

DevOps in der Cloud: Warum die Kultur über das Tempo entscheidet

Von Thomas Faß · 29. Juli 2026 · aktualisiert 9. September 2026

DevOps in der Cloud verbindet die DevOps-Prinzipien – Zusammenarbeit, Automatisierung, kontinuierliche Auslieferung – mit der Elastizität der Cloud, um Software häufiger, schneller und zugleich stabiler bereitzustellen. Die DORA-Forschung zeigt, dass leistungsstarke Teams deutlich häufiger ausliefern, ohne ihre Fehlerrate zu erhöhen. Der entscheidende Hebel ist dabei kultureller Natur: Ohne den Abbau von Silos und geteilte Verantwortung bleibt die beste Werkzeugkette wirkungslos.

Das Wichtigste in Kürze

- Die Kulturthese hat eine Zahl. In der CNCF-Erhebung nennen 47 Prozent den kulturellen Wandel in den Entwicklungsteams als größte Herausforderung – vor fehlender Schulung (36 Prozent), Sicherheit (36 Prozent) und technischer Komplexität (34 Prozent). Die Auswertung fasst es so: „Culture – not complexity or security – is the top barrier to cloud native adoption.“ Die Werte sind Mehrfachnennungen und ergeben eine Rangfolge, keine Aufteilung (CNCF Annual Cloud Native Survey, Auswertung vom 20. Januar 2026, Bezugsjahr 2025; die CNCF veröffentlicht dazu öffentlich kein Erhebungsdesign).
- DevOps in der Cloud verbindet DevOps-Prinzipien – Zusammenarbeit, Automatisierung, kontinuierliche Auslieferung – mit der Elastizität der Cloud.
- So wird Software häufiger, schneller und zugleich stabiler bereitgestellt.
- Die DORA-Forschung zeigt: leistungsstarke Teams liefern deutlich häufiger aus, ohne ihre Fehlerrate zu erhöhen.
- Der entscheidende Hebel ist kulturell: Ohne Abbau von Silos und geteilte Verantwortung bleibt die beste Werkzeugkette wirkungslos.

Einleitung: Die Symbiose von DevOps und Cloud für moderne Softwareentwicklung

Die moderne Geschäftswelt erfordert eine beispiellose Geschwindigkeit und Zuverlässigkeit bei der Softwarebereitstellung. Unternehmen müssen schnell auf Marktveränderungen reagieren, neue Funktionen liefern und gleichzeitig hohe Qualitäts- und Sicherheitsstandards gewährleisten. Traditionelle Softwareentwicklungsmethoden stoßen hier oft an ihre Grenzen, was zu langen Release-Zyklen, Engpässen und einem erhöhten Fehlerrisiko führt. Die Notwendigkeit, Software schneller und zuverlässiger zu liefern, hat die Einführung agiler Methoden und insbesondere von DevOps vorangetrieben.

DevOps ist eine Softwareentwicklungspraxis, die darauf abzielt, traditionelle Silos zwischen Entwicklungs- und IT-Betriebsteams aufzubrechen, um Produkte und Dienstleistungen schnell und effizient bereitzustellen. Diese Methodik fördert die Zusammenarbeit zwischen Teams im gesamten Prozess,

automatisiert Schritte in den verschiedenen Phasen der Produktentwicklung und verbessert kontinuierlich die Überwachung von der Entwicklung bis zur Bereitstellung. Cloud Computing bietet die grundlegende Infrastruktur, die für eine erfolgreiche DevOps-Implementierung unerlässlich ist. Die Cloud ermöglicht es Unternehmen, ihre IT-Ressourcen flexibel und kosteneffizient zu nutzen, was direkt die Prinzipien der Automatisierung und Skalierbarkeit unterstützt, die im Herzen von DevOps liegen. Diese Kombination aus DevOps-Methodik und Cloud-Infrastruktur schafft eine leistungsstarke Synergie, die die Softwareentwicklung erheblich beschleunigt und die Wettbewerbsfähigkeit eines Unternehmens stärkt.

