Admissions & Support:
support@mockattempt.com
Phone / WhatsApp: +91 98765 43210
Course Code: DEVO • **Updated:** 01 Nov 2026
https://mockattempt.com/courses/devops-training

DEVO • DEVOPS CI/CD, GIT, TERRAFORM IAC & SRE **BEGINNER TO PRINCIPAL DEVOPS & PLATFORM ENGINEER**

100% LIVE INDUSTRY EXPERT COHORT + INTERNSHIP LETTER

DevOps & Git CI/CD Training

Master DevOps & Platform Engineering from DORA Metrics, SRE Error Budgets, Git Content-Addressable DAGs & Strict Bash Automation to Jenkins Declarative Pipelines, GitHub Actions OIDC, Idempotent Ansible, Terraform Remote State/HCL & AWS/Azure DevOps — with 200 Mock Tests & Voice Gemini AI Interviews at ₹5,999.

Duration: 16–18 Weeks • Live + Hands-On Projects (210 Hrs) • **Modules:** 16 • **Live Topics:** 42 • **Labs & Tasks:** 84 • **Mocks / AI Vivas:** 244 / 244 • **Projects:** 16 Module + 1 Live Client Prod

Fixed Non-Negotiable Tuition Fee: Base Fee **₹5,999** + 18% GST (₹1,080) = **Total Incl. GST: ₹7,079**

Live Batch Tracks: Weekday Live Online Batch (Mon – Fri) (Monday to Friday (5 Live Interactive Classes / Week) • 1.5 Hours / Day (60 Mins Live Architecture & Coding + 30 Mins Hands-On Lab) • 18–20 Weeks Fast-Track) | Weekend Intensive Live Online Batch (Sat – Sun) (Saturday & Sunday (Designed for Working Professionals & College Students) • 3.0 Hours / Day (2 Hrs Deep-Dive Live Coding + 1 Hr Guided Project Build) • 20–22 Weeks Weekend Track)

Core Technologies & Engineering Stack: Git Internals, Trunk-Based & GitFlow • Strict Bash Scripting & Linux Automation • GitHub Actions OIDC & Jenkins Pipelines • Ansible Playbooks & Molecule Roles • Terraform 1.9 HCL & S3 Remote State • AWS DevOps (OIDC, CodePipeline, ECS/EKS) • Azure DevOps YAML & SRE DORA Metrics

1. PROGRAM MASTERY OUTCOMES & ELIGIBILITY

End-to-End Engineering Outcomes:

1. Measure and optimize elite DORA engineering velocity (Deployment Frequency, Lead Time, Change Failure Rate, MTTR) alongside SRE SLO Error Budgets
2. Navigate Git DAG internals (`blob`, `tree`, `commit`, `reflog`, `bisect`, `rebase`, `stash`) and author defensive `set -euo pipefail` Bash automation
3. Build secretless OIDC CI/CD pipelines across Jenkins, GitHub Actions, AWS, and Azure DevOps, plus idempotent Ansible roles and state-locked Terraform HCL modules
4. Clear 200 Timed Mock Exams (10,000 DevOps, Git & Terraform MCQs) and 200 Spoken Voice + Google Gemini AI Platform Engineering Interviews
5. Linux server administration, Bash automation, systemd services, SSH hardening, DNS/TLS networking, and Nginx reverse proxies.
6. Automated CI/CD pipelines (GitHub Actions) with build caching, automated test gates, SAST/container scans, and environments.
7. Declarative Infrastructure as Code with Terraform modules, remote S3/DynamoDB state locking, and drift detection.
8. Full-stack observability with Prometheus metrics, Grafana dashboards, centralized logs, and incident runbooks.

Eligibility & Prerequisites: Basic familiarity with Linux terminal navigation and editing text files — Git, Bash, CI/CD, and Terraform start from scratch • Any Windows (WSL 2 / Git Bash), macOS, or Linux laptop/PC with Git, Bash, OpenSSH, and Terraform CLI installed • No prior DevOps & Git CI/CD industry experience required — Module 1 starts from environment setup, core mental models, and plain-English fundamentals. • Basic computer & internet familiarity on a Windows, macOS, or Linux laptop (8 GB+ RAM recommended for smooth live coding labs). • Ideal for B.Tech / BCA / MCA / B.Sc IT students, non-CS beginners, career switchers, and working developers upgrading to 2026 production standards. • Commitment to attend live Weekday or Weekend classes, complete daily hands-on lab checklists, and practice topic-wise Mock Tests & AI Interviews.

2. DETAILED MODULE & TOPIC-WISE SYLLABUS (16 MODULES • 42 LIVE TOPICS)

MODULE 1 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 1: DevOps Culture (`CALMS`), DORA Elite Metrics & Immutable Cloud-Native Architecture

Understand how DevOps bridges development agility and operational stability using `CALMS`, the 4 DORA velocity/stability metrics, and immutable pets-vs-cattle delivery architectures.

