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

Presented by Charles Munene

Fundamentals

- Continuous integration is merging developers' changes back to the main branch as often as possible. The changes are then validated through builds, and automated tests run against the builds hence avoiding integration challenges that happen when waiting for release day to merge changes into the release branch.

- Continuous delivery is an extension of continuous integration since, on top of the automated testing, there is an automated release process where you can decide to make releases daily, weekly, fortnightly or any time by clicking a button.

- Continuous deployment goes one step further in that changes that pass all the stages of the production pipeline are released to customers with no human intervention. Only failed tests will prevent a new change from being deployed to production.

Benefits

- With CI/CD automated testing, which can run hundreds of tests in a matter of seconds, less time is spent by developers on issues from new developer code, which drastically reduces the costs of testers needed. , in turn, increases developer productivity as they focus on features.

- With CI/CD automated unit testing, fewer bugs make to production and less time in testing; hence maintenance costs are avoided. It also prevents costly security holes through automated security vulnerability detection.

- Automated infrastructure creation is possible with fewer human errors, resulting in faster deployments and fewer infrastructure costs incurred from unused resources through automated infrastructure cleanup.

Benefits

- With CI/CD, we will have faster and more frequent production deployment as the whole process is automated hence less time needed for a feature to be in the market hence increased revenue as new value-generating features are released more quickly. There is also less pressure on decisions for small changes, hence encouraging iterating faster and therefore, customers see a continuous stream of improvements, as quality increases every day, instead of every month, quarter or year.

- CI/CD results in reduced downtime from deploy-related crashes or major bugs through automated smoke tests and hence maintaining the customer trust and therefore protecting revenue. With automated rollback triggered by job failure, the production can be returned to its original working state and hence protecting more revenue.

Next Steps

- If it’s a new project with no users yet, you can automate every step from commit to production and hence have a great CI/CD process where all new changes are tested before being automatically sent to production.

- On the other hand, having an existing application with customers, you should start with continuous integration and continuous delivery and then later automate deployments ensuring the whole organisation is ready.