Eine wesentliche Erkenntnis ist, dass Cloud Computing nicht nur als Hosting-Plattform für DevOps dient, sondern als ein entscheidender Wegbereiter für dessen volle Reife fungiert. DevOps-Praktiken wie Continuous Integration (CI) und Continuous Delivery/Deployment (CD) sowie die Automatisierung sind grundlegend für eine schnelle Softwarebereitstellung. Cloud-Umgebungen bieten die Elastizität und On-Demand-Ressourcen, die für den effizienten und skalierbaren Betrieb dieser automatisierten Prozesse entscheidend sind. Dies legt nahe, dass der Betrieb von DevOps auf traditioneller On-Premise-Infrastruktur aufgrund inhärenter physischer Einschränkungen nur begrenzte Ergebnisse liefern könnte. Die Cloud hostet DevOps nicht nur; sie verbessert dessen Fähigkeiten grundlegend, indem sie eine wirklich kontinuierliche Integration, Tests und Bereitstellung in einer Geschwindigkeit und einem Umfang ermöglicht, die zuvor unerreichbar waren. Der Fokus verlagert sich von der Verwaltung der zugrunde liegenden Hardware auf die Optimierung der Softwarebereitstellungspipelines. Für Organisationen, die eine hohe DevOps-Reife anstreben, ist die Cloud-Einführung daher keine Option, sondern eine Voraussetzung, um das volle Potenzial der agilen Entwicklung und kontinuierlichen Bereitstellung auszuschöpfen. Dies impliziert einen strategischen Imperativ für die Cloud-Migration als integralen Bestandteil einer umfassenden DevOps-Transformation.

DevOps im Kern: Prinzipien für Agilität und Effizienz

DevOps steht für einen kulturellen und operativen Paradigmenwechsel, der über traditionelle IT-Silos hinausgeht, um Zusammenarbeit und kontinuierliche Verbesserung über den gesamten Softwareentwicklungslebenszyklus hinweg zu fördern. Das Verständnis seiner Kernprinzipien ist entscheidend, um zu erfassen, wie es die Entwicklung beschleunigt.

Die fünf Säulen von DevOps

- **Kollaboration (Collaboration):** DevOps bricht traditionelle Barrieren zwischen Entwicklungs- und Betriebsteams auf und führt sie zu einer kohärenten Einheit zusammen, die über den gesamten Anwendungslebenszyklus hinweg Feedback teilt und zusammenarbeitet. Diese Integration fördert eine „Full-Stack“-Entwicklung, bei der Teams die Verantwortung für eine Funktion von der Idee bis zur Bereitstellung übernehmen, was zu höherer Qualität führt. Offene Kommunikation und Transparenz sind dabei entscheidend, um Missverständnisse zu vermeiden und den Zusammenhalt zu stärken.

- **Automatisierung (Automation):** Ein wesentlicher Bestandteil von DevOps ist die Automatisierung so vieler Schritte des Softwareentwicklungslebenszyklus wie möglich. Dies gibt Entwicklern mehr Zeit für das Schreiben von Code und die Entwicklung neuer Funktionen, reduziert menschliche Fehler und erhöht die Teamproduktivität. Automatisierung ist ein Schlüsselement von CI/CD-Pipelines und ermöglicht kontinuierliche Verbesserungen mit kurzen Iterationszeiten, wodurch schnell auf Kundenfeedback reagiert werden kann.
- **Kontinuierliche Verbesserung (Continuous Improvement):** Dieses Prinzip, das in agilen Praktiken und Lean Manufacturing verwurzelt ist, konzentriert sich auf Experimente, Abfallminimierung und Optimierung von Geschwindigkeit, Kosten und Lieferfreundlichkeit. Es ist eng mit der kontinuierlichen Bereitstellung verbunden und ermöglicht es DevOps-Teams, kontinuierlich Updates zu veröffentlichen, die die Effizienz von Softwaresystemen verbessern. Der ständige Fluss neuer Releases bedeutet, dass Teams konsequent Codeänderungen pushen, die Verschwendung eliminieren, die Entwicklungseffizienz verbessern und einen höheren Kundennutzen schaffen.
- **Kundenzentrierung (Customer-centric Action):** DevOps-Teams nutzen kurze Feedback-Schleifen mit Kunden und Endbenutzern, um Produkte und Dienstleistungen zu entwickeln, die auf deren Bedürfnisse zugeschnitten sind. Echtzeit-Live-Monitoring und schnelle Bereitstellung ermöglichen die sofortige Erfassung und Reaktion auf Benutzerfeedback. Teams erhalten sofortige Transparenz darüber, wie Live-Benutzer mit einem Softwaresystem interagieren, und nutzen diese Erkenntnisse, um weitere Verbesserungen zu entwickeln.
- **End-to-End-Verantwortung (End-to-End Responsibility):** Dieses Prinzip beinhaltet das Verständnis der Kundenbedürfnisse und die Entwicklung von Produkten oder Dienstleistungen, die echte Probleme lösen. DevOps-Teams sollten ein umfassendes Verständnis des Produkts haben, von der Erstellung bis zur Implementierung, anstatt Software basierend auf Annahmen zu entwickeln.

