

DVS Technologies

AI DevOps Using Generative and Agentic AI

Module 1: Foundations of DevOps and Gen AI

- **Introduction to DevOps & AI Integration**
- **AI Basics** for DevOps Engineers
- Tools Overview: **ChatGPT, LangChain, Ollama, Terraform, AWS**
- **GitOps, IaC, CI/CD, Monitoring** – Where AI Fits
- Intro to **Prompt Engineering** for **Infrastructure**

Module 2: AI + Infrastructure as Code

- Generating Terraform code with **LLMs**
- Code Validation: Linting, Testing using **AI Agents**
- Auto-generating Variables & Modules
- Real-World: 2-tier architecture with **AI-generated** Terraform
- On-demand infra generation from high-level description
- Custom **LangChain** agent to convert JSON→Terraform
- Convert existing infra to Terraform using **AI + Recon Tools**
- **AI** for multi-cloud Terraform code generation (**AWS, Azure, GCP**)
- CI/CD pipelines to run Terraform dynamically
- Review: Build complete 2-tier infra from **voice/text input**

Module 3: AI in CI/CD and SRE

- Using **AI to generate CI/CD pipeline** templates (GitHub/GitLab/Azure)
- **AI to fix failed CI/CD builds automatically**
- Monitoring Overview: Prometheus, Grafana, CloudWatch
- Monitoring Overview: Prometheus, Grafana, CloudWatch
- Generate runbooks using AI
- **Self-healing infra via AI** recommendations
- GenAI for reliability scoring & chaos analysis
- **SRE ChatOps**: AI responding to real-time failures
- Auto-documenting infra & pipelines via AI
- Review: **Smart CI/CD + Monitoring with AI tools**

Module 4: Agentic AI Workflows & Automation

- **LangChain Agents** vs OpenAI Functions
- **Build AI agent** to plan and apply Terraform
- Agent to handle **ticket** → **root cause** → **resolution**
- Multi-step agents for app deployment
- Build LLM assistant to generate monitoring dashboards
- Orchestrating workflows with tools like **CrewAI** or **LangGraph**
- Human-in-the-loop (**HITL**) automation patterns
- LLM + GitOps agent for **PRs, merges, test validation**
- Chaining AI agents for full environment provisioning
- End-to-end project: AI agent provisioning + **CI/CD** + **monitor**

Module 5: Advanced Integrations & Real-Time Projects

- **Create voice-based** infra-assistant using Whisper + LLM
- Vision-to-Infrastructure (**diagram** → **terraform**)
- **AI for Cloud Cost Optimization & Budgeting**
- LLMs + Security Automation
- Infra Drift Detection & Resolution using AI
- Build: **Self-service portal with AI infra** requests
- Evaluation: Students present automation bots
- Open Q&A: Debugging, Custom Use Cases
- Final Project Presentation & Certification

Module 6: Working with Infra as a Code (IaC)

- **Create voice-based infra-assistant** using Whisper + LLM
- Vision-to-Infrastructure (diagram → terraform)
- **AI for Cloud Cost Optimization & Budgeting**
- **Terraform** – Your infrastructure sculptor ☐ – learn how to build cloud resources declaratively, manage them consistently, and scale like a pro across AWS, Azure, and GCP.
- **Ansible** – Your automation wizard ☐ – master how to configure, deploy, and manage servers effortlessly with agentless automation.
- Understand state management, providers, and modules in Terraform.
- Write idempotent playbooks in Ansible to ensure systems always stay exactly how you want them.
- **Learn how Terraform provisions infrastructure and Ansible breathes** life into it with configurations.
- Hands-on labs to spin up real cloud environments and automate application deployments.
- Enterprise tips for scalable, repeatable, and secure infrastructure setups.
- Troubleshooting secrets for both tools to handle real-world DevOps challenges.
- See them in action together – **infrastructure + configuration = DevOps superpower!**

Module 7: Getting Started with Pipeline as Code in Jenkins

- Learn what **Pipeline as Code** means and why it's better than manual builds.
- Understand the role of **Jenkins** in automating software delivery.
- Create a **Jenkinsfile** to define your build steps in code.
- Explore the difference between **Declarative** and **Scripted** pipelines.
- Add simple **stages** like Build, Test, and Deploy.
- Connect Jenkins with **Git** so pipelines run automatically on code changes.
- See how to use **Groovy syntax** for pipeline steps.
- Learn how to **store and manage credentials** securely.
- Trigger pipelines manually or automatically.
- Get a basic hands-on demo to run a **complete pipeline from commit to deployment**.

Module 8: Getting Started with Docker

- Understand **what Docker is** and why it's changing how we run applications.
- Learn the difference between **containers** and **virtual machines** (and why containers are faster).
- Install Docker and get familiar with **Docker CLI** commands.
- Pull and run pre-built **Docker images** from Docker Hub.
- Build your own **Docker images** using a **Dockerfile**.
- Learn about **layers** in images and why they make builds faster.
- Learn how to **tag, push, and pull** images from a registry.
- Understand container **lifecycle**: create → start → stop → remove.
- Practice deploying a **sample app** in Docker.

Module 9: Mastering Git Basics for SCM

- Understand **what SCM (Source Control Management)** is and why it's essential in modern software development.
- Learn **what Git is** and why it's the most popular version control tool.
- Install Git and configure your **username & email** for tracking changes.
- Create your first **Git repository** (local & remote).
- Understand **commits** and **commit messages**.
- Learn how to **clone, pull, and push** code to GitHub/GitLab/Bitbucket.
- Work with **branches** for parallel development.
- Merge code and understand **merge conflicts** (and how to resolve them).
- Use **Git log** and **Git diff** to track changes.
- Learn **Git status** and **Git add** to keep your work organized.
- Explore **tags** for marking releases.
- Practice a **team workflow** with pull requests (PRs).

Module 10: Mastering kubernetes

- Introduction to Kubernetes – What is Kubernetes and why do we need it in modern application development?
- Kubernetes Architecture – **Nodes, Pods, Deployments, Services**, Control Plane, and Worker Nodes.
- **Setting up Kubernetes – Installing Kubernetes** using terraform on aws (step-by-step).
- Cluster Initialization – Bootstrapping the control plane and joining worker nodes.
- Installing a Pod Network (CNI Plugin) – Calico/Flannel setup for pod communication.
- Verifying the Cluster – **Checking node status, pods**, and cluster health.
- First Application Deployment – Deploying a sample NGINX application using a Deployment manifest.
- Scaling Applications – Horizontal scaling of pods with kubectl scale.
- Exposing Applications – **Using NodePort/LoadBalancer/Ingress** to make the app accessible.
- Hands-On Demo – **Deploying and accessing the NGINX** web page from Kubernetes cluster.

Real Time Concepts:

- Day to Day Admin Activities
- Frequently Occurring Issues
- Roles and Responsibilities
- Resume Preparation Guidance