**Prerequisite:**

Participants should have basic knowledge of linux scripting

It will be a 80% practical and 20% theory session

**Day 1 (4 Hours)**

1. Why Jenkins?
2. Current issues which Jenkins can resolve
3. Getting Started with Jenkins
   1. Learning Objective
      1. In this module, you will learn about Jenkins, setting up Jenkins environment on your local setup, terminologies used in Jenkins and creating Jenkins jobs.
   2. Topics:
      1. Install Jenkins
      2. Starting Up with Jenkins
      3. Jenkins Architecture and terms of Jenkins
      4. Overview of Jenkins UI
      5. Creating a Jenkins Job
      6. Configuring a Jenkins job
   3. Hands-On:
      1. Installing the Jenkins setup, create a simple Jenkins job, delete, fail, disable the job

**Day 2 (4 Hours)**

1. Introduction to GIT
   1. What is GIT?
   2. Common commands in GIT
   3. Integrate GIT with Jenkins
   4. Git vs GITHub
   5. Create account on GitHub
   6. Create Repo on Github
   7. Clone GitHub repo
2. Plugins and its uses
   1. Learning Objective
      1. In this module, we will learn about various plugins available in Jenkins and their usage. Then we will learn to configure Jenkins to work with other tools and polling for source code changes
   2. Topics:
      1. Introduction to Plugins
      2. Adding Plugins to Jenkins
      3. Commonly used plugins
      4. Configuring Jenkins to work with Python, Git and Docker
      5. Creating a Jenkins Build and Jenkins workspace
      6. Configure Jenkins to check for source code changes periodically.
   3. Hands-On:
      1. Adding the plugins to the Jenkins job, creating a Jenkins job to check for source code changes

**Day 3 (4 Hours)**

1. Setting Up Your Build Jobs and Security
   1. Learning Objective
      1. In this module, we will learn to learn to Build Jobs, add the advanced features to Jobs and Security.
   2. Topics:
      1. Creating a Freestyle Build Job
      2. Introduction to Build Triggers and to Build Steps
      3. Pre-and Post-Build Actions: Adding properties and properties files
      4. Running Your New Build Job
      5. Parameterized Builds
      6. Enabling Security in Jenkins
      7. Different Levels of Authentication
      8. Types of Access and Administration of the Access
   3. Hands-On:
      1. Creating Freestyle jobs, demo on security and authorization in Jenkins, sending email notifications, running the Jenkins job

**Day 4 (4 Hours)**

1. Implementing Automated Testing
   1. Learning Objective
      1. In this module, you will learn to setup Jenkins to Implement Automated Testing and publish reports.
   2. Topics:
      1. Introduction
      2. Automating Your Unit and Integration Tests
      3. Configuring Test Reports in Jenkins
      4. Displaying Test Results
2. Assignment
   1. Assignments will be assignment to the participants as a homework

**Day 5 (4 Hours)**

1. Metrics to Improve Quality
   1. Learning Objective
      1. In this module, we shall learn to check for the code coverage.
   2. Topics:
      1. Looking for foul Code through Code Coverage
      2. Activating and usage of Findbugs Jenkins plugin
      3. Verifying HTML Validity
      4. Reporting
      5. Jenkins with shell script build system
   3. Hands-On:
      1. Jenkins plugin for analyzing the bugs, reporting using Jenkins, Jenkins with a different build system
2. Distributed Jenkins Configuration
   1. Learning Objective
      1. In this module, we shall learn to create a distributed Jenkins system to handle concurrent build triggers, we will learn to set up master and slave configuration in Jenkins.
   2. Topics:
      1. Introduction to Distributed Jenkins Build
      2. Configuring Master Jenkins node
      3. Configuring the Jenkins slave
      4. Managing nodes and distributing jobs over nodes
      5. Binding Jobs on the master and slave setup
      6. Labeling the nodes to run a specific job
   3. Hands-On:
      1. Configuring slave node in your Jenkins, adding labels to nodes, managing the nodes

