

CONTINUOUS INTEGRATION AND CONTINUOUS DEPLOYMENT

DELIVER BETTER FEATURES, FASTER



OVERVIEW OF CI/CD

- CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development. The main concepts attributed to CI/CD are continuous integration, continuous delivery, and continuous deployment. CI/CD is a solution to the problems integrating new code can cause for development and operations teams



CONTINUOUS INTEGRATION

- **Continuous integration (CI)** is the practice of merging all developer working copies to a shared mainline several times a day. ... The main aim of CI is to prevent integration problems.
- **Continuous Integration (CI):** regularly build, test, and merge code changes into main branch.



CONTINUOUS DELIVERY VS. CONTINUOUS DEPLOYMENT

- **Continuous delivery** takes the idea of continuous integration one step further and advances automation along the pipeline. With continuous delivery, code is not only integrated with changes on a regular basis, but as a second stage it is also deployed to a given environment, such as staging or production.
- **Continuous deployment** is a natural progression of continuous delivery. Once all software changes pass automated and functional testing, as well as the build stages, they are automatically deployed to production without human intervention. Continuously deploying code without human intervention is the goal that most companies hope to achieve.

BENEFITS OF CI/CD

- **1. Reduce costs**

Automation in the CI/CD pipeline reduces the number of errors that can take place in the many repetitive steps of CI and CD. Another thing to keep in mind: increasing code quality with automation also increases your ROI.

- **2. Deliver faster**

Teams can build, test and deploy features automatically with almost no manual intervention.

- **3. Generate extensive logs**

With a CI/CD pipeline, extensive logging information is generated in each stage of the development process to get immediate feedback about the system.

- **4. Detection & Isolation**

You can limit potential breakdowns and other critical issues by isolating bugs and vulnerabilities before they damage the entire application.