

Docker & Kubernetes

Course Duration ~90 Hours · 12 Weeks · Live Online · Lab-Heavy · You Will Break Clusters and Fix Them

Curriculum reviewed: July 2026 — Kubernetes ships continuously and the ecosystem changes fast; this syllabus is checked against official documentation

Production Images

Supply-Chain Security

Gateway API Migration

GPU Scheduling

GitOps

Ops, Scale & Cost

In short: Containers are easy. Running them for thousands of users, at 3am, when something breaks — that's the job, and this course teaches the second part. It is the foundation of the entire DevOps track: everything downstream in cloud, platform and SRE work rests on this layer, and the differentiator at hire time is not writing manifests but operating clusters. The urgent item first: the upstream Kubernetes project has **retired Ingress NGINX** — one of the most widely deployed components now receives no further releases, bug fixes or security patches, leaving a very large number of production clusters running an unmaintained, internet-facing component. We teach that migration as first-class, funded work: assessment, alternatives and a staged cutover to the **Gateway API**, now the recommended routing standard. Three more things date this to now: **Kubernetes is where AI workloads run** (declarative, topology-aware GPU and accelerator scheduling has graduated to stable — a bridge to our MLOps course); operational features that changed daily practice (native sidecars, in-place pod resource resizing, declarative admission policy, and deprecations that break upgrades); and **container images as a supply-chain artefact** (distroless, non-root, scanned, SBOM, signed, provenance). Across 11 lab-heavy modules you build signable images, operate core workloads, run the ingress migration, deliver by GitOps, harden a multi-tenant cluster, schedule GPUs, and survive break-fix labs under time pressure — closing with a production-grade platform capstone. It also answers, honestly, whether you actually need Kubernetes at all. Aligned to the CNCF certification path (CKAD, CKA, CKS). The AI question: AI writes Dockerfiles and manifests fluently and produces exactly the artefacts that pass review and fail in production — root containers, missing limits, lying probes, blog-copied manifests for the wrong version. The value is operational judgment.

Duration	12 weeks · ~90 hours · live + hands-on	Format	Live online · lab-heavy · you will break clusters and fix them
Prerequisite	Comfortable on the Linux command line and with basic networking. No container or orchestration experience required. Some scripting ability helps.	Curriculum	11 modules, from containers-from-first-principles to operations, scale, cost and capstone
Certification	Aligned to the CNCF Kubernetes certification path — CKAD, CKA and CKS	Ideal for	Fresh IT graduates, sysadmins moving from VMs to containers, and teams migrating off retired ingress

Do You Actually Need Kubernetes?

An honest question this course answers before it sells you the cluster. For a handful of services, a managed container platform is usually the better answer, and Kubernetes' operational cost — the people, the upgrades, the on-call — is real and permanent. You'll learn to tell the difference and to justify *not* reaching for a cluster as confidently as running one — exactly the judgment that separates a platform engineer from someone who installs Kubernetes because it's on the CV.

Why Learn This When AI Writes Manifests?

AI writes Dockerfiles and Kubernetes manifests fluently — and produces exactly the artefacts that pass review and fail in production: containers running as root, images bloated with build toolchains, missing resource limits, probes that mark a broken pod healthy, and manifests copied from a blog for a different Kubernetes version. None of that fails in a demo; it fails under load, on call. The value is operational judgment, debugging under pressure, and knowing what the cluster will actually do. You'll use AI tooling as a reviewed accelerator and learn the failure patterns generated infrastructure code repeats.

Course Highlights

- Every module leads with what you build, operate or recover from — not what you "learn about"
- The Ingress NGINX retirement handled as urgent, funded migration work — staged cutover to the Gateway API
- GPU and accelerator scheduling for AI/ML workloads — declarative, topology-aware allocation

- Container images as a supply-chain artefact: distroless, non-root, scanned, SBOM, signed, provenance
- Break-fix labs under time pressure — crash-looping pods, DNS failures, failed rollouts, node pressure
- GitOps delivery, progressive rollout, and rollbacks that actually work
- Cost engineering on clusters that typically run far below their provisioned capacity
- An honest "do you actually need Kubernetes?" module — and AI tooling used as a reviewed accelerator