**Day 6 (4 Hours)**

1. Maintain and Use Jenkins
   1. Learning Objective
      1. In this module, we shall learn the Best Practices in Jenkins culture. You will also learn the Do's and Don'ts in Jenkins setup
   2. Topics
      1. How to maintain Jenkins
      2. Do's and Don'ts of Jenkins
      3. Backup of Jenkins and Migrating Jenkins from one server to another.
   3. Hands-On
      1. Backup of Jenkins
2. Assignment
   1. Assignments will be assignment to the participants as a homework

**Day 7 (4 Hours)**

1. Performing Automated Deployment and Continuous Delivery
   1. Learning Objective
      1. In this module, we shall learn to deploy an application to a web server using NGINX and understanding pipeline and parallel builds.
   2. Topics
      1. Deployment Overview
      2. Implementing Automated and Continuous Deployment
      3. Deploying an application to an application server
      4. Install and configure NGINX.
      5. Deployment of Simple Python web application.
      6. Jenkins Build Pipeline
      7. Parallel Jenkins build
      8. Achieve generated Artifacts
      9. Jenkins integrations
      10. Scaling Jenkins
   3. Hands-On
      1. Continuous deployment using Jenkins, parallel and pipeline builds, Jenkins integrations

**Day 8 (4 Hours)**

1. Jenkins Pipeline
   1. Learning Objective
      1. In this module, we shall learn to implement a project’s entire build/test/deploy pipeline in a jenkinsFile and create a fully automated Jenkins pipeline
   2. Topics
      1. Overview of Pipeline as code
      2. Overview of Pipeline Plugin
      3. Automated Jenkins Pipeline
   3. Hands-On:
      1. Jenkins pipeline
2. Pipeline as a code
3. Setting up Environment using Environment variables

**Day 9 (4 Hours)**

1. Jenkins Tips and Tricks
   1. Global environment variables in Jenkins
   2. Create custom global environment variables
   3. Modify the Jenkins URL
   4. Meet the Jenkins' cron: Learn how to execute Jobs automatically
   5. Learn how to trigger Jobs from external sources: Create a generic user
   6. Trigger Jobs from Bash Scripts
2. Introduction to Docker
3. Project/ Case Study: Implement CI/CD with Jenkins Pipeline and Docker
   1. Define the steps for Pipeline
   2. Build: Create a Dockerfile and build an image
   3. Build: Create a Docker Compose file to automate the Image build process
   4. Build: Write a bash script to automate the Docker Image creation process
   5. Build: Add scripts to the Jenkinsfile
   6. Test: Test code using Docker
   7. Test: Create a bash script to automate the test process

**Day 10 (4 Hours)**

1. Project/ Case Study: Implement CI/CD with Jenkins Pipeline and Docker - Continued
   1. Test: Add test script to Jenkinsfile
   2. Create a remote machine to deploy containerized app
   3. Push: Create Docker Hub account
   4. Push: Create a Repository in Docker Hub
   5. Push: Push/Pull Docker images to Repository
   6. Push: Write a bash script to automate the push process
   7. Push: Add push script to Jenkinsfile
   8. Deploy: Transfer some variables to the remote machine
   9. Deploy: Deploy application on the remote machine manually
   10. Deploy: Transfer the deployment script to the remote machine
   11. Deploy: Execute the deploy script in the remote machine
   12. Deploy: Add deploy script to Jenkinsfile
   13. Create a Git Repository to store scripts and the code for the app
   14. Create the Jenkins Pipeline
   15. Modify the path when mounting Docker volumes
   16. Create the Registry Password in Jenkins
   17. Add the private ssh key to the Jenkins container
   18. Add post actions to Jenkinsfile
   19. Execute Pipeline manually
   20. Create a Git Hook to automatically trigger Pipeline
   21. Start the CI/CD process by committing new code to Git!