DevOps Shack

Practical DevSecOps & Cloud DevOps RoadMap

[Click Here To Enrol To Batch-5 | DevOps & Cloud DevOps]

Module-1: Introduction to DevOps

Topics Covered:

1. **What is DevOps?**
   - Definition and origin of DevOps.
   - Core principles of the DevOps culture.
2. **Problems DevOps Solves**
   - Comparison of traditional vs. DevOps approaches.
   - Case studies on efficiency improvements.
3. **Workflow of DevOps in Companies**
   - Typical DevOps lifecycle: Plan, Code, Build, Test, Release, Deploy, Operate, Monitor.
   - Role of automation and collaboration across departments.
4. **Assignment**
   - Research and document the DevOps workflow in a real-world company (e.g., Amazon, Netflix).

Module-2: Linux Fundamentals for DevOps

Topics Covered:

1. **Introduction to Linux OS**
    - Importance of Linux in DevOps.
    - Distribution overview: Ubuntu, CentOS, Debian.
2. **Essential Linux Commands**
    - Daily usage commands: navigation, file manipulation, and system monitoring.
    - Scripting basics: writing simple bash scripts.
3. **Advanced Linux Management**
    - User and group management.
    - File system permissions and security essentials.
    - Basic networking commands and troubleshooting.
4. **Linux Assignments**
    - Debug common Linux errors.
    - Write shell scripts for automating routine tasks.
    - Design a security setup for a Linux server based on given requirements.
    -

Module-3: Version Control with Git

Topics Covered:

1. **Basics of Version Control Systems**
    - Importance of version control in software development.
    - Overview of popular VCS and introduction to Git.
2. **Mastering Git**
    - Git configuration and basic commands (init, add, commit, push, pull).
    - Branching strategies (Git Flow, GitHub Flow).
    - Merging, rebasing, and managing conflicts.
3. **Git Assignments**
    - Practice implementing branching strategies on a sample project.
    - Resolve merge conflicts in a collaborative project.
    - Use Git to manage versions of a personal project.

Module-4: Building with Maven and Node.js

Topics Covered:

1. **Maven for Java Projects**
    - Understanding Maven project structure.
    - Key Maven commands and lifecycle.
    - Building and managing dependencies.
2. **Node.js in DevOps**
    - Setting up a Node.js project.
    - NPM for package management.
    - Scripting with Node.js for automation.
3. **Build Assignments**
    - Troubleshoot a Maven build failure in a Java project.
    - Configure a Node.js project to automatically update dependencies and run tests before deployment.

Module-5: Continuous Integration with Jenkins and GitHub Actions

Topics Covered:

1. **Setting Up Jenkins**
    - Installation and configuration of Jenkins.
    - Creating freestyle, pipeline, and multibranch jobs.
    - Integrating Jenkins with tools like Git, Maven, and Docker.
2. **GitHub Actions for CI/CD**
    - Setting up GitHub Actions for a repository.
    - Writing YAML workflows for build, test, and deploy.
3. **CI Assignments**
    - Configure a Jenkins pipeline to build a multi-stage project.
    - Create a full CI/CD pipeline using GitHub Actions for a web application.

Module-6: Securing DevOps

Topics Covered:

1. **Introduction to Security in DevOps**
    - Importance of security in the DevOps pipeline.
    - Overview of DevSecOps and integrating security into the CI/CD pipeline.
2. **Deep Dive into SonarQube**
    - Setting up SonarQube: Installation and configuration steps.
    - Integrating SonarQube with CI tools like Jenkins.
    - Analyzing reports and managing code quality and security.
    - Enhancing code coverage through automated testing.
3. **Using Trivy for Vulnerability Scanning**
    - Introduction to Trivy and its capabilities.
    - Setting up and running file system and Docker image scans.
    - Interpreting scan results and addressing vulnerabilities.
4. **Prowler for AWS Security**
    - Overview of Prowler: A security tool designed for AWS.
    - Setting up Prowler and conducting security assessments.
    - Best practices for securing AWS environments.
5. **Docker Security with Docker Scout**
    - Introduction to Docker Scout and its role in securing Docker environments.
    - Implementing Docker Scout in your DevOps workflow.
6. **Security Assignments**
    - Create detailed vulnerability reports using SonarQube and Trivy.
    - Implement security best practices in a Docker environment using Docker Scout.
    - Use Prowler to audit an AWS environment and report findings.

