



## Platform Engineering

Platform Engineering ist eine neue Art, DevOps zu betrachten. Es soll Entwicklungs- und Betriebsteams technisch dabei unterstützen, der wachsenden Komplexität Herr zu werden. Dabei weitet Platform Engineering den Blick über die Entwickler-Community aus und bezieht Unternehmensbelange wie Infrastruktur und Sicherheit mit ein.

ab Seite 7

## Platform Engineering

Softwarebereitstellung: Platform Engineering gegen Komplexität

Internal Developer Platform: Mehr Autonomie und Verantwortung

Automatisierung: Selbstständige Softwareagenten

Deklarativ provisionieren: Crossplane hebt GitOps in die Multi-Cloud

Crossplane in der Praxis: IaC in AWS und Azure

OpenGitOps: Es geht auch ohne Kubernetes

Infrastructure-as-Code: Modular arbeiten mit Terraform und OpenTofu

## Developer Experience

Verteilte Anwendungen: Cloud-native Vielfalt für mehr Resilienz

App in die Cloud: Java-EE-Anwendungen mit AWS umziehen

Schwere Entscheidung: Managed Kubernetes oder Managed Development?

Serverless Computing: Kein Server, kein Stress

Container im IoT und Edge-Computing: Podman macht's möglich

## Observability und Security

Eine Frage der Einstellung: Observability braucht mehr als Tools

Verstopfung lösen: Effiziente CI/CD-Pipelines gewährleisten

Debugging im Produktivsystem:  
Chaos Engineering mit eBPF-Tools

Supply Chain Security: Risiken erkennen und beheben

Schwachstellen vermeiden: Sichere APIs entwerfen, entwickeln und betreiben

## Prozesse und Organisation

DevOps: Mehr Qualität, schneller ausliefern und Kosten senken

Kein Geld verschwenden:  
Gezieltes Kostenmanagement in der Cloud

KI aus der Cloud: Angebote im Preis-Leistungs-Vergleich

Machine-Learning-Instanzen von AWS

Azure Compute für KI und ML

Google Cloud ML Engines

Database-as-a-Service: Mit Benchmarking zum optimalen, kostengünstigen Dienst

## Rubriken

Editorial	3
Impressum	97
Inserentenverzeichnis	97



## Developer Experience

Die containerisierte Anwendungsentwicklung und -bereitstellung hat mit Docker und Kubernetes eine rasante Entwicklung erfahren. Development- und Betriebsteams profitieren von mehr Autonomie, sehen sich aber auch mit zusätzlichen Aufgaben und einer spürbar höheren (kognitiven) Last konfrontiert – darunter leidet die Produktivität. Cloud-native Ansätze mit einer abgestimmten Mischung aus frei wählbaren Tools und unterstützenden Managed Services sollen Entwicklungsteams die notwendige Freiheit für effizienteres Arbeiten verschaffen.

ab Seite 55

## Observability und Security

Anders als das auf die Infrastruktur fokussierte Monitoring im klassischen IT-Betrieb nimmt Observability auch Softwareentwickler in die Pflicht, ihre eigenen Anwendungen im Blick zu behalten. Aufbauend auf Logs, Metriken und Traces stehen zahlreiche Tools bereit, um Engpässe in der CI/CD-Pipeline rechtzeitig zu erkennen und zu beseitigen, aber auch sicherheitsrelevante Probleme zu lösen.

ab Seite 85



## Prozesse und Organisation

DevOps bringt Softwareentwicklungs- (Dev) und Betriebsteams (Ops) mit dem Ziel zusammen, die Bereitstellung von Anwendungen zu optimieren und deren Qualität sicherzustellen. Dabei gewinnt die umfassende Kostenkontrolle im Cloud-Native-Zeitalter immer mehr an Bedeutung – auch im Hinblick auf Nachhaltigkeit bei der Auswahl geeigneter Cloud-Dienste.

ab Seite 123

