DevOps Training

# 1. Introduction to DevOps

Definition and Importance

• What is DevOps?

• Benefits of DevOps: Speed, Rapid Delivery, Reliability, Scalability, Improved Collaboration, and Security.

DevOps Lifecycle

• Continuous Integration, Continuous Delivery, Continuous Deployment

• DevOps Tools and Practices

DevOps Culture

• Collaboration between Development and Operations

• Automation and Monitoring

# 2. Linux Basics

Introduction to Linux

• What is Linux? Overview of different distributions

• Why Linux is important in DevOps

Basic Commands

• File system navigation: ls, cd, pwd

• File operations: cp, mv, rm, touch, mkdir

• Permissions: chmod, chown

Shell Scripting Basics

• Writing and running scripts

• Variables, loops, and conditionals

# 3. Introduction to AWS (Amazon Web Services)

Overview of Cloud Computing

• What is cloud computing?

• Types of cloud services: IaaS, PaaS, SaaS

AWS Services Overview

• EC2, S3, RDS, Lambda, VPC, IAM

Setting Up an AWS Environment

• Creating an AWS account

• Launching an EC2 instance

• Basic security practices (IAM, security groups)

# 4. Jenkins for CI/CD

Introduction to Jenkins

• What is Jenkins? Role in CI/CD

• Jenkins Architecture

Setting Up Jenkins

• Installation and configuration

• Jenkins Dashboard overview

Creating and Running Jobs

• Freestyle projects

• Pipelines (Jenkinsfile)

Integration with Other Tools

• Integration with Git, Docker, and AWS

# 5. Docker for Containerization

Introduction to Docker

• What is Docker? Containers vs. Virtual Machines

• Benefits of Docker

Docker Basics

• Docker images and containers

• Dockerfile and building images

Docker Commands

• docker run, docker build, docker images, docker ps, docker stop

Docker Compose

• Multi-container applications

• Writing a docker-compose.yml file

# 6. Maven for Build Automation

Introduction to Maven

• What is Maven? Role in build automation

• Maven Project Object Model (POM)

Building and Managing Dependencies

• Understanding the lifecycle: clean, compile, test, package, install, deploy

Plugins and Extensions

• Commonly used Maven plugins

# 7. Kubernetes for Container Orchestration

Introduction to Kubernetes

• What is Kubernetes? Role in managing containers

• Kubernetes Architecture

Core Components

• Pods, Nodes, Deployments, Services

Basic Operations

• Deploying an application

• Scaling and updating applications

Managing Cluster

• Kubectl basics

• Monitoring and logging

# 8. Terraform for Infrastructure as Code

Introduction to Terraform

• What is Terraform? Role in infrastructure as code (IaC)

• Benefits of using Terraform

Terraform Basics

• Writing configuration files (.tf files)

• Providers and resources

Terraform Workflow

• terraform init, terraform plan, terraform apply, terraform destroy

State Management

• Managing Terraform state files

# 9. Shell Scripting for Automation

Importance of Scripting in DevOps

• Automating routine tasks

Advanced Shell Scripting

• Functions, error handling, debugging

Practical Examples

• Automating deployments, monitoring system health

# 10. Conclusion and Best Practices

Key Takeaways

• Importance of each tool in the DevOps lifecycle

• Continuous learning and improvement

Best Practices

• Security, monitoring, and scalability

• Collaboration and communication

# 11. Projects:

* Setup a DevOps CI/CD pipeline for web application
* Automated Website deployment with Docker
* Create a monitoring dashboard for the web application
* Building a scalable application with docker & Kubernetes Implement CI/CD for DevENV/ProdENV deployments Automated resource allocation
* Accessing application using Ingress Controller
* Upgrade Eks Cluster without Downtime
* 3 tier Architecture setup using terraform
* Deploying application via Helm Chart
* Argo CD Deployment