DEVOPS

Overview: DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality. DevOps is complementary with agile software development; several DevOps aspects came from the agile methodology.

Course Objectives

- Understand the DevOps Concepts
- Understand GIT
- Understand Docker
- Understand Selenium
- Understand Maven
- Understand Jenkins
- Understand Puppet, Ansiable
- Understand Kubernetes, Nagios

Module 1:- - Introduction to DevOps

- What is Software Development
- Software Development Life Cycle
- Traditional Models for SDLC
- Why DevOps?
- What is DevOps?
- DevOps Lifecycle
- DevOps Tools

Module 2:- Infrastructure Setup

- EC2 Walkthrough
- Installation of DevOps Tools on cloud
- Git
- Docker
- Selenium
- Maven
- Jenkins
- Puppet
- Ansible
- Kubernetes
- Nagios

Module 03 - Software Version Control

- What is Version Control
- Types of Version Control System
- Introduction to Git
- Git Lifecycle
- Common Git Commands
- Working with Branches in Git
- Merging Branches
- Resolving Merge Conflicts
- Git Workflow

Module 04:- Continuous Testing

- What is Continuous Testing?
- What is Maven?
- Running Test Cases on Chromium Web Driver
- What is Headless Mode?

Module 05:- Continuous Integration using Jenkins

- Introduction to Continuous Integration
- Jenkins Master Slave Architecture
- Understanding CI/CD Pipelines
- Creating an end to end automated CI/CD Pipeline

Module 06 - Continuous Deployment: Containerization with Docker

- Introduction to Docker
- Understanding Docker Lifecycle
- Components of Docker Ecosystem
- Common Docker Operations
- Creating a DockerHub Account
- Committing changes in a Container
- Pushing a Container Image to DockerHub
- Creating Custom Docker Images using Dockerfile

Module 07 - Containerization with Docker: Ecosystem and Networking

- What are Docker Volumes
- Deploying a Multi-Tier Application using Docker Network
- Using Docker Compose to deploy containers
- What is Container Orchestration
- Container Orchestration Tools
- Introduction to Docker Swarm
- Deploying a 2-Node Cluster using Docker Swarm

Module 08 - Configuration Management using Puppet

- Need of Configuration Management
- Configuration Management Tools
- What is Puppet
- Puppet Architecture
- Setting up Master Slave using Puppet
- Puppet Manifests
- Puppet Modules
- Applying configuration using Puppet
- Puppet File Server

Module 09 - Configuration Management using Ansible

- What is Ansible?
- Ansible vs Puppet
- Ansible Architecture
- Setting up Master Slave using Ansible
- Ansible Playbook
- Ansible Roles
- Applying configuration using Ansible

Module 10 - Continuous Orchestration using Kubernetes

- Introduction to Kubernetes
- Docker Swarm vs Kubernetes
- Kubernetes Architecture
- Deploying Kubernetes using Kubeadms
- Alternate ways of deploying Kubernetes
- YAML Files
- Creating a Deployment in Kubernetes using YAML
- Services in Kubernetes
- Ingress in Kubernetes

Module 11 - Continuous Monitoring using Nagios

- What is Continuous Monitoring?
- Introduction to Nagios
- Nagios Architecture
- Monitoring Services in Nagios
- What are NRPE Plugins
- Monitoring System Info using NRPE plugins

Module 12 - Terraform Modules & Workspaces

- What is Infrastructure as a code?
- Iac vs Configuration Management
- Introduction to Terraform

- Installing Terraform on AWS
- Basic Operations in terraform
- init
- plan
- apply
- destroy
- Terraform Code Basics
- Deploying and end-to-end architecture on AWS using Terraform

At the end of the course participants will be able to 1. Develop a web application which involves database operations.