Hands-on Projects & Portfolio

- **Signed Supply-Chain Image** — minimal, non-root, scanned, with an SBOM and a provenance attestation
- **Multi-Service Deployment** — proper probes, resource limits, config and secrets
- **Ingress Migration Exercise** — move a workload off a retired ingress component onto the Gateway API with zero downtime
- **Stateful Workload + Restore** — persistent storage, backup, and a tested restore
- **GitOps Delivery Pipeline** — environment promotion and an automated rollback
- **Security Hardening Lab** — RBAC, network policy, pod security and admission policy on a deliberately open cluster
- **GPU-Scheduled Workload** — declarative accelerator allocation for an ML workload
- **Break-Fix Labs** — diagnose a crash-loop, an unschedulable pod, a DNS failure, node memory pressure and a failed rollout, under time pressure
- **Cost & Right-Sizing** — cut a cluster's provisioned spend with measured before/after
- **Capstone** — production-grade platform: secured, observable, cost-managed and defended

Tools & Stack

Docker · BuildKit · containerd · Kubernetes · kubectl · Helm · Kustomize · Argo CD · Gateway API · Cilium · cert-manager · Prometheus · Grafana · OpenTelemetry · Loki · Trivy · Cosign · Falco · Kyverno or OPA · KEDA · Velero · Terraform · AWS EKS · GKE · AKS · Git

Certification Alignment

- Developer track — Modules 1–6 and 9 map to the CNCF **Certified Kubernetes Application Developer (CKAD)** scope
- Administrator track — Modules 5–8 and 11 map to the **Certified Kubernetes Administrator (CKA)** scope
- Security track — Modules 4 and 10 map to the **Certified Kubernetes Security Specialist (CKS)** scope
- Presented as preparation and alignment — no official partnership or accreditation implied, and no exam outcome guaranteed

Career Outcomes

Containers and Kubernetes are the substrate almost all modern infrastructure runs on, which makes this the highest-leverage foundational skill in the DevOps track — and increasingly in AI infrastructure too, since GPU-backed training and inference are scheduled the same way. The differentiator at hire time is not writing manifests but operating clusters: debugging live incidents, planning upgrades, enforcing security and controlling cost. This course targets:

- DevOps Engineer · Platform Engineer · Site Reliability Engineer · Cloud Engineer
- Kubernetes Administrator · Infrastructure Engineer · Cloud Architect (with experience) · Backend engineer who owns their own deployment

Who This Course Is For

- Final-year students and fresh IT graduates targeting DevOps, cloud or platform roles
- Developers who want to own their applications through to production
- **System administrators and infrastructure engineers moving from VMs to containers** — a large and motivated group
- Backend, data and ML engineers whose workloads already run on someone else's cluster
- **Teams running unmaintained ingress components who need a migration plan** — urgent, funded work, covered end to end
- SREs and architects responsible for uptime, security and cloud spend

Module 1: Containers From First Principles

8 Hours

YOU BUILD

- A container assembled by hand from namespaces, cgroups and a filesystem — so the runtime stops being magic and every later debugging session becomes tractable

TOPICS

- What a container actually is: namespaces, cgroups, union filesystems and the container runtime
- Images versus layers; the registry model
- Why understanding this makes every later debugging session tractable rather than magical

Assessment: Quiz + Lab**Module 2: Building Production Images**

8 Hours

YOU SHIP

- A build pipeline producing minimal, distroless, non-root, multi-architecture images small and fast enough — and clean enough — that you'd be willing to sign them

TOPICS

- Modern build tooling and caching; multi-stage builds; choosing a base image; minimal and distroless images
- Running as non-root; reproducible builds; multi-architecture images
- Image size and startup time; a build pipeline that produces signable artefacts

Assessment: Quiz + Lab**Module 3: Local Development & Compose**

7 Hours

YOU SHIP

- A multi-container application running locally with real networking, volumes and health checks — development that matches production closely enough to trust

TOPICS

- Multi-container applications; networking and volumes
- Development versus production parity; debugging inside containers; health checks
- Container-based integration testing

