TR-03: Cloud & DevOps Engineering

From Terminal to Production - Build Infrastructure That Scales

Every modern application needs reliable infrastructure. This course builds practical cloud and DevOps engineers from the ground up, starting from Linux fundamentals and building systematically through Docker, Kubernetes, CI/CD, and infrastructure-as-code. In 14 weeks, you'll master the automation, containerization, and deployment skills that companies worldwide are actively hiring for.

The focus is on operational clarity, hands-on classroom work, and building infrastructure habits that work in real production environments.

Why This Course?

The Market Reality

Global Context: Modern software teams rely on cloud infrastructure, repeatable deployments, observability, and automation to ship safely. Engineers who can manage containers, CI/CD, and infrastructure as code are increasingly central to product reliability.

Nepal Context: Nepal's digital infrastructure is modernizing: government services like Nagarik App, banking systems, and e-commerce platforms require reliable CI/CD pipelines, containerized deployments, and cloud infrastructure management. Companies like CloudFactory, Leapfrog, and Deerwalk use AWS and Docker in production.

Your Opportunity: This course positions you for **DevOps engineer, cloud engineer, and SRE roles** - the infrastructure backbone every tech company needs.

Nepal-Relevant Reality	Opportunity
Every Nepali tech company needs deployment infrastructure	DevOps skills are universally demanded
Cloud adoption is accelerating across Nepal	Infrastructure engineers are scarce
Remote DevOps roles pay premium compensation	International access from Kathmandu
Government and fintech are modernizing rapidly	Critical infrastructure roles are opening

Course Snapshot

Parameter	Details
Course Code	TR-03
Title	Cloud & DevOps Engineering
Duration	3.5 Months (14 Weeks)
Schedule	Monday to Friday (Mon–Fri, 5 Days/Week), 2 Hours/Day
Total Hours	140 Hours of Live Training
Batch Size	Maximum 10 Students
Course Fee	NPR 35,000
Prerequisites	Strong computer fluency and willingness to work through setup, logs, and command-line workflows. Laptop with 8GB+ RAM. Saarathi Gate Assessment (diagnostic, no pass/fail) before Day 1.
Self-Study	Minimum 2 hours/day outside class (mandatory)
Outcome	Cloud / DevOps Engineer / Platform Support Engineer

Your Learning Week

Day	Activity
Mon–Fri	2-hour live class session (hands-on, project-based)
Mon–Fri	Minimum 2 hours self-study & practice (mandatory)
Saturday	No classes - flexible self-study, peer collaboration, project work
Sunday	Whole day self-learn time. Classrooms remain fully open for you to come in, study, collaborate with peers, and build projects. (Highly recommended for networking!)

> *Every student MUST spend at least 2 dedicated hours a day on focused classroom practice beyond the classroom at home. This is non-negotiable for success, it is what separates graduates who get hired from those who don't.*

Week-by-Week Curriculum

Phase 1: Linux, Shell, Networking & Operations (Weeks 1–4, 4 Weeks, 40 Hours)

Week	Focus Area	What You'll Master
Week 1	Infrastructure Thinking	Linux mindset, terminal comfort, Git, environment setup
Week 2	Linux Administration	Users, services, permissions, process management, logs
Week 3	Shell Automation	Scripts, pipes, cron, backups, debugging workflows
Week 4	Networking & Reverse Proxy	DNS, ports, HTTP, firewalls, proxy basics, request tracing

Phase 2: Cloud Services, Docker & Monitoring (Weeks 5–8, 4 Weeks, 40 Hours)

Week	Focus Area	What You'll Master
Week 5	Cloud Foundations	IAM, compute basics, account hygiene, access control thinking
Week 6	Storage & Networking	Object storage, block storage, VPC-style concepts, cost awareness
Week 7	Docker Engineering	Images, Dockerfiles, Compose, container networking
Week 8	Monitoring Basics	Logs, metrics, alert awareness, operations review

Phase 3: Kubernetes, CI/CD, IaC & Career Prep (Weeks 9–14, 6 Weeks, 60 Hours)

Week	Focus Area	What You'll Master
Week 9	Kubernetes Foundations	Pods, deployments, services, kubectl habits
Week 10	Kubernetes Delivery	Config, secrets, ingress, rollout troubleshooting
Week 11	Automation & CI/CD	GitHub Actions, build pipelines, release workflow basics
Week 12	IaC Foundations	Terraform concepts, state, modules, provisioning flow
Week 13	Capstone Build	Cloud plus Docker plus K8s integration, documentation, demos
Week 14	Career Launch	Ops storytelling, resume, portfolio, troubleshooting interviews

Skills You'll Gain

Technologies & Platforms

Technology	Proficiency Level
Linux Administration	Professional
Docker & Compose	Container Engineering
Kubernetes	Orchestration Fundamentals
GitHub Actions	CI/CD Pipelines
Terraform	Infrastructure-as-Code
AWS/Cloud Services	Core Service Proficiency
Shell Scripting	Automation

