

GITLAB

Duration: 5 Days (8 hours/Day)

Hands-On Format: This hands-on class is approximately 80/20 lab to lecture ratio, combining engaging lecture, demos, group activities and discussions with comprehensive machine-based practical programming labs and project work.

Module 1 – Introduction to CICD and GitLab

What is CI/CD?

What is GitLab

GitLab Features

GitLab Architecture

Module 2 – Creating Projects

Creating a New Project

Building the Project Locally

Short Introduction to Images and Docker

Building the Project using GitLab CI

Adding a Test Stage

Running a Jobs in Parallel

Running Jobs in the Background

Using Environment Variables for Managing Secrets

Deploying the Project using GitLab CI

Module 3 – Variables in GitLab CI/CD

What are GitLab CI/CD Variables?

Predefined GitLab Variables

Create Custom Variables

Secret Variables in GitLab

Module 4 – Integrating Docker and Kubernetes with GitLab

Build Docker Image with GitLab

Push Image to Central Image Registry

Integrating Kubernetes and GitLab CI

Automatic Deploy Application to Kubernetes

Module 5 – Artifacts

Defining Artifacts

Using Artifacts in Future Stages

Module 6 – GitLab Monitoring

Integrate Prometheus with GitLab

Check Metrics of GitLab Runners

Course Content:

Jenkins (5 Days--8 Hours/Day)

☐ Describe the concept of continuous delivery

☐ Prepare, configure, and set up a deployment of Jenkins

☐ Work with Maven and freestyle job types

☐ Work with the Jenkins user interface and recognize how to monitor jobs

☐ Identify how to organize jobs as the number of jobs grow

☐ Describe how the Jenkins ecosystem works and where to find job examples and plugins

- ❑ Use version control to get code into Jenkins
- ❑ Compare the different ways metrics can be used in Jenkins to measure code quality
- ❑ Recognize the different ways to automate Jenkins through the command line interface and other methods
- ❑ Work with Jenkins jobs and the background identity service
- ❑ Work with Java code and the techniques in promoting it
- ❑ Identify how to add parameters into a workflow
- ❑ Work with Java and Apache Tomcat to automate Java builds
- ❑ Describe how Jenkins can be used to distribute builds to multiple machines
- ❑ Identify the different types of Jenkins jobs
- ❑ Identify how Jenkins can build jobs based on the occurrence of an event
- ❑ Describe how jobs can be scheduled in Jenkins
- ❑ Identify how Jenkins can poll for software changes
- ❑ Describe how Jenkins implements security
- ❑ Identify how Jenkins authenticates applications
- ❑ Identify how Jenkins authorizes users
- ❑ Use Jenkins as a stand-alone application
- ❑ Use Jenkins as a web application
- ❑ Describe the process for installing Jenkins as a Windows service
- ❑ Identify how Jenkins can be used for non-Java applications
- ❑ Describe in detail the Jenkins tool and how Jenkins implements continuous delivery