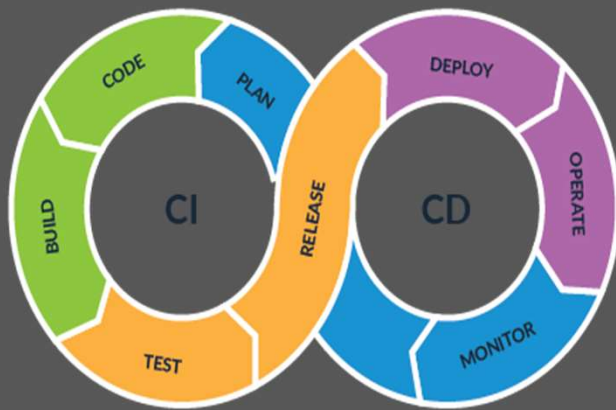


IMPACT OF CI/CD



THE WAY TO CLIENT
SATISFACTION ON
PRODUCT RELEASE

CICD EVOLUTION TO CONVENTIONAL DEPLOYMENT

We realized that continuous integration and deployment has come to stay primarily because of the instant value to your investment in delivering a solution to customer on time.

The evolution of CICD has drastically helps to make agile approach more effective in project delivery

Less time when moving an update to production

It has really helps in collaborating between the development team and the operation team.

The developer and the operation build confidence over-time which helps them to be sure that once deployment move to production, they are assured of client satisfaction because system is in place to ensure clean code move to production.

Continuous integration == Code

Continuous deployment ==
Deployment
Workflow:

- Continuous integration
- Continuous deployment:
- Requires: Continuous integration

Jobs:

Continuous delivery

- Checkout
- Run:
- Ensure you have someone that approve before update are released to production



CONCEPT OF CICD

The evolution of CICD has brought one voice to delivering a project on time and quality.

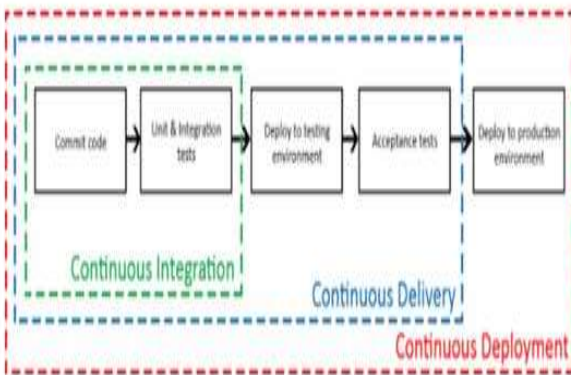
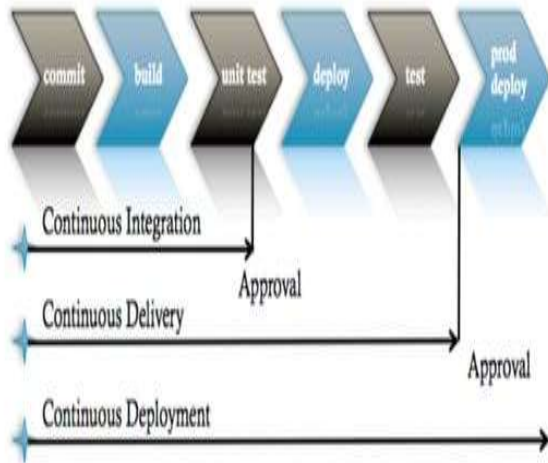
Continuous Integration/Continuous Deployment (CI/CD) describes the key stages in an automated software development and deployment flow

CICD pipeline ensures that raw materials are turned into finished product that is value oriented to the business owner.

In CICD, customer feedback is met because it is easy to deploy an update to production using blue green deployment.

CI/CD leverages lean and agile methodologies that rely heavily on automation to enable efficient development and delivery of high-quality software on a continuous basis.

FUNDAMENTALS OF CICD



CI/CD is a process that always make sure that you deliver project in a timely manner without leaving behind key things like build, test, and any vulnerability exposure.

Without continuous integration, back-tracking errors can be a tedious and time-consuming process, and it's easy to get lost in the maze of code changes.

Integrating as early and often as possible makes it easier to identify where the error occurs. CI also reduces the risk of having errors flare up further down the CD pipeline.

Pipeline is a set of jobs organized in workflow element connected in series where the output of one element is the input of another element

In a standard continuous delivery pipeline, once your application is cleared through testing and declared ready for deployment, it sits in a deployment queue. It is not deployed until manually done so. On the contrary, in a continuous deployment pipeline, as soon as an application passes through testing, it is automatically deployed to end-users.

BENEFIT OF CI/CD

1. When we implement CI/CD, The process ensure that it catch compile errors after merging which in turns reduce cost.
2. It also catch unit test failure by avoiding cost which leads to less bug in production and less time in testing.
3. using CI/CD will help us for faster and more frequent production deployment which will definitely increase revenue. We will achieve this by deploying updates frequently based on customer feedback and the update will be able to have increase in the usage of the application by customers.