

— TR-03

Cloud Computing & DevOps

COURSE BRIEF

SKILLS & TOOLS YOU WILL MASTER



Linux + Bash



AWS



Docker



Kubernetes



Terraform + GitHub Actions

10 Reasons This Works.

Why students choose Saarathi – and how each card helps you ship a real career.

01

10 Max / Batch

Not 50. Not 30. Ten. Every student seen, heard, and unblocked – every class.

02

Sunday Open Classroom

Every Sunday the classroom is open. Come in, pair-program, get unstuck, or just focus.

03

Weekly Detailed Syllabus

Full syllabus from day one. Every week: what to read, what to build, what the outcome is.

04

Revision Every 5th Day

Every fifth class is a revision session. Nothing piles up or slips through.

05

Week 1: Mindset

The first week is about getting into the right headspace before touching a line of code.

06

3 Ways of Learning

Self-learning + collaboration + mentorship. All three built into every course.

07

1:1 Mock Interview

Real mock interview with your mentor in the final week. Actual hiring prep, not fluff.

08

CV + LinkedIn

Resume writing and LinkedIn optimisation. Leave with a profile that gets replies.

09

Real Projects

Every course ends with a deployed, publicly accessible project you can show employers.

10

Gate Assessment

Aptitude test before Day 1. Maps your strengths, weaknesses, learning style, and pace – then builds your personal plan.

saarathiacademy.com.np/cloud-computing-and-devops-course-in-nepal

+977-9744442469 • +977-9761095364

Cloud Computing & DevOps

Why This Course

This course trains the junior cloud / DevOps engineer profile employers actually hire: someone who understands Linux, containers, cloud services, monitoring and automation together. Product teams want engineers who can provision a feature (set up the servers and services it needs), harden it (lock down its security), monitor it and roll it back safely, not just write code. Pay scales with your skills, portfolio, and whether you work locally or for international/remote clients, we don't promise a salary.

This 12-week DevOps-focused program builds that exact profile. The first third (Weeks 1–4) is true foundation, Linux, how the internet, DNS, HTTP and TLS work, and version control, before any cloud, container or Kubernetes depth, so no beginner meets orchestration without the ground under them. It ends with a deployed Kubernetes capstone by Week 12 and a clear runway toward AWS (Cloud Practitioner → SAA, Solutions Architect Associate) and CKA (Certified Kubernetes Administrator) certifications. AWS Cloud Practitioner (CLF-C02) is the course's job-ready gate: passed, or sat with a scheduled re-take date, by course end, with prep distributed across the weeks. The stack, Linux, AWS, Docker, Kubernetes, GitHub Actions, Terraform, Prometheus + Grafana, plus GitOps with ArgoCD and OpenTelemetry, matches what real infrastructure teams run in production today.

AI tools are woven into the workflow the way real infrastructure teams use them: ChatGPT for drafting runbooks (step-by-step operational guides) and incident timelines, Copilot for writing Bash and Terraform faster, and an honest look at where AI-generated infrastructure code requires verification before you run it in production. Every phase includes a note on how AI is changing that part of the role.

Who Is This Course For

- **Absolute beginners** with no Linux background who want a real infrastructure career
- **Support and IT professionals** moving into cloud and DevOps roles
- **Developers** who want to understand what happens after code is pushed
- **System admins** transitioning from on-prem (servers you own and host yourself) to cloud-native stacks
- **CS or IT graduates and career changers** moving into fast-growing infrastructure roles
- **Freelancers** who want to offer hosting, deployment and automation services

Course Snapshot

PARAMETER	DETAILS
Course Code	TR-03
Duration	12 Weeks (3 Months)
Schedule	Monday to Friday, 2 Hours/Day
Total Hours	120 Hours
Batch Size	Maximum 10 Students
Course Fee	NPR 35,500
Level	Beginner → Junior Cloud / DevOps Engineer
Outcome	Junior Cloud Support / Cloud Engineer (DevOps-track), 12–24 month runway to full DevOps Engineer

Prerequisites

- Comfortable using a computer daily and following a technical setup guide
- A laptop with 8GB+ RAM (Linux, macOS, or Windows with WSL2)
- No Linux or cloud experience required, Week 1 starts at the terminal and builds up
- Before Day 1: create a free AWS account at aws.amazon.com/free. It needs a card to verify and can take a day to activate, so start it early. The free tier includes USD 100 in credits, valid 6 months
- Saarathi Gate Assessment before Day 1, diagnostic, no pass or fail

Everything else, the Ubuntu VM, Docker, Git and VS Code, we install together in the first class.