Objectives: ✓ Break down Dev vs Ops wall-of-confusion silos using the `CALMS` framework (Culture, Automation, Lean, Measurement, Sharing) • ✓ Compute the 4 DORA metrics (Deployment Frequency, Lead Time for Changes, Change Failure Rate, Failed Deployment Recovery Time) • ✓ Architect immutable artifact promotion pipelines (`Build Once, Deploy Anywhere`) separating environment config from signed artifacts

1.1 DevOps Foundations — `CALMS` Principles, Wall of Confusion & The 4 DORA Engineering Metrics [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply DevOps Foundations — `CALMS` Principles, Wall of Confusion & The 4 DORA Engineering Metrics in real DevOps & Git CI/CD systems — specifically: Why...

1.2 DevOps Architecture — Immutable Infrastructure, Pets vs Cattle & Build-Once-Deploy-Anywhere [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply DevOps Architecture — Immutable Infrastructure, Pets vs Cattle & Build-Once-Deploy-Anywhere in real DevOps & Git CI/CD systems — specifically: End...

Module 1 Hands-On Capstone Project (100 Marks): DORA Engineering Telemetry & SRE Error-Budget Gate Evaluator — Build an automated Bash/jq release-readiness evaluator that calculates monthly SRE 99.9% SLO error budget burn and DORA Change Failure Rate before approving production auto-promotion.

MODULE 2 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 2: The Continuous DevOps Lifecycle (`7 Cs`) & Shift-Left DevSecOps Principles

Connect all phases of the infinite DevOps loop (`Plan -> Code -> Build -> Test -> Release -> Deploy -> Operate -> Monitor`) and apply Gene Kim's Three Ways and Shift-Left DevSecOps.

Objectives: ✓ Distinguish Continuous Integration (CI), Continuous Delivery (manual business approval gate), and Continuous Deployment (100% automated to prod) • ✓ Apply The Three Ways (Flow left-to-right, Fast Feedback right-to-left, Continual Chaos/Learning experimentation) • ✓ Shift security left by embedding secret scanning (`gitleaks`), SAST, SCA dependency checks, and policy-as-code directly in PR checks

2.1 The Infinite DevOps Lifecycle — Continuous Integration vs Continuous Delivery vs Continuous Deployment [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply The Infinite DevOps Lifecycle — Continuous Integration vs Continuous Delivery vs Continuous Deployment in real DevOps & Git CI/CD systems — specif...

2.2 DevOps Workflow & Principles — The Three Ways, Value-Stream Mapping, WIP Limits & Shift-Left Security [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply DevOps Workflow & Principles — The Three Ways, Value-Stream Mapping, WIP Limits & Shift-Left Security in real DevOps & Git CI/CD systems — specifi...

Module 2 Hands-On Capstone Project (100 Marks): Shift-Left Pre-Merge DevSecOps Gate (Secret Scan + Lockfile Audit + Smoke Check) — Create a fast `<30s` pre-merge CI gate script that blocks leaked AWS/API credentials, verifies deterministic lockfiles, and enforces Continuous Delivery vs Continuous Deployment promotion ru...

MODULE 3 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 3: CI/CD Pipeline Architecture, Testing Pyramids & The Cloud-Native DevOps Toolchain

Design fast fail-early CI/CD DAG pipelines following the Test Pyramid, and map the CNCF/DevOps landscape cleanly across SCM, CI, IaC, Configuration Management, Containers, and Observability.

Objectives: ✓ Order pipeline stages by speed & signal (`Lint/Secret Scan [15s] -> Unit Tests [45s] -> Build/Scan Image [90s] -> Ephemeral Integration -> Canary CD`) • ✓ Avoid the fragile "Ice-Cream Cone" E2E anti-pattern by enforcing the Testing Pyramid (`70% Unit`, `20% Integration/Testcontainers`, `10% E2E`) • ✓ Select the right tool for each job: Terraform (provisioning cloud infra) vs Ansible (OS configuration) vs Kubernetes/ArgoCD (workload orchestration)

3.1 CI/CD Pipeline Methodology — Fail-Fast Stage Ordering, Fan-Out/Fan-In DAGs & The Test Pyramid [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply CI/CD Pipeline Methodology — Fail-Fast Stage Ordering, Fan-Out/Fan-In DAGs & The Test Pyramid in real DevOps & Git CI/CD systems — specifically: S...

3.2 The DevOps Toolchain Matrix — Terraform vs Ansible vs Docker vs Kubernetes vs Prometheus [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply The DevOps Toolchain Matrix — Terraform vs Ansible vs Docker vs Kubernetes vs Prometheus in real DevOps & Git CI/CD systems — specifically: Unders...

