

LINUX FOUNDATION

Kubernetes Security Essentials (LFS260) Preparation Training

Delivered from Hacky Security's own curriculum, by instructors who also work our security operations centre. Classroom, online live, or at your office.

ADVANCED

32 HOURS

4 DAYS

CONTAINER ORCHESTRATION

At a glance

COURSE	Kubernetes Security Essentials (LFS260)
TRACK	Linux Foundation
DOMAIN	Cyber Security
LEVEL	Advanced
DURATION	32 hours (about 4 days)
DELIVERY	Classroom · Online live · On-site at your office
LANGUAGE	English
CLASS SIZE	Small cohorts — everyone gets lab time

WHAT IS INCLUDED

- ✓ Instructor-led delivery by practitioners who also work our SOC
- ✓ Hands-on lab access for the duration of the course
- ✓ Course materials and lab guides
- ✓ Hacky Security certificate of completion
- ✓ Post-course support by email

Who this is for

This is an advanced course. It suits people already working in or moving into Cyber Security who want structured, hands-on preparation rather than a video library. If you are unsure whether the level fits your team, send us the job titles and we will tell you honestly — including if a different course would serve you better.

Course outline

About this course map

Nine chapters, mapped one-for-one to the Linux Foundation's LFS260 Kubernetes Security Essentials course outline and taught in the same order, so nothing on the course is left uncovered.

LFS260 is the Linux Foundation's preparation course for the Certified Kubernetes Security Specialist exam. It is organised by the lifecycle of a cluster rather than by the exam's domain list — you decide what to install, install it, secure the API server, secure the network, then secure what runs on top — and it closes with a review chapter that walks the CKS domains explicitly. That ordering is deliberate and we keep it, because the exam is performance-based and thinking in cluster-lifecycle order is what actually gets tasks finished. A separate section below maps every chapter onto the published CKS domains and their weightings, so you can see the coverage both ways.

This is a hands-on course for a hands-on exam. The CKS gives you a terminal, a set of clusters and two hours, and asks you to change things. Nothing is multiple choice. The teaching is built for that — you break a cluster, you fix it, and you do it against the clock often enough that the muscle memory is there on the day.

Chapter 01 — Course Introduction

How the course and the lab environment work, and what you need in place before anything else.

- What the course covers, how it is structured, and how it relates to the CKS exam
- The lab environment: a Linux host, a cloud provider or local virtualisation, and what each needs to have available
- Assumed starting point — comfort with Linux, the command line, package managers and Git
- Why the course expects existing Kubernetes administration ability rather than teaching it
- Orientation to the tooling used throughout: kubectl, manifests, and working entirely from a terminal
- Setting expectations about exam format, allowed documentation and time pressure
- How the chapters map onto the cluster lifecycle, and why that order was chosen

Chapter 02 — Cloud Security Overview

The threat picture for container platforms, before any of it is made concrete.

- What changes about security when workloads are containers scheduled by a control plane
- The attack surface of a Kubernetes cluster: control plane, nodes, workloads, network and supply chain
- Multi-tenancy and the isolation guarantees containers do and do not provide
- Shared responsibility when the cluster is managed by a cloud provider rather than by you
- Threat modelling a cluster, and identifying which components are worth an attacker's time
- Security standards and benchmarks relevant to Kubernetes, including the CIS Benchmark
- Defence in depth across build, deploy and runtime, and why controls at only one stage fail
- Common misconfigurations that turn a working cluster into an exposed one

Chapter 03 — Preparing to Install

The decisions made before installation that determine what is possible to secure afterwards.

- Choosing a cluster topology and understanding the security consequences of each
- Node operating system selection, minimising the host footprint and reducing installed attack surface
- Verifying platform binaries and images before they are deployed

- Certificate authority planning and how cluster trust is established
- Network design decisions taken up front, including CNI selection and its effect on network policy support
- Bootstrapping credentials, tokens and the risks in the join process
- Host-level hardening, kernel configuration and least-privilege identity for node access
- Planning for upgrades, because an un-upgradeable cluster becomes a vulnerable one

Chapter 04 — Installing the Cluster

Building a cluster whose defaults are defensible.

