

INTRODUCTION TO DEVOPS FOR BEGINNERS

Module 1. Understanding DevOps

- Introduction to DevOps
- Evolution of DevOps
- Key Principles of DevOps

Module 2. Collaboration and Communication

- Collaboration between Development and Operations Teams
- Effective Communication in DevOps
- Building a DevOps Culture

Module 3. Continuous Integration (CI)

- Introduction to CI/CD
- Benefits of Continuous Integration
- Setting up a Basic CI Pipeline

Module 4. Version Control

- Overview of Version Control Systems
- Git Basics and Best Practices
- Collaborative Development with Git

Module 5. Continuous Deployment (CD)

- Introduction to Continuous Deployment
- Automated Testing and Deployment
- Blue-Green Deployments

Module 6. Infrastructure as Code (IaC)

- Understanding IaC
- Introduction to Terraform
- Creating and Managing Infrastructure with Terraform

Module 7. Configuration Management

- Overview of Configuration Management
- Introduction to Ansible
- Writing Ansible Playbooks

Module 8. Monitoring and Logging

- Importance of Monitoring and Logging
- Overview of Monitoring Tools
- Introduction to ELK Stack (Elastic search, Log stash, Kibana)

Module 9. Containerization

- Introduction to Containers
- Docker Basics and Concepts
- Managing Containers with Docker

Module 10. Orchestration with Kubernetes

- Introduction to Kubernetes
- Deploying Applications with Kubernetes
- Managing Kubernetes Clusters

Module 11. Continuous Monitoring and Feedback

- Importance of Continuous Monitoring
- Feedback Loops in DevOps
- Real-time Analytics and Dashboards

Module 12. Security in DevOps

- DevSecOps: Integrating Security in DevOps
- Best Practices for Secure DevOps
- Compliance and Auditing

Module 13. DevOps Tools Landscape

- Overview of Popular DevOps
- Tools Tool chain Integration
- Selecting the Right Tools for Your Environment

Module 14. DevOps Culture and Best Practices

- Fostering a DevOps Culture
- Best Practices for Successful DevOps Implementation
- Case Studies and Success Stories

Module 15: Let's Perform Projects, 10 Projects for Beginner

Project 1: Version Control Project

Objective: Set up a Git repository for a simple web application.

Tasks:

1. Create a repository on GitHub or GitLab.
2. Clone the repository to your local machine.
3. Add, commit, and push changes to the repository.
4. Collaborate with a team member using branches and pull requests.

Project 2: Continuous Integration Project

Objective: Implement a basic CI pipeline for a simple application.

Tasks:

1. Set up a CI/CD pipeline using a CI tool (e.g., Jenkins, GitLab CI).
2. Configure the pipeline to trigger on code commits.
3. Include stages for build and basic unit testing.

Project 3: Infrastructure as Code (IaC) Project

Objective: Use Terraform to provision cloud resources

Tasks:

1. Write Terraform code to create a simple infrastructure (e.g., a virtual machine).
2. Apply the Terraform configuration to provision the infrastructure.
3. Destroy the infrastructure using Terraform

Project 4: Configuration Management Project

Objective: Configure a server using Ansible.

Tasks:

1. Write an Ansible playbook to install and configure software on a server.
2. Run the playbook to apply the configuration.
3. Verify the server's state matches the desired configuration.

Project 5: Configuration Management Project

Objective: Configure a server using Ansible.

Tasks:

1. Write an Ansible playbook to install and configure software on a server.
2. Run the playbook to apply the configuration.
3. Verify the server's state matches the desired configuration.

Project 6: Docker Project

Objective: Containerize a simple web application using Docker

Tasks:

1. Write a Dockerfile for the web application.
2. Build a Docker image.
3. Run a container from the Docker image.

Project 7: Kubernetes Project

Objective: Deploy a multi-container application on Kubernetes

Tasks:

1. Set up a Kubernetes cluster (locally or using a cloud provider)
2. Define Kubernetes manifests for the application.
3. Deploy and scale the application on Kubernetes.

Project 8: Monitoring and Logging Project

Objective: Set up monitoring and logging for a web application

Tasks:

1. Configure a monitoring tool (e.g., Prometheus) to collect metrics.
2. Set up logging using the ELK Stack (Elastic search, Log stash, Kibana).
3. Create dashboards for monitoring and exploring logs.

Project 9: Security Automation Project

Objective: Implement security checks in the CI/CD pipeline

Tasks:

1. Integrate security scanning tools (e.g., SonarQube) into the CI pipeline.
2. Automate security checks for vulnerabilities and code quality.

Project 10: Container Orchestration Scaling Project

Objective: Implement auto-scaling for a containerized application

Tasks:

1. Deploy a containerized application on Kubernetes.
2. Configure Horizontal Pod Auto scaling (HPA) for the application.

Full DevOps Pipeline Project

Objective: Build an end-to-end DevOps pipeline for a sample project

Tasks:

1. Version control with Git.
2. CI/CD pipeline with Jenkins or GitLab CI.
3. IaC with Terraform.
4. Configuration management with Ansible.
5. Docker containerization.
6. Deployment and scaling on Kubernetes.
7. Monitoring and logging setup.
8. Security checks in the pipeline.