Der kulturelle Wandel: Von Silos zu integrierten Teams

Der Übergang zu DevOps ist eine grundlegende kulturelle Transformation, die von starren Aufgabenbeschreibungen und Abteilungsengpässen wegführt. Er erfordert einen Mentalitätswandel, der geteilte Verantwortung und Eigenverantwortung fördert. Herausforderungen umfassen kulturellen Widerstand und die Überwindung bestehender Silos, da traditionell Entwicklungsteams auf schnelle Feature-Bereitstellung fokussiert waren, während Betriebsteams Stabilität priorisierten, was zu Zielkonflikten und Kommunikationsproblemen führte. Eine starke Führung, klare Kommunikation und funktionsübergreifende Teams sind entscheidend, um diesen Wandel erfolgreich zu gestalten.

Der Abstand zwischen den Reifegruppen zeigt, woran es liegt. Unter den Organisationen, die die CNCF als „cloud native innovators“ führt, setzen 58 Prozent GitOps-Prinzipien umfassend ein. Unter den „adopters“ sind es 23 Prozent – 35 Prozentpunkte Abstand bei einer Praxis, die frei verfügbar ist und nichts kostet (CNCF, Januar 2026). Beide Bezeichnungen sind Reifegruppen der CNCF-Auswertung, keine Größenklassen.

Das ist der Beleg für die Kulturthese in einer Zahl. Was die beiden Gruppen trennt, ist nicht das Budget und nicht der Zugang zur Technik. Es ist die Bereitschaft, den gewünschten Zustand versioniert festzuschreiben und Abweichungen davon sichtbar zu machen – also die Bereitschaft, sich selbst überprüfbar zu machen.

Eine wichtige Beobachtung ist die wechselseitige Abhängigkeit von kulturellem Wandel und Automatisierung für eine nachhaltige Beschleunigung. DevOps-Prinzipien betonen sowohl die kulturelle Zusammenarbeit, wie das Aufbrechen von Silos und geteilte Verantwortung, als auch die technische Automatisierung durch CI/CD und Infrastructure as Code (IaC). Die Praxis zeigt, dass kultureller Widerstand ein erhebliches Hindernis darstellt. Wenn Teams in Silos verbleiben oder sich dem Wandel widersetzen, werden selbst die fortschrittlichsten Automatisierungstools ihr volles Potenzial nicht entfalten können, da Prozesse fragmentiert oder falsch ausgerichtet bleiben.

Effektive Automatisierung erfordert eine kollaborative Kultur, um konsistente Prozesse und gemeinsame Ziele zu definieren. Umgekehrt kann erfolgreiche Automatisierung den kulturellen Wandel verstärken, indem sie greifbare Vorteile wie schnellere Bereitstellungen und weniger Fehler aufzeigt, Vertrauen aufbaut und manuelle Arbeit reduziert, wodurch die Zusammenarbeit attraktiver wird. Dies bedeutet, dass Organisationen DevOps-Tools nicht einfach „kaufen“ und eine Transformation erwarten können. Eine nachhaltige Beschleunigung der Softwareentwicklung hängt von einer gleichzeitigen und integrierten Anstrengung ab, eine kollaborative, vertrauensbasierte Kultur zu fördern und gleichzeitig Arbeitsabläufe systematisch zu automatisieren. Das Vernachlässigen eines dieser Aspekte führt zu suboptimalen Ergebnissen oder sogar zum Scheitern der DevOps-Initiative.

Cloud-native Architektur: Die technologische Basis der Beschleunigung

Cloud-native Architektur ist das technologische Fundament, das die schnelle, skalierbare und resiliente Softwareentwicklung ermöglicht, die für modernes DevOps charakteristisch ist. Es ist ein Ansatz, der von Grund auf darauf ausgelegt ist, die einzigartigen Fähigkeiten von Cloud-Umgebungen zu nutzen.

