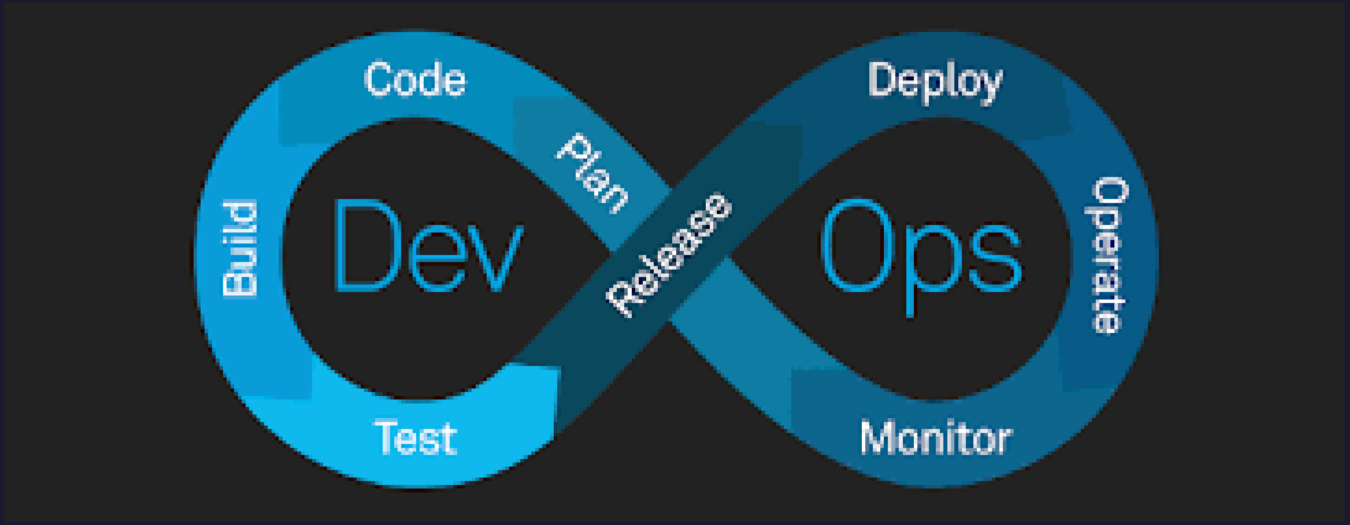
CI/CD

Continuous integration (CI)
&
continuous delivery (CD)

Alsafa Wagdy

CI/CD

time-to-market, tighter feedback loops, and greater interaction with internal and external customers are all benefits of CI/CD processes.



Continuous Integration (CI):

Consider an application that has its code stored in a Git repository in GitLab. Developers push code changes every day, multiple times a day. For every push to the repository, you can create a set of scripts to build and test your application automatically. These scripts help decrease the chances that you introduce errors in your application

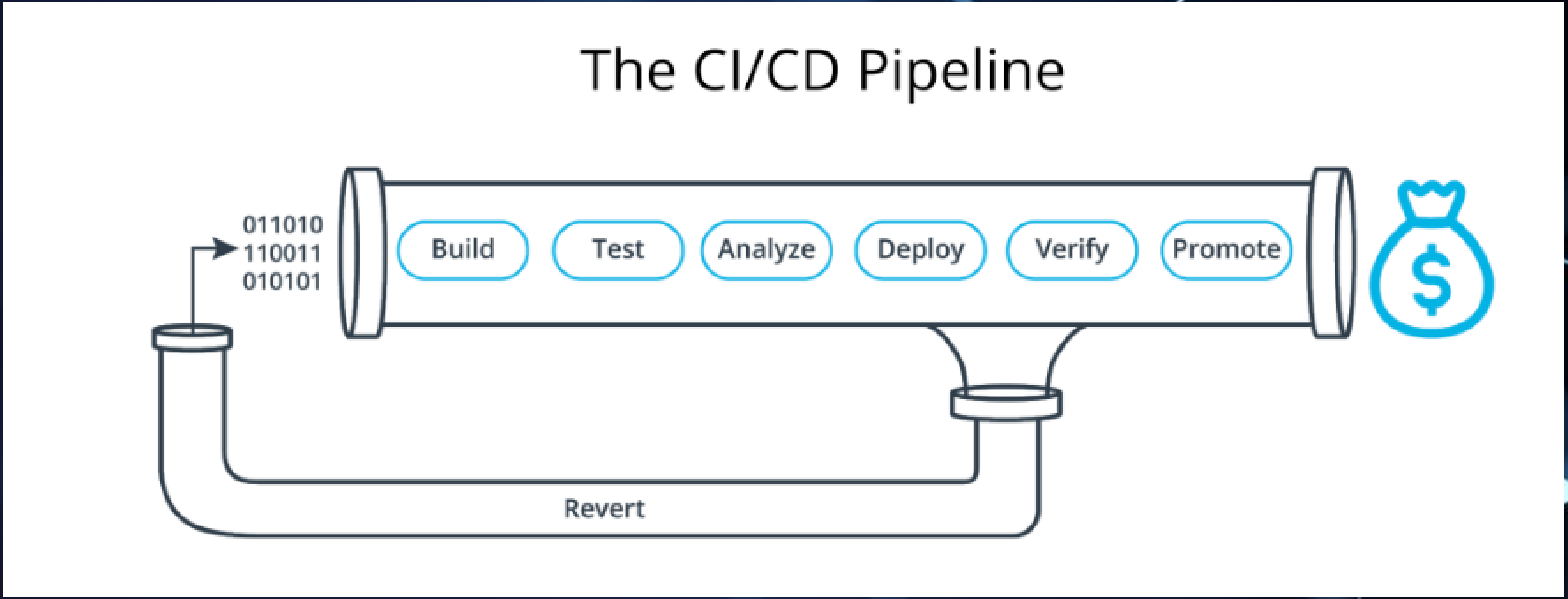
Continuous Delivery (CD):

