CICD

Continuous Integration

Continuous Deployment

Continuous Integration

- The practice of merging all developers' working copies to a shared mainline several times a day to avoid conflicts in the code in the future.
- Everything related to the code fits here.
- Some of the steps in this stage include compiling, testing, running static analysis, checking for vulnerabilities in our dependencies and storing the code artifacts.

Continuous Deployment

- A software engineering approach in which the value is delivered frequently through automated deployments.
- Everything related to deploying the artifact fits here. It's the process of "Moving" the artifact from the shelf to the spotlight.
- Some steps in this stage include setting up infrastructure, provisioning servers,
- copying files, smoke testing, promoting to production and even rolling back a
- change if something did not look right.

Benefits of CICD to our company

- **Automate Infrastructure Creation and clean up**: Eliminating human errors and avoid unnecessary cost of unused.
- **Superior Quality Code:** Developers release code into a shared repository in small batches, which enables them to conduct parallel testing. Rather than working in isolation.
- **Loyal Customer Base:** which it minimizes release risk and maximizes developer time. A stable operating environment and faster resolution to production requirements enable you to cement customer loyalty and create great experiences.
- **Smoke Tests:** Automate smoke test after deployment and automatic rollback in case of failure which will decrease downtime and reduce cost.
- **Detect Security Vulnerabilities**: This would enable us to easily detect serious security flaws that would be embarrassing if it had made it to the public. This would save us money trying to win back the customers' trust and rebuilding our image.
- **Deploy to Production Without Manual Checks**: Less time to market would help us to increase our revenue.