
DevOps im Unternehmen.

CI/CD, Container und Automatisierung - der Praxisleitfaden für Teams, die schneller und zuverlässiger liefern wollen

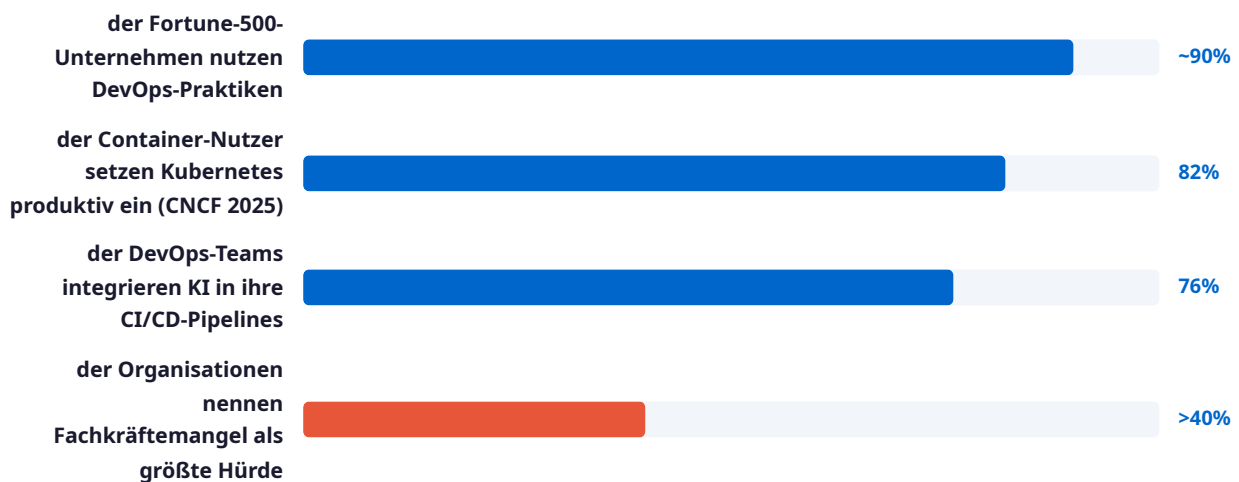
DOCKER · KUBERNETES · CI/CD · TERRAFORM · ANSIBLE · GITOPS · OBSERVABILITY · DEVSECOPS

Warum DevOps?

Aktuelle Zahlen zeigen: Wer DevOps beherrscht, liefert schneller und stabiler

Der DORA State of DevOps Report 2024 zeigt: Elite-Teams deployen mehrfach täglich, erholen sich in unter einer Stunde von Ausfällen und erreichen Change-Failure-Rates von nur 5%. Gleichzeitig setzen 83% der IT-Entscheider auf DevOps, um Geschäftswert zu steigern. Die Frage ist nicht mehr ob DevOps, sondern wie gut.

Was die Daten sagen



Die vier DORA-Metriken - dein Kompass

DF

Deployment Frequency
Wie oft wird deployt?

LT

Lead Time for Changes
Commit bis Produktion

CFR

Change Failure Rate
Anteil fehlgeschlagener Änderungen

MTTR

Mean Time to Restore
Wiederherstellungszeit

182x

Elite-Teams deployen laut DORA 2024 über 100-mal häufiger als Low-Performer

<5%

Change Failure Rate bei Elite-Performern (DORA 2024)

Marktentwicklung

Der globale DevOps-Markt wächst von 16,1 Mrd. USD (2025) auf voraussichtlich 51,4 Mrd. USD bis 2031 - das entspricht einer jährlichen Wachstumsrate von über 21%. DevOps ist kein Trend, sondern Industriestandard.

DevOps-Kultur und Prinzipien

CALMS - das Framework für nachhaltige Veränderung

DevOps ist kein Tool und kein Job-Titel. Es ist eine Kultur, die Entwicklung und Betrieb zusammenbringt. Das CALMS-Modell beschreibt die fünf Säulen, auf denen erfolgreiche DevOps-Transformationen stehen.

