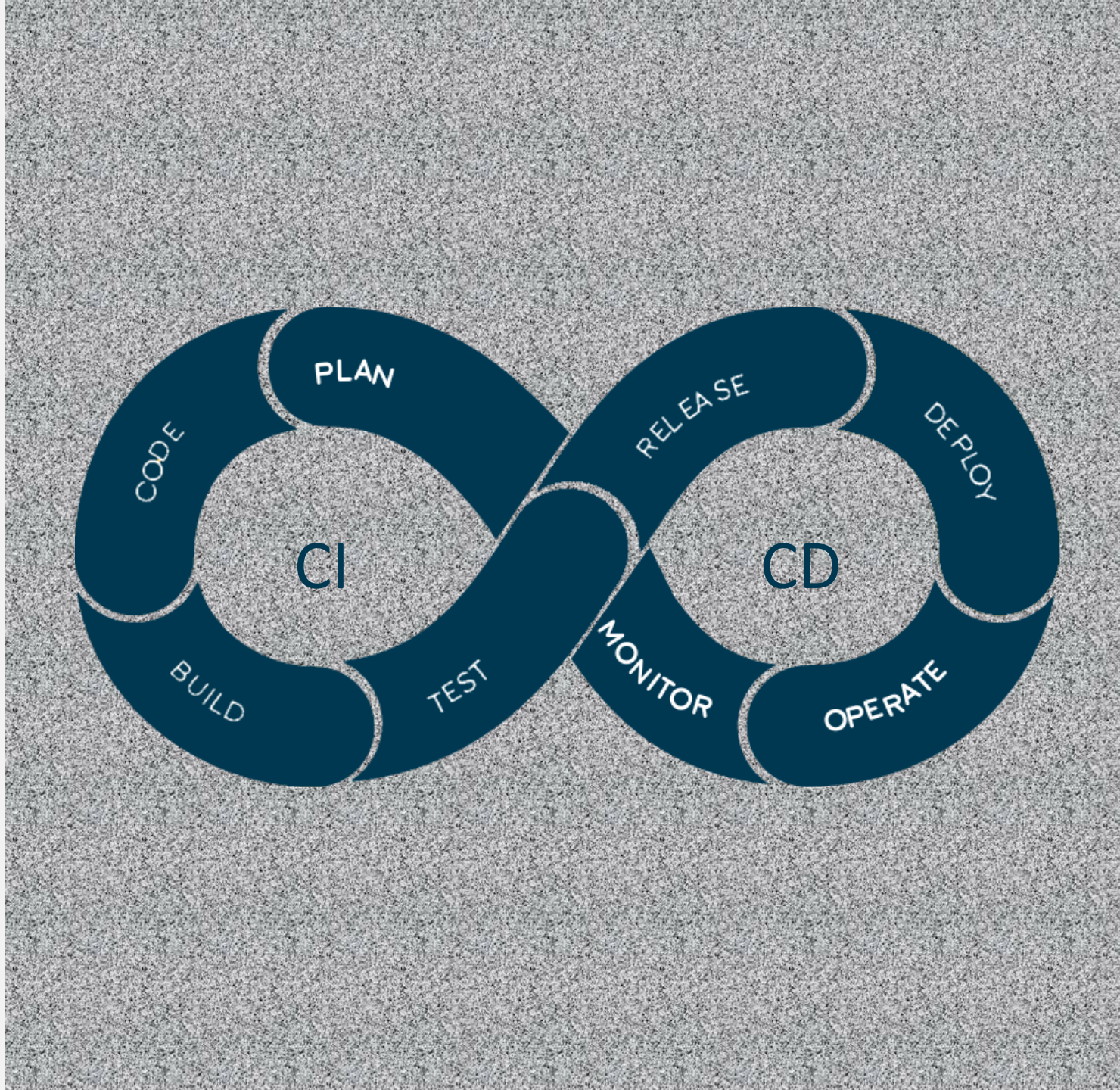
# EMBRACING AGILITY WITH CI/CD

Continuous Integration and Continuous Deployment of Cloud-Based Applications for Agiliy and better Productivity.

- By Emeka Anachebe



# What Is CI/CD

- CI/CD stands for Continuous Integration and Continuous Deployment or Delivery. It is a two-step process that significantly streamlines code development and deployment using automation in order to frequently deliver software.
- CI automates and improves developer tasks like source code integration and version control, so software can get into production faster.
- CD automates software testing deployment and promotion.
- Together, CI/CD is a powerful and unmatched engine of modern software development and it has untold benefits for businesses.

