

CI/CD Fundamentals and Benefits

By: Mahmoud Ahmed

Fundamentals of CI/CD

- ▶ **Continuous Delivery:** An engineering practice in which teams produce and release value in short cycles.
- ▶ **Continuous Integration:** An engineering practice in which teams produce and release value in short cycles.
- ▶ **Continuous Deployment:** A software engineering approach in which the value is delivered frequently through automated deployments.

Fundamentals of CI/CD

- ▶ **Pipeline:** A set of data processing elements connected in series, where the output of one element is the input of the next one.
- ▶ **Infrastructure as Code:** The management of infrastructure using code.
- ▶ **Provisioning:** The process of setting up IT infrastructure.
- ▶ **Artifact:** A product of some process applied to the code repository.
- ▶ **DevOps:** A set of practices that works to automate and integrate the processes between software development and IT teams.
- ▶ **Testing:** A practice that seeks to ensure the quality of the software.

Benefits of CI/CD

- ▶ Catch compile Errors After Merge: help reducing cost as it leads to Less developer time on issues from new developer code
- ▶ Catch Unit Test Failures: it help avoiding extra costs due to Less bugs in production and less time in testing
- ▶ Detect security vulnerabilities: Prevent embarrassing or costly security holes
- ▶ Automate infrastructure cleanup : Less infrastructure costs from unused resources
- ▶ Faster and more frequent production deployments: New value-generating features released more quickly which help increasing Revenue
- ▶ Deploy production without manual checks : it increasing the revenue due to “less time to market” effect
- ▶ Automated Rollback triggered by job failure: Quick undo to return production to working state which help increase revenue

Best Practices for CI/CD

- ▶ **Fail Fast:** Set up your CI/CD pipeline to find and reveal failures as fast as possible. The faster you can bring your code failures to light, the faster you can fix them.
- ▶ **Measure Quality:** Measure your code quality so that you can see the positive effects of your improvement work (or the negative effects of technical debt).
- ▶ **Only Road to Production:** Once CI/CD is deploying to production on your behalf, it must be the only way to deploy. Any other person or process that meddles with production after CI/CD is running will inevitably cause CI/CD to become inconsistent and fail.
- ▶ **Maximum Automation:** If it can be automated, automate it. This will only improve your process!
- ▶ **Config in Code:** All configuration code must be in code and versioned alongside your production code. This includes the CI/CD configuration files!