CI/CD - A better, faster approach to build and ship products to market

..tapping into the business of the cloud and deliverable of goods more consistently

CI/CD means Continuous Integration (CI) and Continuous Delivery (CD)

In an evolving world of meeting consumers expectations and satisfactions. The choice of shortening software development, meeting markets demands on time is crucial.

There are tons of benefits of using CI/CD to achieve, build and deploy automation within our company's products

This gives us the company an added advantage in getting her products to the market within a snap and gives our customers satisfactory services and also be ahead of our competitors

Additionally, it brings both the developers and operational team in other to avoid any misalignment. Which is a synergy that has not been achieve over the years

Continuous Integration (CI)

Continuous Integration integration ensures that software components work together with the help of version control repository. It detects any changes made to the source code repository

Pipeline

Pipeline is simply a series of connected jobs together in a sequence where each sequence or job is dependent on the output of the previous stage to complete.

The practice of collating all developers' working copies to a shared pipeline several times a day.

Everything related to the code fits here, and it all culminates in the ultimate goal of CI: a high quality, deployable artifact!

Some common CI-related phases might include:

- Compile and Build
- Unit Test
- statistics of jobs in the process
- Dependency vulnerability testing
- Store artifact
- Notification of results via email or chat system

Continuous Delivery (CD)

Here is the chief minister called "On timely technique" to deliver the products for user's experience. it deals with Provisioning Infrastructure, Configuring Infrastructure, Promotion, etc.

This is an software engineering approach in which the value is delivered frequently through automated deployments.

Everything related to deploying the artifact autonomously fits here. It's the process of "Moving" the artifact from the shelf to the spotlight without human intervention.

- Some common CD-related phases might include:

- Creating and configuring infrastructure using IAC(infrastructure as Code)
- Promoting the automated jobs to production
- Smoke Testing (verifying the effectiveness of your jobs)
- Destroy and reverts to previous configuration in case if any failure

Some of the Benefits of CI/CD at the Business level

The rationale of CI/CD is the famous saying: ‘a penny saved is a penny earned', so here's a preview of the benefits of setting up a CI/CD pipeline:

- Automate Infrastructure Creation

This will help to avoid cost by providing less human error, which means faster deployments using already-made template

- Faster and More Frequent Production Deployments:

This would help to increase revenue by releasing new value-generating features more quickly

This allow us to predict and project costs of spending on infrastructures

- More Automated Tests Reliability

This would help protect revenue by reducing downtime from a deploy-related crash or a major bug.

This breads transparency and accountability in the company

- Detect Security Vulnerabilities

This would help to avoid cost by preventing embarrassing or costly security holes.

- Deploy to Production Without Manual Checks

This would help to increase revenue by making new products or features be in the market on time

- Smaller Backlog

The benefits of solving non-critical issues ahead of time are many.

This gives developers to focus their energy on larger problems or trying to improve the old systems.

- Customer Satisfaction

It does not allow the comapny falls into technical glitches when there are more customers or a new customer trying out a products.

- Open source tools

There are tons of open source framework and tools to select from when employing the evolving complicated software systems