Fundamentals and Benefits of CI/CD to Achieve, Build, and Deploy Automation for Cloud-Based Software Products

A guide by UBAH Jeremiah

The fundamentals of CI/CD can be broken down into several key components:

1. Intro

Continuous Integration: This is the practice of automatically integrating code changes from multiple developers into a single repository multiple times a day..

Continuous Testing: Automated tests are run against the integrated codebase to ensure that all changes are working as expected and that the software remains functional.

Continuous Delivery: The software is packaged and deployed to a staging environment, where it can be further tested and validated before being released to production.

1. Intro

The fundamentals of CI/CD can be broken down into several key components:

Continuous Deployment: The final step in the CI/CD pipeline, this involves automatically deploying the software to production after it has been validated in the staging environment.

The benefits of implementing CI/CD for cloud-based software products are numerous. By automating the development, testing, and deployment processes, organizations can reduce the risk of errors and downtime, while increasing the speed, reliability, and quality of software development.

The Fundamentals and Benefits of CI/CD for UdaPeople

- ➔ Increased speed of delivery: CI/CD enables faster delivery of high-quality software products, which can lead to increased revenue by bringing new features to market sooner.
- ➔ Improved quality: Automated testing and continuous integration ensure that software is thoroughly tested and meets high standards, reducing the risk of costly bugs and downtime.

The Fundamentals and Benefits of CI/CD for UdaPeople

- ➔ Better collaboration: CI/CD promotes collaboration between development and operations teams, leading to better communication and improved efficiency, which can result in cost savings.
- ➔ Better scalability: Automated testing and continuous deployment allow organizations to scale their software development processes quickly and efficiently, which can help control costs.
- ➔

Technical Language	Value	Translation
Catch Compile Errors After Merge	Reduce Cost	Less developer time on issues from new developer code
Catch Unit Test Failures	Avoid Cost	Less bugs in production and less time in testing
Detect Security Vulnerabilities	Avoid Cost	Prevent embarrassing or costly security holes
Automate Infrastructure Creation	Avoid Cost	Less human error, Faster deployments
Automate Infrastructure Cleanup	Reduce Cost	Less infrastructure costs from unused resources

In conclusion

In conclusion, CI/CD is a crucial component of modern software development, particularly for cloud-based software products. By streamlining the development, testing, and deployment processes, organizations can reduce the risk of errors and downtime, while increasing the speed, reliability, and quality of their software products.