Module 3 Hands-On Capstone Project (100 Marks): Fail-Fast Multi-Stage GitHub Actions OIDC CI/CD Pipeline Blueprint — Author a production GitHub Actions CI/CD workflow YAML using parallel fan-out lint/test jobs, dependency caching, and short-lived OIDC cloud authentication (zero long-lived AWS keys).

MODULE 4 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 4: Git Content-Addressable Internals (`.git/objects` DAG) & Branching Workflows (`GitFlow` vs Trunk-Based)

Understand Git from first principles as an immutable Merkle-tree DAG (`blob`, `tree`, `commit`, `tag` + lightweight pointer branches), and compare GitFlow against modern Trunk-Based Development.

Objectives: ✓ Inspect `.git/objects` SHA-1/SHA-256 content-addressable storage (`blob` file contents, `tree` directories, `commit` snapshots + parent pointers) • ✓ Trace `Working Tree -> Index (Staging Area) -> Local Repository -> Remote` and understand `HEAD` & fast-forward vs 3-way merges • ✓ Choose accurately between **Trunk-Based Development** (high-velocity SaaS microservices) and **GitFlow** (multi-versioned mobile/on-prem enterprise software)

4.1 Git Internals — Snapshots (Not Diffs!), `.git/objects` (`blob`, `tree`, `commit`) & Branch Pointers [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Git Internals — Snapshots (Not Diffs!), `.git/objects` (`blob`, `tree`, `commit`) & Branch Pointers in real DevOps & Git CI/CD systems — specifica...

4.2 Git Branching Strategies — `GitFlow` (`main`/`develop`/`release`/`hotfix`) vs Trunk-Based Development [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Git Branching Strategies — `GitFlow` (`main`/`develop`/`release`/`hotfix`) vs Trunk-Based Development in real DevOps & Git CI/CD systems — specifi...

Module 4 Hands-On Capstone Project (100 Marks): Automated Branch Protection, Conventional Commit Hook & Release Tagging Workflow — Implement a local/CI Git validation suite enforcing Conventional Commits (`feat:`, `fix:`, `chore:`), preventing direct force-pushes to `main`, and generating annotated SemVer release tags.

MODULE 5 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 5: Surgical Git Operations (`rebase`, `reflog`, `bisect`, `revert`) & Safe `git stash` Workflows

Recover from any "lost" commit using `git reflog`, hunt regressions in `O(log N)` steps with `git bisect`, understand `revert` vs `reset` & `--force-with-lease`, and master `git stash -u` / `git worktree`.

Objectives: ✓ Choose accurately between `git revert` (safe new inverse commit on shared branches) and `git reset` (local pointer rewrite) • ✓ Recover accidentally hard-reset commits or deleted branches via `git reflog` and automate regression hunting via `git bisect run` • ✓ Stash tracked AND untracked files safely (`git stash push -u -m "..."`) without losing newly created files

5.1 Production Git Commands — `revert` vs `reset`, `rebase`, `--force-with-lease`, `reflog` & `bisect` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Production Git Commands — `revert` vs `reset`, `rebase`, `--force-with-lease`, `reflog` & `bisect` in real DevOps & Git CI/CD systems — specifical...

5.2 Mastering `git stash` & `git worktree` — Untracked `-u` Trap, `apply` vs `pop` & `stash branch` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Mastering `git stash` & `git worktree` — Untracked `-u` Trap, `apply` vs `pop` & `stash branch` in real DevOps & Git CI/CD systems — specifically:...

Module 5 Hands-On Capstone Project (100 Marks): Automated `git bisect run` Regression Finder & Safe Hotfix Stash/Worktree Lab — Write an automated non-interactive Git regression hunter (`git bisect run ./test.sh`) that pinpoints the exact offending commit across 64 commits in 6 steps, plus safe `--force-with-lease` &...

MODULE 6 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 6: Defensive Production Bash Scripting (`set -euo pipefail`) & Declarative Jenkins CI/CD

Eliminate silent shell automation bugs using strict-mode Bash (`set -euo pipefail`, `trap`, quoted expansions, `shellcheck`), and engineer scalable Declarative `Jenkinsfile` pipelines with ephemeral K8s/Docker agents.

Objectives: ✓ Enforce `set -euo pipefail`, `trap cleanup EXIT`, `[[ ... ]]` conditionals, and `"$var"` double-quoting in every DevOps script • ✓ Understand why bare `set -e` alone STILL ignores failing commands on the left side of a pipe (`curl_fail | jq`) without `set -o pipefail` • ✓ Author Declarative `Jenkinsfile` pipelines (`agent`, `environment`, `credentials()`, `stages`, `parallel`, `post { always { cleanWs() } }`)

6.1 Defensive Bash for DevOps — `set -euo pipefail`, `trap EXIT`, Quoting Rules & Retry Backoff [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Defensive Bash for DevOps — `set -euo pipefail`, `trap EXIT`, Quoting Rules & Retry Backoff in real DevOps & Git CI/CD systems — specifically: Why...

