DevOps Training (Duration: 2 Days – 8 Hours/Day)

Day 1: Introduction to DevOps and Source Code Management

1. Introduction to DevOps

- Definition and significance of DevOps.
- The DevOps culture and principles.
- Benefits of DevOps in software development.
- DevOps lifecycle stages.

2. Source Code Management with Git and GitHub

- What is Source Code Management (SCM)?
- Introduction to Git and its key concepts (repositories, commits, branches).
- **Handson:** Setting up a Git repository.
- Using Git for version control.
- **Collaboration with GitHub:** Creating repositories, pushing, and pulling code.
- Branching and merging in Git.
- Resolving merge conflicts.
- Collaborative coding on GitHub.
- Best practices for version control.

Day 2: Continuous Integration, Infrastructure as Code, and Continuous Deployment

1. Build Automation with Maven

- Understanding the need for build automation.
- Introduction to Apache Maven.
- Creating a simple Maven project.
- Configuring dependencies and plugins.
- Building, packaging, and deploying with Maven.

2. Continuous Integration with Jenkins

- Exploring Continuous Integration (CI) principles.
- **Introduction to Jenkins:** Installation and setup.
- Creating Jenkins jobs for automated builds.
- Integration with Git repositories.
- Scheduling and triggering builds.

3. Infrastructure as Code with Terraform and Ansible

- Understanding Infrastructure as Code (IaC).
- **Introduction to Terraform:** Writing and applying infrastructure code.
- **Introduction to Ansible:** Writing playbooks for configuration management.
- **Handson:** Provisioning and configuring infrastructure with Terraform and Ansible.

4. Continuous Deployment with Docker and Kubernetes

- Introduction to Docker and containerization.
- Creating Docker images and containers.
- Introduction to Kubernetes and container orchestration.
- Deploying applications on Kubernetes clusters.
- Monitoring and scaling in Kubernetes.

Conclusion:

This two-day outline covers the Concepts of DevOps, source code management, build automation, continuous integration, infrastructure as code, and continuous deployment.

P.T.O

Delivery Method:

- This training program is delivered in a Virtual instructor Led Training (VILT) or Classroom Training format, emphasizing real-world application.
- It comprises approximately 70% hands-on labs, and all participants are expected to complete them.

Prerequisites:

- Basic understanding of Linux commands.
- Basic knowledge of a Cloud platform such as AWS.
- It's good to have an AWS Free Tier Account for Practice.

Required Software/Tools:

- A stable internet connection with open ports for SSH (22) and HTTP (80, 443).
- Access to the AWS Management Console and Playgrounds.
- **For Windows platform users:** Putty, Puttygen, MobaXterm, PowerShell, etc.
- **For Mac users:** Terminal.