- C Culture - Kultur des Vertrauens**

Silos zwischen Dev und Ops aufbrechen. Gemeinsame Verantwortung für den gesamten Lebenszyklus einer Anwendung. Blameless Post-Mortems statt Schuldzuweisungen. Laut DORA 2025 ist die größte Hürde für Cloud-Native-Adoption nicht Technologie, sondern Kultur (47% der Befragten).
- A Automation - Automatisierung als Standard**

Alles, was wiederholt wird, wird automatisiert: Builds, Tests, Deployments, Infrastruktur-Provisionierung. Manuelle Schritte sind Fehlerquellen und Engpässe. CI/CD-Pipelines sind das Rückgrat moderner Softwarelieferung.
- L Lean - Verschwendung eliminieren**

Kleine Batch-Größen, kurze Feedback-Zyklen, Work-in-Progress-Limits. Lean-Prinzipien aus der Fertigung auf die Softwareentwicklung übertragen. Weniger Handoffs, schnellere Durchlaufzeiten.
- M Measurement - Messen statt raten**

Ohne Metriken keine Verbesserung. DORA-Metriken, Cycle Time, Error Budgets und SLOs geben Orientierung. Datengetriebene Entscheidungen statt Bauchgefühl.
- S Sharing - Wissen teilen**

Dokumentation, interne Tech-Talks, gemeinsame Runbooks. Wissen darf nicht in einzelnen Köpfen stecken bleiben. Sharing reduziert den Bus-Faktor (das Risiko, wenn Schlüsselwissen bei einer Person liegt) und beschleunigt Onboarding.



Seminartipp: DevOps-Kultur verstehen

LPIC-OT DevOps Tools Engineer Komplettausbildung - die umfassende Ausbildung verbindet Theorie, Kultur und Praxis in einem Kurs.

Container-Grundlagen

Docker, Podman und der OCI-Standard

Container haben die Art verändert, wie Software paketiert, ausgeliefert und betrieben wird. Container garantieren: Was auf deinem Laptop läuft, läuft auch in Produktion. Über 90% der Organisationen setzen Container produktiv ein oder evaluieren sie aktiv.

Warum Container?

- Reproduzierbare Umgebungen - Schluss mit "Works on my machine"
- Schneller Start, geringer Overhead (Sekunden statt Minuten)
- Immutable Infrastructure - keine Konfigurationsdrift
- Portabilität zwischen Cloud, On-Premises und Hybrid
- Microservices-fähig durch Isolation auf Prozessebene

Docker vs. Podman

- **Docker:** De-facto-Standard, großes Ökosystem, Docker Compose für Multi-Container
- **Podman:** Daemonless, Rootless, Drop-in-Replacement für Docker CLI
- **OCI-Standard:** Open Container Initiative sorgt für Kompatibilität zwischen Runtimes
- **Registries:** Docker Hub, GitHub Container Registry, Harbor, Quay.io

Docker Schulung und Container-Seminare

- 01 Docker Grundkurs - Der komplette Einstieg**
Images, Container, Volumes, Networking, Docker Compose
- 02 Einführung in Container mit Podman**
Rootless Container, Pods, Systemd-Integration, Podman Compose
- 03 Linux Container Workshop - Docker und Podman Intensivkurs**
Vertiefter Vergleich beider Runtimes, Security, Best Practices
- 04 KI-Apps containerisieren mit Podman, Docker & Kubernetes**
GPU-Workloads, Model Serving, ML-Pipelines in Containern
- 05 Docker und Java - Container für Java-Applikationen**
JVM-Tuning in Containern, Multi-Stage Builds, Spring Boot

Kubernetes

Die Plattform für Container-Orchestrierung im Unternehmen

82% der Container-Nutzer setzen Kubernetes produktiv ein (CNCF 2025). Kubernetes ist das Betriebssystem der Cloud - und zunehmend auch die Plattform für KI-Workloads. 66% der Organisationen nutzen Kubernetes für Inference-Workloads.