6.2 Jenkins Pipeline-as-Code — Declarative `Jenkinsfile`, Ephemeral Agents & Groovy String Interpolation Security [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Jenkins Pipeline-as-Code — Declarative `Jenkinsfile`, Ephemeral Agents & Groovy String Interpolation Security in real DevOps & Git CI/CD systems —...

Module 6 Hands-On Capstone Project (100 Marks): Idempotent Zero-Leak Bash Deployment Script + Declarative Ephemeral `Jenkinsfile` — Build a ShellCheck-clean deployment health script with `trap` temp-file cleanup and retry backoff, invoked from a Declarative `Jenkinsfile` running inside ephemeral container agents with mas...

MODULE 7 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 7: Idempotent Configuration Management (`Ansible`) & `Terraform` Remote State Internals

Automate fleet OS hardening with agentless idempotent Ansible Roles & Handlers, and master Terraform's core engine: `.tfstate` state locking, S3/Cloud backends, and zero-plaintext-secret state hygiene.

Objectives: ✓ Write idempotent FQCN (`ansible.builtin.*`) Ansible Playbooks where running a playbook 10 times produces `changed=0` on runs 2–10 • ✓ Configure Terraform Remote Backends (`backend "s3"` with encryption + state locking) so teams never overwrite concurrent infrastructure changes • ✓ Understand why `terraform.tfstate` ALWAYS stores sensitive resource attributes in plaintext JSON (and why `.tfstate` must NEVER touch Git!)

7.1 Ansible Automation — Agentless SSH, Idempotent FQCN Modules (`changed=0`), Handlers & Ansible Vault [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Ansible Automation — Agentless SSH, Idempotent FQCN Modules (`changed=0`), Handlers & Ansible Vault in real DevOps & Git CI/CD systems — specifica...

7.2 Terraform State (`terraform.tfstate`) — Desired vs Real-World Graph, Remote Locking & Plaintext Secret Risk [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Terraform State (`terraform.tfstate`) — Desired vs Real-World Graph, Remote Locking & Plaintext Secret Risk in real DevOps & Git CI/CD systems — s...

Module 7 Hands-On Capstone Project (100 Marks): Idempotent Ansible SSH Hardening Playbook + Encrypted Locked S3 Terraform Backend — Author an idempotent FQCN Ansible Playbook validating config syntax before restarting `sshd` via `handlers`, plus a locked & KMS-encrypted Terraform S3 backend configuration.

MODULE 8 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 8: Dynamic Terraform HCL (`for_each`, `for` Expressions, `dynamic`) & Safe `output` Contracts

Eliminate brittle index-shifting `count` bugs using keyed `for_each` + HCL `for` comprehensions, and expose clean cross-module contracts using typed, precondition-checked, `sensitive` Terraform `output`s.

Objectives: ✓ Choose `for_each` (keyed map/set identity: `aws_subnet.app["az-a"]`) over `count` (fragile positional index `[0], [1], [2]`) • ✓ Transform lists and maps cleanly in `locals` using HCL `for` expressions (`{ for s in var.services : s.name => s if s.enabled }`) • ✓ Export module `output` values safely (`sensitive = true`, `precondition`) and query them via `terraform output -raw` / `-json`

8.1 Terraform Loops — `for` Expressions vs `for_each` Meta-Argument vs The `count` Index-Shift Trap [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Terraform Loops — `for` Expressions vs `for_each` Meta-Argument vs The `count` Index-Shift Trap in real DevOps & Git CI/CD systems — specifically:...

8.2 Terraform Outputs — Cross-Module Interfaces, `sensitive = true`, `ephemeral` & `terraform output -raw` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Terraform Outputs — Cross-Module Interfaces, `sensitive = true`, `ephemeral` & `terraform output -raw` in real DevOps & Git CI/CD systems — specif...

Module 8 Hands-On Capstone Project (100 Marks): Zero-Destruction Multi-AZ Subnet Module (`for_each` + Filtered `for` Map + `output`s) — Refactor a dangerous `count`-indexed subnet configuration into a stable `for_each` map keyed by Availability Zone so removing middle items never destroys and recreates surviving subnets!

MODULE 9 OF 16 • 11 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 9: Terraform Code Quality Gates (`fmt`, `validate`, `tflint`) & AWS Cloud DevOps

Enforce deterministic IaC pull-request quality gates (`terraform fmt -check`, `validate`, `tflint`, `trivy config`), and architect secretless AWS DevOps pipelines using OIDC Federation, ECR, and ECS/EKS Blue-Green deployments.