a step beyond Continuous Integration. Not only is your application built and tested each time a code change is pushed to the codebase, but the application is also deployed continuously. However, with continuous delivery, you trigger the deployments manually. it checks the code automatically, but it requires human intervention to manually and strategically trigger the deployment of the changes.

Continuous Deployment (CD):

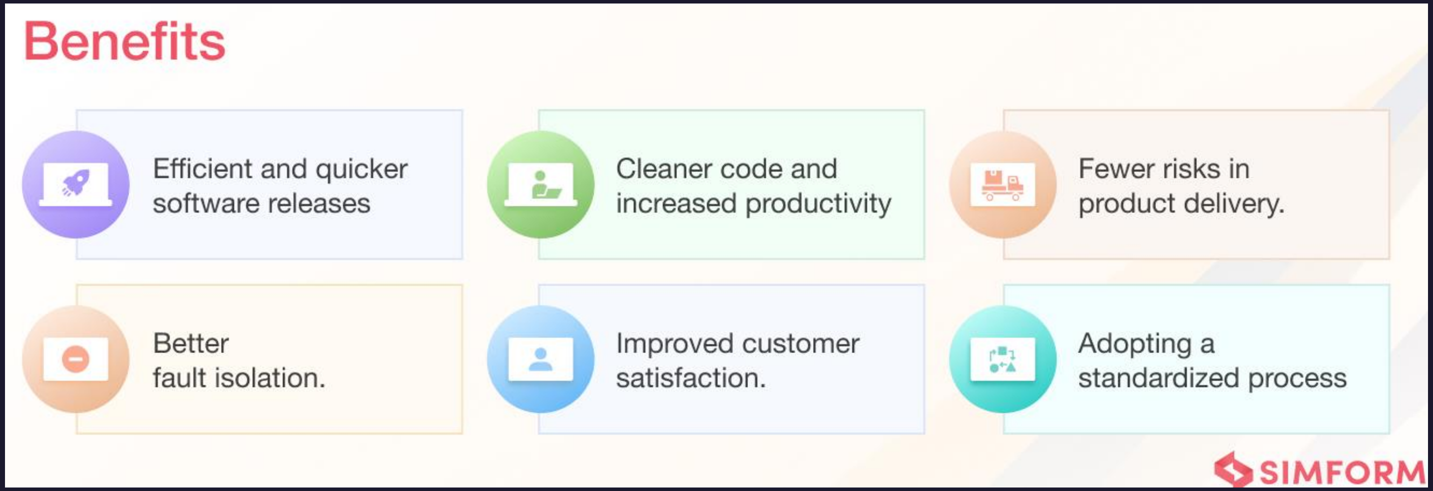
is another step beyond Continuous Integration, like Continuous Delivery. The difference is that instead of deploying your application manually, you set it to be deployed automatically. Human intervention is not required.

CI/CD PIPELINE



Benefits OF CI/CD

1) Higher efficiency
2) Reduced risk of defects
3) Faster product delivery
4) Log generation
5) Quick rollback if required
6) Better planning
7) Efficient testing & monitoring
8) Cost-effectiveness
9) Smaller Code Changes
10) Fault Isolations
11) Faster Mean Time To Resolution (MTTR)
12) More Test Reliability
13) Faster Release Rate
14) Customer Satisfaction
15) Increase Team Transparency and Accountability
16) Easy Maintenance and Updates



Benefits

- Efficient and quicker software releases
- Cleaner code and increased productivity
- Fewer risks in product delivery.
- Better fault isolation.
- Improved customer satisfaction.
- Adopting a standardized process

SIMFORM

Benefits OF CI/CD cont.

- Avoid Cost : Automation of infrastructure creation hence faster deployment and less human error.
- Reduce Cost : Automation of infrastructure cleanup prevents unwanted cost on unused resources. Catching compile errors after merging reduces time spent on issues from new developer code.
- Fail Fast : The faster we detect the errors, the faster we act and fix the issues even before it occurs on production, and that would save a lot of time debugging and testing also will save money.
- Make Revenue – Faster and More Frequent Production Deployments ensures more quicker releases. Removal of manual checks before deployment means less time to market.
- Protect Revenue – Automated smoke test reduces downtime due to deploy related crash or a major bug. Automated rollback due to a job failure means a fast undo from production to working state.