Kernkomponenten und ihr Beitrag zur Beschleunigung

Microservices-Architektur: Microservices zerlegen monolithische Anwendungen in kleine, unabhängige und lose gekoppelte Dienste, die über leichtgewichtige APIs kommunizieren. Jeder Dienst hat eine spezifische Geschäftsfunktion und kann unabhängig entwickelt, bereitgestellt und skaliert werden. Diese Modularität ermöglicht es verschiedenen Entwicklungsteams, gleichzeitig an verschiedenen Diensten zu arbeiten, ohne sich gegenseitig zu behindern, was die Entwicklungszyklen erheblich verkürzt und die Time-to-Market beschleunigt. Änderungen oder Fehlerbehebungen in einem Dienst erfordern keine erneute Bereitstellung der gesamten Anwendung, – Updates werden schneller, die Bereitstellung kontinuierlich. Beispiele wie Netflix und Spotify zeigen diesen Übergang von Monolithen zu Microservices für verbesserte Skalierbarkeit und Zuverlässigkeit.

Containerisierung (Docker) und Orchestrierung (Kubernetes): Die Containerisierung verpackt eine Anwendung und alle ihre Abhängigkeiten (Bibliotheken, Dateien, Konfigurationen, Binärdateien) in ein einziges, isoliertes und ausführbares Image. Docker ist ein beliebtes Tool dafür. Container sind leichtgewichtig und portabel, teilen nur den OS-Kernel mit der Host-Maschine und sind daher ressourceneffizienter als VMs. Kubernetes (K8s) ist eine Open-Source-Container-Orchestrierungsplattform, die die Bereitstellung, Skalierung, Verwaltung und den Betrieb von containerisierten Anwendungen automatisiert. Container gewährleisten Konsistenz über verschiedene Umgebungen hinweg (das Problem „Es funktioniert auf meiner Maschine“ wird gelöst), beschleunigen die Bereitstellung und reduzieren die Debugging-Zeit. Kubernetes automatisiert Skalierung, Selbstheilung und

Ressourcenzuweisung, wodurch Entwickler von Infrastrukturproblemen entlastet werden und schnellere, zuverlässigere Bereitstellungen ermöglicht werden. Dies ermöglicht es Teams, sich auf den Code statt auf die Infrastrukturverwaltung zu konzentrieren, was die Entwicklungszyklen beschleunigt.

Serverless Computing: Serverless ist ein Cloud-natives Entwicklungsmodell, bei dem Entwickler Anwendungen erstellen und ausführen können, ohne Server verwalten zu müssen. Der Cloud-Anbieter übernimmt die Bereitstellung, Wartung und Skalierung der Infrastruktur. Entwickler verpacken Code in Containern zur Bereitstellung, und Serverless-Anwendungen reagieren auf Nachfrage, indem sie automatisch hoch- oder herunterskaliert werden. Die Abrechnung erfolgt in der Regel nach Verbrauch, d.h. es fallen keine Kosten an, wenn die Anwendung im Leerlauf ist. Dies ermöglicht Entwicklern, sich ausschließlich auf das Schreiben von Geschäftslogik zu konzentrieren, was die Entwicklungsgeschwindigkeit durch die Auslagerung der Infrastrukturverwaltung beschleunigt. Automatische Skalierung und ereignisgesteuerte Ausführungsmodelle stellen sicher, dass Anwendungen immer verfügbar und leistungsfähig sind, ohne manuellen Eingriff, was zu einer schnelleren Markteinführung neuer Funktionen führt.

Infrastructure as Code (IaC): IaC ist die Praxis, IT-Infrastruktur mithilfe von Code anstatt manueller Prozesse bereitzustellen und zu verwalten. Konfigurationsdateien definieren den gewünschten Zustand der Infrastruktur (z. B. VMs, Netzwerke, Datenbanken). IaC automatisiert die Infrastruktur-Einrichtung, reduziert menschliche Fehler und gewährleistet konsistente Umgebungen in Entwicklung, Test und Produktion. Dies beschleunigt die Bereitstellung, ermöglicht die schnelle Duplizierung von Umgebungen und erleichtert schnelle Rollbacks, wodurch die gesamte Entwicklungs- und Bereitstellungs-pipeline beschleunigt wird. Es ist ein integraler Bestandteil von CI/CD-Pipelines und DevOps-Praktiken.

CI/CD-Pipelines (Continuous Integration/Continuous Delivery/Deployment): CI/CD ist eine Softwareentwicklungsmethodik, die den Softwarelieferlebenszyklus automatisiert und rationalisiert. Continuous Integration (CI) beinhaltet das häufige Zusammenführen von Codeänderungen in ein gemeinsames Repository mit automatisierten Tests. Continuous Delivery/Deployment (CD) automatisiert die Freigabe von Code in ein Repository (Delivery) oder direkt in die Produktion (Deployment). CI/CD-Pipelines ermöglichen schnelle, häufige und zuverlässige Code-Updates, was die Time-to-Market erheblich verkürzt. Automatisierte Tests erkennen Fehler frühzeitig, reduzieren kostspielige Probleme und beschleunigen die Fehlerbehebung. Diese Automatisierung minimiert menschliche Fehler und gewährleistet einen konsistenten, wiederholbaren Bereitstellungsprozess.

