

# CORE BENEFITS OF CI/CD

**Basil Ihuoma**

---

## **Outline** **Major of the Core Benefits**

### **Continuous Integration (CI)**

- Compilation
- Unit testing
- Static Analysis
- Dependency Vulnerability testing
- Store artifact

### **Continuous Development**

- Compilation
- Unit testing
- Static Analysis
- Dependency Vulnerability testing
- Store artifact

### **Continuous Deployment**

- This is an engineering practice in which teams produce and release value in short cycles. This happens by achieving CI/CD technology.



**CI/CD** or **CICD** is the combined practice of continuous integration (CI) and either continuous delivery or continuous deployment (CD). The idea behind it is that we should create jobs that perform certain operations like building, testing, deploying, and so on. Those jobs should be chained together to create a CI/CD pipeline. CI/CD bridges the gaps between development and operation activities and teams by enforcing automation in building, testing and deployment of applications. The process contrasts with traditional methods where all updates were integrated into one large batch before rolling out the newer version. Modern day DevOps practices involve continuous development, continuous testing, continuous integration, continuous deployment and continuous monitoring of software applications throughout its development life cycle. The CI/CD practice, or CI/CD pipeline, forms the backbone of modern day DevOps operations. CI/CD 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;** The concepts of continuous integration fits beautifully inside of the continuous delivery paradigm. This is the merging of all developers working copies to a shared main line several times a day. This is a practice that involves

developers making small changes and checks to their code. Due to the scale of requirements and the number of steps involved, this process is automated to ensure that teams can build, test, and package their applications in a reliable and repeatable way. CI helps streamline code changes, thereby increasing time for developers to make changes and contribute to improved software.

**Continuous Delivery** is an engineering paradigm, in which teams produce and release value in short cycles. CD is the automated delivery of completed code to environments like testing and development. CD provides an automated and consistent way for code to be delivered to these environments.

**Continuous Deployment;** this also fits nicely in the continuous delivery paradigm, since it is a software engineering approach in which value is delivered frequently through automated deployments. Continuous deployment is seen as the next step of continuous delivery. Every change that passes the automated tests is automatically placed in production, resulting in many production deployments.

**DevOps** is a set of practices and tools designed to increase an organization's ability to deliver applications and services faster than traditional software development processes. The increased speed of DevOps helps an organization serve its customers more successfully and be more competitive in the market.

---

## Benefits

**Detailed benefits from the technical and Business point of view:**

Technical	Benefit	Business
Unit test failure	Less cost	Avoidance of cost of Security holes
Automate infra creation	Less cost	Less cost of unused hardware
Go live without manual push	Increase revenue	Faster time on market

Automatic rollback	Save revenue	Faster recovery from failure
Automatic smoke test	Save revenue	Reduce downtime
Faster production code release	Increase revenue	New feature release more quickly
Security Vulnerability	Less cost	Less human error
Compile error	Reduce cost	Less developer time on making code

---

## Summary

**CI/CD ensures that the development team releases new codes rapidly and effectively by;**

- Less Backlog
- Improve transparency
- Increase test cycle
- Faster release cycle
- Reduced change time
- Ensure code quality
- Improve mean time to resolution
- Increase monitoring metrics
- Raise customer satisfaction level
- Fault Detection and Recovery