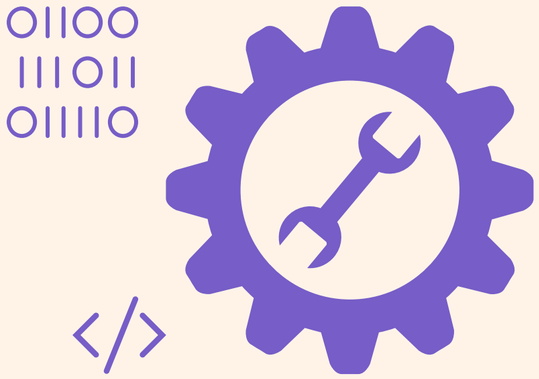
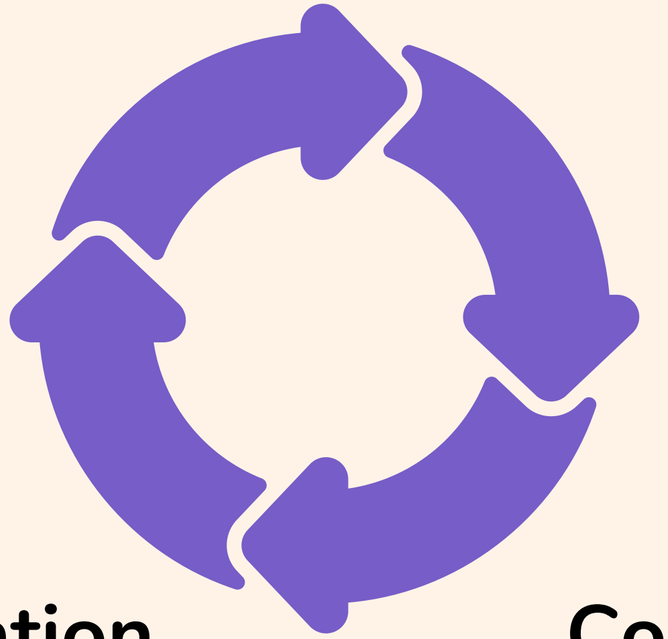
UDA PEOPLE

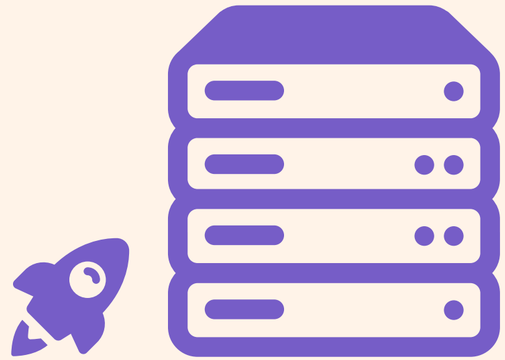
CONTINUOUS DELIVERY

A case for CI/CD

DEFINITIONS







Continuous Integration

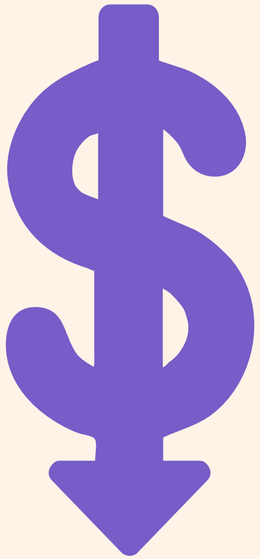
The practice of **merging** all developers' working copies to a **shared** mainline several times a day.

Continuous Deployment

A software engineering approach in which the value is **delivered** frequently through **automated deployments**.

CI/CD BENEFITS

REDUCED COSTS



Reduced bodged merges will result to **reduced costs** as code will be integrated into a single unified build process

Minimizing cost of spinning up resources in the case of defective software artifacts as defects will be identified earlier in the pipeline.

There will be only **one way to production** which means that the boring, repetitive process of deploying software will be automated thus controlling costs for the business.

CI/CD BENEFITS

INCREASED REVENUE

Developers will **focus on features** rather than deployments/spinning up infrastructure which will translate to **increased revenue** as features will be released fast to customers

Automation of the process of delivering software will result in **increased revenue** as features will be available to the user at a faster pace.

As **done means deployed,** stakeholders confidence will increase and this will protect revenue for the business. The discrepancy of code working on the developer machine but not working on the server will be reduced by automated CI/CD pipelines.