Tech Aspirants

DevOps
With Multi Cloud

by

MR. T.SRINIVASA RAJU
20+ Yrs. I.T.Exp.

Flat No:109,1st Floor, Annapoorna Block, ADITYA ENCLAVE,
Kumar Basti,Ameerpet, Hyderabad-500003

+91- 9100549370

https://techaspirants.com

DevOps With Multi Cloud

Introduction To DevOps

- What Is SDLC
- What Are the Different SDLC Models
- Understanding The SDLC Models
- Limitations In Waterfall-Model
- Need of Agile & Why Industry Adapts the Agile Model
- Limitations In the Agile Process

What Is DevOps

- Understanding DevOps Movement.
- Why We Need DevOps
- Introduction To DevOps Life Cycle
- Buzz-Words: Ci/Cd
- Tools/Scripting/Languages Need to Effective Implementation of DevOps Linux
- Administration Topics for DevOps Engineers

Introduction to Linux

- Linux Architecture
- importance of Linux Administration
- Linux distributions and their differences
- Linux command line basics
- Basic file management (ls, cp, mv, mkdir, rm, etc.)
- File permissions and ownership (chmod, chown)

System Administration Fundamentals

- Managing users and groups (useradd, usermod, groupadd, etc.)
- Filesystem management and disk usage (df, du, mount, etc.)
- Process management (ps, top, kill, etc.)
- service management
- System monitoring and performance tuning Package management (apt, yum/dnf, zypper, etc.)
- Memory & swap management job Automation

Storage and Filesystem Administration

- Disk management (fdisk, parted, LVM)
- Filesystem types (ext4, XFS, Btrfs)
- RAID configuration and management
- Network File System (NFS) and
- SMB/CIFS configuration

Networking Basics

- Network configuration (ifconfig, ip, route, etc.)
- DNS configuration (resolv.conf, nslookup, dig)
- Firewall management (iptables, firewalld)
- SSH configuration and key management

System Logging and Monitoring

- Logging mechanisms and log rotation
- Monitoring tools (top, htop, sar, Nagios, etc.)
- Understanding system logs (syslog, journalctl)

Backup and Restore

- Backup Types and Strategies Backup Tools and Utilities
- What is snapshot
- Restoring filesystems
- How to take a backup on production servers

DNS (Domain Naming System) Management

- What is DNS?
- What is DNS server?
- How it works
- What is the format of domain name files to edit to configure the DNS
- What are the DNS records?
- What is the profile of the DNS?
- How to configure the DNS server and DNS client

DHCP (Dynamic Host Configuration Protocol)

- What is DHCP?
- How the DHCP works
- What is the disadvantage to assign the static IP address?
- What is the profile of DHCP?
- How to configure DHCP server and client

Database Management

- MySQL Server or MariaDB
- what is MySQL or MariaDB?
- what is the profile of MySQL or MariaDB?
- How to Configure MySQL or MariaDB

Database operations:

- create, insert, update, delete and backup of database.

Web Server/App server Management

- Apache httpd, nginx, tomcat, jboss
- What is webserver and app server management
- How these servers work?
- How to configure webserver

- How to configure app server on Linux servers deploying apps in to servers
- Different way to host websites
- Clustering: deploying app to multiple servers

Advanced Topics

- Kernel management and tuning SELinux / AppArmor basics
- High Availability (HA) and clustering concepts

Troubleshooting and Debugging

- Common Linux issues and debugging techniques
- Performance profiling and optimization

Introduction to Shell Scripting

- variables
- constants
- keywords,
- shell built-in commands
- operators
- conditional,
- looping,
- jumping and selection statements
- arrays and strings
- file handling (xml, yaml, json)
- regular expressions, sed, awk commands
- handling errors
- handling functions

Project 1- Real time Scripting

- Deployment of web applications using shell scripting
- building custom libraries using shell scripting
- Check disk usage and alert if it exceeds a threshold.
- Monitor CPU and memory usage and log data periodically.
- Monitor service uptime and restart if it fails.
- Update system packages and send email notifications.
- Restart services at scheduled intervals.
- Clean up temporary files older than a certain date.
- Ping servers and services to ensure they are responsive.
- Send alerts if services are down or response times are high.