# Why Should Udapeople Adopt CI/CD?

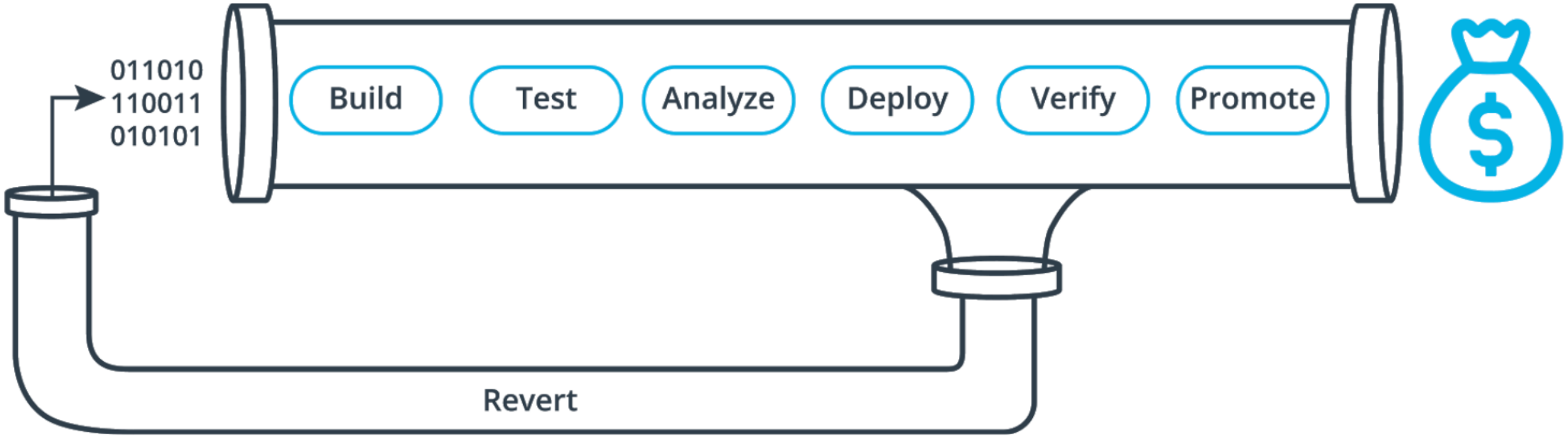
CI/CD would enable us:

- Ship to market quicker and more efficiently: The primary goal of a CI/CD pipeline is to deliver working software to users quickly and frequently with automated pipelines that require little or no human intervention.
- Significantly Increase our Return on Investment: With CI/CD, we'll ship to production quickly, and my doing so, we'll start making money quicker. Also, thanks to CICD's frequent releases, we can quickly release new features to address customer complaints.
- Reduce Cost: Automation reduces cost. Anytime a human does not have to intervene in the software development process, time, and thus money, are saved. Also, by automating the deletion of resources we do not need using IAC tools, we'll considerably save cost.
- Improve our MTTR: With CI/CD, our development teams will see issues quicker and fix them faster because they'd be pin-pointed. Thus, CI/CD would greatly improve our MTTR, which translates to increased trust from our customers and increased revenue.
- Reduce Errors: By automating the merge, test, delivery, and deployment processes in Udapeople, we would be eliminating human error from these processes. Thus, our delivery would be error-proof.

# The Principles of CI/CD?

- Repeatable Reliable Process
- Automate Everything
- Version Control Everything
- Bring the Pain Forward
- Build-in Quality
- “Done” Means Released
- Everyone is Responsible
- Continuous Improvement

# CI/CD Pipeline



# The Benefits of CI/CD

- Ensure superior code quality: With test automation offered by CI/CD, developers can know about code problems nearly in real time. This leads to a concept called “failing fast”. By doing so, teams wouldn’t  wasting time or resources with buggy code, and developers wouldn’t be burdened with endless “fix” requests when they’ve moved on to other projects; thus, eliminating context switching.
- Deliver faster with an accelerated release rate: Automation of the entire SDLC process leads to faster and more frequent releases.
- Fault isolation: Before CI/CD, development teams spent a lot of time troubleshooting to identify faults. With CI/CD, Developers would quickly identify and isolate faults. Bad code snippets and description of infrastructure faults would be included in the alerts that are automatically sent to developers.
- Simplified rollback: Roll backs are automated with CI/CD
- Optimum transparency and accountability: Thanks to continuous feedback, CI/CD makes the software development process completely transparent to the business side.
- Improved mean time to resolution (MTTR): CI/CD greatly improves the mean time to resolve faults, which translates to increased trust from our customers and increased revenue.
- Reduction of non-critical defects in backlog: CI/CD gives developers time to work on things they wouldn’t normally be able to, such as going back to fix older code and make it cleaner and more efficient.
- Cost Reduction: With the inclusion of Infrastructure as Code the CI/CD pipelines, infrastructure cost is greatly reduced. Also, the reduction of time taken to detect and fix errors, translates to less cost of building software, and more time to work on new features.

# Conclusion

The benefits of an CI/CD range from practical considerations like code quality and rapid bug fixes, to ensuring you're building the right thing for your users and improving your entire software development process.

At Udapeople, implementing CI/CD would improve our processes and revenue greatly.

I'll be happy to organise a DevOps Dojo to demonstrate these practically, and to simulate a proper CI/CD compliant environment where every member of the development team would be involved. This would aid our transition.

Thank you!