Kernkonzepte

- **Pods:** Kleinste Deployment-Einheit, ein oder mehrere Container
- **Deployments:** Deklarative Updates, Rolling Releases, Rollbacks
- **Services:** Stabile Netzwerk-Endpunkte, Load Balancing
- **Helm:** Paketmanager für Kubernetes-Applikationen
- **Operators:** Automatisierte Verwaltung komplexer Anwendungen

Managed vs. Self-Hosted

- **EKS:** Amazon Elastic Kubernetes Service
- **AKS:** Azure Kubernetes Service
- **GKE:** Google Kubernetes Engine
- **OpenShift:** Red Hats Enterprise-Kubernetes
- **Rancher/RKE2:** Multi-Cluster-Management
- **Self-Hosted:** Volle Kontrolle, höherer Aufwand

Kubernetes Schulung bei cmt

Seminar	Fokus	Zertifizierung
Kubernetes Grundkurs	Einstieg, Pods, Services, Deployments	-
CKA Training	Cluster-Administration, Troubleshooting	CKA
CKAD Training	Applikationsentwicklung auf Kubernetes	CKAD
CKS Training	Cluster Security, Supply Chain, Runtime	CKS
LFS458 Kubernetes Administration	Linux Foundation Curriculum	CKA
Kubernetes für Plattform-Teams	Plattform-Betrieb, Multi-Tenancy	-

Weitere Kubernetes-Seminare

[Air-Gapped Kubernetes](#), [Amazon EKS](#), [Azure Container Apps](#), [Rancher RKE2](#) und [LFS258 + CKA Bundle](#).

CI/CD-Pipelines

Automatisiert von Commit bis Produktion

Continuous Integration und Continuous Delivery sind das Herz jeder DevOps-Praxis. Eine gut gebaute Pipeline erkennt Fehler in Minuten, nicht Tagen - und liefert Änderungen zuverlässig in Produktion.

Pipeline-Stufen

- 1 Build**
Code kompilieren, Dependencies auflösen, Container-Image bauen
- 2 Test**
Unit Tests, Integration Tests, Security Scans, Linting
- 3 Deploy**
Staging, Canary, Blue/Green, Produktions-Rollout
- 4 Verify**
Smoke Tests, Health Checks, Observability

Plattformen im Vergleich

- **GitLab CI:** Integriert in GitLab, YAML-basiert, Auto DevOps
- **GitHub Actions:** Marketplace mit 20.000+ Actions, Matrix Builds
- **Jenkins:** Open Source, Plug-in-Ökosystem, Pipeline-as-Code
- **Azure DevOps:** Tiefe Microsoft-Integration, YAML & Classic Pipelines
- **Jenkins X:** Cloud-Native CI/CD für Kubernetes (Projekt archiviert, Konzepte leben in anderen Tools weiter)

CI/CD-Seminare bei cmt

- 01 Git - Einführungskurs in Git, GitLab und CI/CD**
Versionierung, Branching, Merge Requests, GitLab CI Pipelines
- 02 Jenkins Grundkurs / Jenkins Kompaktkurs**
Pipelines erstellen, Plug-ins, Blue Ocean, Pipeline-as-Code
- 03 GitHub Actions: Automate your Workflow**
Workflows, Custom Actions, Secrets, Matrix Strategies
- 04 CI/CD with Jenkins X (LFS268)**
Kubernetes-native CI/CD, Preview Environments, GitOps

Infrastructure as Code

Terraform, Ansible und GitOps - Infrastruktur versioniert und automatisiert

Infrastructure as Code (IaC) bedeutet: Infrastruktur wird nicht geklickt, sondern geschrieben, versioniert und automatisiert ausgerollt. GitOps-Adoption hat laut CNCF End User Survey 2025 die 77%-Marke erreicht, und 81% der Adopter berichten von höherer Infrastruktur-Zuverlässigkeit und schnellerem Rollback.