Assessments at a glance

Two exams bracket the course. The **Saarathi Gate Assessment** runs before Day 1 as a diagnostic with no pass or fail; it maps your learning style and pace. The **Saarathi Certification Exam** is sat at the end of the course, 2 hours and around 30 questions specific to this course in mixed formats (single choice, multi-select, true/false, numeric, ordering, matching, fill-in-the-blank, short or long answer, code where relevant, and short case studies). Pass mark is 60 percent and a pass is mandatory for the Saarathi Certificate of Completion. The exam is

proctored: it runs in fullscreen, switching tab or app and pasting are logged, and three strikes end it. You can answer the questions in any order and come back to any of them, but every question needs an answer before you submit. Recommendation letters are discretionary, mentor-issued only to learners who pass the Certification Exam, and never guaranteed.

Your Learning Week

DAY	ACTIVITY
Mon–Fri	2-hour live class session (hands-on, project-driven) with mentor
Mon–Fri	2-hour self-study at home (assignments, practice, debugging)
Sunday	Open classroom for focused build time or rest
Day 5 (every week)	Within the 2-hour session, the last 1 hour is a built-in revision block plus a broken-then-fixed drill for consolidation and doubt-clearing

Real improvement happens when you build, break, debug and re-ship between classes. The classroom gives structure; the hours between classes give depth.

Week-by-Week Curriculum

Phase 1 · Linux, Internet & Networking Foundations (Weeks 1–4, 40 Hours)

The first third of the course is true foundation: Linux, how the internet works, and version control, taught for the why before any cloud, container or Kubernetes depth.

WEEK	FOCUS AREA	TOPICS
Week 1	Linux & Terminal Foundations	- What is the cloud and why: the problem it solves, IaaS vs PaaS vs SaaS, and where DevOps sits - Why Linux runs almost every server and container - Terminal basics (commands, not clicks) and how the Linux file system is organised - File operations (copy, move, delete), grep to search inside files, find to locate files anywhere - Users and permissions (who can read or change each file), SSH keys for password-less login, remote access to your own server - Environment setup (VS Code Remote-SSH) and documentation habits

WEEK	FOCUS AREA	TOPICS
		- How AI is changing this role: ChatGPT and Copilot in daily DevOps work
Week 2	Linux Administration & Scripting	- Server accounts: user and group management, <code>sudo</code> (temporary admin power), least privilege (only needed access) - What runs on a server: processes (running programs), services (background programs), <code>systemctl</code> to start and stop them, <code>journalctl</code> to read their logs - Cron jobs (scheduled, automatic tasks) and package management (clean software installs) - Bash scripting: commands in a runnable file, with variables, conditionals, loops, functions - Redirects (routing command output), logging, debugging with <code>set -e</code> (script stops at the first failed step) - Backup and healthcheck automation: copy data somewhere safe, confirm the server is alive - Linux security hardening basics: fail2ban (blocks repeated break-ins), unattended-upgrades (automatic security patches), what attackers look for on an exposed server
Week 3	How the Internet Works & Version Control	- How a request travels: client → DNS lookup → TCP handshake → request → response, the map every later debug uses - IP addresses, ports (numbered doors), TCP/UDP (reliable vs fast delivery) - DNS in depth: how names like google.com resolve, record types (<code>A</code> , <code>CNAME</code> , <code>MX</code>), TTLs and why propagation feels slow - HTTP/HTTPS and TLS: request/response anatomy (methods, status codes, headers) and how certificates encrypt it - Git and GitHub from zero: <code>commit</code> , <code>push</code> , <code>pull</code> , reading <code>git log</code> and <code>git diff</code> , <code>.gitignore</code> - Branching and the pull-request workflow: feature branches, resolving a conflict, review-then-merge - Why version control is the source of truth that GitOps and IaC build on later