Objectives: ✓ Gate every IaC Pull Request with `terraform fmt -check -recursive`, `terraform init -backend=false && terraform validate`, and `trivy config` • ✓ Eliminate static `AWS_ACCESS_KEY_ID` secrets in CI runners using AWS IAM OpenID Connect (`OIDC` `sts:AssumeRoleWithWebIdentity`) • ✓ Automate zero-downtime AWS CodeDeploy / ECS / EKS Blue-Green traffic shifting with CloudWatch alarm auto-rollback

9.1 Terraform Formatting & Static Quality Gates — `fmt -check`, `validate`, `check {}` Blocks & `tflint` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Terraform Formatting & Static Quality Gates — `fmt -check`, `validate`, `check {}` Blocks & `tflint` in real DevOps & Git CI/CD systems — specific...

9.2 AWS DevOps Engineering — IAM OIDC Federation (Zero Static Keys!), ECR Immutable Tags & CodeDeploy Blue-Green [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply AWS DevOps Engineering — IAM OIDC Federation (Zero Static Keys!), ECR Immutable Tags & CodeDeploy Blue-Green in real DevOps & Git CI/CD systems —...

Module 9 Hands-On Capstone Project (100 Marks): Automated Terraform PR Linter + Secretless AWS OIDC AssumeRole Trust Policy — Create a zero-network-credential CI check validating HCL formatting/types (`fmt -check` + `validate`) paired with a strict `repo`+`environment`-scoped AWS IAM OIDC Trust Policy.

MODULE 10 OF 16 • 11 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 10: Enterprise `Azure DevOps` Multi-Stage YAML Pipelines & Self-Healing SRE Automation

Build governed multi-stage `azure-pipelines.yml` workflows with Workload Identity Federation, Templates, and Environment Approvals, and eliminate operational toil through idempotent event-driven SRE automation.

Objectives: ✓ Structure multi-stage `azure-pipelines.yml` (`stages -> jobs -> deployment -> strategy: canary/runOnce`) with reusable YAML `templates` • ✓ Authenticate Azure Pipelines to AKS/ACR/KeyVault via Workload Identity Federation (zero expiring client-secret outages!) • ✓ Eliminate repetitive SRE toil (<50% ops cap) using idempotent, circuit-breaker-guarded self-healing automation & ChatOps runbooks

10.1 Azure DevOps Pipelines — Multi-Stage `azure-pipelines.yml`, `deployment` Environments, Templates & Workload Identity [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Azure DevOps Pipelines — Multi-Stage `azure-pipelines.yml`, `deployment` Environments, Templates & Workload Identity in real DevOps & Git CI/CD sy...

10.2 DevOps & SRE Automation — Eliminating Toil, Circuit-Broken Auto-Remediation, Policy-as-Code & Drift Detection [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply DevOps & SRE Automation — Eliminating Toil, Circuit-Broken Auto-Remediation, Policy-as-Code & Drift Detection in real DevOps & Git CI/CD systems —...

Module 10 Hands-On Capstone Project (100 Marks): Reusable `azure-pipelines.yml` Governance Template + Bounded SRE Auto-Remediation Engine — Author a multi-stage `azure-pipelines.yml` enforcing gated `production` `deployment` approvals and Workload Identity Federation, paired with a circuit-broken SRE self-healing script that pre...

MODULE 11 OF 16 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Reusable CI/CD Pipelines, OIDC Cloud Auth & Supply-Chain DevSecOps

Production-grade DevOps & Git CI/CD Training engineering module covering Reusable CI/CD Pipelines, OIDC Cloud Auth & Supply-Chain DevSecOps with hands-on labs and capstone build.

Objectives: ✓ Reusable Workflows, Matrix Caching & Concurrency Cancel-in-Progress — Purpose: Cut CI pipeline feedback from 18 minutes down to 3 minutes across microservices. • ✓ Keyless OIDC Federation to AWS/GCP & Ephemeral Environment Previews — Purpose: Eliminate leaked long-lived cloud secrets and spin up live PR preview URLs automatically. • ✓ Shift-Left DevSecOps: Secret Scanning, SAST (Semgrep), Trivy & SBOM Signing — Purpose: Block leaked keys, vulnerable dependencies, and unsigned images before deployment. • ✓ Semantic Release Automation, Immutable Artifact Promotion & Changelogs — Purpose: Promote the exact tested SHA artifact from staging to production without rebuilding.

11.1 Reusable Workflows, Matrix Caching & Concurrency Cancel-in-Progress [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Reusable Workflows, Matrix Caching & Concurrency Cancel-in-Progress in real DevOps & Git CI/CD systems — specifically: Cut CI pipeline feedback fr...

11.2 Keyless OIDC Federation to AWS/GCP & Ephemeral Environment Previews [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Keyless OIDC Federation to AWS/GCP & Ephemeral Environment Previews in real DevOps & Git CI/CD systems — specifically: Eliminate leaked long-lived...

11.3 Shift-Left DevSecOps: Secret Scanning, SAST (Semgrep), Trivy & SBOM Signing [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Shift-Left DevSecOps: Secret Scanning, SAST (Semgrep), Trivy & SBOM Signing in real DevOps & Git CI/CD systems — specifically: Block leaked keys,...

