous Integration/Continuous Delivery or Deployment)는 애플리케이션 개발 단계를 자동화하여 애플리케이션을 더욱 짧은 주기로 고객에게 제공하는 방법론



CI = 지속적인 통합(Continuous Integration) = “빌드와 테스트 자동화”

**CD** = 지속적인 전달 또는 제공(Continuous Delivery) 또는 지속적인 배포(Continuous Deployment) = “배포 자동화”

- **지속적 통합(Continuous Integration, CI):** 자동화 프로세스를 통한 지속적인 통합을 의미. 애플리케이션 코드의 새로운 변경 사항이 정기적으로 빌드 및 테스트를 거쳐 공유 리포지토리에 병합한다. 공유 리포지토리를 사용할 경우, 여러 명의 개발자가 작업할 경우 발생하는 소스 코드 충돌 문제를 해결이 가능하다.
- **지속적 제공(Continuous Delivery, CD):** 변경 사항을 버그 테스트를 거쳐 리포지토리에 자동으로 업로드하는 것을 의미. 코드 변경 사항 병합부터 프로덕션에 적합한 빌드 제공에 이르는 모든 단계에는 테스트 자동화와 코드 릴리스 자동화가 포함되어야 하며, 이 프로세스를 완료하면 운영팀이 더욱 빠르고 손쉽게 애플리케이션을 프로덕션으로 배포할 수 있다.
- **지속적 배포(Continuous Deployment, CD):** 리포지토리에서 고객이 사용 가능한 프로덕션 환경까지 자동으로 릴리스하는 것을 의미. 지속적 배포 환경이 구성되면, 수동 배포로 인한 프로세스를 줄이며, 개발자와 운영자가 간의 가시성 및 의사 전달에 도움이 된다. 실제 사례에서 지속적 배포란 개발자가 애플리케이션에 변경 사항을 작성한 후 몇 분 이내에 클라우드 애플리케이션을 자동으로 실행할 수 있다.

## 왜 필요할까

### *Quality at Speed*

시장 변화 및 고객 요구에 신속하고 유연하게 대응할 수 있는 개발안을 구축하는 중추적인 솔루션인 Agile 문화와 DevOps의 한 부분으로써 CI/CD 개념이 출현

- **코드 품질 보장 :** CI/CD 파이프라인은 테스트 자동화를 제공하기 때문에 개발자들은 거의 실시간으로 코드 문제를 파악할 수 있음
- **릴리스 속도를 단축하여 더 신속하게 제공 :** 코드를 더 빨리 릴리스하고, 더 많은 새로운 코드를 개발할 수 있게 됨에 따라 끝없이 릴리스가 가능, 기업 차원에서도 고비용의 개발자 리소스를 최대한 활용할 수 있어 이득을 얻을 수 있음
- **간소화된 롤백 :** CI/CD 파이프라인을 통해 오류를 신속하게 해결하고, 훨씬 더 빠르게 복구할 수 있음, 팀의 시간과 에너지, 리소스를 절약하고, 문제가 있는 코드를 보다 신속하게 수정할 수 있도록 함
- **지속적인 피드백 :** 비즈니스 이해관계자들을 비롯해 팀의 모든 사람들이 무슨 일이 일어나고, 어디에서 일어나는지, 그리고 무엇이 잘못되었는지 확인할 수 있는 방법을 제공

## 도구들

### 젠킨스



- 중앙 빌드 및 지속적인 통합 프로세스가 가능
- Windows, macOS 및 기타 Unix 계열 운영 체제용 패키지가 포함된 독립형 Java 기반 프로그램
- 소프트웨어 개발 프로젝트의 빌드 파이프라인 구성, Build 자동화의 확립, 배포 및 테스트 자동화 등을 지원

### 팀시티



- JetBrains의 빌드 관리 및 지속적 통합 서버
- Java 환경에서 실행되며 Visual Studio 및 IDE와 통합될수 있음
- Windows 및 Linux 서버에 모두 설치할 수 있음
- .NET 및 개방형 스택 프로젝트도 지원