Introduction to Source Code Management- GIT/GUTHUB

- What is version controlling system(vcs)?
- Types of vcs
- What is source code management?
- What is git flow

- role of DevOps engineer in the source code management what is git and GitHub
- how to configure the git on Linux host
- How to create local repo(git) and remote repository (GitHub) Accessing GitHub central repository via HTTPS, ssh and CLI
- Working with git â Important Git commands
- what is branch in git
- branching strategies in git
- what is merging and its strategies
- tagging in git
- advanced git commands
- Integration with GitHub Actions, slack
- GitHub Repository Handling operations and administration
- collaborating with Git
- Introduction to Webhooks in Git
- git workflows
- Handling Large files in git repositories
- working with trees and modules in Git
- troubleshooting in git

Project 2:Deploying web applications as well other kind of automation tasks using GitHub actions.

Introduction to Build Automation

- What is build automation
- What is build, types of builds
- What are the diff types of build tools
- What is maven
- Setting-up of maven on Linux host
- Maven architecture
- Build Lifecycle in maven
- What are the diff types of lifecycles
- Project Object Model (POM)
- Effective POM
- Maven Repositories
- Maven Repository Structure
- Dependency Management
- Plugins Management
- Goals
- Dependency types and scope
- Build Profiles
- SNAPSHOT Versions

- Parent POM
- Dependency Exclusions
- Integrating Maven for Code quality checks, SAST
- Build Directory Structure
- Managing versions and configurations of plugins used in the project Release
- Management
- Deployment automation
- Continuous Integration (CI)
- Integration-Jenkins, GitLab CI
- Site Generation and Reporting
- importance of Unit and Integration Testing for DevOps Engineers

Project 3:Implement Deployment of application to multiple environments using Maven.

Introduction to Gradle

- What is a Build Tool?
- (Ant, Maven, Gradle evolution)
- Gradle Architecture: Tasks, Plugins, Build Scripts,
- Installing Gradle & Project Structure
- Create a simple Gradle project (Java or Groovy)
- Gradle Build Scripts
- Gradle build lifecycle (Initialization, Configuration, Execution)
- Understanding build.
- gradle (Groovy DSL) & settings.gradle
- Common Gradle tasks (build, clean, assemble, check)
- Write a custom Gradle task
- Dependencies & Repositories
- Dependency Management basics
- Repositories: Maven Central, JCenter, local repo
- Dependency configurations (implementation, Api, compile Only, runtime Only, test Implementation)
- Hands-on: Add external libraries to project + manage versions
- Plugins & Multi-Project Builds
- Built-in plugins (Java, Application, War)
- Third-party plugins (Spring Boot, Docker, etc.)
- Multi-module (multi-project) builds
- Configure a Java + Spring Boot project with Gradle plugin
- Testing & Code Quality
- Running tests with JUnit/TestNG in Gradle
- Test reporting

- Code quality plugins
- (Checkstyle, PMD, SpotBugs, Jacoco for coverage)
- Configure a unit test + generate a test report
- Gradle Performance & Advanced Concepts
- Incremental builds & build caching
- Parallel execution
- Gradle Wrapper (gradlew) for consistent builds
- Build scans
- Run builds with Gradle Wrapper + analyze performance
- Gradle in CI/CD
- Using Gradle with GitHub Actions
- Versioning & Release management in Gradle

Project 4: Configure a CI pipeline that builds & tests a Gradle project

Introduction to SonarQube

- Introduction to SonarQube
- Installation and Setup
- Project Setup
- Running SonarQube Analysis
- Dashboard Overview
- Integrating SonarQube with Build tools-Maven
- Code Quality Gates
- Issues Management
- Rule Sets and Profiles
- Custom Rules and Quality Profiles
- Code Duplication Analysis
- Security Hotspots
- Project Branches and Pull Requests
- Quality Metrics and Reports
- Notifications and Alert

Introduction to Jfrog-Artifactory

- Introduction to Jfrog Artifactory
- Installation and Setup
- Repository Management
- Artifact Management
- Artifact Metadata
- Access Control and Permissions
- Integration with Build Tools (e.g., Maven, Gradle)
- JFrog-cli
- Proxying Remote Repositories

- Repository Layouts
- Artifact Promotion
- Search and Metadata Queries
- Checksum-Based Storage
- Content Distribution
- Security and Vulnerability Scanning
- Backup and Restore

Project5: Integrating all above tools like git, maven, JUnit, SonarQube and Jfrog to implement the ci with Shell Scripting as well as groovy with Jenkins.