Observability (Logs, Metriken, Traces): Observability ist die Fähigkeit, den internen Zustand eines Systems aus seinen externen Ausgaben abzuleiten, und bietet eine vollständige Sicht auf die Systemgesundheit und -leistung. Ihre drei Säulen sind: Logs (detaillierte Ereignisaufzeichnungen), Metriken (numerische Messungen wie CPU-Auslastung, Fehlerraten) und Traces (End-to-End-Karten von Anfragen durch verteilte Systeme). Observability hilft DevOps-Teams, Probleme schnell zu finden und zu beheben, die Leistung zu verbessern und Vorfälle zu verhindern, indem sie eine vollständige Ansicht des Systemverhaltens unter verschiedenen Bedingungen liefert. Sie ermöglicht die proaktive Erkennung von Anomalien und die Diagnose von Grundursachen, was zu einer schnelleren Problemlösung und optimierten Ressourcen führt. Die Integration von Observability in CI/CD-Pipelines und die Nutzung von Tools wie Prometheus, Grafana oder Datadog sind Best Practices, um eine kontinuierliche Überwachung und schnelle Reaktion auf Probleme zu gewährleisten.

APIs (Application Programming Interfaces): APIs sind die Verbindungselemente in einer Microservices-Architektur und definieren Regeln und Spezifikationen für die Interaktion von Softwarekomponenten und den Datenaustausch. Sie koppeln Dienste lose – so lassen sich diese unabhängig bereitstellen, entwickeln und skalieren, wiederverwendbar und interoperabel. Deklarative APIs beschreiben den gewünschten Zustand eines Systems, anstatt die Schritte zu dessen Erreichung, was die Verwaltung komplexer Systeme vereinfacht. Durch API-First-Design können Teams parallel entwickeln und Datenzugriffsengpässe beseitigen, was die Feature-Bereitstellung beschleunigt.

Service Meshes: Ein Service Mesh ist eine dedizierte Infrastrukturschicht, die die Kommunikation zwischen Microservices in einer Microservices-Architektur steuert. Es übernimmt Aufgaben wie Lastausgleich, Datenverschlüsselung und Service Discovery. Service Meshes entlasten Entwickler von der Komplexität der Service-zu-Service-Kommunikation, sodass sie sich auf die Geschäftslogik konzentrieren können. Sie verbessern die Sicherheit durch automatische Verschlüsselung und Authentifizierung der Kommunikation zwischen Microservices (Zero Trust Security) und bieten End-to-End-Transparenz durch die automatische Erfassung von Metriken und Logs. Dies ermöglicht eine präzise Fehlerbehebung und dynamische Traffic-Kontrolle, was die Zuverlässigkeit und Agilität erhöht.

Immutable Infrastructure: Immutable Infrastructure ist ein Ansatz, bei dem Server und Container nach der Bereitstellung nicht mehr verändert, sondern bei Bedarf durch neue, aktualisierte Instanzen ersetzt werden. Dies gewährleistet eine konsistente Bereitstellung und eliminiert Konfigurationsdrift, ein häufiges Problem in traditionellen Systemen. Dieser Ansatz verbessert die Sicherheit, da manuelle Updates und Ad-hoc-Korrekturen vermieden werden, und vereinfacht die Fehlerbehebung und Rollbacks, da jede Instanz eine versionierte, vordefinierte Konfiguration hat. Es integriert sich nahtlos mit IaC und Automatisierung, was die Bereitstellung und Skalierung beschleunigt.

Edge Computing: Edge Computing verlagert die Datenverarbeitung und -speicherung näher an die Datenquellen, wodurch Latenzzeiten reduziert und die Echtzeitverarbeitung verbessert werden. Dies ist besonders wichtig für IoT-Anwendungen und Szenarien, die sofortige Reaktionen erfordern. Edge Computing ergänzt Cloud Computing, indem es die schnelle Verarbeitung am Rand des Netzwerks ermöglicht, während die Cloud für langfristige Speicherung und komplexe Analysen genutzt wird. Diese Synergie optimiert die Effizienz, Skalierbarkeit und Kosten, indem sie den Datenverkehr zur Cloud reduziert.

