CI/CD

Save money & time with automation super powers!

Presented by Shrouk Ramadan

What is CI/CD ?

is the combined practices of continuous integration and continuous delivery or continuous deployment. They are sometimes referred to collectively as continuous development or continuous software development.allows organizations to ship software quickly and efficiently.CI/CD facilitates an effective process for getting products to market faster than ever before, continuously delivering code into production, and ensuring an ongoing flow of new features and bug fixes via the most efficient delivery method.

Continuous Integration (CI)

Continuous integration (CI) is the practice of automating the integration of code changes from multiple contributors into a single software project. allowing developers to frequently merge code changes into a central repository where builds and tests then run. Automated tools are used to assert the new code’s correctness before integration.

What is CD?

CD can stand for either continuous delivery or continuous deployment. Both involve taking the code continuously integrated and getting it able to deploy to an environment either QA or production. Continuous deployment takes the process one step further and performs the actual deployment to an environment.

CI/CD Benefits

- **Make Revenue**

  Faster and More Frequent Production Deployments ensures more quicker releases. Removal of manual checks before deployment means less time to market.

- **Protect Revenue**

  Automated smoke test reduces downtime due to deploy related crash or a major bug.
  Automated rollback due to a job failure means a fast undo from production to working state.

CI/CD Benefits (Cont.)

- **Avoid Cost**

  Automation of infrastructure creation hence faster deployment and less human error. Catch unit test failure ensures less bugs in production environment and less time testing. Detecting security vulnerabilities avoids future embarrassment from security attacks.

- **Reduce Cost**

  Automation of infrastructure cleanup prevents unwanted cost on unused resources. Catching compile errors after merging reduces time spent on issues from new developer code.

CI/CD Benefits (Cont.)

- **Reduction of non-critical defects in backlog**

  By now it's clear CI/CD is a time and money saver, so much so that it gives developers time to work on things they wouldn't normally be able to, such as going back to fix older code and make it cleaner and more efficient. The idea that developers cannot only tackle the backlog (it's called a backlog for a reason after all – who has time for this?), but also work on non-critical defects, is a game -changer brought to teams by DevOps and CI/CD.

- **Fail Fast**

  The faster we detect the errors, the faster we act and fix the issues even before it occurs on production, and that would save a lot of time debugging and testing also will save money.

CI/CD Benefits (Cont.)

- **Smaller Code Changes**

  One technical advantage of continuous integration and continuous delivery is that it allows you to integrate small pieces of code at one time. These code changes are simpler and easier to handle than huge chunks of code and as such, have fewer issues that may need to be repaired at a later date.

- **Fault Isolations**

  Fault isolation refers to the practice of designing systems such that when an error occurs, the negative outcomes are limited in scope. Limiting the scope of problems reduces the potential for damage and makes systems easier to maintain.

CI/CD Benefits (Cont.)

- **Faster Mean Time To Resolution (MTTR)**

  CI/CD reduces the MTTR because the code changes are smaller and fault isolations are easier to detect. One of the most important business risk assurances is to keep failures to a minimum and quickly recover from any failures that do happen. Application monitoring tools are a great way to find and fix failures while also logging the problems to notice trends faster.

CI/CD Benefits (Cont.)

- **More Test Reliability**

  Using CI/CD, test reliability improves due to the bite-size and specific changes introduced to the system, allowing for more accurate positive and negative tests to be conducted. Test reliability within CI/CD can also be considered *Continuous Reliability*. With the continuous merging and releasing of new products and features, knowing that quality was top of mind throughout the entire process assures stakeholders their investment is worthwhile.

CI/CD Benefits (Cont.)

- **Faster Release Rate**

  CI/CD continuously merges codes and continuously deploys them to production after thorough testing, keeping the code in a release-ready state. It's important to have as part of deployment a production environment set up that closely mimics that which end-users will ultimately be using. Containerization is a great method to test the code in a production environment to test only the area that will be affected by the release.

THANKS !