WEEK	FOCUS AREA	TOPICS
Week 4	Network Tooling, Hardening & Reverse Proxy	- Network fault-finding tools: <code>ping</code> (is it reachable), <code>curl</code> (fetch a page), <code>dig</code> (check DNS), <code>ss</code> (open ports), <code>traceroute</code> (where the path breaks) - Firewall basics (UFW) to control incoming traffic, SSH hardening so only the right keys get in, fail2ban tied into the firewall - Server hardening checklist: disabled root SSH, automatic security patches, log review, the baseline every cloud host needs - Nginx reverse proxy: one front door that passes each request to the right app behind it (upstream routing) - TLS termination with Let's Encrypt: a free, auto-renewing certificate so the site serves real HTTPS - Structured debugging: walking DNS → port → firewall → service until the fault shows

Phase 2 · AWS Cloud & Docker (Weeks 5–6, 20 Hours)

WEEK	FOCUS AREA	TOPICS
Week 5	AWS Cloud Foundations	- AWS account setup done safely: MFA (a second login step) and billing alerts, no surprise bills - IAM, the AWS permission system: users, groups and roles with least privilege - Launching cloud servers with EC2: instance types, AMIs, security groups (per-server traffic filter), key pairs and tagging discipline - Storage: EBS volumes (cloud disks) and snapshots, plus S3 buckets (cloud file containers) with versioning and lifecycle rules - RDS managed database inside a VPC: public/private subnets, route tables, IGW (Internet Gateway), the three-tier (web/app/database) layout - AWS CLI workflows: driving AWS from the terminal, not the console - Cost control and FinOps: Cost Explorer, Trusted Advisor, Budgets, Reserved vs Spot vs On-demand, plus GCP/Azure cross-walk awareness - AWS CLF-C02 exam runway starts: weekly Day 5 practice sets through Week 12
Week 6	Docker Engineering	- Containers vs VMs: an app packaged with everything it needs, runs the same anywhere, lighter than a full VM, run by the Docker engine

WEEK	FOCUS AREA	TOPICS
		- Dockerfile (the image-build recipe), layering and caching for fast rebuilds, multi-stage builds for small final images - Runtime settings: environment variables (config from outside), port mapping (container door to host) and volumes (storage that survives restarts) - Container logs and failure diagnosis - Docker Compose for multi-service stacks: app, database and web server up with one command - Registry push/pull (Docker Hub / ECR): publishing images so any machine can fetch them

Phase 3 · Kubernetes, CI/CD & Infrastructure as Code (Weeks 7–10, 40 Hours)

WEEK	FOCUS AREA	TOPICS
Week 7	Kubernetes Foundations	- Kubernetes building blocks: cluster (machines acting as one), node (each machine), pod (smallest running unit), deployment (keeps the right copy count running), service (stable address for pods) - Driving the cluster with <code>kubectl</code> : <code>get</code> , <code>describe</code> , <code>logs</code> , <code>exec</code> , the daily Kubernetes commands - YAML manifests (describe what should run) and declarative workflows (state the goal, Kubernetes makes it true) - Deployments, replicas (copy count) and rollouts (gradual release of a new version, with rollback) - Exposing apps: ClusterIP (cluster-only), NodePort (quick external port), LoadBalancer (production entrance) services - CrashLoopBackOff diagnosis: fixing the classic restarting-pod error step by step - Capstone project selection
Week 8	Kubernetes Workload Delivery	- Settings out of code: ConfigMaps for app config, Secrets for passwords and keys - AWS Secrets Manager / SSM Parameter Store: live credentials at runtime, never in your repo - Namespaces and workload isolation: walls between environments inside one cluster - Ingress controllers, the cluster's front door, host/path routing to the right service - PersistentVolume and PVC basics: storage that outlives any pod - Health and readiness probes: automatic checks that a pod is