Beschleunigung der Softwareentwicklung durch Cloud-DevOps

Belegt wird dieser Zusammenhang durch die DORA-Forschung (DevOps Research and Assessment), die im jährlichen „State of DevOps“-Report die Software-Auslieferung tausender Teams auswertet. Sie misst Leistung an vier Kennzahlen – Deployment-Häufigkeit, Vorlaufzeit für Änderungen (Lead Time for Changes), Änderungsfehlerrate (Change Failure Rate) und Wiederherstellungszeit (Time to Restore Service). Ihr zentrales Ergebnis: Leistungsstarke Teams liefern deutlich häufiger und schneller aus und halten zugleich eine niedrigere Fehlerrate – Tempo und Stabilität sind keine Gegensätze, sondern steigen gemeinsam.

Die Integration von DevOps-Prinzipien mit Cloud-nativen Architekturen führt zu einer signifikanten Beschleunigung des gesamten Softwareentwicklungslebenszyklus. Diese Beschleunigung manifestiert sich

in verschiedenen Schlüsselbereichen.

Schnellere Time-to-Market und höhere Release-Frequenz

Cloud-native Entwicklung, kombiniert mit DevOps-Praktiken, ermöglicht eine drastische Reduzierung der Time-to-Market für neue Produkte und Funktionen. Durch die Nutzung von Microservices und Containerisierung können Teams Funktionen und Updates nahtlos und unabhängig voneinander bereitstellen. Dies ermöglicht eine kontinuierliche Bereitstellung von Software, oft mehrmals täglich oder wöchentlich, anstatt monatlicher oder quartalsweiser Releases, die in traditionellen Umgebungen üblich sind.

Was die aktuelle DORA-Erhebung zeigt. Für den State-of-DevOps-Report 2025 wurden 4.867 IT-Fachleute befragt. 90 Prozent setzen KI-Werkzeuge bei der Arbeit ein, mehr als 80 Prozent berichten von höherer Produktivität, und die Befragten verbringen im Median rund zwei Stunden ihres Arbeitstages in der Interaktion mit KI – etwa ein Viertel eines Achtstundentages. Häufigster Einsatzzweck ist das Schreiben von neuem Code (71 Prozent).

Die Zahl, die daneben gehört: 30 Prozent der Befragten haben wenig oder kein Vertrauen in den von KI erzeugten Code.

Neun von zehn setzen das Werkzeug ein, knapp jeder Dritte traut dem Ergebnis nicht. Für eine Auslieferungskette heißt das etwas sehr Konkretes: Der Engpass wandert vom Schreiben zum Prüfen. Wer die Erzeugung beschleunigt und die Prüfung lässt, wie sie war, verschiebt den Stau nur nach hinten.

Diese Beschleunigung wird durch automatisierte CI/CD-Pipelines ermöglicht, die den gesamten Prozess von der Code-Erstellung bis zur Produktion rationalisieren. Automatisierte Tests, einschließlich Unit-, Integrations- und End-to-End-Tests, werden frühzeitig im Entwicklungszyklus durchgeführt, um Fehler schnell zu erkennen und zu beheben, bevor sie kostspielig werden. Dies minimiert die Notwendigkeit manueller Eingriffe und beschleunigt die Fehlerbehebung, was zu schnelleren und zuverlässigeren Releases führt.

Verbesserte Skalierbarkeit und Resilienz

Cloud-native Anwendungen sind von Natur aus für Skalierbarkeit und Resilienz konzipiert. Die Cloud bietet eine scheinbar unbegrenzte Ressourcenzuweisung, die es Anwendungen ermöglicht, dynamisch auf schwankende Anforderungen zu reagieren. Auto-Scaling-Mechanismen passen die Ressourcen automatisch an die Echtzeit-Nachfrage an, wodurch sichergestellt wird, dass Anwendungen auch bei Spitzenlasten optimal funktionieren und gleichzeitig Kosten während ruhigerer Perioden optimiert werden.

Resilienz, die Fähigkeit einer Anwendung, Störungen zu widerstehen oder sich von ihnen zu erholen, ist ein weiteres zentrales Merkmal. Microservices-Architekturen, kombiniert mit Container-Orchestrierung durch Kubernetes, fördern die Fehlertoleranz, da der Ausfall eines Dienstes nicht zum Zusammenbruch der gesamten Anwendung führt. Service Meshes tragen zusätzlich zur Resilienz bei, indem sie die Kommunikation zwischen Microservices absichern und überwachen, was Kaskadenfehler verhindert. Redundanzstrategien, wie die geografische Verteilung von Daten und Systemen über mehrere Cloud-Regionen, minimieren das Risiko von Ausfallzeiten und gewährleisten die Geschäftskontinuität.

