UDAPEOPLE

CI/CD BENEFITS PROPOSAL

OVERVIEW

- ★ CI/CD Explained.
- ★ Problems CI/CD Seeks to Solve.
- ★ Benefits of Cloud DevOps Principles
- ★ CI/CD Challenges

CI/CD Explained

CI/CD is an acronym for two distinct development processes; Continuous Integration (CI), and Continuous Delivery (or Deployment). It is made up of three components which are:

- ➢ Continuous Integration: The process of merging developer branches to the main branch several times a day. CI puts an emphasis on test automation and finally generates a high quality, deployable artifact(s).
- ➢ Continuous Delivery: ensure that changes to a software product can be released quickly to customers in an automated way and at any point in time.
- ➢ Continuous Deployment: is an extension of **continuous delivery**, and allows frequent automated deployments without any human interference.

Problems CI/CD Seek to Solve

CI/CD provides solutions to a number of problems which includes:

1. Manual Builds: Manually compiling source files, packaging them and updating database schemas cost software teams a lot – both in terms of development effort, as well as development time and cost.
2. Unhappy Clients: Any software application that is developed with longer periods between integrations is sure to make it exponentially more difficult to find and fix problems. This can easily knock a project off-schedule or cause it to fail altogether – resulting in unhappy clients.
3. Insufficient testing & Visibility: Unclear testing responsibilities, inadequate test resources, and infrequent communication often cause applications to be delivered with defects – which is a major cause for unhappy users.

Benefits of Cloud DevOps Principles

Using the earlier mentioned problems in the last section, the adoption and implementation of Cloud DevOps helps proffer the following solutions to the respective problems:

1. Manual Builds: CI automates the process of running the scripts after a build finishes. By integrating code into a shared repository several times a day, CI verifies every check-in using an automated build and prevents defects or regressions.
2. Unhappy Clients: Since CI enables frequent integration, there is significantly less backtracking to discover where things went wrong. By catching issues early and nipping them in the bud, developers can spend more time building features and less time in waiting to find out if the code is going to work.
3. Insufficient testing & Visibility: CI solves the problem of insufficient testing by laying the foundation of automated test suites. By monitoring the central code repository and running tests on every change that is made, CI collects test results and communicates them to the entire team working on the project.

CI/CD Challenges and how to Overcome Them

Some of the challenges of CI/CD include:

★ Cost Implication: establishing CI/CD comes with a high amount of initial cost and learning.

★ Continuous Development and Improvement: delivering CI/CD pipelines is not a one-off process, but requires constant support and maintenance.

It is noteworthy to mention that, the adoption of CI/CD will improve overall business processes and dramatically reduce costs on the long run. Hence it is strongly recommended for engineering teams and organizations to embrace CI/CD processes to optimize their production.