### CircleCI



- 소스코드를 통합하고 빌드, 배포를 자동화하는 파이프라인을 구성하기 위한 툴

- 코드 구축, 테스트에서 배포에 이르기까지 사용자의 파이프라인 전반에 걸쳐 신속한 소프트웨어 개발 및 게시를 지원하는 CI/CD 자동화 도구
- GitHub, GitHub Enterprise 및 Bitbucket과 통합하여 사용할 수 있다는 장점이 있음

## CircleCI 로 CI/CD 구축해보기



**Build anything fast.  
The CI/CD platform for the  
AI future.**

Start for free. Scale without limits.

**Start Building for Free**

**Watch a Demo**

# You're almost there. Now let's add some code.

Signing up with CircleCI is **free**. Next, you'll be taken to GitHub or Bitbucket to authenticate so you can start shipping faster.

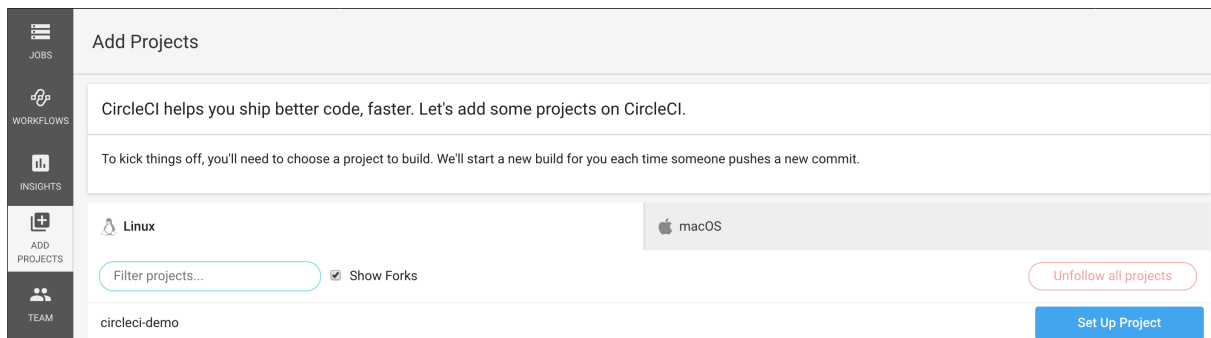
 **Authorize GitHub**

 **Authorize Bitbucket**

By clicking on "Authorize GitHub" or "Authorize Bitbucket" you are agreeing to our [Terms of Service](#) and [Privacy Policy](#).

## 가입하고 깃허브 계정과 연동



## 프로젝트 설정



**Add Projects**

CircleCI helps you ship better code, faster. Let's add some projects on CircleCI.

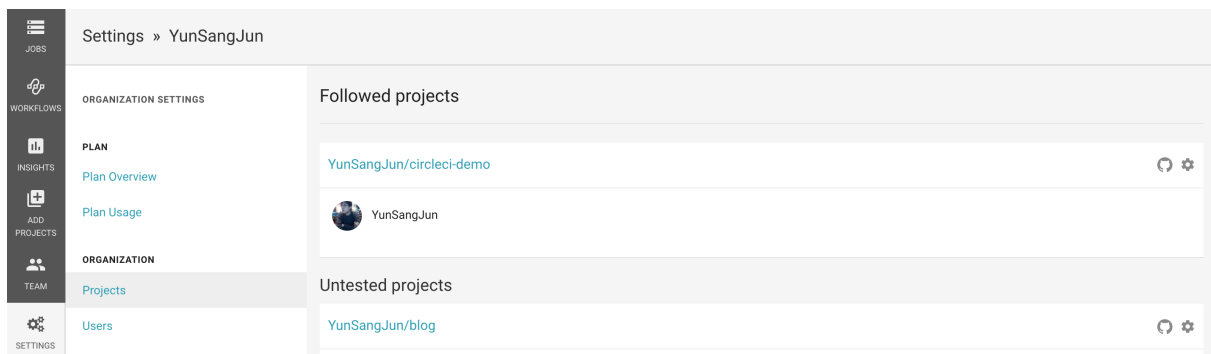
To kick things off, you'll need to choose a project to build. We'll start a new build for you each time someone pushes a new commit.