Optimierte Kosten und Ressourceneffizienz

Die Cloud-Migration und die Einführung von DevOps können zu erheblichen Kosteneinsparungen führen, insbesondere durch die Optimierung der Ressourcennutzung. Das Pay-as-you-go-Modell der Cloud bedeutet, dass Unternehmen nur für die tatsächlich genutzten Ressourcen zahlen, wodurch hohe Vorabinvestitionen in Hardware entfallen.

Strategien zur Kostenoptimierung umfassen:

- **Rightsizing:** Anpassung der Cloud-Ressourcen (z. B. VMs, Speicher) an den tatsächlichen Workload-Bedarf, um Überprovisionierung und damit verbundene Verschwendung zu vermeiden. Tools wie AWS Compute Optimizer und Azure Advisor geben Empfehlungen für die optimale Ressourcengröße.
- **Eliminierung ungenutzter Ressourcen:** Identifizierung und Abschaltung von inaktiven oder verwaisten Ressourcen, die weiterhin Kosten verursachen.
- **Nutzung von Rabattmodellen:** Einsatz von Reserved Instances (RIs), Savings Plans oder Spot Instances für vorhersehbare oder fehlertolerante Workloads, um erhebliche Rabatte zu erzielen.
- **Automatisierung von Kostenkontrollen:** Implementierung von FinOps-Praktiken, die Finanz-, Technologie- und Geschäftsteams zusammenbringen, um Cloud-Ausgaben gezielt zu steuern. Dies umfasst die Einrichtung von Budgetwarnungen und die Automatisierung der Ressourcenverwaltung.

Verbesserte Sicherheit und Compliance

Cloud-native Architekturen integrieren Sicherheitsmaßnahmen von Grund auf, anstatt sie nachträglich hinzuzufügen. Dies führt zu einer „Security by Design“-Philosophie, die potenzielle Bedrohungen frühzeitig antizipiert.

Wichtige Sicherheitsaspekte umfassen:

- **Identity and Access Management (IAM):** Strikte Zugriffskontrollen und das Prinzip der geringsten Rechte (Least Privilege) stellen sicher, dass nur autorisierte Benutzer und Dienste auf Ressourcen zugreifen können. Multi-Faktor-Authentifizierung (MFA) und rollenbasierte Zugriffskontrolle (RBAC) sind hierbei entscheidend.
- **Verschlüsselung:** Daten werden sowohl während der Übertragung (in transit) als auch im Ruhezustand (at rest) verschlüsselt, um die Vertraulichkeit und Integrität zu gewährleisten.
- **Automatisierte Sicherheitsüberwachung und Compliance-Checks:** Tools zur Cloud Security Posture Management (CSPM) scannen die Infrastruktur kontinuierlich auf Fehlkonfigurationen und Compliance-Verletzungen. Policy as Code (PaC) ermöglicht die Automatisierung von Governance- und Compliance-Regeln, die in den CI/CD-Pipelines durchgesetzt werden, wodurch menschliche Fehler reduziert und Konsistenz über Umgebungen hinweg gewährleistet wird.
- **Zero Trust Security Model:** Dieses Modell verifiziert jede Zugriffsanfrage, unabhängig vom Ursprung, und minimiert das Risiko unbefugten Zugriffs.

Die Einhaltung von Compliance-Vorschriften wie der DSGVO oder HIPAA ist in **Multi-Cloud**-Umgebungen komplex, da jedes Land und jeder Cloud-Anbieter eigene Regeln haben kann. Eine zentralisierte Sichtbarkeit und Verwaltung sowie die Standardisierung von Sicherheitsrichtlinien über alle Cloud-Plattformen hinweg sind entscheidend, um diese Herausforderungen zu meistern.

