CI/CD A world of Automation

- Continuous Integration (CI) allows you to continuously integrate code into a single shared and easy to access repository. Continuous Delivery (CD) allows you to take the code stored in the repository and continuously deliver it to production. CI/CD creates a fast and effective process of getting your product to market before your competition as well as releasing new features and bug fixes to keep your current customers happy.

Continuous Integration

- Continuous integration is a process in devops where changes are merged into a central repository after which the code is automated and tested. The continuous integration process is a practice in software engineering used to merge developers' working copies several times a day into a shared mainline.
- It refers to the process of automating the integration of code changes coming from several sources. The process comprises several automation tools that emphasize on the code's correctness before Integration.
- phases to successful CI/CD pipelines: ***Plan, Cod, build, test, Release, Deploy, Operate, Monitor & Optimize***

Continuous Deployment

- Continuous Delivery (CI) is a DevOps practice that refers to the building, testing, and delivering improvements to the software code. The most important part of the CD is that the code is always in a deployable state.
- **CD Stages**

1-Source

2-Build

3-Test

4-Deploy

The business value and benefits of CI/CD

- 1. Superior code quality
- 2. Shorter time-to-market
- 3. Cost reduction thanks to automation
- 4. Fast feedback
- 5. Easier communication
- 6. Accurate measurement
- 7. Greater customer satisfaction
- 8. Increased business sustainability

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
- - Faster and More Frequent Production Deployments:
- This would help to increase revenue by releasing new value-generating features more quickly
- - Automated Smoke Tests:
- This would help protect revenue by reducing downtime from a deploy-related crash or a major bug
- - 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 features take less time to market.
- - Detect Security Vulnerabilities:
- This would help to avoid cost by preventing embarrassing or costly security holes.
- And many more …