🚀 Guide2Code - DevOps & Automation Roadmap 🔧

🔰 Beginner Level - Getting Started with DevOps

Required Programming Languages:

- **Bash** 💻
- **Python** 🐍
- **YAML** 📝

Required Skills:

- **Linux & Command Line Basics** 💻
- **Version Control (Git & GitHub)** 🔄
- **Basic Scripting** (Bash, Python) 📝
- **Continuous Integration (CI) Basics** ⚙️
- **Cloud Basics** ☁️

Learn the Fundamentals:

- **Introduction to DevOps**: Learn the core principles and culture of DevOps, and how it integrates development and operations.
- **Version Control with Git**: Understand Git basics, including commit, branches, merges, and collaboration on GitHub.
- **Scripting & Automation**: Learn basic scripting to automate repetitive tasks, such as file management and process automation.
- **CI/CD Basics**: Learn how to set up continuous integration (CI) pipelines using GitHub Actions or Jenkins.
- **Cloud Platforms**: Understand the basics of cloud computing (e.g., AWS, Azure) and how DevOps leverages cloud infrastructure.

Beginner Projects 🚀:

1. **GitHub Portfolio**: Create a personal GitHub repository to showcase your work.
2. **Simple Backup Script**: Automate file backups using Bash or Python scripts.
3. **CI Pipeline Setup**: Set up a basic CI pipeline using GitHub Actions to run tests and deploy applications.

4. **Dockerizing a Simple App**: Create a Docker container for a simple web application to understand containerization.
5. **Automated Log Monitoring**: Write a script to monitor log files for specific errors or patterns.

🔄 Intermediate Level - Expanding DevOps & Automation Knowledge

Required Programming Languages:

- **Python (Advanced)** 🐍
- **Bash/Shell Scripting** 💻
- **Terraform** 🏗️

Required Skills:

- **Infrastructure as Code (IaC)** 🏗️
- **Containerization & Orchestration** 🐳
- **Advanced CI/CD Pipelines** 🔄
- **Automation with Ansible/Terraform** 🔧
- **Monitoring & Logging** 📊

Expanding Your Knowledge:

- **Infrastructure as Code (IaC)**: Learn Terraform and Ansible to automate infrastructure provisioning and configuration management.
- **Containerization**: Understand how to package applications into containers using Docker.
- **Container Orchestration**: Learn Kubernetes for managing and scaling containerized applications.
- **Advanced CI/CD**: Implement multi-stage CI/CD pipelines for automating testing, build, and deployment of applications.
- **Monitoring & Logging**: Explore tools like Prometheus, Grafana, or the ELK stack for system monitoring and log analysis.

Intermediate Projects 🔄:

1. **Multi-Stage CI Pipeline**: Create a multi-stage CI pipeline that includes build, test, and deploy stages using Jenkins or GitHub Actions.
2. **Automated Infrastructure Setup**: Use Terraform to set up cloud infrastructure for a basic web application.
3. **Containerize a Full-Stack App**: Build a full-stack application with a frontend, backend, and database, and run it in Docker containers.
4. **Kubernetes Cluster**: Set up a Kubernetes cluster and deploy a containerized application to it.
5. **Monitoring and Alerting**: Set up Prometheus and Grafana to monitor and visualize system metrics.

🔝 Advanced Level - Mastering DevOps & Automation

Required Programming Languages:

- **Python (Advanced)** 🐍
- **Go** 💻
- **Ruby** 💎 (Optional)

Required Skills:

- **Advanced CI/CD Pipelines** 🔄
- **Cloud-Native DevOps (AWS, Azure, GCP)** ☁️
- **Site Reliability Engineering (SRE)** 🔧
- **Automation with ArgoCD, FluxCD** 🔄
- **Security in DevOps (DevSecOps)** 🔐

Deep Dive Into Advanced Topics:

- **Advanced CI/CD with Kubernetes**: Implement complex, scalable CI/CD pipelines using Kubernetes and containerized microservices.
- **Site Reliability Engineering (SRE)**: Learn about building and managing reliable, scalable systems with a focus on availability, latency, and change management.
- **Cloud-Native DevOps**: Dive into using AWS, Azure, or Google Cloud for infrastructure, scaling, and resource management.

- **GitOps with ArgoCD and FluxCD**: Use GitOps tools to automate the management of Kubernetes clusters and infrastructure.
- **DevSecOps**: Learn how to integrate security practices into the DevOps pipeline to build secure applications.

Advanced Projects :

1. **Microservices Architecture on Kubernetes**: Build and deploy a microservices architecture with Kubernetes, Docker, and Helm.
2. **CI/CD for Microservices**: Implement a CI/CD pipeline specifically designed for deploying microservices to Kubernetes clusters.
3. **Self-Healing System**: Create a system that automatically recovers from failures using Kubernetes and automated remediation tools.
4. **Automated Cloud Infrastructure with Terraform**: Use Terraform to provision and manage multi-cloud infrastructure for a production-ready system.
5. **DevSecOps Pipeline**: Build an end-to-end CI/CD pipeline with integrated security checks (e.g., static code analysis, vulnerability scanning) using tools like SonarQube, Snyk, or Aqua Security.

Thank You for Visiting Guide2Code!

"Automate everything, optimize relentlessly, and scale with confidence!"