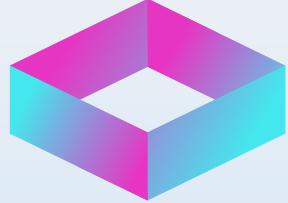


Created by
Team

CI/CD



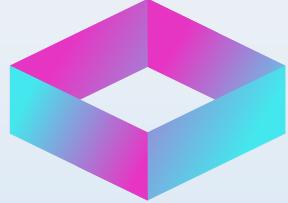


What is Continuous Integration?



Continuous integration (CI) is the practice of automating the integration of code changes from multiple contributors into a single software project. It's a primary DevOps best practice, allowing developers to frequently merge code changes into a central repository where builds and tests then run. Automated tools are used to assert the new code's correctness before integration.





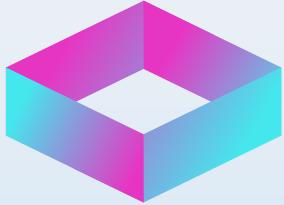
What is Continuous Delivery?



Continuous delivery (CD) picks up where continuous integration is over. While CI is the process to build and test automatically, CD deploys all code changes to the testing or staging environment in the build.

CD enables builds to be released to the production environment when needed. Allowing the team to deploy on its own, the CD effectively reduces time on the market.





Pipeline



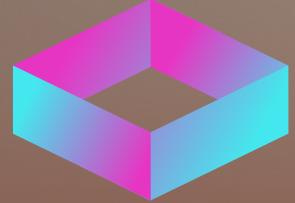
- Commit
- Build
- Automate tests
- Deploy



When the developers make a change, they commit the change to the repository.

Commit





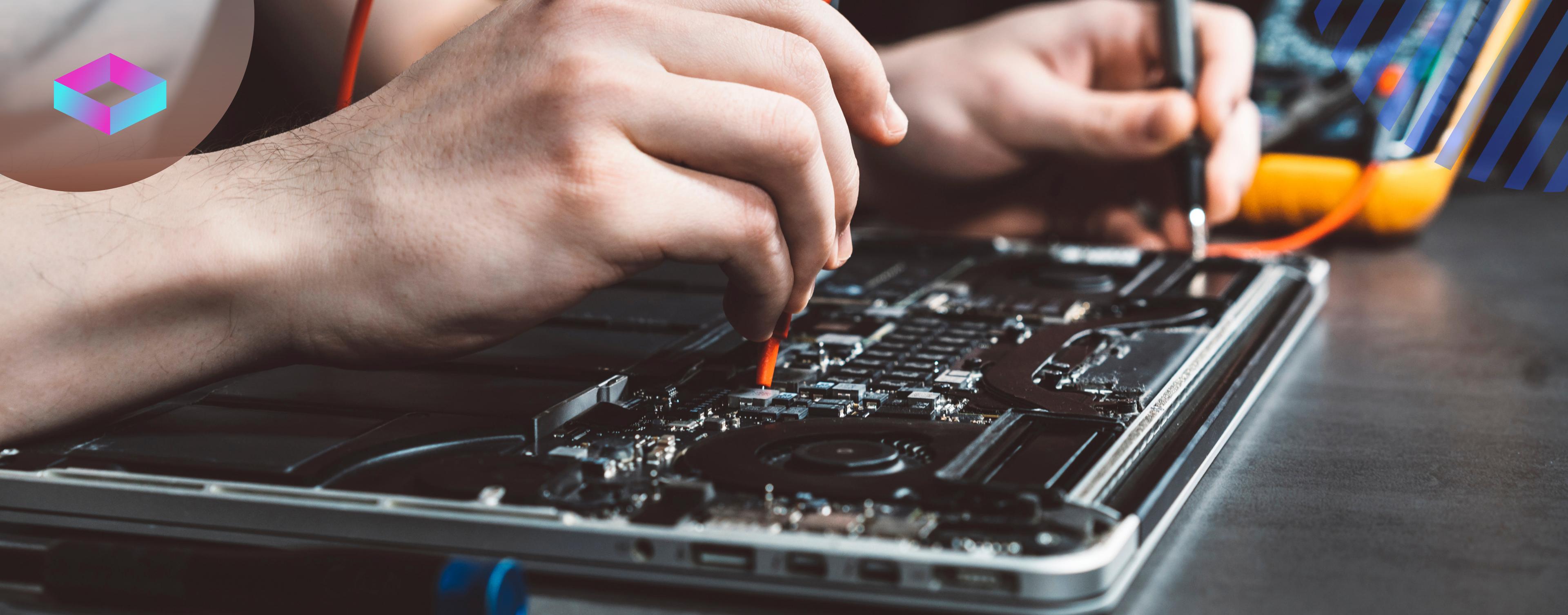
```
IFormatProvider is not used; the default culture is [en-US]:  
    'No' format string:          11876.54  
    'N5' format string:         11.876.54000  
    'E' format string:          1.187654E+004  
    'E5' format string:         1.18765E+004  
  
A CultureInfo object for [nl-NL] is used for the IFormatProvider:  
    'No' format string:          11876.54  
    'N5' format string:         11.876.54000  
    'E' format string:          1.187654E+004  
  
A NumberFormatInfo object with digit group size = 2 and  
digit separator = ',' is used for the IFormatProvider:  
    'N' format string:          1,18,76,54  
    'E' format string:          1,187654E+004  
Press any key to continue . . .
```



The source code in the repository is integrated into the build.

Build





Automated tests are being run against the build. Test automation is a key component of any CI / CD pipeline

Automate Tests





The built version will be delivered
to production

Deploy





Benefits of CI/CD



- Cost Reduction
- Users Satisfaction
- Maintainable Backlog

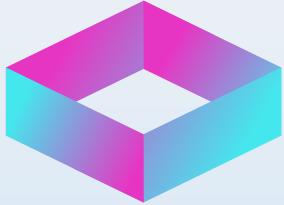


Cost Reduction



Teams could use CI/CD pipelines to automate builds, tests, and merges in a common repository. This drastically minimizes the likelihood of human error in the repetitious development stages. Furthermore, developers can concentrate more on product development because the codebase will require fewer changes in the future if problems are discovered early.



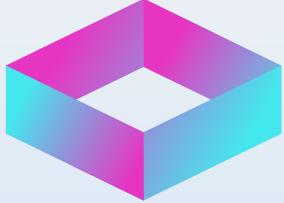


Users Satisfaction



A product's reputation can be tarnished by persistent bugs. A CI/CD pipeline encourages faster release cycles for new features, bug fixes, or both. As a result, by continuously updating product technology and responding to feedback with rapid modifications, teams can ensure that they are consistently meeting their customers' requests.





Maintainable Backlog



Developers can use CI/CD pipelines to cut down on the number of non-critical issues in their backlog. Solving non-critical issues before they become destructive to the team has numerous advantages. Developers, for example, might concentrate on larger issues like improving the overall product. End-users would also be happier if there were fewer errors reaching them.





Conclusion



Two of DevOps' recommended practises for managing misalignment between developers and the operational team are continuous integration and continuous delivery. Developers may deploy modifications and new features more frequently using automation, while operations teams have improved overall stability.