Provisioning: Terraform

- **Deklarativ:** Beschreibe den Soll-Zustand, Terraform berechnet den Plan
- **Multi-Cloud:** AWS, Azure, GCP, VMware, Kubernetes
- **State Management:** Zustandsverwaltung, Drift Detection
- **Module:** Wiederverwendbare Infrastruktur-Bausteine
- **OpenTofu:** Open-Source-Fork, Community-getrieben

Konfiguration: Ansible

- **Agentless:** Kein Agent auf Zielsystemen nötig
- **Idempotent:** Playbooks können beliebig oft laufen
- **YAML:** Leicht lesbar, flache Lernkurve
- **Roles & Collections:** Modularer Aufbau, Ansible Galaxy
- **AAP:** Ansible Automation Platform für Enterprise

GitOps: Git als Single Source of Truth

Bei GitOps beschreibt ein Git-Repository den gewünschten Zustand der gesamten Infrastruktur und Applikationen. Operator-Tools wie **ArgoCD** oder **Flux** gleichen den Cluster kontinuierlich mit dem Repository ab. Jede Änderung läuft über Pull Requests, ist nachvollziehbar und prüfbar. Laut CNCF-Umfrage 2025 ist Argo CD die meistgenutzte GitOps-Lösung.

IaC-Seminare bei cmt

- 01 Terraform mit AWS**
Einführung, Kompaktkurs und Aufbaukurs verfügbar
- 02 Terraform mit Azure**
Azure Provider, ARM-Migration, Remote State in Azure Storage
- 03 IaC mit KI: Terraform & GitOps sauber skalieren**
KI-gestützte IaC-Entwicklung, Policy-as-Code, GitOps Workflows
- 04 Ansible Kompaktkurs**
Playbooks, Roles, Inventory, Vault, Ansible Automation Platform
- 05 Red Hat Automation with Ansible (AU294)**
Offizieller Red-Hat-Kurs, RHEL-Automatisierung, AAP

Observability & Monitoring

Sehen, was in Produktion passiert - bevor es der Kunde merkt

Observability geht über klassisches Monitoring hinaus. Es geht nicht nur darum, ob ein System läuft, sondern warum es sich so verhält. Die drei Säulen: Metriken, Logs und Traces - zunehmend vereint durch OpenTelemetry als offenen Standard.

Metriken

Zahlen über Zeit: CPU, Memory, Request Rate, Error Rate, Latenz. Prometheus ist der Standard.

Logs

Strukturierte Events: Was ist passiert? Loki, Elasticsearch, Fluentd für Aggregation.

Traces

Request-Pfade durch verteilte Systeme. Jaeger, Tempo, OpenTelemetry.

Der Observability-Stack



Seminartipp

01 Prometheus, Grafana und OpenTelemetry für Kubernetes

Metriken sammeln, Dashboards bauen, Alerting konfigurieren, Traces korrelieren

Prometheus Queries (PromQL), Grafana-Dashboards, OpenTelemetry Collector, Alertmanager, ServiceMonitors in Kubernetes

Praxis: Kompletter Observability-Stack auf einem echten Kubernetes-Cluster

SRE und Incident Management

Für Teams, die Observability mit SRE-Praktiken verbinden wollen: [SRE Deep Dive: Incident Automation](#) - Error Budgets, Runbook Automation, Post-Incident Reviews.

Cloud Native & Platform Engineering

Microservices, Plattform-Teams und Developer Experience

Cloud Native bedeutet: Anwendungen werden für die Cloud entworfen, nicht nur dorthin migriert. Laut Gartner setzen 2026 rund 80% der großen Software-Organisationen auf Platform Engineering. Der Schlüssel: Plattform-Teams, die Entwicklern eine Self-Service-Infrastruktur bieten, ohne dass jedes Team Kubernetes-Experte sein muss.

Microservices-Architektur

- Unabhängig deploybare, lose gekoppelte Services
- API-First-Design, Event-Driven Architecture
- Jeder Service hat seine eigene Datenbank
- Service Mesh (Istio, Linkerd) für Observability und Traffic Management
- Herausforderung: Distributed Tracing, Datenkonsistenz

Platform Engineering

- Internal Developer Platform (IDP) als Produkt
- Self-Service für Infrastruktur, CI/CD, Environments
- Paved Paths statt Vorschriften
- Deutlich höhere Deployment-Frequenz bei reifen Plattformen (DORA 2024)
- 40-50% weniger Cognitive Load für Entwickler

