

PRESENTATION BY MOHAMMED TAWFIQ SANI

BENEFITS OF CICD TO THE ORGANIZATION

DATE : 15<sup>TH</sup> JUNE,2022

CICD , Is an acroynm for Continuous integration and Continuous Deployment.

Continuous integration (CI) and continuous delivery (CD) are the processes that are used to build, code analysis,package and deploy your application. Ideally, it outlines some practices to follow in order for the code you write to be built and deployed more quickly and safely get to your users and ultimately generate value.

Continuous integration is the practice of continually integrating updates into a codebase. CI offers a consistent, automated process of building, packaging, and testing new software.

With CI, developers commit code changes (whether corrective or innovative) into a shared repository. The updates are always small, making them easy to track. Each new integration triggers an automatic build-and-test sequence. This process provides quick feedback to the developer and notifies about any errors.

Ideally, the CI feedback loop should be done in the shortest possible time. Builds and merges should happen as often as possible.

Continuous delivery starts where continuous integration ends. CD enables developers to deploy regular software changes (new features, improvements, bug fixes) to different environments and end-users at any time.

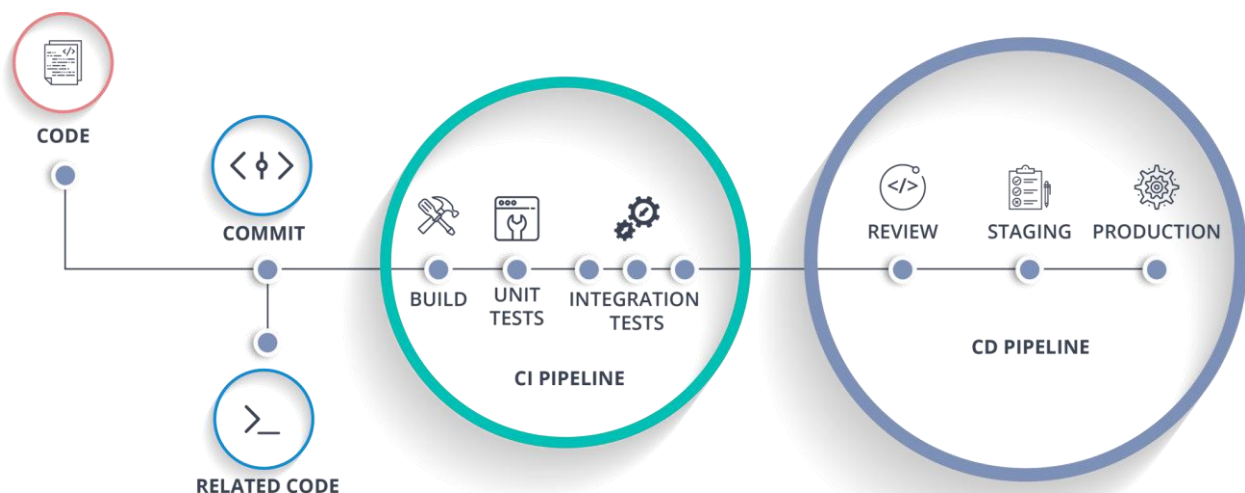
All code that enters the CD process must go through CI first.

Smaller, more frequent software releases are less disruptive and are easier to troubleshoot or rollback in case of an issue. The team also has the ability to rapidly deliver new features, helping the company better meet customer needs.

Base on the diagram Below . As soon as the developers commit to the Control Version Service ( Github) the trigger is set in the CI tool ( Circleci or Jenkins) to pull the commit and build on it

Then after building is done successfully with no error or vulnerabilities found in the code

The code is push to Production and if there is error in our latest build we can roll back to the previous version built since we version all our builds this roll back is done in such a manner the user hardly see .



With the above illustration we can see CICD helps to reduce rundown to near zero . What i mean by rundown to be near zero is thus it reduces the risk of an application going offline for a longer time which in turn is going to cost out organization profits and prospective client . For example should there be an update on out website and due to syntax error or vulnerabilities in the code which

by the way will be taken care of in the CI process by tools like Jenkins with addition integration with other tools like SonarQube which will help detect the vulnerabilities and errors in the code .But assuming without using the CICD pipeline we deploy an update to our website and for one or two of the above scenario we need to roll-over to previous version of our website which might take few hours to be up and running by then most of our prospective client would have lost interest coming to our website to buy or order which in turn we are going to lose revenues . But with CICD that profit lost will be taken of by this three mechanism in the CICD pipeline

1. The CICD pipeline makes room for building a Docker image which will be versioned for easy roll-over so should the updated version be down all the Devop-enginer has to do his update his Kubernetes deployment file to the older version which was stable and the website will be back running this process is done in a matter of seconds which will never be known by the client
2. The second important aspect of the Pipeline is the Code Quality analysis which has been integrated in the deployment process which will always help us eliminate vulnerabilities and errors in the code
3. The automation aspect of the deployment . By this i mean the process is going to be triggered as soon as the developer makes or pushes a code in the version control service repository

CICD practices matters to organization as it helps get continuous feedback not only our customers but also from our team. Moreover it can also lead to big advantages. Some of the benefits of implementing CICD pipelines to our everyday software development process are numerous some are :

1. Using automation in the CICD pipeline helps reduce the number of errors that can take place in the many repetitive steps of CI and CD thereby reducing cost to the organization

2. One major technical advantage of CI and CD is that it allows you to integrate small pieces of code at one time. This helps developers to recognize a problem before too much work is completed afterward, which will help reduce the runtime as earlier explained

3. Faster release and fixing of bugs and releasing the code which will help reduce lost of revenue due to the shutdown of the website

4. Bring Products to Market Faster

when our Organizations effectively implemented CI/CD we can bring new products and features to market faster and immediately start generating revenue from the features they deploy rather than waiting for the entire app to be completed before they can launch.

we will always know the code is in good shape because we have automated testing, and continuous delivery means code is automatically deployed if it meets predefined criteria. Back-to-back releases are easier and less time-consuming, and, if something isn't working, we can pull features with a roll-back

5. By being able to make quicker updates and deploy faster to the market we will be able to meet the needs of the customers in the very shortest possible time which will help us increase revenue by gaining more clients

In conclusion I would say CI/CD is a system that was created to make life easier for developers. It allows for quicker system updates, easier problem solving, better customer experience, and faster input and output of new systems than ever before. These valuable benefits make CI/CD an amazing tool to add efficiency to our business and increase your bottom line.

diagram credit :

<https://www.edureka.co/blog/content/ver.1531719070/uploads/2018/07/Asset-33-1.png>