- Deploying a cluster and reviewing what the installer chose on your behalf
- Control plane component configuration and the flags that carry security meaning
- etcd security: encryption at rest for cluster data, access restriction and backup handling
- Kubelet configuration, authentication and authorisation, and disabling anonymous access
- Reviewing the installed configuration against the CIS Benchmark, and using kube-bench to do it repeatably
- Protecting node metadata endpoints from workloads that should never reach them
- Certificate management, rotation and the lifetimes involved
- Upgrading Kubernetes to remove known vulnerabilities, and doing it without losing the cluster

Chapter 05 — Securing the kube-apiserver

The API server is the cluster, so this is where most of the meaningful access control lives.

- The request path through the API server: authentication, then authorisation, then admission
- Authentication methods available, and eliminating the weak ones
- Role-based access control in practice — roles, cluster roles and the bindings that connect them
- Minimising exposure through RBAC rather than granting cluster-admin and moving on
- Service accounts: disabling defaults, limiting token automounting and keeping permissions narrow on new ones
- Restricting access to the Kubernetes API itself, including network-level restriction
- Admission controllers, and using admission control to enforce policy that RBAC cannot express
- Auditing API server access, configuring an audit policy, and reading the resulting log
- Diagnosing why a request was allowed or denied

Chapter 06 — Networking

Controlling what can talk to what, inside and outside the cluster.

- The Kubernetes network model, and why every pod being reachable is the default problem
- Network policies for ingress and egress, and restricting access at cluster level
- Designing default-deny policies and adding exceptions deliberately
- CNI plugin behaviour, and confirming your plugin actually enforces the policies you write
- Ingress objects configured correctly with TLS, and certificate handling at the edge
- Minimising external access to the cluster network and its services
- Service meshes and pod-to-pod encryption with tools such as Cilium and Istio
- Protecting cluster-internal endpoints and services that were never meant to be public

Chapter 07 — Workload Considerations

Securing what actually runs, from image build through to the running container.

- Minimising base image footprint and the vulnerabilities that come with unnecessary packages
- Understanding and securing the supply chain: software bills of materials, CI/CD, artifact repositories
- Restricting deployment to permitted registries, and signing and validating artifacts
- Static analysis of workloads and images with tools such as Kubesec and KubeLint
- Pod security standards and applying the right level to the right namespace
- Managing Kubernetes secrets, limiting their exposure and avoiding them in manifests and images
- Isolation techniques: multi-tenancy patterns and sandboxed container runtimes
- Kernel hardening for workloads with AppArmor and seccomp profiles
- Resource limits, security contexts and dropping capabilities a container does not need

Chapter 08 — Issue Detection

Finding out that something is wrong while it is still happening.

- Behavioural analytics to detect malicious activity in running workloads
- Runtime security tooling and the syscall-level signals it works from
- Detecting threats across infrastructure, applications, network, data, users and workloads
- Ensuring container immutability at runtime, and alerting when a container is modified
- Using Kubernetes audit logs to monitor access and reconstruct what an identity did
- Logging architecture for a cluster, and retaining what you will need during an investigation
- Investigating an incident and identifying the phases of an attack and the actor behind it
- Responding within the cluster: isolating a workload, revoking access and recovering state

Chapter 09 — Domain Review

A structured pass back over the material, organised the way the exam is organised.

- Revisiting each CKS domain in turn and locating the course material that covers it
- Identifying gaps between what you have practised and what the domain expects
- Consolidating the command patterns that recur across tasks
- Practising against the clock, since time is the binding constraint in the exam
- Working efficiently with the permitted Kubernetes documentation during the exam
- Common task types and the fastest correct route through each
- Verifying your own work before moving on, because unverified changes score nothing
- Building a personal revision list from what you actually got wrong

How this maps to the CKS exam domains

The CKS curriculum is published by CNCF with six domains and fixed weightings. LFS260's chapters cut across them, so this is the crosswalk we teach to.

