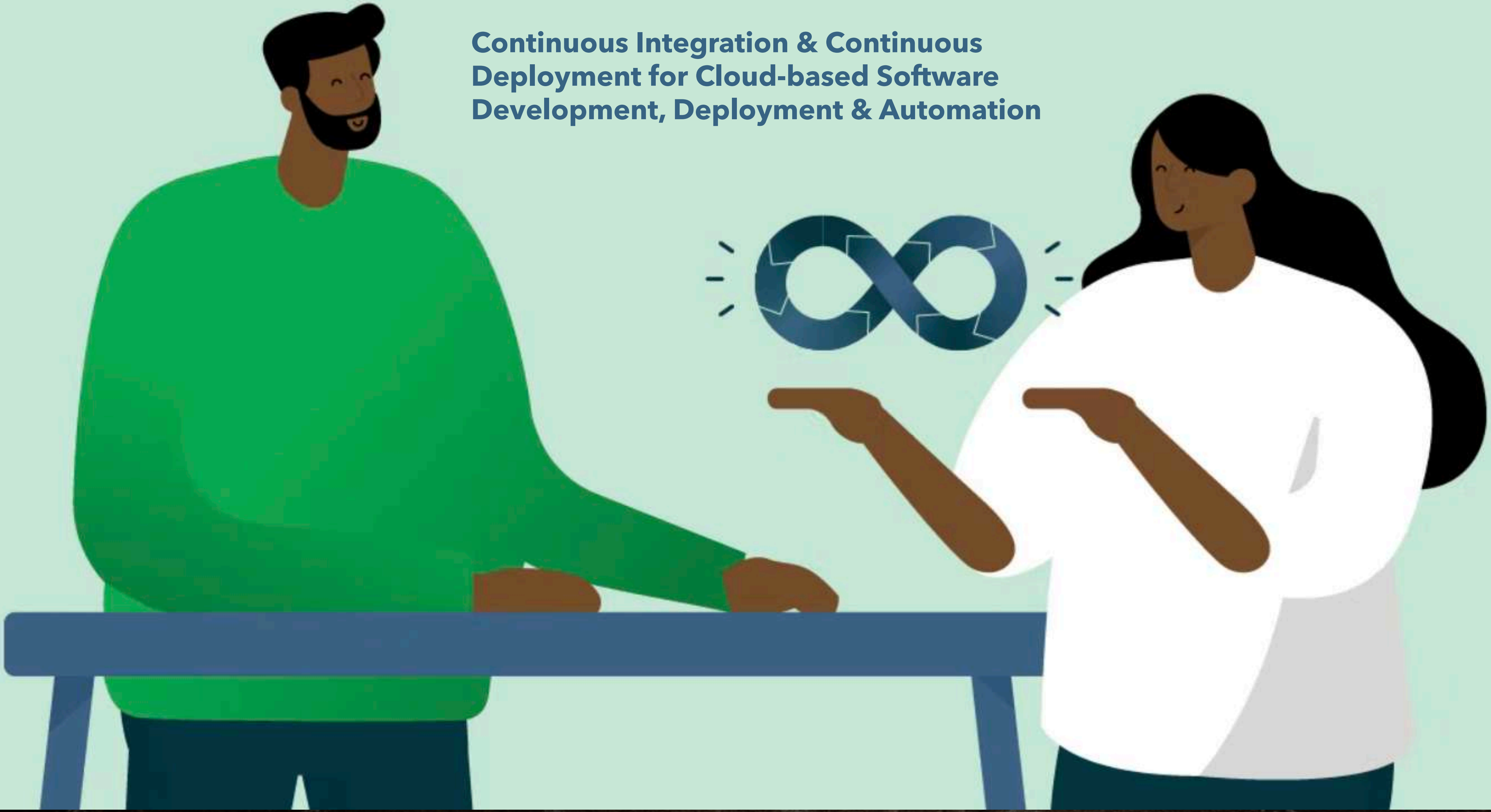
The Continuos Delivery Mindset with CICD!

Continuous Integration & Continuous Deployment for Cloud-based Software Development, Deployment & Automation



First of, what is CI/CD

CICD is automation! But automation of what?
CICD is automation of all the processes needed to frequently deliver software. It is divided into two (2) main parts, CI and CD

Continuous Integration (CI) is a development practice that helps ensure that software components work together. CI allows you to continuously integrate code into a single shared and easy to access repository.

Continuous delivery (CD) is the ability to deploy your integrated code into production without the need of human intervention. CD allows you to take the code stored in the repository and continuously delivery it to production.

What this means in summary is that CI/CD is the continuous development, deployment and delivery of software.

It is important to note that CI is different from CD. While CI has to do with everything that concerns the code, CD has to do with everything that concerns deployment and infrastructure.

Both processes can exist as stand-alones, but for the best results it would be best to have both processes running together.