Introduction to Docker:

- Docker architecture
- setting up docker on Linux hosts
- differences between containers and vmâ
- docker daemon configurations
- discussion on container platforms
- docker file system
- images and containers
- building custom images using docker file
- introduction to docker hub
- integrating docker hub with GitHub
- webhooks in docker
- docker networking
- docker compose
- docker volumes
- introduction to ECR, ECR
- introduction to docker security
- docker host security
- docker image security
- auditing and analyzing vulnerabilities in docker containers
- docker secrets
- container monitoring
- Introduction to docker swarm
- limitation in docker

Project 6: Integrating docker with CI

Project 7: Deploying containers in to ECS using code commit, codebuild, code deploy & code pipeline.

Introduction to Kubernetes[k8s]

- What is Kubernetes
- Advantages of Kubernetes
- Kubernetes Architecture
- Kubernetes Components
- Kubernetes Cluster Setup-using EKS
- Kubernetes Objects
- Pod
- Replication Controller
- ReplicaSet
- namespaces
- DaemonSet
- Deployment -blue/green deployment
- configmaps and secrets
- Service
- Volumes-PV, PVC
- ingress
- ingress controller
- liveness probe
- readiness probe
- stateful sets
- Horizontal pod autoscaling
- vertical pod autoscaling
- taints & tolerance
- node selector
- k8s security
- logging, observability and Monitoring in k8s
- ELK/EFK
- Helm Charts

Project 8: Deploying app in to k8s cluster using k8s, Helm,EKS,EFK,JENKINS,GITHUB,MAVEN,DOCKER.

Prometheus & Grafana

- Helm Charts
- Monitoring Fundamentals
- What is Monitoring & Observability
- Metrics, Logs, and Traces overview
- Types of monitoring: System,
  Application,
  Infrastructure,
  Prometheus Basics

Introduction to Prometheus

- Prometheus architecture: Server, exporters, push gateway, Alert manager
- Key concepts: Metrics, Labels, Jobs, Targets
- Prometheus data model & query language (PromQL)
- Installing Prometheus (Linux/Docker)
- Configuring scrape targets & exporters
- Prometheus Exporters & Metrics Collection
- Node Exporter (CPU, Memory, Disk, Network)
- cAdvisor (Docker container metrics)
- Custom metrics for applications
- Alerting rules & firing
- Alerts with Alert manager

Grafana Basics

- Introduction to Grafana
- Grafana architecture & data sources
- Installing Grafana (Linux/Docker)
- Connecting Grafana to Prometheus as a data source

Dashboards & Visualization

- Creating dashboards & panels
- Using queries (PromQL) in Grafana panels
- Visualizations: Graph, Table, Gauge, Heatmap
- Templating & variables in dashboards
- Sharing & exporting dashboards

Alerts & Notifications

- Grafana alerting vs Prometheus Alert manager
- Configuring alert rules
- Notification channels: Slack, Email, Teams, Webhook
- Testing & managing alerts

Advanced Prometheus & Grafana

- Recording rules & aggregation
- Prometheus federation & scaling
- Grafana Loki integration for logs (optional)
- Dashboards for Jenkins, Kubernetes, Docker, Applications
- Best practices for metrics, dashboards, and alerting

Real-World Hands-On Scenarios

- Monitor system metrics (CPU, RAM, Disk)
- Monitor Docker containers with cAdvisor & Node Exporter
- Monitor Kubernetes cluster metrics
- Create an integrated dashboard with alerts for critical metrics

Python Scripting:

- Introduction to Python
- Variables, DataTypes, Input /Output, Comments, DocString, Typecasting
- Control Structures: if, elsemelif, for, while, break, continue, pass
- Data structures: Strings, lists, tupes, sets, dictionaries
- Functions, Modules and Packages
- Error Handling
- Iterators, generators and Decorators

Introduction to OOPs:

- Encapsulation,
- inheritance,
- polymorphism and
- Abstraction

Project 9: Deploying application using CI/CD Pipeline-with-Python Introduction to Ansible

What is Ansible?

- Configuration Management
- Ansible Features
- Ansible Architecture
- Ansible Installation
- Ansible Commands
- Ansible Play Books
- Tags
- Handlers
- Group Variables and Host Variables
- Loops
- Conditional Statements
- Ansible Vault
- Ansible Roles
- Ansible Galaxy
- Ansible Modules

Project 10: Deploying application in to k8s cluster using ansible-playbooks tools: git, GitHub, maven, JUnit, sonar, Jfrog, docker, k8s, ansible, Jenkins scripting: groovy.