11.4 Semantic Release Automation, Immutable Artifact Promotion & Changelogs [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Semantic Release Automation, Immutable Artifact Promotion & Changelogs in real DevOps & Git CI/CD systems — specifically: Promote the exact tested...

Module 11 Hands-On Capstone Project (100 Marks): Zero-Trust GitHub Actions CI/CD Pipeline with SAST/Container Gates — Design, implement, test, and document 'Zero-Trust GitHub Actions CI/CD Pipeline with SAST/Container Gates' applying every production concept from Module 11 (Reusable CI/CD Pipelines, OIDC CI...

MODULE 12 OF 16 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Production Terraform IaC, Ansible Hardening & Secrets Management

Production-grade DevOps & Git CI/CD Training engineering module covering Production Terraform IaC, Ansible Hardening & Secrets Management with hands-on labs and capstone build.

Objectives: ✓ Modular Terraform Design, Remote State Locking, Imports & Drift Checks — Purpose: Manage VPCs, clusters, DNS, and databases cleanly across dev, stage, and prod. • ✓ Policy-as-Code Checks (Checkov / OPA) & Automated Terraform Plan PR Reviews — Purpose: Catch accidental public S3 buckets or destructive resource replacements in PR review. • ✓ Idempotent Linux Host Hardening & Patching with Ansible Roles — Purpose: Standardize SSH, firewall (ufw/nftables), sysctl, and systemd baselines across fleets. • ✓ Centralized Dynamic Secrets with HashiCorp Vault / Cloud Secrets Manager — Purpose: Issue

short-lived leased DB credentials with instant revocation capabilities.

12.1 Modular Terraform Design, Remote State Locking, Imports & Drift Checks [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Modular Terraform Design, Remote State Locking, Imports & Drift Checks in real DevOps & Git CI/CD systems — specifically: Manage VPCs, clusters, D...

12.2 Policy-as-Code Checks (Checkov / OPA) & Automated Terraform Plan PR Reviews [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Policy-as-Code Checks (Checkov / OPA) & Automated Terraform Plan PR Reviews in real DevOps & Git CI/CD systems — specifically: Catch accidental pu...

12.3 Idempotent Linux Host Hardening & Patching with Ansible Roles [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Idempotent Linux Host Hardening & Patching with Ansible Roles in real DevOps & Git CI/CD systems — specifically: Standardize SSH, firewall (ufw/nf...

12.4 Centralized Dynamic Secrets with HashiCorp Vault / Cloud Secrets Manager [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Centralized Dynamic Secrets with HashiCorp Vault / Cloud Secrets Manager in real DevOps & Git CI/CD systems — specifically: Issue short-lived leas...

Module 12 Hands-On Capstone Project (100 Marks): Modular Multi-AZ Cloud Stack Provisioned 100% via Terraform — Design, implement, test, and document 'Modular Multi-AZ Cloud Stack Provisioned 100% via Terraform' applying every production concept from Module 12 (Production Terraform IaC, Ansible Harden...

MODULE 13 OF 16 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

GitOps Kubernetes Delivery (ArgoCD), Progressive Rollouts & Autoscaling

Production-grade DevOps & Git CI/CD Training engineering module covering GitOps Kubernetes Delivery (ArgoCD), Progressive Rollouts & Autoscaling with hands-on labs and capstone build.

Objectives: ✓ Declarative ArgoCD GitOps Sync, Health Waves & Drift Self-Healing — Purpose: Converge cluster state automatically to Git commits with full audit history. • ✓ Blue/Green & Metric-Gated Canary Rollouts with Automated Rollback — Purpose: Shift 5% -> 25% -> 100% live traffic while verifying latency and 5xx thresholds. • ✓ Ingress TLS Automation (cert-manager), ExternalDNS & Network Isolation — Purpose: Provision HTTPS certificates and DNS records automatically when services launch. • ✓ Cluster Cost & Reliability Engineering: HPA, Karpenter Spot Mix & PDBs — Purpose: Maintain 99.95% availability while cutting compute bills using diversified instances.

13.1 Declarative ArgoCD GitOps Sync, Health Waves & Drift Self-Healing [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Declarative ArgoCD GitOps Sync, Health Waves & Drift Self-Healing in real DevOps & Git CI/CD systems — specifically: Converge cluster state automa...

13.2 Blue/Green & Metric-Gated Canary Rollouts with Automated Rollback [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Blue/Green & Metric-Gated Canary Rollouts with Automated Rollback in real DevOps & Git CI/CD systems — specifically: Shift 5% -> 25% -> 100% live...

13.3 Ingress TLS Automation (cert-manager), ExternalDNS & Network Isolation [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Ingress TLS Automation (cert-manager), ExternalDNS & Network Isolation in real DevOps & Git CI/CD systems — specifically: Provision HTTPS certific...