WEEK	FOCUS AREA	TOPICS
		alive and ready for traffic - Runbooks for broken deployments - Helm charts awareness: K8s manifests as versioned packages, installed in one command - Service mesh awareness (Istio, Linkerd): mTLS (encrypted service-to-service traffic), traffic shifting and observability between services
Week 9	CI/CD Pipelines	- Python scripting for ops automation: small scripts for cloud chores - CI/CD: machines checking your code (continuous integration) and shipping it (continuous delivery), through build, test, package, deploy - GitHub Actions workflows, jobs, secrets: pipelines that run on every push - Docker image build and publish in CI - Catching vulnerabilities before they ship: a Trivy scan failing the build on HIGH/CRITICAL CVEs (known security flaws), plus SBOM generation, a software bill of materials for your image (DevSecOps) - Supply-chain hygiene: pin CI actions to immutable commit SHAs so a hijacked tool cannot poison your pipeline, the March 2026 trivy-action compromise lesson - Environment promotion (dev → staging → prod) and deployment verification at each gate - Pipeline troubleshooting - GitOps with ArgoCD, deployed hands-on: it pulls declarative state from Git into the cluster with self-healing, the default delivery model juniors are now expected to show
Week 10	Terraform, Ansible & IaC	- Terraform: infrastructure as code, rebuilt exactly from providers (cloud plugins), resources, HCL (its config language) and state (record of what exists) - The init, plan, apply workflow: prepare, preview, make it real - OpenTofu awareness (v1.12, the CNCF open-source Terraform fork, same HCL, adds built-in state encryption) - Reusability: variables, outputs, modules (packaged infrastructure blocks) and team-shared remote state - Infrastructure change review and drift detection (manual changes outside Terraform) - Ansible in depth: configuring many servers from one set of files, with playbooks (to-do lists), inventory (which servers), roles

WEEK	FOCUS AREA	TOPICS
		(reusable bundles), a real server-provisioning playbook hands-on - Azure DevOps awareness: pipeline YAML compared with GitHub Actions, for Azure clients - Capstone architecture planning

Phase 4 · Observability, Capstone & Career Launch (Weeks 11–12, 20 Hours)

WEEK	FOCUS AREA	TOPICS
Week 11	Monitoring & Observability	- Logs vs metrics vs traces vs alerts, plus OpenTelemetry , the 2026 vendor-neutral standard for emitting all three from one instrumentation - Prometheus setup: the standard metrics collector, with metric types and scraping config (what to measure, where to fetch) - Grafana dashboards and visualisation: numbers turned into live charts for the team - Alerting rules and threshold thinking (which numbers mean trouble), plus on-call awareness (the engineer who gets paged) - CloudWatch integration: wiring in AWS's built-in monitoring - Incident response basics - AI for DevOps: ChatGPT for runbooks and incident timelines, Copilot for Bash and Terraform, reviewing AI-generated infra code before running it - Capstone: monitoring stack integration
Week 12	Capstone & Career Launch	- Capstone project, cloud + Docker + K8s + CI/CD - Architecture presentation and peer review - Resume, LinkedIn , GitHub portfolio (self-study with mentor review, class time stays on the capstone) - Troubleshooting interview practice - Certification roadmap (AWS CLF-C02 → SAA → CKA v1.35); CLF-C02 passed or re-take scheduled, the job-ready gate - Career planning and job search strategy

Skills You Will Gain

CATEGORY	TECHNOLOGIES
Linux & Networking	Bash, Linux admin, security hardening, Nginx reverse proxy
Cloud (AWS)	EC2, S3, IAM, VPC, RDS, CloudWatch, Cost Explorer
Containers	Docker, Docker Compose, multi-stage builds
Orchestration	Kubernetes, kubectl, Ingress, ConfigMaps
CI/CD & GitOps	GitHub Actions pipelines, environment promotion, Trivy image scanning, SBOM, ArgoCD GitOps deploy
IaC	Terraform / OpenTofu (depth), Ansible (hands-on playbooks)
Observability	Prometheus, Grafana, OpenTelemetry, structured logging
AI Tools	ChatGPT for runbooks, Copilot for Bash and Terraform
Tools	Git, GitHub, Python for automation

Topic Depth and Awareness

Depth means repeated practice on real infrastructure. **Awareness** means you are introduced clearly enough to read and compare later.

