DevOpsSchool

Course curriculum · November 2026

LEVEL 03 OF 04 · INTERMEDIATE

Docker Security Training Intermediate Training

Detect runtime threats with Falco, add code-level scanning, and build hardened, audited images that follow Docker security best practices.

LIVE ONLINE **Max 8-10 hours**	CORPORATE (DAY-WISE) **3 days · Days 1-3**	SELF-PACED VIDEO **1-year LMS access**
BATCH SIZE **Max 10 learners**	NEXT BATCHES **1st of every month · weekends: first Saturday**	CERTIFICATE **Verifiable certificate of completion**

Foundation	Essential	Intermediate	Advanced
1 day · Included	2 days · Included	3 days · This curriculum	5 days · Next step

INSIDE THIS CURRICULUM

What you will achieve	Outcomes and level focus
Who should attend	Audience and prerequisites
Agenda	37 topics · Foundation to Intermediate
Hands-on labs	3 labs
Formats and fees	Live, 1:1, self-paced, corporate
Trainer and reviews	Rajesh Kumar
Enrolment	Batches, payment, contacts

WHY DEVOPSSCHOOL

- Rated 4.6 on Google
- 1,500+ verifiable certificates issued in the public registry
- Trainer Rajesh Kumar: 20 years in industry
- Small batches: max 10 learners, every session recorded

Course page: www.devopsschool.com/courses/docker/docker-security-training-intermediate-training.html
Talk to us: WhatsApp +91 99057 40781 · +1 (469) 756-6329 · contact@DevOpsSchool.com · reply within 1 hour

OUTCOMES

What you will achieve

By the end of the Docker Security Training course path, you will be able to:

1

Apply kernel-level controls

Use namespaces, cgroups, capabilities, seccomp, AppArmor and SELinux to isolate containers on a Docker host.

2

Build trusted, hardened images

Write secure Dockerfiles, scan and audit images, and pull only signed images from trusted registries.

3

Monitor and respond at runtime

Detect runtime threats with Falco and logging, and respond to container incidents with a repeatable process.

WHAT THE INTERMEDIATE LEVEL ADDS

Application runtime, code security and image hardening

Hands-on: Harden an image and watch runtime. Rebuild an application image to a hardened, non-root, read-only standard, scan it, and trigger Falco alerts by spawning a shell inside the container.

WHO SHOULD ATTEND

- Regular Docker Security Training users who want to automate, integrate and manage it at a deeper level.
- Specialists who support Docker Security Training for colleagues or customers.
- Leads preparing Docker Security Training for wider use across projects.

BEFORE YOU START

- The Essential level, or equivalent hands-on experience.
- You can already: Apply namespaces, cgroups, capabilities, seccomp and AppArmor controls to containers and add logging, monitoring and signed image pulls.
- Basic computer skills and a laptop with a stable internet connection for the live sessions and labs.
- This curriculum includes every earlier level, so it is complete on its own.

AGENDA · FOUNDATION TO INTERMEDIATE

Docker Security Training Intermediate Training agenda

37 topics. Each level builds on the one before it, so this curriculum lists every earlier level in full. Days are for corporate (day-wise) delivery; live online batches cover the same agenda in max 8-10 hours.

DAY 1 Foundation: Docker security challenges and architecture

Included from Foundation

1. Understanding Docker security challenges: shared kernel, image supply chain, daemon privileges and container sprawl
2. Protecting your Docker containers: defence in depth across host, daemon, image, application runtime and code
3. Understanding Docker architecture: Docker daemon, CLI, containerd, runc and the Docker Registry from a security view
4. Kernel namespaces: pid, net, mnt, ipc, uts and user namespaces and the isolation each one provides
5. Control groups: CPU, memory, pids and block I/O controllers and how Docker applies them
6. Linux kernel capabilities (libcap): capsh and getpcaps, the default Docker capability set
7. SELinux, AppArmor and seccomp: mandatory access control and system call filtering in plain terms
8. Docker security levels: host and runtime security, application runtime security and code level security
9. Installing Docker securely: official repositories, package signature checks and the rootless install option
10. First posture checks: docker info security options, docker inspect for Privileged, CapAdd and SecurityOpt
11. Running in super privileged mode: what --privileged grants, device access and why to avoid it

DAY 2 Essential: Host and runtime security controls

Included from Essential

12. Kernel namespaces and security aspects: user namespace remapping, risks of --pid=host and --ipc=host
13. Control groups and security aspects: limit resource utilization, manage memory and CPU usage with --memory, --cpus and --pids-limit
14. Docker daemon attack surface: root daemon, docker.sock permissions, remote API exposure and TLS on port 2376
15. Linux kernel capabilities and security: --cap-drop ALL, adding only what is needed, no-new-privileges
16. Seccomp security profiles for Docker: the default profile, custom JSON profiles and --security-opt seccomp
17. AppArmor security profiles for Docker: the docker-default profile and custom profiles loaded with apparmor_parser
18. Other kernel security features: SELinux labels with :z and :Z, masked and read-only /proc paths
19. Docker Content Trust signature verification: DOCKER_CONTENT_TRUST=1, signing and pulling signed tags
20. Docker Logging: logging drivers, log rotation and daemon events as an audit trail
21. Docker Monitoring: docker stats and docker events, cAdvisor and Prometheus metrics to spot abnormal behaviour
22. Container privileges: USER in images, --user at runtime and rootless Docker mode
23. Restart policy: set an on-failure container restart policy with a maximum retry count

