

Certified DevOps Master's Program – Complete Edition

Program Overview

- This Master's Program provides complete mastery of DevOps practices including CI/CD, Infrastructure as Code, Containerization, Monitoring, Security, and Production-grade Deployments.
- It focuses on real-world implementation using Git, Jenkins, Maven, SonarQube, JFrog, Docker, Kubernetes, Terraform, Ansible, Shell Scripting, Prometheus, Grafana, ELK Stack, AquaTrivy, and Complete Linux System Administration.

Complete Linux System & Server Administration

- Linux Architecture & Kernel Fundamentals
- File System Hierarchy & Disk Management (ext4, xfs)
- User & Group Management
- File Permissions, ACLs & Special Permissions
- Process Management (top, htop, ps, nice, kill)
- System Monitoring & Logs (/var/log)
- Package Management (RPM, YUM, DNF, APT)
- Boot Process & GRUB Configuration
- Systemd & Service Management
- Networking Configuration (IP, DNS, Routing, Bonding)
- SSH Hardening & Security
- Firewall Management (iptables, firewalld)
- SELinux Concepts
- LVM & Storage Management
- Swap Management
- Crontab & Job Scheduling
- NFS & Samba
- Apache & Nginx Installation
- Reverse Proxy Configuration
- Performance Tuning & Troubleshooting
- Server Hardening & Best Practices

Shell Scripting – Automation & Administration

- Introduction to Bash & Shell Environments
- Variables & Environment Variables
- User Input Handling
- Conditional Statements (if, case)
- Loops (for, while, until)
- Functions in Shell
- Arrays & String Manipulation

- Exit Codes & Error Handling
- Process Automation Scripts
- Log Monitoring Scripts
- Backup Automation Scripts
- Cron Job Automation
- System Health Monitoring Script Development
- Writing Production-ready Deployment Scripts

Git – Version Control System

- Git Architecture & Distributed Model
- Repository Initialization
- Branching & Merging Strategies
- Rebasing & Cherry Picking
- Git Hooks
- Conflict Resolution
- Git Workflows (Feature, GitFlow)
- Git Tags & Releases

Maven – Build Automation

- Maven Lifecycle
- POM Structure
- Dependencies & Plugins
- Multi-module Projects
- Artifact Packaging
- Maven Repositories (Local, Central, Nexus/JFrog)

SonarQube – Code Quality & Static Analysis

- Static Code Analysis
- Quality Gates
- Code Coverage
- Vulnerability Detection
- Integration with Jenkins

JFrog – Artifact Management

- Artifact Repository Concepts
- Versioned Artifacts
- Docker Image Registry
- Integration with CI/CD Pipelines

Jenkins – Continuous Integration & Delivery

- Jenkins Architecture (Master/Agent)
- Pipeline as Code (Declarative & Scripted)
- CI/CD Workflow Design
- Integration with Git, Maven, SonarQube, Docker
- Credentials Management
- Webhooks
- Distributed Builds

Docker – Containerization

- Docker Architecture
- Docker Images & Containers
- Dockerfile Creation
- Volumes & Networking
- Docker Compose
- Private Registries
- Container Security Best Practices

AquaTrivy – Container Security

- Vulnerability Scanning
- Image Scanning
- Kubernetes Security Scanning
- CI/CD Security Integration

Kubernetes – Container Orchestration

- Kubernetes Architecture (Master & Worker Nodes)
- Pods, ReplicaSets, Deployments
- Services & Ingress
- ConfigMaps & Secrets
- Volumes & Persistent Storage
- Namespace & RBAC
- Helm Charts
- Auto Scaling (HPA)
- Rolling Updates & Rollbacks
- Cluster Troubleshooting

Terraform – Infrastructure as Code

- Terraform Architecture

- Providers & Resources
- Variables & Outputs
- Remote State
- Workspaces
- Modules
- State Management
- Provisioners
- Real-world Infrastructure Deployment

Ansible – Configuration Management

- Agentless Architecture
- Inventory Management
- Playbooks & Roles
- Variables & Templates (Jinja2)
- Handlers
- Dynamic Inventory
- Integration with CI/CD

Prometheus – Monitoring

- Metrics Collection
- PromQL
- Alertmanager
- Kubernetes Monitoring
- Node Exporter

Grafana – Visualization

- Dashboard Creation
- Prometheus Integration
- Alert Visualization
- Monitoring Best Practices

ELK Stack – Logging & Observability

- Elasticsearch Architecture
- Logstash Pipelines
- Kibana Dashboards
- Centralized Logging
- Log Analysis & Alerting

CI/CD Project Implementation

- End-to-End DevOps Pipeline Design
- Code → Build → Scan → Package → Deploy → Monitor
- Production Deployment Strategy
- High Availability & Scaling

Course Duration

- Total Duration: 80 Hours
- Hands-on Practical Implementation Included

Who Can Attend

- Software Developers
- System Administrators
- DevOps Engineers
- IT Managers
- Project Managers
- Cloud Engineers
- Architects