Introduction to Jenkins:

- Introduction & Fundamentals
- What is Jenkins and CI/CD concepts

- Jenkins architecture: Master, Agent (Slave), Executors, Nodes
- Installation methods: Docker, Linux, Windows
- Jenkins UI: Dashboard, Jobs, Blue Ocean overview

Jobs & Pipelines Basics

- Freestyle Jobs
- Pipeline Jobs (Declarative vs Scripted)
- Multi-branch and Folder Jobs
- Jenkins file syntax & structure

Source Code Management (SCM)

- Git/GitHub/GitLab integration
- Poll SCM vs Webhooks
- Branch and tag management in pipelines
- Build & Test Automation
- Build tools: Maven, Gradle, npm, Python pip
- Running unit tests and generating test reports
- Archiving build artifacts
- Build environment setup and cleanup

Advanced Pipeline Features

- Parallel and sequential stages
- Conditional execution (when, input)
- Parameterized builds for dynamic pipelines
- Post-build actions and triggers

Deployment & Continuous Delivery

- Continuous Deployment vs Continuous Delivery
- Deploying apps to Tomcat
- Deploying Docker images
- Deploying on Kubernetes
- Automating deployments with pipelines

Jenkins Security & Credentials

- User roles, RBAC, and matrix-based security
- Credentials management: passwords, tokens, SSH keys
- CSRF protection and securing pipelines
- Using credentials in pipelines (with Credentials)

Plugins & Ecosystem

- Essential plugins: Git, Pipeline Utility Steps, Slack, SonarQube, Blue Ocean
- Shared libraries for reusable pipeline code
- Jenkins monitoring: metrics, logs, performance tuning

Backup & restore Jenkins

- Best Practices &
- Real-World Scenarios

Introduction to AWS

- What is cloud computing
- What are the diff types of cloud models (public, private, hybrid)
- What are the broader services in cloud
- What is IAAS, SAAS, PAAS
- What is cloud architecture
- Introduction to Json and yaml
- Introduction to basic terms:
- Region
- az
- edgelocations
- outposts
- localzones
- wavelength zones
- Introduction To IAM
- what users, groups, policies, roles
- introduction to types of policies
- AWS organizations
- Introduction to EC2
- what is AMI
- what is instance
- diff between instance and AMI
- types of instances
- What is key-pair
- importance of ppk, pem files
- diff ways to launch ec2
- cost explorer on ec2
- Introduction block-level storage-EBS
- what is block
- how blocks stores the data
- what are the importance devices
- encryption and decryption
- backups and snapshots
- life cycle manager
- Introduction to VPC
- what is VPC
- what are subnets
- what is Internet gateway
- what is routing table
- what is NAT Gateway

- What is NACL
- Design VPC with public subnet
- Design VPC with public, private subnets using NAT Gateways
- What is EIP
- What is Endpoint and endpoint services
- Peering connections
- Vpn connections
- Introduction to route 53
- Introduction to CDN-CloudFront
- Introduction to ELB
- What is osi layers
- What are TCP layers
- Types of loadbalancers
- Introduction to Application load balancers
- Introduction to target groups
- Introduction to Auto scaling
- Introduction to Object-level storage-s3
- What is bucket
- Bucket policies
- Encryption-using KMS
- Static webstitehosting
- Introduction to WAF
- Introduction to ACM
- Introduction to Lambda
- Introduction to Developer-tools
- Codecommit &codebuild
- Code artifact &code deploys
- Code pipeline
- Introduction to database services
- SQLBASED-RDS
- NOSQL-DYNAMODB
- Introduction to Cloudwath
- Introduction to SNS
- Introduction to API-Gateways & APP-Mesh

Project 11: AWS Project: Design, Implementing & Deploying 3-Architecture

Project 12: MicroservicesDeployment using Cloud-native Deployment techniques

Introduction to IAC-Terraform

- what is IAC
- What is terraform
- terraform architecture
- what is terraform lifecycle
- setting up terraform on Linux Host
- Integrating Terraform with VSCODE
- workspace in terraform
- what is local and remote locking files
- what are the providers and plug-ins in terraform
- Defining resources (resource block)
- Managing resource attributes
- Using variables and expressions
- Organizing code with modules
- what are the terraform variables
- blocks in terraform
- expressions, conditions, looping statements in terraform
  modules in terraform