 Linux  macOS

Filter projects... ☒ Show Forks Unfollow all projects

circleci-demo Set Up Project

## 서클 CI 콘솔에서 프로젝트 설정



**Settings » YunSangJun**

**ORGANIZATION SETTINGS**

**PLAN**

[Plan Overview](#)



[Plan Usage](#)


**ORGANIZATION**

[Projects](#)



[Users](#)

**Followed projects**

YunSangJun/circleci-demo  

 YunSangJun

**Untested projects**

YunSangJun/blog  

Settings » YunSangJun » circleci-demo

PROJECT SETTINGS

Overview

Org Settings

BUILD SETTINGS

Environment Variables

Advanced Settings

NOTIFICATIONS

Chat Notifications

Status Badges


PERMISSIONS

Checkout SSH keys

SSH Permissions

API Permissions

AWS Permissions

 These project settings are no longer available. Please u

### AWS keys for YunSangJun/circleci-demo

Set the AWS keypair to be used for authenticating against AWS  
~/.aws/credentials properties files. These are read by comr

We recommend that you create a unique IAM user for use by Cir

Access Key ID

Secret Access Key

Save AWS keys

Clear AWS keys

AWS 사용한 경우 AWS Permissions 에서 Access Key ID, Secret Access Key 입력

Settings » YunSangJun » circleci-demo View circleci-demo »

PROJECT SETTINGS

Overview

Org Settings

BUILD SETTINGS

Environment Variables

Advanced Settings

NOTIFICATIONS

Chat Notifications

Environment Variables

Environment Variables for YunSangJun/circleci-demo

Import Variables

Add Variable

Add environment variables to the job. You can add sensitive data (e.g. API keys) here, rather than placing them in the repository.

Name	Value	Remove
AWS_ACCOUNT_ID		×
AWS_DEFAULT_REGION		×
AWS_RESOURCE_NAME_PREFIX		×

Build Settings 의 Environment Variables 에 필요한 환경 변수 입력

AWS_ACCOUNT_ID	AWS 계정 ID
AWS_DEFAULT_REGION	AWS 리전
AWS_RESOURCE_NAME_PREFIX	프로젝트 이름

## 빌드 설정

프로젝트 root에 `.circleci` 디렉토리 생성

`.circleci` 디렉토리 하위에 `config.yml` 파일을 생성 후 필요한 설정 입력

설정 예시

- Version : 버전 설정
- Orbs (CircleCI 플랫폼을 빠르게 사용할 수 있도록 도와주는 패키지) :
- working directory : 프로젝트 디렉토리 설정
- Access key
- Region

등등 설정후 변경사항 Github에 업로드

Github에 변경사항이 발생하면 CircleCI에서 이를 감지하여 빌드 및 배포를 수행

## 확인

The screenshot shows the CircleCI 'Workflows' page for a project named 'YunSangJun'. The selected workflow is 'circleci-demo' on the 'master' branch, with a specific commit hash. The workflow is currently in a 'RUNNING' state. Below this, it shows '2 jobs in this workflow': 'build' (status: success, duration: 01:02) and 'deploy' (status: in progress, duration: 00:57). The 'build' job is highlighted with a green checkmark, and the 'deploy' job is highlighted with a blue plus sign.

CicleCI 콘솔-Workflows 메뉴에서 빌드 및 배포 상태를 확인

빌드 또는 배포 상태를 선택하면 해당 Job의 상세 정보를 확인할 수 있음

# CI/CD를 잘 구축해보아요