- Cluster Setup, 15% — chapters 3, 4 and 6: network policy, CIS benchmark review of Kubernetes components, Ingress with TLS, node metadata protection, binary verification
- Cluster Hardening, 15% — chapters 4 and 5: RBAC, service account discipline, restricting API access, upgrading to remove vulnerabilities

- System Hardening, 10% — chapters 3 and 7: minimising host OS footprint, least-privilege identity and access management, minimising external network access, AppArmor and seccomp
- Minimize Microservice Vulnerabilities, 20% — chapters 6 and 7: pod security standards, secrets management, isolation and sandboxing, pod-to-pod encryption
- Supply Chain Security, 20% — chapter 7: base image minimisation, understanding and securing the supply chain, permitted registries, artifact signing, static analysis of workloads and images
- Monitoring, Logging and Runtime Security, 20% — chapter 8: behavioural analytics, threat detection across the environment, attack phase investigation, container immutability, audit logs

Chapter 2 underpins all six domains, and chapter 9 revisits them in the exam's own order.

Who should take this

Kubernetes administrators taking on security responsibility, platform and DevOps engineers who run clusters in production, cloud security engineers who now have to secure workloads rather than infrastructure, and SREs who are being asked to sign off on cluster hardening.

The Linux Foundation expects candidates to be proficient with Kubernetes and CKA-certified before starting. That is not decoration: the exam gives you no time to work out how to do routine administration, and the course begins well past that point.

You will need strong Linux fundamentals, real command-line fluency, familiarity with package managers, Git and GitHub, and access to a Linux machine and either a cloud provider account or local virtualisation for lab work.

The exam

The Certified Kubernetes Security Specialist is set and awarded by the Linux Foundation in partnership with CNCF. It is performance-based: you are given a remote terminal onto live clusters and asked to complete tasks. There are no multiple-choice questions.

- Performance-based, solving multiple tasks from a command line running Kubernetes
- Two hours
- A score of 67% or above is required to pass
- Online and remotely proctored, with webcam and screen monitoring
- Passing the Certified Kubernetes Administrator exam is a prerequisite, and must be done before you attempt the CKS
- The certification is valid for two years
- Exam content is updated on a quarterly cycle to follow the Kubernetes release cadence
- The exam registration is 445 USD and includes one free retake

The exam registration is purchased separately from the Linux Foundation. We can arrange it on your behalf and send you the code, and you book your own exam window. Hacky Security is not a Linux Foundation training partner, and this course is taught from our own material rather than official Linux Foundation courseware.

How we deliver it

Three days, classroom or online live, or at your office for a private group. Almost all of it is hands-on against clusters we provide, one per delegate, with full administrative access — including the freedom to break them, which is most of the point.

Chapter checkpoints throughout, and a timed performance-based mock on the final afternoon, built to the CKS domain weightings and sat under exam conditions with only the permitted Kubernetes documentation available. It

is followed by a walkthrough of where time was lost, which for most candidates matters more than what they did not know.

Mapped to the LFS260 Kubernetes Security Essentials course outline and to the CNCF CKS exam curriculum.
Taught from Hacky Security's own curriculum by instructors who run and secure production Kubernetes for a living.

Sitting the official exam

Exam voucher sold separately. We buy it from the certification body on your behalf and send you the code — you book your own exam date. This course is Hacky Security's own preparation training. The certification itself is awarded by the certification body, not by us.

Investment

FROM

₹90,000

per person

INR	₹90,000 per person
USD	\$1,079 per person
GBP	£849 per person
AED	AED 3,999 per person

Group, corporate and repeat-cohort pricing on request. Prices are reviewed quarterly and exclude the exam voucher.

Book this course

Tell us how many people, where, and when — we will come back with dates and a quote within one working day.

+91-8091900310 · info@hackysecurity.com

<https://hackysecurity.com/courses/hacky-kubernetes-security-essentials-course/>

CHANDIGARH · SHIMLA · DELHI · DUBAI · LONDON

Independent preparation training delivered from Hacky Security's own curriculum. Hacky Security Private Limited is not affiliated with or endorsed by the certification body named in this document. Certification names and codes are the trademarks of their respective owners and are used only to describe what this course prepares you for. Prices, durations and outlines are indicative and may change; the quotation we send you is the binding one. Issued 12 September 2026.