Cloud-Native-Seminare bei cmt

- 01 Vom Monolith zum Cloud Native System**
Migrationsstrategie, Strangler Fig Pattern, Domain-Driven Design
- 02 Container, Microservices & Plattform Engineering**
Plattform-Design, Developer Portals, Golden Paths
- 03 Cloud Native Infrastruktur & Betrieb: Plattformfähig**
Kubernetes-Plattform aufbauen, Multi-Tenancy, Networking
- 04 Microservices mit Spring / Cloud-Native Go**
Sprachspezifische Microservice-Entwicklung mit Spring Boot oder Go
- 05 FinOps & Cloud Native Effizienz**
Cloud-Kosten messen, optimieren und steuern

Security in DevOps (DevSecOps)

Shift Left - Security früh einbauen statt spät aufdecken

87% der Organisationen verfolgen einen Shift-Left-Ansatz bei Security, 80% nutzen Container-Security in ihren DevSecOps-Pipelines, und 82% setzen SBOMs in ihren Build-Pipelines ein. DevSecOps ist kein Add-on mehr, sondern integraler Bestandteil jeder CI/CD-Pipeline.

Die vier Säulen von DevSecOps

- 01 Shift Left - Security in der Pipeline**
SAST, DAST und SCA direkt im Build-Prozess. Sicherheitslücken werden beim Commit gefunden, nicht erst beim Pen-Test vor dem Release.
- 02 Container Security**
Image Scanning, signierte Images, minimale Base Images (Distroless, Alpine), Runtime Security mit Falco oder Sysdig. 87% der Container in Produktion enthalten High/Critical CVEs - Scanning ist Pflicht.
- 03 Supply Chain Security**
SBOM (Software Bill of Materials) für Transparenz über alle Abhängigkeiten. SLSA-Framework für Provenance Tracking. Sigstore für kryptografische Signierung von Artefakten. Regulatorischer Druck steigt.
- 04 Policy as Code**
OPA/Gatekeeper, Kyverno oder Sentinel für automatisierte Compliance-Prüfungen. Policies werden versioniert und getestet wie Applikationscode.

Security-Seminare bei cmt

- 01 CKS: Certified Kubernetes Security Specialist**
Cluster Hardening, Supply Chain, Runtime Security
- 02 LFS460: Kubernetes Security Fundamentals**
Linux Foundation, RBAC, Network Policies, Secrets
- 03 GitHub Advanced Security (GH-500)**
Code Scanning, Secret Scanning, Dependabot, SBOM
- 04 Cloud Native Security & Governance**
Zero Trust, Policy as Code, Compliance Automation

Red Hat OpenShift

Enterprise-Kubernetes mit integrierter CI/CD und Security

OpenShift ist Red Hats Enterprise-Distribution von Kubernetes. Es bringt CI/CD (Tekton Pipelines), eine integrierte Registry, Entwickler-Console und strikte Security-Defaults (SCCs) mit. Viele Unternehmen in regulierten Branchen setzen auf OpenShift als Plattform-Grundlage.

OpenShift-Seminare bei cmt

- 01 DO180 Red Hat OpenShift Administration I**
Cluster-Setup, Basis-Administration, Operator-Lifecycle
- 02 DO280 Red Hat OpenShift Administration II**
Multi-Tenancy, Networking, Storage, Cluster-Scaling
- 03 DO188 OpenShift Development I: Introduction to Containers**
Container mit Podman, Deployment auf OpenShift, S2I Builds
- 04 DO288 Red Hat OpenShift Developer II**
Fortgeschrittene Entwicklung, Helm Charts, Pipelines, Operators

Azure DevOps - die Microsoft-Plattform

- 05 AZ-400: Designing and Implementing Microsoft DevOps Solutions**
Offizielle Microsoft-Zertifizierung, Azure Pipelines, Repos, Boards
- 06 DevOps mit Azure DevOps Workshop**
Praxis-Workshop: Boards, Repos, Pipelines, Test Plans, Artifacts

Weitere Azure-DevOps-Kurse

Azure DevOps für Tester/Entwickler/Projektmanager und Azure DevOps Verwaltung Kompaktkurs - rollenspezifische Kurse für verschiedene Zielgruppen.