13.4 Cluster Cost & Reliability Engineering: HPA, Karpenter Spot Mix & PDBs [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Cluster Cost & Reliability Engineering: HPA, Karpenter Spot Mix & PDBs in real DevOps & Git CI/CD systems — specifically: Maintain 99.95% availabi...

Module 13 Hands-On Capstone Project (100 Marks): Zero-Downtime Canary Delivery Platform on Kubernetes — Design, implement, test, and document 'Zero-Downtime Canary Delivery Platform on Kubernetes' applying every production concept from Module 13 (GitOps Kubernetes Delivery (ArgoCD), Progressiv...

MODULE 14 OF 16 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

SRE Observability Stack (Prometheus, Grafana, Loki, OTel) & Incident Ops

Production-grade DevOps & Git CI/CD Training engineering module covering SRE Observability Stack (Prometheus, Grafana, Loki, OTel) & Incident Ops with hands-on labs and capstone build.

Objectives: ✓ SLIs, SLOs & Multi-Window Error-Budget Burn-Rate Alerting — Purpose: Eliminate alert fatigue by paging engineers only when real user SLO budgets burn. • ✓ Prometheus PromQL, Grafana Dashboards, Loki Logs & OpenTelemetry Traces — Purpose: Correlate metrics, structured logs, and distributed

traces in one unified pane. • ✓ Load Testing (k6) Capacity Planning & Chaos Fault-Injection GameDays — Purpose: Find breaking points and verify failover runbooks weeks before peak sale traffic. • ✓ Automated Backup Restore Verification, Runbooks & Blameless Postmortems — Purpose: Turn every incident or drill into permanent automated platform resilience.

14.1 SLIs, SLOs & Multi-Window Error-Budget Burn-Rate Alerting [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply SLIs, SLOs & Multi-Window Error-Budget Burn-Rate Alerting in real DevOps & Git CI/CD systems — specifically: Eliminate alert fatigue by paging eng...

14.2 Prometheus PromQL, Grafana Dashboards, Loki Logs & OpenTelemetry Traces [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Prometheus PromQL, Grafana Dashboards, Loki Logs & OpenTelemetry Traces in real DevOps & Git CI/CD systems — specifically: Correlate metrics, stru...

14.3 Load Testing (k6) Capacity Planning & Chaos Fault-Injection GameDays [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Load Testing (k6) Capacity Planning & Chaos Fault-Injection GameDays in real DevOps & Git CI/CD systems — specifically: Find breaking points and v...

14.4 Automated Backup Restore Verification, Runbooks & Blameless Postmortems [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Automated Backup Restore Verification, Runbooks & Blameless Postmortems in real DevOps & Git CI/CD systems — specifically: Turn every incident or...

Module 14 Hands-On Capstone Project (100 Marks): Full SRE Golden-Signals, SLO Burn-Rate & Disaster Drill Suite — Design, implement, test, and document 'Full SRE Golden-Signals, SLO Burn-Rate & Disaster Drill Suite' applying every production concept from Module 14 (SRE Observability Stack (Prometheus, G...

MODULE 15 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

DevSecOps Supply Chain, OIDC Keyless Cloud Deploys & Terraform State Governance

Self-explanatory list-and-bullet production mastery module covering DevSecOps Supply Chain, OIDC Keyless Cloud Deploys & Terraform State Governance in DevOps & Git CI/CD Training, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken Gemini AI Viva.

Objectives: ✓ Keyless OIDC Cloud Authentication, Least-Privilege Workflow Permissions &Pinned Actions — Purpose: Eliminate long-lived cloud secrets in CI/CD and block supply-chain workflow injection. • ✓ Automated SAST, Secret Scanning, Dependency Audits & Container CVE Gates — Purpose: Catch leaked keys and vulnerable packages on every pull request before merge. • ✓ Remote Terraform State Locking, Drift Detection & Reusable Environment Modules — Purpose: Manage dev, staging, and production infrastructure safely across multi-engineer teams. • ✓ Build, verify, and document 'Keyless GitHub Actions OIDC -> AWS/Docker/Terraform Pipeline with Policy Gates' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + Google Gemini AI Technical Interviews.

15.1 Keyless OIDC Cloud Authentication, Least-Privilege Workflow Permissions &Pinned Actions [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Keyless OIDC Cloud Authentication, Least-Privilege Workflow Permissions &Pinned Actions in real DevOps & Git CI/CD systems — specifically: Elimina...

15.2 Automated SAST, Secret Scanning, Dependency Audits & Container CVE Gates [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Automated SAST, Secret Scanning, Dependency Audits & Container CVE Gates in real DevOps & Git CI/CD systems — specifically: Catch leaked keys and...

15.3 Remote Terraform State Locking, Drift Detection & Reusable Environment Modules [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Remote Terraform State Locking, Drift Detection & Reusable Environment Modules in real DevOps & Git CI/CD systems — specifically: Manage dev, stag...