Assessment: Quiz + Lab**Module 4: Container Security & Supply Chain**

8 Hours

YOU SHIP

- A hardened supply chain: an image scanned, an SBOM generated, a provenance attestation produced and the artefact signed and verified before it can be deployed

TOPICS

- Vulnerability scanning; SBOM generation; provenance attestations and image signing
- Base image lifecycle and patching; secrets that never belong in an image; registry access control
- The runtime hardening options

Assessment: Quiz + Lab**Module 5: Kubernetes Architecture**

8 Hours

YOU BUILD

- A cluster stood up from its components, then read like a practitioner — tracing one request through the API server, scheduler, kubelet and runtime until the reconciliation model is obvious

TOPICS

- The control plane and its components; the declarative reconciliation model that explains everything else
- The API server and resource model; scheduling; kubelet and the container runtime interface; etcd
- Reading cluster state like a practitioner rather than guessing

Assessment: Quiz + Lab

Module 6: Core Workloads**9 Hours****YOU SHIP**

- A multi-service workload with native sidecars, honest probes, quality-of-service-aware requests and limits, and graceful shutdown that drains cleanly under a rolling update

TOPICS

- Pods and the pod lifecycle; deployments, replica sets, stateful sets, daemon sets, jobs and cron jobs
- Native sidecar containers; init containers; configuration and secrets; resource requests and limits and quality-of-service classes
- Probes that tell the truth; graceful shutdown

Assessment: Quiz + Lab

Module 7: Networking, Routing & Service Exposure**9 Hours****YOU SHIP**

- The ingress migration: a workload moved off a retired ingress component onto the Gateway API with zero downtime, TLS managed, and network policy segmenting the namespace

TOPICS

- The cluster networking model; services and their types; DNS; the CNI layer; network policy for segmentation
- The Gateway API as the modern routing standard; TLS and certificate management
- Migrating off retired or unmaintained ingress components — assessment, alternatives and staged cutover

Assessment: Quiz + Lab

Module 8: Storage & Stateful Workloads**7 Hours****YOU SHIP**

- A stateful workload on persistent storage with a tested backup-and-restore and a safe stateful upgrade — plus a written call on when a database does not belong on the cluster

TOPICS

- Volumes; persistent volumes and claims; storage classes and dynamic provisioning; the container storage interface
- Running databases on Kubernetes and when you genuinely shouldn't
- Backup and restore; stateful upgrade patterns

Assessment: Quiz + Lab

Module 9: Configuration, Delivery & GitOps**8 Hours****YOU SHIP**

- A GitOps delivery pipeline: templated config promoted across environments by a reconciliation controller, with a canary rollout and an automated rollback that works when it must

TOPICS

- Templating and configuration management with Helm and Kustomize; environment promotion
- GitOps delivery with a reconciliation controller; progressive delivery including canary and blue/green; rollbacks that work
- Secret management integration; operators and custom resources for extending the platform

Assessment: Quiz + Lab

Module 10: Security, Multi-Tenancy & Governance

8 Hours

YOU SHIP

- A hardened, multi-tenant cluster: RBAC and workload identity, pod security standards, declarative admission policy and image-provenance enforcement, checked against a hardening checklist

TOPICS

- RBAC done properly; service accounts and workload identity; pod security standards
- Admission control including declarative policy; namespace and tenancy strategy; image provenance enforcement
- Audit logging; the cluster hardening checklist

Assessment: Quiz + Lab**Module 11: Operations, Scale, Cost & Capstone**

10 Hours

YOU SHIP

- Your capstone: a production-grade platform — observable, autoscaled, right-sized, GPU-capable, upgrade-planned, cost-managed and defended — plus a cost exercise cutting provisioned spend with measured before/after

TOPICS

- Observability with metrics, logs and distributed tracing; autoscaling horizontally, vertically and at the node level; in-place resource resizing and right-sizing practice
- GPU and accelerator scheduling for AI/ML workloads; cluster upgrades and deprecation planning; disaster recovery
- Capacity and cost optimisation on clusters that typically run far below provisioned capacity; incident response and debugging under pressure; capstone build and defense

Assessment: Capstone Project + Review