FUNDAMENTALS AND BENEFITS OF CI/CD TO ACHIEVE, BUILD, AND DEPLOY AUTOMATION FOR CLOUD BASED SOFTWARE PRODUCTS

JUDE IGWE

WHAT IS CICD

- Continuous integration (CI) and continuous delivery (CD), also known as CI/CD, embodies a culture, operating principles, and a set of practices that application development teams use to deliver code changes more frequently and reliably.
- CI/CD is a best practice for DevOps teams. It is also a best practice in agile methodology. By automating integration and delivery, CI/CD lets software development teams focus on meeting business requirements while ensuring code quality and software security.

THE OLD WAY

• Before ci/cd, the release cycle began with change management (cm) tickets, which each team was required to file eight days before the deploy. Product owners and development teams totaling about 30 people would meet to discuss the current set of releases every Wednesday at a cm coordination meeting. In this meeting, we would determine the order in which each of the 12 code bases would be deployed to avoid errors and dependency failures. Why the eight-day lead time? Each team needed time to determine the impact and dependencies of their work. Releases would only happen on Thursdays. If we needed a deployment outside of the normal cycle, we would have to file a cm exception. The amount of overhead and complexity involved in this process meant that product managers would spend significant portions of their time project managing their releases. Building things was no fun.

INTRODUCTION OF CICD

- With the introduction of CI/CD, deployment became fun. The development and operations teams collaborated more efficiently and the process became more predictable. CI/CD is our new coordination process. There are no more meetings to facilitate dependencies, no more deployment schedules, and no more lead time.
- Getting code out of the door with the risk of discovering showstoppers like critical bugs found late, infrequent releases, too much manual work, or late feedback has spawned a need for adopting Continuous Integration and Continuous Deployment/Delivery (CI/CD) technologies. For any competent software house, CI/CD is evolving as the modus operandi for rapid software delivery.
- The "automate everything" mindset is gaining momentum as the need to streamline sprawling delivery ecosystems amplifies. Leveraging the benefits of deployment at the push of a button has transformed the software distribution and delivery paradigm over the last few decades. CI/CD, in unison, automates the flow from design to deployment offering you better integration, delivery, and accelerated value.

BENEFIT OF CICD TO UDAPEOPLE

Cost Deduction

- As per one of the recent Forbes Insights surveys, "Three out of four executives agree that the amount of time, money, and resources spent on ongoing maintenance and management— versus new project development or new initiatives—is affecting the overall competitiveness of their organization."
- Adopting CI/CD will drastically reduce the running cost and increase revenuw of the organization by reducing the amount spent on resources to run a manual deployment.

Accelerated Release Cycles

• With CI/CD, you can visualize the development cycle from commit to production, continuously merge code, and repair detected bugs. You can release software to production multiple times after thorough testing without compromising quality parameters. This will ensure that will deliver our services to our customer in a timely manner and keep our customers happy. More happy customers, more revenue for the organization.

Loyal Customer Base

• The on-demand delivery experience that customers insist on leaves little room for errors and bugs. Matching them with the best fulfillment is critical to capturing market share. Implementing CI/CD reduces both lead time and response time, which in turn minimizes release risk and maximizes developer time. A stable operating environment and faster resolution to production requirements enable you to cement customer loyalty and create great experiences.

Superior Quality Code

• Developers release code into a shared repository in small batches, which enables them to conduct parallel testing. Rather than working in isolation, they share their builds with the entire team frequently. Teams collaborate to identify critical bugs, which ensures that bad code does not make it to production. Thus CI/CD implementation drives business growth by providing high-quality releases that have fewer errors and bugs. This will ensure lesser time for our developers to review their codes, which translates to efficient push to production and less vulnerability to attacks.

CONCLUSION

• CI/CD will ultimately become imperative for Udapeople – the benefits are simply too many. Migrating from the legacy software release cycles to CICD won't just accelerate the time-to-market, but also result in a productive and satisfied development team.