Zertifizierungen

Anerkannte Nachweise für DevOps-Kompetenz

Zertifizierungen belegen Kompetenz gegenüber Arbeitgebern, Kunden und Auditoren. Die wichtigsten DevOps-Zertifizierungen auf einen Blick - und die passende DevOps Weiterbildung bei cmt.

Zertifizierung	Anbieter	Fokus	cmt-Kurs
CKA	CNCF / Linux Foundation	Kubernetes-Administration	CKA Training
CKAD	CNCF / Linux Foundation	Kubernetes-Entwicklung	CKAD Training
CKS	CNCF / Linux Foundation	Kubernetes Security	CKS Training
LPI-OT DevOps	LPI	DevOps Tools & Praktiken	Komplettausbildung
AZ-400	Microsoft	Azure DevOps Engineer	AZ-400 Training
DevOps Engineer Expert	Microsoft	Azure DevOps (Aufbau)	DevOps Engineer Expert

DevOps-Komplettausbildungen

01 LPI-OT DevOps Tools 2.0 Engineer Komplettausbildung

Umfassende Ausbildung: Container, Orchestrierung, CI/CD, IaC, Monitoring

Docker, Kubernetes, Git, Jenkins, Ansible, Terraform, Prometheus - alle relevanten Tools in einem Kurs. Vorbereitung auf die LPI-Zertifizierung.

02 LPI DevOps Tools Engineer - Prüfungsvorbereitung (DO6)

Gezielter Workshop zur Prüfungsvorbereitung

Prüfungsrelevante Themen, Übungsaufgaben, Beispielprüfungen, Tipps zur Prüfungsstrategie

03 DevOps Engineering on AWS

AWS CodePipeline, CodeBuild, CloudFormation, ECS/EKS

Offizieller AWS-Kurs für DevOps-Praktiken auf der AWS-Plattform. Ideal als Vorbereitung auf die AWS DevOps Engineer Professional-Prüfung.

Seminarübersicht

Alle DevOps-Seminare nach Themenbereich

Thema	Seminar
Container	Docker Grundkurs - Der komplette Einstieg
	Docker Fundamentals
	Einführung in Container mit Podman
	Linux Container Workshop - Docker und Podman Intensivkurs
	Container Management und Orchestrierung (DO2)
Kubernetes	Kubernetes Grundkurs
	CKA: Certified Kubernetes Administrator
	CKAD: Certified Kubernetes Application Developer
	CKS: Certified Kubernetes Security Specialist
	Kubernetes für Plattform-Teams
	Air-Gapped Kubernetes
CI/CD	Git, GitLab und CI/CD
	Jenkins Grundkurs / Kompaktkurs
	GitHub Actions (GH-200)
	CI/CD with Jenkins X (LFS268)
IaC	Terraform mit AWS / Azure
	Ansible Kompaktkurs
	IaC mit KI: Terraform & GitOps
OpenShift	DO180 OpenShift Administration I
	DO280 OpenShift Administration II
	DO188 OpenShift Development I
	DO288 OpenShift Developer II
Azure DevOps	AZ-400 Microsoft DevOps Solutions
	DevOps mit Azure DevOps Workshop
	Azure DevOps für Tester/Entwickler/PM
	Azure DevOps Verwaltung Kompaktkurs
Cloud Native	Vom Monolith zum Cloud Native System
	Container, Microservices & Plattform Engineering

Seminarübersicht (Fortsetzung)

Weitere Seminare und Spezialthemen

Thema	Seminar
Microservices	Microservices mit Spring
	Cloud-Native Go: Skalierbare Microservices
	Legacy Java EE zu Cloud Native Spring Boot
Security & Governance	LFS460 Kubernetes Security Fundamentals
	GH-500 GitHub Advanced Security
	Cloud Native Security & Governance
	Firecracker: MicroVMs sicher betreiben
Observability	Prometheus, Grafana & OpenTelemetry für Kubernetes
	SRE Deep Dive: Incident Automation
Zertifizierung	LPIC-OT DevOps Tools 2.0 Engineer Komplettausbildung
	LPI DevOps Prüfungsvorbereitung (DO6)
	Microsoft Certified: DevOps Engineer Expert
AWS	DevOps Engineering on AWS
	Containers on Amazon EKS
Azure/Microsoft	Azure Container Apps Workshop
Sonstiges	KI-Apps containerisieren
	VM und Container Deployment mit Vagrant (DO5)
	FinOps & Cloud Native Effizienz

