CI/CD

By Gaurav kakkar

Why CI/CD

- **Faster time to market -** The primary goal of a CI/CD pipeline is to deliver working software to users quickly and frequently.
- **Better code quality -** Testing your code's behavior is an essential step in the software release process but doing it thoroughly can also be extremely time consuming.
- **Maximized creativity -** As we've seen, building a CI/CD pipeline eliminates waste and helps create a leaner, more efficient software development and release process.
- **More Test Reliability** - Using CI/CD, test reliability improves due to the bite-size and specific changes introduced to the system, allowing for more accurate positive and negative tests to be conducted
- **Smaller Backlog -** Incorporating CI/CD into your organization's development process reduces the number of non-critical defects in your backlog. These small defects are detected prior to production and fixed before being released to end-users.
- **Customer Satisfaction** - The advantages of CI/CD do not only fall into the technical aspect but also in an organization scope. The first few moments of a new customer trying out your product is a make-or-break-it moment.
- **Increase Team Transparency and Accountability -** CI/CD is a great way to get continuous feedback not only from your customers but also from your own team. This increases the transparency of any problems in the team and encourages responsible accountability.

CI/CD principles

- Repeatable Reliable Process
- Automate Everything
- Version Control Everything
- Bring the Pain Forward
- Build-in Quality
- “Done” Means Released
- Everyone is Responsible
- Continuous Improvement
- Don’t Forget the Database

What is CI/CD

- **Continuous Integration:**

Developers practicing continuous integration merge their changes back to the main branch as often as possible. The developer's changes are validated by creating a build and running automated tests against the build. By doing so, you avoid integration challenges that can happen when waiting for release day to merge changes into the release branch.

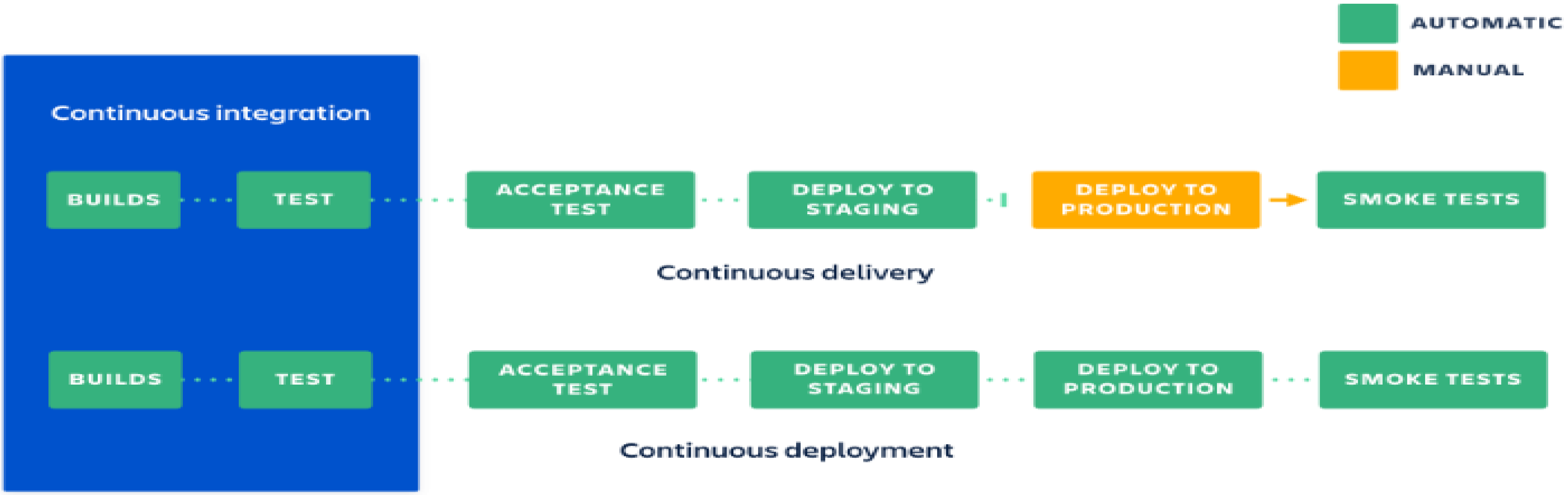
- **Continuous Deployment:**

Continuous deployment goes one step further than continuous delivery. With this practice, every change that passes all stages of your production pipeline is released to your customers. There's no human intervention, and only a failed test will prevent a new change to be deployed to production.

- **Continuous Delivery:**

Continuous delivery is an extension of continuous integration since it automatically deploys all code changes to a testing and/or production environment after the build stage.

CI Vs CD


AUTOMATIC
MANUAL
Continuous integration
BUILDS
TEST
ACCEPTANCE TEST
DEPLOY TO STAGING
DEPLOY TO PRODUCTION
SMOKE TESTS
Continuous delivery
BUILDS
TEST
ACCEPTANCE TEST
DEPLOY TO STAGING
DEPLOY TO PRODUCTION
SMOKE TESTS
Continuous deployment