What is CI/CD and why should we care?

Definitions

- CI/CD stands for continuous integration and continuous deployment and/or continuous deployment

- Continuous integration is the practice of frequently and reliably merging code into a main shared repository from where it is compiled, unit tested, vulnerability tested, analyzed and the artifact produced are stored automatically with the help of automation

- Continuous integration tries to solve the merge problem and code coherence issue that teams faces when they are all working on the same code base

- It improves coordination in teams by using continuous feedback and automation

Definitions (contd.)

- Continuous deployment is the process of frequently delivering value through automated deployments eliminating the need for human intervention

- It reduces the lead time which represents how difficult it is to deploy code to production and with a short lead time, we get quicker results of ideas

- Continuous delivery is the practice whereby teams produce and release value in short cycles

- Basically, CI/CD means automatically and continuously test, build and release code changes to a deployment environment

Why it matters

- Implementing CI/CD in a software life-cycle could have positive impact to business cost and revenue

- Some of the benefits of CI/CD include:

  - Frequent and faster deployment to production making new features available to customer faster which leads to an increase in revenue

  - Automated infrastructure creation and cleanup reducing cost from human errors and avoiding cost from unused resources

  - Automatically run tests and catch failures helping the business avoid cost and reduce time spent on testing

Why it matters (cond.)

Studies have revealed that high performing teams who use CI/CD and practice DevOps showed frequent and faster code and change deployment, shorter lead time, higher change success rates, faster mean time to restore service amongst other benefits, thereby outperforming their non-high performing teams