FOCUS AREA	LEARNING FOCUS
Linux admin, Bash, networking, Nginx	Depth
Linux security hardening (fail2ban, log hardening)	Depth
AWS (IAM, EC2, S3, VPC, RDS, CloudWatch)	Depth
AWS cost optimisation (Cost Explorer, Reserved vs Spot vs On-demand)	Depth
Docker, Compose, multi-stage builds	Depth
Kubernetes (Deployments, Services, Ingress)	Depth

FOCUS AREA	LEARNING FOCUS
GitHub Actions CI/CD	Depth
Image scanning (Trivy) + secrets manager in pipeline	Depth
SBOM generation	Awareness-to-hands-on (generated once in CI)
Supply-chain signing	Awareness
OpenTofu (Terraform fork)	Awareness
Terraform (HCL, state, modules, drift detection)	Depth
Ansible (playbooks, roles, hands-on provisioning)	Depth
Prometheus + Grafana observability	Depth
OpenTelemetry (vendor-neutral telemetry)	Awareness-to-hands-on
AI for DevOps (ChatGPT runbooks, Copilot for Bash/Terraform)	Depth
GitOps / ArgoCD deploy	Awareness-to-hands-on
Azure DevOps pipeline YAML	Awareness
Helm charts, service mesh	Awareness
GCP, Azure	Awareness (AWS is depth)

Project Pool

Each student picks **one** capstone with mentor in Week 7. Pool rotates per batch:

#	PROJECT	FOCUS
1	Containerised 3-tier app deployed on AWS with CI/CD	Docker, K8s, GitHub Actions
2	Zero-downtime blue/green deployment pipeline	K8s, Argo Rollouts, GitHub Actions

#	PROJECT	FOCUS
3	Terraform-provisioned production AWS stack	Terraform, IAM, VPC, modules
4	Observability-first service with SLOs and alerting	Prometheus, Grafana, alerting
5	SRE-style incident runbook + chaos test	Incident response, on-call
6	Trainer-scoped client brief	Whatever the brief requires

Career Paths & Trajectory

ROLE	TIMELINE	FOCUS
Junior Cloud Support / DevOps Engineer	0–2 yrs	Basic infra, incidents, pipelines
Cloud / DevOps Engineer	2–4 yrs	Own pipelines, K8s ops, cost control
Senior DevOps / Platform Engineer	4–7 yrs	Architecture, SLOs, team mentoring
Platform Engineer / SRE	7+ yrs	Reliability, on-call, global scale

FAQ

What does Cloud Computing & DevOps cost and how do I pay?

NPR 35,500. Pay by cash at our Old Baneshwor campus, bank transfer, or Fonepay. Email info@saarathiacademy.com.np for the payment plan.

Do I need prior experience?

Prerequisites are listed above. The Saarathi Gate Assessment runs before Day 1 to map your starting point, diagnostic, not pass-or-fail. Below prerequisites? We suggest a short foundation track first.

How long is the course?

12 Weeks (3 Months). The capstone runs inside the course, scoped with your mentor in Week 7 and shipped live in Week 12, with a 1:1 mock interview in the final week.

What tools will I use?

Docker, Kubernetes, Terraform, AWS, GitHub Actions, Prometheus (plus Grafana, ArgoCD and OpenTelemetry). Every module ships a public, deployed artefact for your portfolio, no toy projects.

What roles does this prepare me for?

Roles are listed in the Career Paths section. We do not promise jobs. We build your portfolio, run mock interviews, polish your CV and LinkedIn, and refer you into our hiring network.

How does the batch and mentorship work?

Maximum 10 students per batch, every student gets direct mentor time and project review. Hybrid by default, in-person at Old Baneshwor with online recordings and office hours. Sunday Open Classroom is free for enrolled students: pair-program, get unstuck, or just focus.

Enrollment

- **Next Batch:** see the live course page
 - **Location:** Old Baneshwor Chowk, Kathmandu, Nepal
 - **Mode:** Online + Offline
 - **Contact:** 9744442469, 9761095364
 - Limited to 10 seats per batch
-

Cloud Computing & DevOps — full details, fees and next batch dates:
saarathiacademy.com.np/cloud-computing-and-devops-course-in-nepal

Version 1.0 · Updated July 2, 2026