DAY 3 Intermediate: Application runtime, code security and image hardening

New at this level

24. Threats during application runtime: shells in containers, cryptominers, privilege escalation and lateral movement
25. Why Docker application runtime security matters: zero-days and misconfigurations that image scans cannot catch
26. Tools for Docker application runtime: Sysdig Falco installation, default rules and writing custom Falco rules
27. Runtime enforcement: SecComp, AppArmor and SELinux profiles generated from application behaviour
28. What is code level security: why code level security matters and shifting checks left into container builds

29. Tools for Docker code level security: SonarQube quality gates and Fortify static analysis
30. Dockerfile security: non-root USER, pinned base images, no secrets in layers, COPY over ADD and HEALTHCHECK
31. Docker image hardening standard: minimal or distroless bases, removing shells and package managers, multi-stage builds
32. Use trusted images: Docker Official Images, Verified Publishers and approved private registries
33. Auditing Docker images: docker history, Trivy or Docker Scout vulnerability scans and SBOM review
34. Network exposure: do not expose unused ports, do not share the host network namespace, limit unbounded network access from containers
35. Container access: do not run SSH within containers, use docker exec and centralised logs instead
36. Read-only containers: mount the container root filesystem as read only with tmpfs for writable paths
37. Use overlay networks: encrypted overlay networks to protect traffic between hosts

PRACTICE

Hands-on labs

You build these yourself during the course, guided by the trainer.

FOUNDATION

Inspect container isolation on a host

Start containers with default and privileged settings, compare their namespaces, capabilities and security options, and record the differences.

ESSENTIAL

Lock down a running container

Run a web container with dropped capabilities, a custom seccomp profile, AppArmor, resource limits and a non-root user, then prove each control works.

INTERMEDIATE

Harden an image and watch runtime

Rebuild an application image to a hardened, non-root, read-only standard, scan it, and trigger Falco alerts by spawning a shell inside the container.

HOW YOU CAN LEARN

Formats and fees

The same agenda in four formats. Fees are in Indian rupees as of November 2026; the course page shows them in your currency.

FORMAT	DURATION	WHAT YOU GET	FEE
Live & interactive (online)	Max 8-10 hours	Instructor-led public batch of max 10 learners with demos, labs, Q&A and assignment review. Weekday batches start on the 1st of every month, weekend batches on the first Saturday of every month.	**INR 34,999**
1:1 live training (online)	Max 8-10 hours, one-to-one	The same agenda, taught one-to-one by the trainer. Starts immediately, without waiting for the next batch; sessions planned around your working hours, at your pace.	**INR 99,999**
Self-paced video	1-year LMS access	All four Docker Security Training levels, plus 20+ more courses and 50+ tools, at your own pace.	**INR 833/month** billed yearly at INR 9,996
Corporate (day-wise)	3 days for this level	A private batch for your team, on demand. Foundation 1 day, Essential 2, Intermediate 3, Advanced 5 (cumulative).	**Quote on request**

ASSESSMENT AND CERTIFICATE

Your progress is checked through hands-on labs and assignments at every level. On completion you receive a DevOpsSchool certificate of completion with a unique credential ID that anyone can verify at devopsschool.com/certificates.

INCLUDED WITH EVERY LIVE BATCH

- Every live session is recorded and stays in your LMS for a year.
- Can't attend your batch? Move to next month's batch.
- Free demo class before you pay.
- Free 10-15 minute call with an advisor or the trainer to check the course fits you.

YOUR TRAINER

Learn from Rajesh Kumar

Principal Architect, trainer and mentor in DevOps, SRE, DevSecOps and platform engineering, with 20 years across 8+ software MNCs, including Staff DevOps/SRE Engineer for a leading Japanese mobile-payment platform and Principal DevOps Architect and Manager in Europe.

Profile: www.rajeshkumar.xyz · Videos: youtube.com/@TheDevOpsSchool

20 years in industry	**200+** corporate trainings
10,000+ engineers mentored	**70+** organisations coached

REVIEWS

What learners say

Rated 4.6 on Google · more reviews at devopsschool.com/testimonials and in our video testimonials.

> “Rajesh is very good trainer. Rajesh was able to resolve our queries and question effectively. We really liked the hands-on examples covered during this training program.”
>
> Indrayani, India

> “The training was very useful and interactive. Rajesh helped develop the confidence of all.”
>
> Abhinav Gupta, Pune

START YOUR DOCKER SECURITY TRAINING TRAINING

Join the next batch

Weekday batches start on the 1st of every month and weekend batches on the first Saturday of every month, with max 10 learners each. 1:1 training can start immediately; corporate batches run on demand. We reply within 1 hour.

Pay by UPI, credit and debit cards, net banking or PayPal. Before you pay, read our refund policy. All levels of this course: www.devopsschool.com/courses/docker/

ENROL OR RESERVE A SEAT
Docker Security Training Intermediate Training

WHATSAPP · INDIA
+91 99057 40781

WHATSAPP · USA
+1 (469) 756-6329

EMAIL
contact@DevOpsSchool.com

Next step after this level: Docker Security Training Advanced Training - Run a complete open-source Docker security toolchain, automate it in CI/CD and govern container security across teams and environments.