Über 60 DevOps-Seminare bei cmt

Alle Seminare sind als offener Termin, Inhouse-Training oder Online-Live buchbar. Garantie-Termine, kleine Gruppen und erfahrene Trainer. Das vollständige Programm findest du unter www.cmt.de.

Einstiegsfahrplan

Von Null auf DevOps in sechs Monaten

DevOps-Transformation passiert nicht über Nacht. Dieser Fahrplan zeigt einen realistischen Weg, wie dein Team in sechs Monaten von manuellen Prozessen zu automatisierten Pipelines kommt - Schritt für Schritt.

M1 Monat 1-2: Fundament legen

Versionierung, Container-Basics, erste Pipeline

[Git](#), [GitLab](#) und [CI/CD](#) - Versionierung und erste Pipeline-Erfahrung

[Docker Grundkurs](#) - Applikationen containerisieren, Images und Registries verstehen

Ergebnis: Code ist versioniert, erste CI-Pipeline baut und testet automatisch, Applikation läuft im Container

M2 Monat 3-4: Orchestrierung und Infrastruktur

Kubernetes, IaC, Deployment-Strategien

[Kubernetes Grundkurs](#) - Pods, Services, Deployments, Helm

[Terraform mit AWS](#) oder [Ansible Kompaktkurs](#) - Infrastruktur als Code

Ergebnis: Container laufen orchestriert auf Kubernetes, Infrastruktur wird per Code provisioniert

M3 Monat 5-6: Observability und Security

Monitoring, Security-Integration, Zertifizierung

[Prometheus](#), [Grafana](#) & [OpenTelemetry](#) - Metriken, Dashboards, Alerting

[CKA Training](#) - Kubernetes-Kompetenz zertifizieren lassen

Ergebnis: Vollständige DevOps-Pipeline mit Observability und Zertifizierung als Nachweis



Individueller Lernpfad

Dieser Fahrplan ist ein Vorschlag. Je nach Vorwissen, Technologie-Stack und Zielen passen wir den Lernpfad individuell an. Sprich uns an - wir beraten dich gerne.



Warum cmt?

Dein Partner für DevOps und Cloud Native

2.100+

Seminare im Programm

Von IT-Grundlagen bis Spezialisierung - eines der umfassendsten Schulungsangebote in Deutschland.

25+

Jahre Erfahrung

Seit über 25 Jahren begleiten wir Unternehmen bei der technologischen Weiterentwicklung.

60+

DevOps- & Container-Seminare

Vom Docker-Grundkurs bis zur CKA-Zertifizierung - alles aus einer Hand.

100%

Garantie-Termine

Alle offenen Termine finden garantiert statt - keine Absage wegen zu weniger Teilnehmer.

Formate, die zu dir passen

Offene Seminare

Feste Termine, kleine Gruppen (4-8 TN), garantierte Durchführung

Inhouse-Training

Maßgeschneidert auf deinen Stack und deine Prozesse, bis 12 TN

Online-Live

Interaktiv, ortsunabhängig, gleiche Qualität wie Präsenz

Dein Ansprechpartner

Yves Hoppe - IT & Open Source

Telefon: 0800 71 20 000 · E-Mail: yves.hoppe@cmt.de

Du möchtest ein individuelles DevOps-Schulungsprogramm für dein Team? Yves berät dich persönlich.



**DevOps-Schulung planen?
Sprich uns an.**



Yves Hoppe

IT & Open Source

0800 71 20 000 · yves.hoppe@cmt.de

cmt GmbH · Telefon: 0800 71 20 000 · info@cmt.de

www.cmt.de