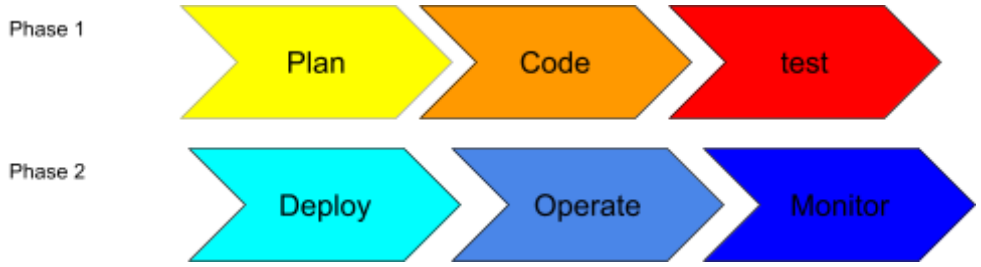
# Why CI/CD ?

For a software company, there are few steps to go through in order to profit from any software it delivers.
Will try to explain each step in brief so you get a better idea where does CI/CD fit in this process.

Typical Software Development LifeCycle:




How did it used to be:

1. Gathering requirements from client
2. Developers write code
3. Testers run tests to check for syntax and performance issues.
4. Security checks for code and packages vulnerabilities.
5. Infrastructure Engineer to provision and configure servers.
6. System Admin to operate and monitor.

Each of these steps might take up to weeks to get from step 1 through step 6, that is if every thing works perfect, which in most cases they don't; usually due to human error, whether in writing code , starting/ ending tests, getting back to devs in case of a bug to fix it, time to provision infrastructure and the list keeps going. Those weeks mean falling back in delivering new feature or fixing a bug.

## From a business view:

1. spending more money on infrastructure and other costs that come with opening a business.
2. not meeting customers requirements or expectations and even might lose customers.

This is where Automation steps in, to eliminate or reduce human error and speed up the delivery of a product.

## What is CI/CD ?

By definition, CI/CD is a term for Continuous integration/ continuous deployment.

CI/CD means to automate the software development lifecycle, starting at writing code and end up delivering it to customers.

## How does that happen ?

Will split it in 2 parts:

1. Continuous Integration: developers continuously and frequently integrate code in a shared repository or main branch multiple times a day. But the changes that are made goes through several automated tests to rectify problems and get checks done.
2. Continuous Deployment: provisioning infrastructure needed to host the code, configure environment, run your application and finally setup monitoring system to keep an eye on it.


## From a business view:

CI/CD creates a fast and effective process of getting your product to market before your competition, as well as releasing new features and bug fixes to keep your current customers happy.

## Pros of CI/CD

- Smaller codes are simple and have less unintended consequences.
- Mean time to resolution (MTTR) is faster and shorter.
- Fault isolation is smaller and faster.
- Improved test reliability due to smaller and specific changes.
- Increased release rates help detect and repair failures faster.
- Number of non-critical defects in your backlog can be reduced by incorporating CI/CD.
- CI/CD help in getting customer and employee feedback.
- Automation in CI/CD reduces the number of errors that can occur in the multiple steps in  CI/CD pipeline.