Module-7: Artifact Management with Nexus and AWS CodeArtifacts

Topics Covered:

1. **Understanding Artifact Management**
    - The role of artifact management in DevOps.
    - Overview of artifact repositories and their importance.
2. **Nexus Repository Manager**
    - Setting up Nexus: Installation and initial configuration.
    - Managing artifacts: Publishing, retrieving, and storing artifacts.
    - Creating and configuring a Docker registry and Maven repository within Nexus.
    - Implementing cleanup policies to manage repository storage efficiently.
3. **AWS CodeArtifact**
    - Introduction to AWS CodeArtifact and its integration into the AWS ecosystem.
    - Setting up an artifact repository on AWS CodeArtifact.
    - Managing permissions and policies for secure access to artifacts.
4. **Azure Artifacts**
    - Overview and setup of Azure Artifacts.
    - Integrating Azure Artifacts with Azure DevOps pipelines.
    - Best practices for versioning and managing dependencies.
5. **Artifact Management Assignments**
    - Configure Nexus to manage Java and Docker artifacts, setting up appropriate cleanup policies.
    - Set up an AWS CodeArtifact repository and demonstrate how to publish and consume artifacts.
    - Use Azure Artifacts in a CI/CD pipeline to automate artifact publishing and dependency management.

Module-8: Mastering Docker

Topics Covered:

1. **Docker Deep Dive**
    - Introduction to Docker and its role in DevOps.
    - Understanding Docker components: images, containers, Dockerfiles, and Docker Hub.
    - Creating Dockerfiles and building images with multi-stage builds for optimization.
2. **Working with Docker Containers**
    - Running and managing containers: start, stop, remove, and inspect.
    - Debugging containers and understanding container logs.
    - Advanced container manipulation techniques including exec and attach commands.
3. **Docker Networking and Volumes**
    - Overview of Docker networks: bridge, host, and overlay networks.
    - Managing network configurations and connecting multiple containers.
    - Using Docker volumes for persistent data storage and sharing data between containers.
4. **Docker Compose for Multi-container Deployment**
    - Introduction to Docker Compose: defining and running multi-container Docker applications.
    - Writing Docker Compose files and managing service configurations.
    - Practical use cases of Docker Compose in development and production environments.
5. **Docker Troubleshooting and Assignments**
    - Resolve common Docker container issues: network problems, volume errors, and resource constraints.
    - Optimize a Dockerfile for performance and security.
    - Orchestrate a multi-container deployment using Docker Compose, simulating a real-world application setup.

Module-9: Kubernetes Fundamentals

Topics Covered:

1. **Kubernetes Overview**
    - Introduction to Kubernetes and its significance in DevOps.
    - Understanding Kubernetes architecture: Clusters, Nodes, Pods, and Control Plane.
2. **Setting Up Kubernetes**
    - Options for Kubernetes setups: Minikube, self-hosted, EKS (Elastic Kubernetes Service), and AKS (Azure Kubernetes Service).
    - Installation and initial configuration of a Kubernetes cluster.
3. **Core Kubernetes Components**
    - Deep dive into Pods, Services, Deployments, and ReplicaSets.
    - Exploring ConfigMaps, Secrets, and StatefulSets for managing configuration and storage.
4. **Deploying Applications on Kubernetes**
    - Writing and managing Kubernetes manifest files (YAML).
    - Deploying a sample application with a database and persistent volumes.
    - Using Liveness and Readiness Probes for health checks and efficient management.
5. **Scaling and Resource Management**
    - Techniques for scaling applications horizontally and vertically.
    - Managing resource allocations (CPU and memory) for Pods and Nodes.
6. **Troubleshooting Kubernetes**
    - Common issues in Kubernetes environments and how to resolve them.
    - Debugging Pods and services, analyzing logs, and effective use of kubectl commands.
7. **Kubernetes Assignments**
    - Set up a small Kubernetes cluster and deploy a multi-component application.
    - Implement resource management and scaling for an application.
    - Troubleshoot deployment issues and optimize the application deployment.

