Continuous Integration & Continuous Delivery known as CI/CD

BY: Matembu Emmanuel Dominic

Summary

A project design or better workflow always requires to be of less cost involved with efficient delivery to meet the clients needs while apply the Continuous Integration and Continuous Delivery known as CI/CD.

CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development. The main concepts attributed to CI/CD are continuous integration, continuous delivery, and continuous deployment.

Background

At one time developers would spend a few months building a new feature then they would go through the gargantuan effort of integration. That is, merging their changes into an upstream code repository, which had inevitably changed since they started their work. This task of integration would often introduce regressions, bugs, and in some cases might even be impossible or irrelevant, leading to months of lost work.

This continuous integration was greatly aided by the now ubiquitous automated test pipeline, which helps ensure that each incremental change doesn’t introduce unintentional behavioral regressions.

Objective

The main objective to CI/CD is to allow the organization to ship software quickly and efficiently since it facilitates an effective process for getting products to market faster than ever before, continuously delivering code into production, and ensuring an ongoing flow of new features and bug fixes via the most efficient delivery method.

Note: Continuous Delivery and Continuous Deployment are logical extensions of Continuous Integration, that doesn’t mean they need to be chronological extensions.

Scope/Benefits

- Less developer time on issues from new developer code thus reducing on the costs.
- Less bugs in production and less time in testing while avoiding high costs.
- Prevent embarrassing or security holes that might be costly.
- Less human error and faster deployments leading to low costs.
- Less infrastructure costs from unused resources.
- New value-generating features released more quickly thus increasing revenue.
- Less time to market while increasing revenue.
- Reduced downtime from a deploy-related crash or major bug while protecting revenue.
- Quick undo to return production to working state to protect revenue.