

# **The Fundamentals and Benefits Of CI/CD To Achieve, Build, And Deploy Automation for Cloud-Based Software Products.**

## **The Fundamentals Of CI/CD**

CI/CD is a fundamental practice for DevOps implementation in bridging the gap and allowing for synergy between the development team and the operations team. The modern development environment relies on CI/CD for the best delivery and optimum operations of software.

CI/CD is an acronym for Continuous Integration/Continuous Deployment

## **Continuous Integration**

Continuous Integration is the practice of automating the integration of code changes from multiple contributors into a single software project.

Continuous Integration includes;

- Code Compiling
- Unit Test
- Static Analysis
- Dependency vulnerability testing
- Store artifact

## **Continuous Deployment**

Continuous Deployment is a software engineering approach where value is delivered frequently through automated deployments. It's the process of "*Moving*" the artifact from the shelf to the spotlight.

Continuous Deployment includes;

Creating infrastructure

Provisioning servers

Copying files

Promoting to production

Smoke Testing (aka Verify)

Rollbacks

## **The Terminologies of CI/CD**

### **Pipeline:**

Just as expected of pipelines to be devices of transportation that take in inputs from an end and deliver the inputs as outputs on the other end, pipelines in CI/CD take a set of data as the output of a preceding element and give the data as input for the next element.

### **Continuous Integration:**

This is a practice of merging the works of all the developers into one mainstream at several intervals.

### **Continuous Delivery:**

This is a practice in engineering where teams create values and release the values in short successions.

### **Continuous Deployment:**

This is an engineering approach to frequently implement and deploy the deliveries in an automated manner.

### **Infrastructure as Code:**

This is the management of infrastructure through codes.

### **Provisioning:**

Provisioning is the process of setting up the IT infrastructure

### **Artifact:**

This is the process applied to the code repositories.

### **DevOps:**

This is a practice that automates and integrates processes between software development and IT teams

### **Testing:**

This is a practice that ensures the quality delivery of software. Here, the software is being tested before deployment to production.

### **The Benefits Of CI/CD**

The benefits of CI/CD in today's software engineering are enormous. As a software company, it is always a desire to have quality output without wastage of resources (Money, manpower and other resources). Hence, the need for CI/CD for cloud-based software.

The number one benefit of CI/VD is that it helps **save costs**. Errors in codes are detected before the codes make it to production. This will help in avoiding unnecessary costs which could be incurred if those errors are not corrected before the software makes it to production.

With CI/CD, security vulnerabilities are easily detected and this helps in avoiding the cost of fixing bridged security in the software when it is in production.

Reducing downtime to the barest minimum in the lifecycle of software is a way to make more profit as users can be confident of always accessing the software. This is what the Smoke test will help to achieve in CI/CD. With Smoke Test, possible errors are easily discovered before they could cause downtime.

With CI/CD, the failed job can easily be rolled back and returned to the previous working state while the errors are being fixed. This will help to avoid downtime which may in turn result in lost revenue.