Prerequisite Topics

These are the foundational topics that should be understood before starting DevOps projects:

1. **Linux Basics**
    1. File system and permissions
    2. Basic shell scripting
    3. Package management
2. **Version Control**

    1. Git basics (clone, commit, push, pull, branches)
    2. Git workflows (feature branching, pull requests, merging)
3. **CI/CD Concepts**

    1. Continuous Integration vs Continuous Deployment
    2. Overview of Jenkins, GitLab CI/CD, or GitHub Actions
4. **Containerization**

    1. Docker fundamentals
    2. Creating, managing, and running Docker containers
    3. Docker Compose basics
5. **Container Orchestration**

    1. Kubernetes basics
    2. Pods, Deployments, Services, ConfigMaps, and Secrets
6. **Infrastructure as Code (IaC)**

    1. Terraform or Ansible introduction
    2. Writing basic IaC scripts
7. **Monitoring and Logging**

    1. Introduction to Prometheus, Grafana, ELK/EFK stack
    2. Setting up alerts and dashboards
8. **Cloud Services**

    1. Overview of AWS, Azure, or GCP
    2. Setting up basic cloud infrastructure (EC2, S3, IAM)

Mini Project: DevOps Pipeline for a Basic Application

Objective:

Build and deploy a small application using a CI/CD pipeline.

Project Flow:

Plan:

1. Define the pipeline stages: Build, Test, Deploy.
2. Use GitHub or GitLab as the version control repository.

Develop:

1. A simple Node.js or Python application (e.g., a REST API or a calculator app).

Implementation:

1. Create a CI/CD pipeline using **Jenkins** or **GitLab CI/CD**.
    1. **Build Stage:** Build the application using Docker.
    2. **Test Stage:** Run unit tests.
    3. **Deploy Stage:** Deploy the container to a Kubernetes cluster.
2. Use **Docker Compose** to run the application locally.
3. Use **Kubernetes** for deployment in a cloud cluster (e.g., Minikube, AWS EKS, or GCP GKE).

Outcome:

1. Automated pipeline triggers on every Git commit.
2. Application is deployed and accessible via a public endpoint.

Big Project: End-to-End DevOps Workflow for E-Commerce Application

Objective:

Build a robust DevOps pipeline for a multi-service e-commerce application.

Project Flow:

Plan:

1. Break the application into microservices:
    1. User Management
    2. Product Catalog
    3. Order Management
2. Define environments: Dev, Staging, Production.

Develop:

1. Use a **microservices architecture** (Golang, Python, or Java).
2. Each service has its own repository and CI/CD pipeline.

Implementation:

1. **Infrastructure Setup:**
    1. Use **Terraform** to provision cloud infrastructure on AWS.
    2. Setup services like EC2, RDS, S3, and IAM.
2. **CI/CD Pipeline:**
    1. Use **Jenkins** for a multi-branch pipeline.
    2. Use **Docker** to containerize all services.
    3. Deploy to a **Kubernetes cluster** on AWS EKS.
3. **Monitoring & Logging:**
    1. Configure **Prometheus** for metrics and **Grafana** for dashboards.
    2. Use the **ELK stack** for centralized logging.
4. **Security:**
    1. Scan containers with tools like **Trivy**.
    2. Use **Vault** or **AWS Secrets Manager** for managing sensitive data.
5. **Load Testing:**
    1. Simulate real-world traffic using **JMeter** or **Locust**.

Outcome:

1. Fully automated CI/CD workflow from code commit to deployment.
2. Scalable microservices architecture hosted on the cloud.
3. Complete monitoring and logging setup for troubleshooting.

Detailed Syllabus for DevOps Project Course

Week 1-2: Prerequisite Setup

- Linux commands and shell scripting.
- Git basics and workflows.
- Introduction to Docker and Kubernetes.
- Basics of CI/CD concepts.

Week 3-4: Mini Project

- Building and deploying a simple application using Docker.
- Configuring Jenkins pipelines for CI/CD.
- Deploying the app to Kubernetes using Minikube.

Week 5-8: Big Project

- **Week 5: Microservices Design**
  - Designing and containerizing microservices.
- **Week 6: Kubernetes Deployment**
  - Setting up Kubernetes clusters on AWS EKS or GCP GKE.
  - Deploying microservices and configuring services.
- **Week 7: Monitoring & Security**
  - Configuring Prometheus, Grafana, and ELK.
  - Scanning containers for vulnerabilities.
- **Week 8: Final Workflow**
  - End-to-end CI/CD pipeline setup.
  - Load testing and performance optimization.

Deliverables

1. Mini project files (code, YAML configurations, CI/CD pipeline scripts).
2. Big project repository with:
   - Codebase for all microservices.
   - Terraform/Ansible scripts for infrastructure setup.
   - Kubernetes manifests and Helm charts.
   - CI/CD pipeline scripts.

    - Monitoring and logging dashboards.

3. Detailed project documentation and deployment guide.