Project-13: Terraform-deployments

Azure & Azure DevOps Fundamentals

- Introduction to Azure
- Azure Global Infrastructure (Regions, Availability Zones, Resource Groups)
- Core Azure Services:
- Compute (VMs, App Services, Functions)
- Storage (Blob, Table, File, Queue)
- Networking (VNet, NSG, Load Balancer)
- Create a free Azure account + deploy a VM

Azure Core Services Deep Dive

- Azure Identity & Access Management (IAM, RBAC, Azure AD basics)
- Monitoring & Security (Azure Monitor, Log Analytics, Security Center)
- Cost Management and Pricing
- Create a Storage Account + Upload & Access Data

Introduction to Azure DevOps

- What is Azure DevOps and Why Use It?
- Overview of Azure DevOps Services:
- Azure Repos (Git & TFVC)
- Azure Pipelines (CI/CD)
- Azure Boards (Agile project mgmt)
- Azure Artifacts (package mgmt)

- Azure Test Plans
- Azure Repos & Pipelines (CI/CD Basics)
- Git fundamentals inside Azure Repos
- Build Pipelines: Concepts, YAML vs Classic
- Continuous Integration with Azure Pipelines
- Setup repo + configure a build pipeline (e.g., simple .NET/Java app)
- Release Management & Deployment

Release Pipelines basics

- Continuous Deployment to Azure App Service
- Deployment Strategies (Canary, Blue-Green basics)
- Deploy an app from Repo Pipeline Azure App Service

Agile Project Management & Artifacts

- Azure Boards: Epics, Features, Stories, Tasks
- Agile, Scrum, Kanban in Azure DevOps
- Azure Artifacts: Packages & Feeds (NuGet, npm, Maven)
- Create a backlog + sprint plan + use Boards for tracking

Integration & Best Practices

- Integrating Azure DevOps with GitHub, Slack, Teams
- Monitoring CI/CD with Azure Monitor & Application Insights

Project 14: Working CI/CD pipeline deploying a sample app to Azure App Service

AIOps & MLOps for DevOps Engineers

- What is AIOps? & What is MLOps?
- how they extend DevOps
- AIOps focus Monitoring, anomaly detection, automation
- MLOps focus â Model lifecycle, CI/CD for ML models
- Hands-on: Explore logs/metrics in ELK or Prometheus
- Observability & Data in AIOps
- Logs, Metrics, Traces (data sources for AIOps)
- Event correlation & noise reduction
- Send app logs,
- ELK/EFK stack & visualize anomalies
- ML Basics for DevOps
- ML workflow: Data, Training, Model, Deployment & Monitoring
- Role of DevOps in ML pipelines
- Run a simple ML model (e.g., scikit-learn model training in Python)
- Automation & Incident Handling (AIOps)
- Automated incident detection &
- Self-healing

- Alert fatigue reduction with AIOps tools (Moogsoft, AWS DevOps Guru)
- Create a Grafana alert
- send Slack notification
- CI/CD for ML Models (MLOps)
- Model versioning (MLflow, DVC basics)
- CI/CD pipeline for ML (training + testing + deploy)
- Build a Jenkins/Azure DevOps pipeline that deploys a simple ML model (Dockerized)

Deployment & Monitoring

- AIOps: Proactive monitoring with anomaly detection
- MLOps: Model deployment to Kubernetes / Docker
- Monitoring models: Drift, accuracy, performance
- Deploy ML model to Kubernetes + monitor logs/metrics

Integration & Best Practices

- Where AIOps & MLOps meet (smart monitoring + model pipelines)
- Tools overview:
- AIOps, Dynatrace, Datadog, Splunk ITSI, AWS DevOps Guru
- MLOps, MLflow, Kubeflow, SageMaker, Azure ML

Project:15
App + Logs, anomaly detection (AIOps)
ML model, trained + deployed with versioning (MLOps)

Benefits Taking Course in Tech Aspirants

- Realtime scenarios
- Realtime projects
- Mock tests
- Mock interviews
- Interview questions
- Resume preparation
- Physical / Digital Course Completion Certificate
- Total duration 5 months