DevOps Fundamentals
Course duration: 3 days (24 hrs)

About DevOps

DevOps is a software development and delivery process that helps organizations to increase their ability to deliver application and services at high velocity. Implement process automation of every function in SDLC.

What will you gain from this training?

- Understand DevOps concepts like Continuous Integration, Delivery and Deployment, Infrastructure as code.
- How to setup end to end delivery pipeline.
- Understand Containerization of Application and orchestration of containerized applications.

Training Pre-requisites:

1) Linux Awareness.
    a. Linux commands, File System. etc
2) Basic Networking concepts.
3) SDLC awareness.

Day wise schedule:

Day-1

- **INTRODUCTION TO DEVOPS**
    - Define DevOps
    - What is DevOps
    - SDLC models, Lean, ITIL, Agile
    - Why DevOps?
    - History of DevOps
    - DevOps Stakeholders
    - DevOps Goals
    - Important terminology
    - DevOps perspective
    - DevOps and Agile
    - DevOps Tools
    - Configuration management
    - Continuous Integration and Deployment
- **CULTURE OF CONTINUOUS DELIVERY**
    - The deployment pipeline concept
    - A new way of testing
    - The role of automation
    - Shifting towards CD culture

- **INTRODUCTION TO CI/CD**
  - Agile Development
  - What is Continuous Integration
  - What is Continuous Deployment
  - Typical Setup for CI/CD
- **DEVOPS TOOLS LANDSCAPE**
  - The Choice of Cloud Platform
  - IaaS for DevOps
  - PaaS for DevOps
  - Containerization Tools
  - System Configuration Automation and Management
  - Continuous Integration (CI) Systems
  - Build and Dependency Management Systems
  - Selection of DevOps Tools
- **Git – DVCS**
  - Introduction
  - Version control systems
  - Local, Centralized and distributed
  - Installing Git
    - Installing on Linux
    - Installing on Windows
    - Initial setup
  - Git Essentials
    - Creating repository
    - Cloning, check-in and committing
    - Fetch pull and remote
    - Branching
- **JENKINS – CI / CD TOOL**
  - Introduction.
    - Understanding continuous integration
    - Introduction about Jenkins
    - Build Cycle
    - Jenkins Architecture
  - Installation
    - Obtaining and installing Jenkins
    - Installing and configuring GIT
    - Java installation and configuration
    - Maven Installation
    - Exploring Jenkins Dashboard.
  - Jobs
    - Creating Jobs
    - Running the Jobs
    - Adding and updating Plugins
    - Disabling and deleting jobs
  - Build Deployments
    - Understanding Deployment.
    - Tomcat installation and configuration
    - Deployment Plugins
    - Deploying a war file from Jenkins to Tomcat

Day-2

Jenkins continued…

- JIRA, Artifactory integration
- Securing Jenkins
  - Authentication
  - Jenkins Plugin
  - Authorization
  - Confidentiality
  - Creating users
  - Best Practices for Jenkins

• Ansible – Configuration Management

- Ansible Introduction:
  - Describe the terminology and architecture of Ansible.
  - Understanding requirement.
- Introduction to Ansible Modules.
- Using Ansible in Ad-Hoc mode.
- What are playbooks?
  - Implement playbooks.
  - Write Ansible plays and execute a playbook.
  - Manage variables and inclusions
  - Describe variable scope and precedence; manage variables and facts in a play.
- Variable handling in Ansible.
  - Variable declaration and hierarchy
- Implement task control
  - Manage task control, handlers, and tags in Ansible playbooks.
- What are Ansible Roles?
  - Create and manage roles.
- Talking about Ansible in practical environment. Implement Ansible in a
- DevOps environment.
  - Integrate with CI / CD tools etc.

• CHEF – PUPPET – ANSIBLE COMPARISON.

- Introduction to chef and Puppet
- Chef to Ansible comparison
- Puppet to Ansible comparison

• DOCKER AND KUBERNETES

- Introduction
  - What is a Docker
  - Dockers vs Virtualization
- Architecture
  - Docker Architecture.
  - Important Docker components

DAY-3

- Provisioning
  - Docker Hub.
  - Downloading Docker images.
  - Running commands in container.
- Custom images
  - Using Dockerfile for image creation
  - Creating a custom image.
  - Running a container from the custom image.
  - Publishing the custom image.
- Docker Networking
  - Accessing containers
  - Exposing container ports

        - Volume mapping and file sharing.
    - Docker Compose
        - Introduction to Compose
        - Using compose for Microservices driven application.
        - Running application with docker compose
        - Integrate Docker with Jenkins
- **KUBERNETES FOR Container Deployments**
    - Container Cluster management
    - Kubernetes objects
    - Writing for Manifest and object deployment
    - Replication controller.
    - Service
    - Deployment
    - PODS
    - Persistent Volume and Persistent Volume Claim.
- **INFRASTRUCTURE MONITORING**
    - Application and infra monitoring
    - VM and Container monitoring