CI/CD Pipelines

By
Jayaram Aita

Agenda

- What is CI
- What is CD
- Benefits
- Conclusion

What is CI

- Continuous integration is all about the source code.
- New changes to the code need to be validated, verified, exercised, worked over, massaged and squeezed to see if there are leaks. We do this by compiling, transpiling, linting, running unit tests, performing static analysis, checking dependencies for security vulnerabilities and other things.
- Example:

  When multiple developers are working on same project we would end up multiple working copies. All these copies of the code are integrated into a single mainline. This process is called Continuous Integration.

  Some command CI-related might include

  - Compile
  - Unit Test
  - Static Analysis
  - Dependency vulnerability testing
  - Store artifact

What is CD

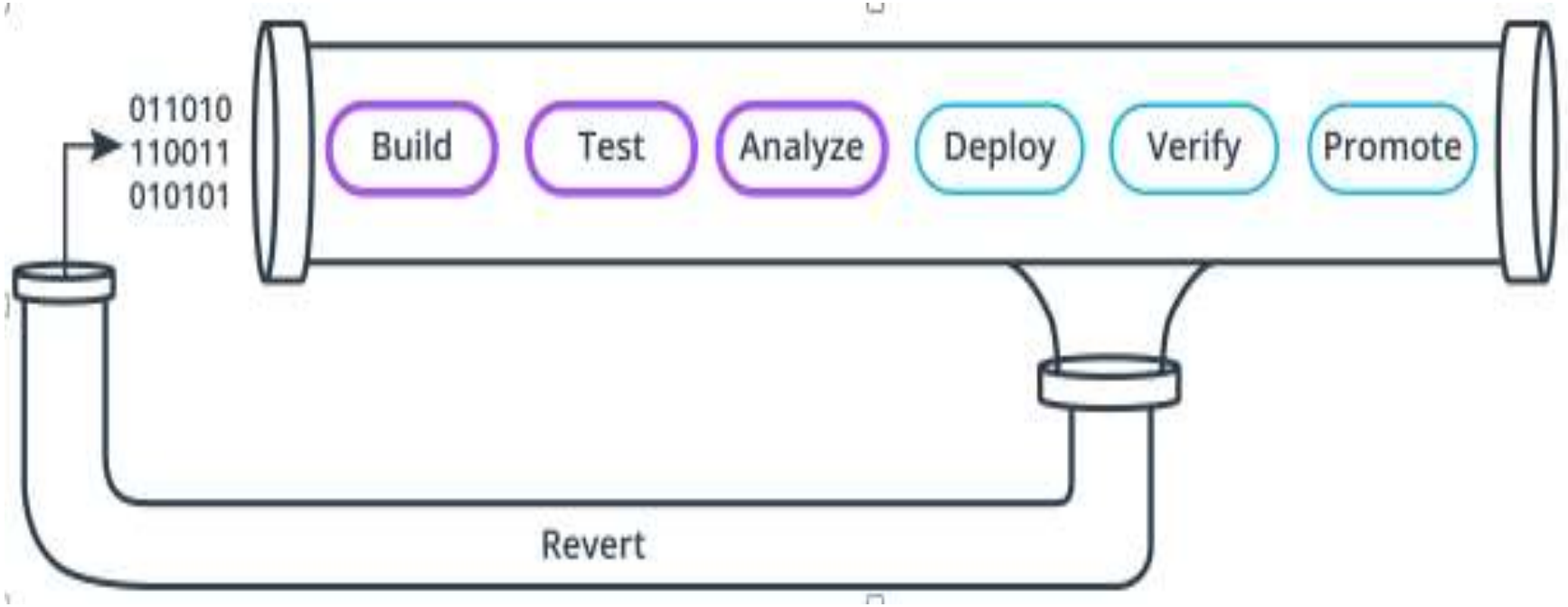
- Continuous deployment is all about built code and deployment.
- Once the source code has been built in CI, we're ready to ship it to servers and devices either in the same network or elsewhere. Depending on your team's delivery process and deployment strategy, you might deploy to a staging or pre-production server for final testing or you might deploy to production right away.
- Before doing so, CD can run scripts to prepare the infrastructure, run smoke tests, and handle rollbacks and reverts if something doesn't go as planned.

CI/CD Pipeline.

CI/CD pipelines are a series of steps that must be completed to deliver a new software release. The aim is to improve the software delivery process by introducing monitoring and automation to improve the development and delivery process.

The CI/CD pipeline typically breaks down into the following stages:

Build
Test
Release
Deploy
Validate



CI/CD Phases/Flow

Few of the Business Benefits

CI/CD plays a crucial role in shortening time to value.

According to 3Pillar Software Engineer, Paul Estrada, the strategy “fosters a culture of innovation, allowing developers to experiment with new technologies and try out new ideas. Teams can test different features with real users in parallel and use their findings to ensure

Reduce Costs and Boost Profits

CI/CD is also good for the bottom line. It standardizes deployment processes across all projects, and, done right, it enables teams to systematically test every change made to the source code.

As a result, this process stands to dramatically reduce the likelihood that any bugs or errors slip through the cracks and cause problems down the line. Done right, this practice can lower development costs by eliminating many of the costs incurred while building and testing code changes.

Teams spend less time on testing and bug fixes, meaning organizations spend less money on tasks that don’t provide any value to the business or its customers.

CI/CD Improves App Quality

One of the most notable examples of CI/CD's business value is that it allows dev, ops, and QA to focus on what they do best. Developers can spend more time writing code that solves real problems rather than worrying about what's going on in the production environment.

Automate Repeatable Processes.

While writing code is still a very "human" process, many aspects of the development lifecycle can be automated. CI/CD automation streamlines the software development cycle, allowing teams to work through the feedback cycle faster and consistently deliver high-quality products with fewer instances of code errors/bugs.

CI/CD supports customer outcomes from a technical standpoint.

Frequent updates, new feature releases, rapid response to feedback, and quick bug fixes all play a major role in customer satisfaction and long-term loyalty. By using CI/CD, organizations can continuously build on applications and enhance the experience without the risk of downtime or interruptions.

Bring Products to Market Faster

Organizations that have effectively implemented CI/CD can bring new products and features to market faster and immediately start generating revenue from the features they deploy rather than waiting for the entire app to be completed (and checked manually) before they can launch.

Conclusion

The benefits of CI/CD impact all ends of the development lifecycle, the customer experience, and the big-picture business strategy.

It plays a critical role in software development and delivery and helps smaller teams move faster, respond to constant changes, and incorporate real time feedback—all of which contribute to cost savings, profitability, and a higher-quality end-product.