Fundamentals and Benefits of CI/CD to Achieve, Build, and Deploy Automation for Cloud-Based Software Products.

By Peter Ojo.

Content

- What is CI/CD?
- Definition of key terminology:
    - Continuous Integration
    - Continuous Delivery
    - Continuous Deployment
- Benefits of CI/CD.

What is CI/CD

CI/CD is a devops methodology which uses an automated approach to deliver applications to clients. CI/CD is the short form for continuous integration, continuous delivery and/or continuous deployment (both words have different meaning). CI/CD addresses issue that may arise as a result of a new application feature which makes the development team to add more or remove a part of the existing code for the implementation of the feature.

Continuous integration is a software development practise where developers incorporate code changes into a master repository. The adjustments of the developer are confirmed by building and performing automated tests against the build. This helps in avoiding the big difficulty when we wait till the release day to integrate all changes to the release branch. Continuous integration puts a strong emphasis on automated testing to ensure that new commits in the main branch will not break down the program (Atlassian).

Continuous delivery often entails that an operations team may deploy a developer's changes to a live production environment after they have been automatically checked for bugs and submitted to a repository (such as GitHub or a container registry). In other words, it is a continuous integration advancement to ensure you can easily and reliably make new improvements available for your customers. Theoretically, you can release the product increment per day, week, fortnight, etc with continuous delivery. However, to get the gains of continuous delivery, the product should be released to production as soon as possible.

Continuous deployment is same as continuous delivery. Only difference is that it replaces the manual deployment of the code changes of the developer's to production by the operations team with an automated pipeline process.

As can be seen in the image above, the changes are released automatically to the repository and manually released to production in continuous delivery. The release to production is done automatically in continuous deployment.

Benefits of CI/CD

- Errors are caught easily and fast with automated unit tests. This helps the development team to identify the issue early on in the application development lifecycle and look out for possible fix. Time saved will help the business to save costs of operation.

- CI/CD helps to save time it would normally take to create and destroy infrastructure. With CI/CD, a well architected cloud infrastructure can be provisioned within few minutes. Likewise, the destruction of the infrastructure.

- With CI/CD, unsuccessful changes to application can be easily rolled back to its previous state (successful) in a fast way. This helps to minimize/avoid production downtime.