

## **Udapeople CI/CD**

In this presentation we are going to discuss Utilizing CI/CD concepts in our project “UdaPeople”, starting from Creating the Infrastructure, Configuring the Infrastructure, successive Deployments, Applying Smoke tests, Failover scenarios and high availability, Monitoring Environments and defining Alerts, along with operational features for Maintaining KPIs and Performance.

### **What and Why CI/CD ?**

CI/CD concept is established essentially based on three techniques :

- Continuous Integration
- Continuous Deployment
- Continuous Delivery

### **Continuous Integration:**

A practice of merging all developers working copies to a shared main line several times a day which makes it easier for the Development and Operations Teams to :

- Compile New Codes
- Apply periodic Unit Tests
- Build Static Analysis
- Test Dependency Vulnerability
- Store Artifacts

### **Continuous Deployment:**

A software Engineering Approach in which the value is delivered frequently through Automated Deployments, which will enable the Company to introduce New Projects and deliver to maturity faster than before through “Develop Fast / Fail fast / Enhance fast”, applying these Concepts will facilitate frequent processing for :

- Creating Infrastructure
- Provisioning Servers
- Copying Files
- Promotion to Production
- Smoke Testing “verification”
- Rollbacks

### Continuous Delivery:

An Engineering Practice in which teams produce and release value in short cycles ,  
Triggering automatic compile sequences applying the following Concepts and techniques :

- Analysis of Code
- Repeatable reliable Process
- Automate Everything
- Test runs
- Version Control Everything
- Bring the Pain Forward
- Build-in Quality
- Done Means Released
- Everyone is Responsible
- Continuous Improvement

Using Each of the Preceding Concepts of CI/CD will lead the company and the project of Udapeople to Catch more Revenue and Save more cost as we can :

- Catch Compile Errors After Merge : less cost due to less time on issues from New developer Code
- Catch Unit Test Failures : Avoid cost due to less bugs in production and less time in testing .

- Detect Security Vulnerabilities : Avoid Cost by Preventing Costly security Holes
- Automatic Infrastructure Creation : Avoid Cost due to less human error and faster deployments .
- Automate Infrastructure Cleanup : Reduce Cost Due to less Infrastructure cost from unused resources
- Faster and more frequent Production Deployments : Increase Revenue with new features introduced and released more quickly .
- Deploy to production without Manual Check : Increase Revenue due to less time to market .
- Automated Smoke Tests : Reduced Downtime from a deploy-related crash or major bug that will protect revenue and Cost .
- Automated Rollback Triggered by Job Failure : Quick undo to return production to working state will save money due to failures .

Finally , It's been very Clear after this Illustration for Benefits of Applying CI/CD to our Project "UdaPeople" , that it's much more profitable and revenue assuring to use CI/CD Concepts in our project , that will save us so much time , Effort , Money and Headache .

It will cut ways short from development to production and cut them way more short on failure and disaster recovery , which is the Target for any Mature Company to Achieve.