Development Tools

Tool	Application
Git & GitHub	Version control, collaboration
Nginx	Reverse proxy configuration
Prometheus & Grafana	Monitoring & alerting
kubectl	Kubernetes management

Topic Depth and Awareness

Section	Guidance
Purpose	This course intentionally separates what you need to master in depth from what you only need to understand with working awareness.
Depth	The Linux, container, cloud, and deployment workflows practiced repeatedly in class and classroom exercises The execution areas you are expected to perform independently in junior DevOps and cloud work The troubleshooting habits most likely to matter in real infrastructure environments

Section	Guidance
Awareness	Adjacent tools, optional stretch topics, and industry context introduced for broader understanding Concepts you should be able to explain, compare, and recognize even if you are not yet executing them independently Advanced directions for later specialization, higher-level tracks, or guided self-study
How to use this syllabus	Spend most of your self-study time strengthening the depth topics first. Treat awareness topics as context builders that help you make better decisions and understand the larger professional landscape.

Project Pool

> *All options below are **intermediate-level final projects**. Each student chooses **one** final project from this pool. Trainers may run smaller guided exercises during the course, but public phase-wise project sections are intentionally removed so the completion standard stays clear and consistent.*

#	Final Project Choice	What You Will Build	Core Stack / Tools
1	Linux & Nginx Operations Stack	Build and harden a Linux-based service stack with reverse proxy, logs, services, and operational runbooks.	Linux, Bash, Nginx, systemctl, logs
2	Dockerized AWS Service Platform	Deploy a Dockerized application on AWS with secure access, storage, and basic monitoring.	AWS, Docker, EC2, security groups, CloudWatch
3	Kubernetes Service Pack	Ship a small Kubernetes deployment with services, ingress, config, and troubleshooting notes.	Kubernetes, kubectl, Ingress, ConfigMaps, deployment flow
4	CI/CD & Rollback Pipeline	Build a multi-step deployment pipeline with test gates, environment promotion, and rollback steps.	GitHub Actions, Docker, deployment automation, release controls
5	Monitoring & Incident Response Pack	Instrument a service with metrics, alerts, dashboards, and a simple incident-response playbook.	Prometheus, Grafana, logging, alerting, runbooks

Career Paths & Trajectory

Role Path	Focus and Proof	Stage and Timeline	What Actually Matters
Junior Cloud Support Engineer	Support Linux hosts, cloud resources, and deployment troubleshooting with better operational discipline. Proof you leave with: Linux notes, AWS setup proof, and stronger runtime inspection habits	Entry role - first 0–12 months	Terminal confidence, clear documentation, safe access handling, and calm first-response troubleshooting.
Junior DevOps Engineer	Maintain CI/CD workflows, container deployments, and basic cloud automation for real teams. Proof you leave with: Docker delivery flow, GitHub Actions practice, and deployment checklists	Growth role - 1–3 years	Reproducible deployments, log-based debugging, and reducing manual steps without hurting reliability.
Cloud / Platform Engineer	Own environment setup, monitoring, Kubernetes support, and service reliability across a product area. Proof you leave with: Runbooks, monitoring notes, and stronger service-awareness	Senior individual contributor - 3–5 years	Service ownership, cost and security awareness, and better automation judgment under pressure.
Platform Engineer / SRE	Improve reliability standards, rollback discipline, and operational tooling across multiple teams. Proof you leave with: Incident habits, architecture reasoning, and support-ready documentation	Leadership path - 5+ years	Reliability mindset, mentoring, and making delivery safer and more observable across the platform.

Saarathi Gate & Completion Review

Before You Start: Saarathi Gate Assessment

All students complete the **Saarathi Gate Assessment** before Day 1. It is a short diagnostic review of aptitude, learning behaviour, and thinking style. It has **no pass/fail** and is used only to tailor support from the start.

After Course Completion: Saarathi Completion Review

The **Saarathi Academy Certificate** is issued after the selected final project is completed, documented, and reviewed by the trainer. There is **no separate certification exam** for this course.

Completion Requirements:

1. **Attendance:** Minimum 80% attendance
2. **Weekly Work:** Core deliverables, revision work, and practice tasks completed
3. **Final Project:** One intermediate-level project selected from the project pool and completed to trainer-approved quality
4. **Portfolio Proof:** Screenshots, documentation, case-study notes, or equivalent proof assets updated
5. **Trainer Review:** Practical execution, consistency, communication, and overall growth signed off by the trainer

Enrollment & Next Steps

Next Batch: Starting soon (contact for exact dates) **Offline Location:** Old Baneshwor Chowk, Kathmandu, Nepal **Mode:** Online + Offline **Contact (Call/WhatsApp):** 9761095364, 9744442469

» **[ENROLL NOW]** - Limited to 10 seats per batch

Every app needs infrastructure. In 14 weeks, you'll have the skills and the portfolio to prove it.

Last Updated: Mar 30, 2026