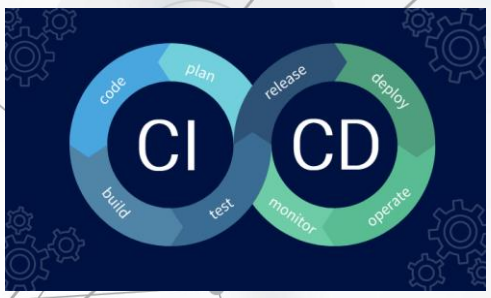
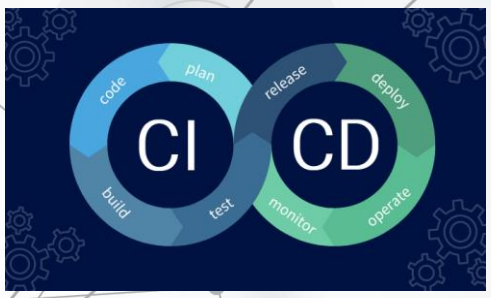


UDACITY GIVE YOUR APPLICATION AUTO DEPLOY SUPERPOWERS

Afolabi Fatai Kayode (ALX-T Cloud DevOps Engineer Nanodegree Program)

EXISTING FRAMEWORK/METHODOLOGY



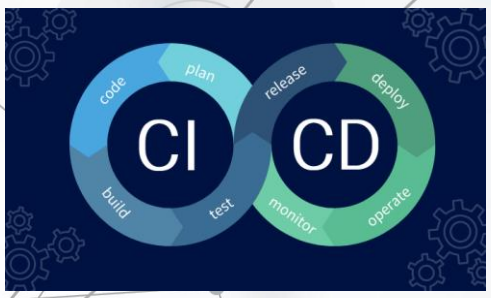
Before the advent of CICD software engineers uses traditional waterfall models, rapid prototyping, agile methodology among others.

With all these framework we still take a longer time to deliver our softwares with low quality and bugs. Although there are manual / reactive monitoring which is also laborious.

Nevertheless the process is still taking lots of time before the software product could get to the users.

DevOps as a set of practices, tools, and a cultural philosophy automate and integrate the processes between software development and IT teams.

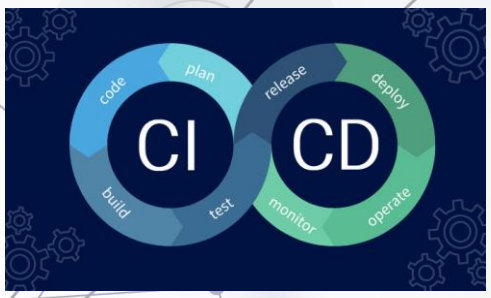
CICD INTRODUCTION



Continuous Integration is a coding philosophy and set of practices that drive development teams to frequently implement small code changes and check them in to a version control repository.

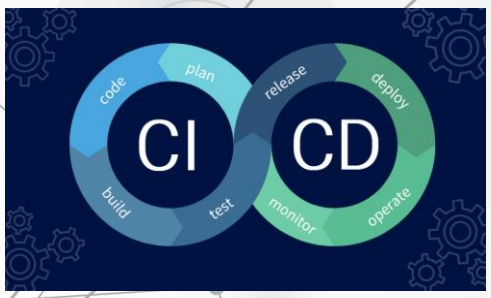
Continuous delivery picks up where continuous integration ends, and automates application delivery to selected environments, including production, development, and testing environments.

BENEFITS OF CI/CD TO THE BUSINESS



1. **Reduce risk:** Finding and fixing bugs late in the development process is expensive and time-consuming.
2. **Deliver faster:** Organizations are moving toward releasing features multiple times a day.
3. **Expend less manual effort:** To align with the shift-left paradigm, we need automation right from the start.
4. **Generate extensive logs:** Observability is one of the biggest aspects of DevOps and CI/CD integration.
5. **Make easier rollbacks:** One of the biggest advantages of a CI/CD pipeline is you can roll back changes quickly.
6. **Monitoring:** the environment metrics with options of integrating rich graphics.

ROI OF CI/CD



In current times, if a software organization is not yet adopting CI/CD, its leadership and management should realize that they are already behind and at a huge disadvantage.

I also like to expand CI/CD abbreviation into CI/CT/CD, meaning continuous integration / continuous testing / continuous delivery. This makes testing and meeting quality standards more obvious in this process.

In very simple terms, when you adopt CI/CT/CD, every dev work — new feature, bug fix, improvement — is continuously tested and integrated into your "ready to ship" branch and is, well, ready to be released to your customers based on your criteria for delivery.