Module-10: Infrastructure as Code (IaC) with Terraform and Ansible

Topics Covered:

1. **Introduction to IaC**
    - Overview of Infrastructure as Code and its benefits in automating and managing infrastructure.
    - Comparison of different IaC tools and methodologies.
2. **Terraform Setup and Basics**
    - Installing and configuring Terraform.
    - Writing Terraform scripts to define infrastructure components.
    - Managing state files and understanding Terraform's execution plan and resource graph.
3. **Advanced Terraform Usage**
    - Creating and managing a dynamic infrastructure with Terraform modules.
    - Implementing best practices for writing and organizing Terraform code.
    - Creating an Amazon EKS (Elastic Kubernetes Service) cluster using Terraform.
4. **Ansible Introduction and Setup**
    - Overview of Ansible and its role in configuration management.
    - Setting up Ansible and configuring environments for automation.
5. **Working with Ansible**
    - Writing Ansible playbooks to automate routine server configuration tasks.
    - Using Ansible roles and collections to modularize and reuse code.
    - Integrating Ansible with Jenkins for automated deployments.
6. **IaC Assignments**
    - Provision a complete infrastructure setup using Terraform, including networking, servers, and security configurations.
    - Configure and manage a multi-server environment using Ansible, demonstrating dynamic inventory use.
    - Integrate Terraform and Ansible to provision infrastructure and then configure it, simulating a real-world workflow.

Module-11: Mastering Azure DevOps

Topics Covered:

1. **Azure DevOps Overview**
    - Introduction to Azure DevOps and its components.
    - Benefits of using Azure DevOps for project management, source code control, and CI/CD.
2. **Setting Up Azure Account and Organization**
    - Steps to set up an Azure account and create an organization.
    - Best practices for managing users and permissions in Azure DevOps.
3. **Working with Azure Repos**
    - Overview of Azure Repos for version control.
    - Git integration and repository management techniques.
4. **Implementing CI/CD with Azure Pipelines**
    - Setting up build and release pipelines.
    - Integrating various tools and services, such as Kubernetes, Docker, and SonarQube.
    - Strategies for multi-environment deployments.
5. **Azure Artifacts for Package Management**
    - Using Azure Artifacts to host and manage packages.
    - Configuring feeds for NuGet, npm, and Maven packages.
6. **Azure Kubernetes Service (AKS) and Container Registry**
    - Integrating AKS for container orchestration.
    - Managing container images with Azure Container Registry.
    - Best practices for containerized deployments using AKS.
7. **Azure WebApps**
    - Deploying web applications using Azure WebApps.
    - Auto-scaling and performance monitoring.
8. **Azure DevOps Assignments**
    - Create a full CI/CD pipeline using Azure Pipelines for a full-stack project.
    - Configure Azure Artifacts to automate package versioning and dependencies.

- Deploy a containerized application using AKS and monitor its performance.

Let's conclude our detailed DevOps roadmap with a focus on monitoring, utilizing Prometheus and Grafana to ensure robust and responsive systems:

Module-12: Monitoring with Prometheus and Grafana

Topics Covered:

1. **Introduction to Monitoring**
    - The importance of monitoring in maintaining system reliability and performance.
    - Overview of Prometheus and Grafana as monitoring tools.
2. **Setting Up Prometheus**
    - Installing and configuring Prometheus.
    - Understanding Prometheus metrics, queries, and alert rules.
    - Integrating Prometheus with various applications and infrastructure components.
3. **Exploring Prometheus Exporters**
    - Using Node Exporter for hardware and OS metrics.
    - Setting up BlackBox Exporter for probing HTTP, HTTPS, DNS, TCP, and ICMP endpoints.
4. **Deep Dive into Grafana**
    - Installing and configuring Grafana.
    - Connecting Grafana to Prometheus as a data source.
    - Creating and managing dashboards for visual data representation.
5. **Advanced Monitoring Techniques**
    - Setting up alerts in Grafana based on data from Prometheus.
    - Using Grafana for real-time analytics and historical data analysis.
6. **Monitoring Assignments**
    - Set up a monitoring stack using Prometheus and Grafana for a sample application.

- Create custom alerts and dashboards tailored to specific operational requirements.
- Analyze performance metrics and simulate problem-solving scenarios based on alerts.

Educational Objective: By completing this series, learners will have a solid foundation across a wide range of DevOps practices and tools, ready to tackle real-world challenges with confidence.