CICD

Continuous Integration Continuous Deployment

Overview

Continuous Integration

The practice of integrating all developers working copies to a shared mainline codebase several times a day. It's the process of "Making". Everything related to the code fits in here, and it all culminates in the ultimate goal of CI (a high quality, deployable artifact).It is all about code and "Dev" workflow. Some CI related phases are Build, Unit Test, and Static Analysis

Continuous Deployment

A software engineering approach in which the value is delivered frequently through automated deployments. Everything related to deploying the artifact fits in here. It's the process of "Moving" the artifact from the shelf to the spotlight. It is all about deployment and "Ops" workflow. Some CD related phases are provisioning infrastructure, configuring servers, promoting to production

Benefits of CICD

1. Automate Infrastructure Creation and clean up: Eliminating human errors and avoiding unnecessary cost of unused or invalid infrastructure.
2. Faster to production: By automating the pipeline to production, this way we can deploy features as soon as created which will help increase revenue.
3. Automated rollback triggered by Job Failure: Automate the process of rolling back and cleaning any infrastructure which would help in reducing cost and lower down time.

Benefits of CICD

4. Catch Compile Errors After Merge: Discover errors as soon as developer makes his commit which will help reduce the time of developers and reduce cost
5. Catch Unit Test Failures: Unit tests are not neglected with CICD which will increase code quality and catch errors early before production, this would decrease cost.
6. Automated Smoke Tests : Automate smoke test after deployment and automatic rollback in case of failure which will decrease downtime and reduce cost