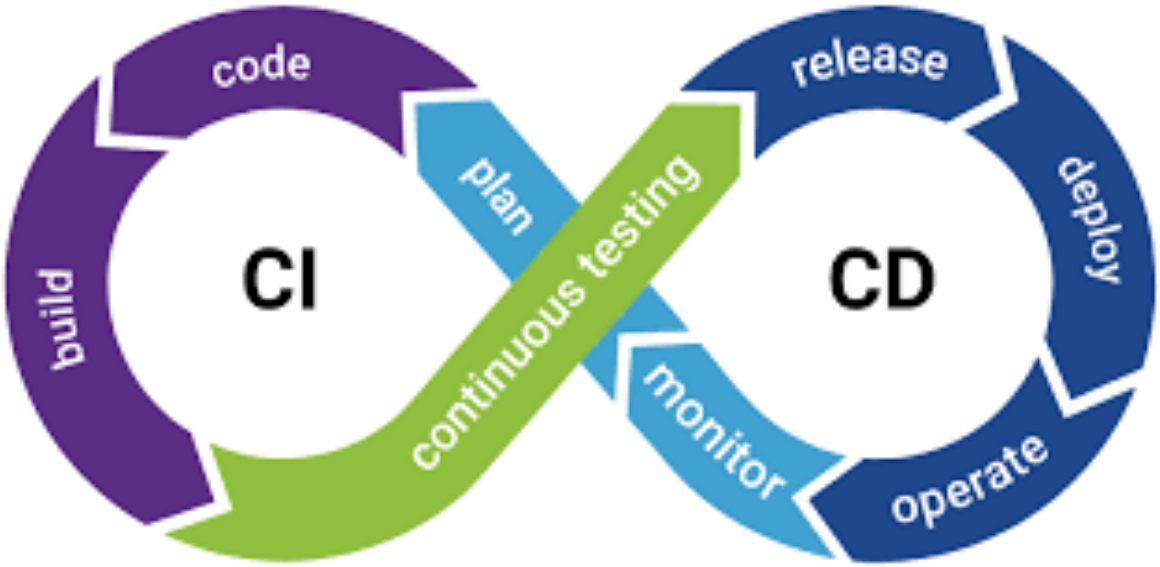
code
build
CI
plan
continuous testing
release
deploy
CD
monitor
operate

What is CI and CD ?

- **Continuous Integration (CI)** is a set of practices to integrate developer's work into the main source code to be deployed later. It contains of an automated process to build, test and analyse the code before merging it into the source code. That way developers can focus on writing codes and discussing best practices and use cases instead of trying to manually build and test the code before approving it for merge.

- **Continuous Deployment(CD)** is a set of practices to automate the deployment process. It handles the infrastructure creation, configuration, and even quick smoke tests before promoting a new code into production.

Why should we implement CI/CD Pipeline ?

- The less code we release in a single release, the better. With an automated deploy process we could reach dozens of small deploys weekly. This is more value being delivered at a constant pace.

- Fault isolation is quicker and the time spent by each developer catching issues in other developer's code is huge. Automating this would allow developers to focus on what really matters to deliver value.

- The number of issues we produce after a new release is much bigger than it should. With a well structured pipeline we can ensure to reduce the number of issues, which improves the user experience and helps o retain our users

- But what if a deploy fails? We can easily roll back to previous stable version of the software.

- All that infrastructure that we are not using anymore but it is still being charged by AWS, that can also be cleaned up in an automated deploy process.

Business Benefit

Four ways one can translate the technical benefits of CICD into business terms relating to monetary win

- Reduce Cost - not having to spend money in the future.
- Avoid Cost - money you didn't have to spend on something now
- Increase Revenue - Generating profits
- Protect Revenue - Protecting the money to not waste

Reduce Cost

- Less developer time on issues from new developer code
- Less infrastructure costs from unused resources

Avoid Cost

- Less bugs in production and less time in testing
- Less human error, Faster deployments
- Prevent embarrassing or costly security holes

Increase Revenue

- New value-generating features released more quickly
- Less time to market

Protect Revenue

- Quick undo to return production to working state
- Reduced downtime from a deploy-related crash or major bug