Module 15 Hands-On Capstone Project (100 Marks): Keyless GitHub Actions OIDC -> AWS/Docker/Terraform Pipeline with Policy Gates — Architect, implement, test, and document 'Keyless GitHub Actions OIDC -> AWS/Docker/Terraform Pipeline with Policy Gates' applying every bullet-wise engineering concept from Module 15 (DevSe...

MODULE 16 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Progressive Delivery (Blue-Green/Canary), SLO Observability & Incident Response

Self-explanatory list-and-bullet production mastery module covering Progressive Delivery (Blue-Green/Canary), SLO Observability & Incident Response in DevOps & Git CI/CD Training, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken Gemini AI Viva.

Objectives: ✓ Blue-Green, Canary & Rolling Releases with Automated Health-Check Rollbacks — Purpose: Ship multiple production deployments per day while keeping blast radius near zero. • ✓ Four Golden Signals (Latency, Traffic, Errors, Saturation), SLIs/SLOs & Error Budgets — Purpose: Replace noisy CPU pager fatigue with actionable symptom-based SRE alerts. • ✓ Blameless Postmortems, Runbook Automation & DevOps/SRE Interview Scenarios — Purpose: Walk through broken CI/CD pipelines, DNS/TLS outages, and scaling incidents calmly. • ✓ Build, verify, and document 'Prometheus + Grafana Golden-Signals Alerting Stack & Zero-Downtime Rollback Drill' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + Google

Gemini AI Technical Interviews.

16.1 Blue-Green, Canary & Rolling Releases with Automated Health-Check Rollbacks [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Blue-Green, Canary & Rolling Releases with Automated Health-Check Rollbacks in real DevOps & Git CI/CD systems — specifically: Ship multiple produ...

16.2 Four Golden Signals (Latency, Traffic, Errors, Saturation), SLIs/SLOs & Error Budgets [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Four Golden Signals (Latency, Traffic, Errors, Saturation), SLIs/SLOs & Error Budgets in real DevOps & Git CI/CD systems — specifically: Replace n...

16.3 Blameless Postmortems, Runbook Automation & DevOps/SRE Interview Scenarios [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Blameless Postmortems, Runbook Automation & DevOps/SRE Interview Scenarios in real DevOps & Git CI/CD systems — specifically: Walk through broken...

Module 16 Hands-On Capstone Project (100 Marks): Prometheus + Grafana Golden-Signals Alerting Stack & Zero-Downtime Rollback Drill — Architect, implement, test, and document 'Prometheus + Grafana Golden-Signals Alerting Stack & Zero-Downtime Rollback Drill' applying every bullet-wise engineering concept from Module 16 (Pr...

3. LIVE CLIENT PRODUCTION PROJECT, GLOBAL CERTIFICATIONS & CAREER SCOPE

End-of-Course Live Client Production Project + Official Internship Confirmation Letter

Project: ShipFast Automated GitSecOps CI/CD Pipeline, Terraform Cloud & Prometheus Stack • **Confirmed Role:** DevOps, CI/CD & Cloud Infrastructure Automation Intern

Domain & Production Infra: Cloud-Native SaaS Release & Reliability Engineering Client (Linux + GitHub Actions + Docker + Terraform IaC + AWS + Prometheus & Grafana)

- Built multi-stage GitHub Actions CI/CD pipelines running lint, tests, Trivy container scans, and automated rollouts.
- Provisioned reproducible AWS compute, networking, and security resources using modular Terraform state workflows.
- Configured Nginx reverse-proxy TLS, systemd services, zero-downtime blue/green deployments, and secret injection.
- Deployed Prometheus metrics, Grafana golden-signal dashboards, and Alertmanager incident hooks to production.

Aligned Global Vendor Certifications: **TA-003** (HashiCorp Certified: Terraform Associate (003)) • **GH-200** (GitHub Actions Certified CI/CD Engineer) • **DCA** (Docker Certified Associate (Container Engineering)) • **DOP-C02** (AWS Certified DevOps Engineer – Professional)

Target Job Roles: DevOps Engineer, CI/CD & Release Automation Engineer, Junior Site Reliability Engineer (SRE), Infrastructure & Platform Engineer

Realistic Annual CTC Scope: **Fresher / Associate DevOps Engineer (0–1 Yr):** ₹5.5 LPA – ₹10.0 LPA **Mid-Level DevOps / SRE Engineer (1–3 Yrs):** ₹11.0 LPA – ₹22.5 LPA **Lead DevSecOps / Principal Platform Architect (3–6+ Yrs):** ₹23.0 LPA – ₹45.0+ LPA

MockAttempt Academy Pvt. Ltd. • Email: support@mockattempt.com • Phone/WhatsApp: +91 98765 43210 • Web: https://mockattempt.com/courses/devops-training

Try 2 Free Topic-Wise 50-MCQ Mock Tests & Spoken Gemini AI Interviews daily on every topic above before enrolling.