Why we should adopt CI/CD

For Udapeople, CI/CD will enable us:

Ship software quickly and efficiently: The use of CI/CD pipelines will move software from the coding to deployment phases at scale, ensuring that the pace of development matches the needs of the business. This means we can deliver at speed to our clients and even have more time to build for new clients with the same staff strength.

Increase productivity: By implementing automated CICD processes, we can effectively merge the work between the development and operations teams and ensure we no longer spend time merging, building, testing, releasing, and deploying software manually. Instead, we can focus on writing better code and monitoring deployments for issues.

Reduce risk on delivery: We can automatically test every stage and feature before release. This would mean happier clients for us.

Standardize processes and eliminate human error: Automating the merge, test, delivery, and deployment processes means that we can eliminate human error from this processes.

Build confidence: When we achieve all the above, we would experience less errors, faster time to track errors and fix if any. This will translate to confidence in ourselves and client retention will go up which will further increase revenue for us.

Principles of CICD

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

Benefits of CICD

There are so many benefits of CI/CD but I will focus on few which are explained below.

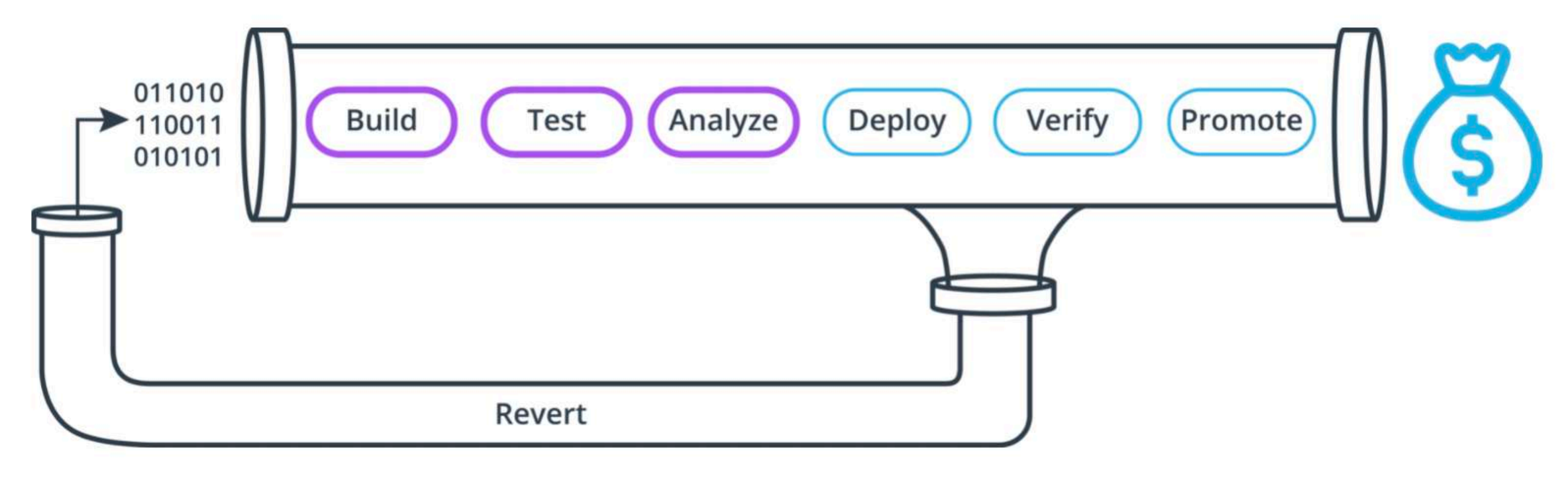
1.Reduce Cost: By practicing continuous delivery, errors will be detected faster and automatically. This will reduce the time spent on fixing code issues and bugs. When we reduce the time spent on detecting errors, cost of building code will reduce drastically and this is a plus for Udapeople.

2. Avoid Cost / Reduce Expenditure: In continuous delivery, there is something unit test. This makes it very easy to detect bugs during integration before it gets to production. Cost of deploying codes with bugs to production will be avoided. Only error-free codes will be get to production. Security Vulnerabilities will also be detected before it gets to production. This will help the organization to avoid cost of resolving security vulnerabilities in the production code. The beauty of this process is that all this will be done automatically.

3. Protect revenue: Smoke tests in CICD reduces downtime due to bugs during deployment because failed jobs will trigger automated rollback quickly from production back to working state. If the downtime is reduced during deployment, our client will be pleased with us and keep trusting us for projects. This means Udapeople can protect their revenue and paying clients.

4. create revenue: With CICD pipeline we can produce more software and push them quickly to the market, meaning we can generate more money for the firm!

The CICD Pipeline



This is the standard process an automation for building and deployment will pass through. Let us think of it as a pipeline that converts code to cash.

Conclusion

The emergence of DevOps has triggered the dire need for CI/CD. CI/CD is the future of software development and I believe it is a step in the right direction for Udapeople.

I am happy to organise a workshop where I can elucidate on how we can evolve to CICD and improve our processes and revenue.

Thank you.