

## **Agenda Jenkins CI/CD Training – 2 days:**

### **Day1:**

- Understanding the DevOps movement & culture
- DevOps Lifecycle – All About ‘Continuous’
  - Continuous Development
  - Continuous Testing
  - Continuous Integration
  - Continuous Deployment
  - Continuous Monitoring
  - Continuous Feedback
- Continuous Integration and its definition, overview of various CI Servers/Tools - Jenkins, Travis City, CircleCI, Bamboo etc
- Jenkins/Hudson history, Cloudbees introduction
- Overview of various types of Jenkins Installations
  - Windows – MSI Installer (as a service)
  - Linux Ubuntu or AMI (AWS or any other Linux box)
  - Generic war file
  - Docker container (AWS or any other Linux box)
- Installation on Jenkins MSI on all participants machine, Sample free style Jobs, Repeatable Jobs
- Significance of JENKINS\_HOME and its location for various installation types
- Jenkins Generic Configurations & Tool set-up
  - LDAP (open LDAP for setting up Authorization)
  - SMTP (emails & notifications)
  - Setting up - Maven, Gradle, Ant, Git, JDK
- Installation of Git, JDK & Maven, configure Jenkins to use Java & Maven correctly
- User Administration & Global Security
  - Adding new users, setting up fine grain access using Role Based Authorization Plugin
- Git Job – Put up a Free Style Jenkins job to connect to GitHub Repository, pull up a Maven based Java repository and build it
- Parameterized build Job for checking out a GitHub Repository with multiple branches and injecting branch parameter as an input and significance of passing parameters to various Jenkins Jobs

### **Day 2:**

- Sample Java & Maven Job, Integration with Static Code Analyzers (SonarQube and/or Jacoco, Cobertura, PMD etc) for analyzing Code repositories, code coverage reports for Jacoco, JUnit reports & PMD analysis reports
- Continuous Deployment of Maven based Java web application to a Tomcat Web Server using Deployment to Container plugin
- Automated Continuous Deployment using GitHub web-hook for automatically triggering Jenkins job based upon code changes in GitHub Repository

- Distributed Builds and Need for Agent/Node set-up, setting up Master-Slave configuration using JNLP protocol and delegating a new job to run on Slave/Agent node
- Introduction to Pipelines, setting up simple pipeline job using UpStream – DownStream Jobs, visualizing Pipeline job using Delivery Pipeline Plugin
- Jenkins 2.0 pipeline syntax using Scripted & Groovy DSL, sample pipeline job for setting up simple multi-stage job
- Jenkinsfile syntax builder and setting up pipeline job for checking out Maven based web-app from GitHub Repository, running Unit Tests, deployment and sending appropriate notifications
- Jenkins – Back-up & restoration, shutting down gracefully, Plugin Management
- Docker Jobs (if required)
  - Demo of a Job that can build DockerFile Image and push it to DockerHub Public Registry