Überwindung des Fachkräftemangels

Der Fachkräftemangel, insbesondere im IT-Bereich, stellt Unternehmen vor erhebliche Herausforderungen. Die Bitkom-Arbeitsmarktstudie 2026 zählt **79.000 offene IT-Stellen** in Deutschland – der Wert ist gegenüber den Vorjahren gesunken, die Besetzbarkeit aber nicht besser geworden: **97 Prozent** der Unternehmen geben an, IT-Stellen nur schwer besetzen zu können. Weniger Stellen bei gleich schwieriger Besetzung heißt, dass der Engpass sich nach oben verschoben hat, nicht aufgelöst. DevOps in der Cloud bietet verschiedene Lösungsansätze, um diesem Mangel entgegenzuwirken:

- **Attraktivität als Arbeitgeber steigern:** Flexible Arbeitsmodelle wie Homeoffice, Remote Work, Gleitzeit und Vertrauensarbeitszeit sind entscheidende Faktoren bei der Gewinnung und Bindung von Talenten. Unternehmen, die solche Modelle anbieten, können auf einen größeren Talentpool zugreifen und die Mitarbeiterzufriedenheit sowie -bindung erhöhen. Beispiele wie Afilio und Inosoft AG zeigen, wie flexible Arbeitszeiten die Arbeitgeberattraktivität steigern und die Fluktuation senken können.
- **Zugang zu einem globalen Talentpool:** Remote Work und Outsourcing-Modelle wie Staff Augmentation oder Dedicated Teams ermöglichen Unternehmen den Zugriff auf qualifizierte Fachkräfte weltweit, unabhängig von geografischen Grenzen. Dies ist besonders relevant für IT-Positionen, die oft schwer zu besetzen sind. Osteuropäische Länder wie Polen, Ukraine und Rumänien bieten einen großen Pool an hochqualifizierten IT-Fachkräften mit guten Englischkenntnissen und kultureller Kompatibilität zu westlichen Unternehmen, oft zu wettbewerbsfähigen Kosten.
- **Fokus auf Upskilling und Reskilling:** Angesichts des schnellen technologischen Wandels ist die kontinuierliche Weiterbildung der bestehenden Belegschaft entscheidend. DevOps und Cloud-native Entwicklung erfordern neue Fähigkeiten, und Investitionen in Schulungsprogramme helfen, die Qualifikationslücke zu schließen und Mitarbeiter an das Unternehmen zu binden.
- **Effizienzsteigerung durch Automatisierung:** Automatisierung in DevOps-Pipelines reduziert den Bedarf an manuellen Tätigkeiten und entlastet IT-Mitarbeiter von Routineaufgaben. Dies ermöglicht es den Teams, sich auf höherwertige, innovative Aufgaben zu konzentrieren, was die Produktivität steigert und die Auswirkungen des Fachkräftemangels abmildert.

Herausforderungen und Lösungsansätze

Die Einführung von DevOps in der Cloud bringt zwar erhebliche Vorteile, ist aber auch mit spezifischen Herausforderungen verbunden, die proaktiv angegangen werden müssen.

Kulturelle und organisatorische Hürden

Der Übergang zu DevOps und Cloud-nativen Architekturen erfordert einen tiefgreifenden kulturellen Wandel, der oft auf Widerstand stößt. Dass die Kultur und nicht die Technik der Engpass ist, sagen die

Betroffenen selbst: In der CNCF-Erhebung für 2025 steht der **Kulturwandel mit 47 Prozent** an der Spitze der Hürden bei der Cloud-native-Einführung, vor Schulung und Sicherheit mit je **36 Prozent** und der Komplexität mit **34 Prozent**. Traditionelle Silos zwischen Entwicklungs- und Betriebsteams, Ängste vor Kontrollverlust oder Jobunsicherheit sowie mangelndes Verständnis für die neuen Arbeitsweisen können die Implementierung behindern.

Lösungsansätze:

- **Führung und Kommunikation:** Führungskräfte müssen den Wandel aktiv vorleben und die Vorteile der Cloud-Migration und DevOps klar kommunizieren. Transparente Kommunikation über das „Warum“ der Migration und die Auswirkungen auf die Mitarbeiter ist entscheidend.
- **Cross-funktionale Teams und Zusammenarbeit:** Förderung der Zusammenarbeit zwischen Teams und das Aufbrechen von Silos durch die Bildung funktionsübergreifender Teams. Regelmäßige Meetings und der Aufbau von Vertrauen sind essenziell für Remote-Teams.
- **Schulung und Upskilling:** Investition in umfassende Schulungsprogramme, um Mitarbeiter mit den notwendigen Cloud- und DevOps-Fähigkeiten auszustatten. Peer-Mentoring und flexible Lernmodelle können den Übergang erleichtern.

Technische Komplexität und Integration

Die Verwaltung von Multi-Cloud-Umgebungen und die Integration von Legacy-Systemen stellen erhebliche technische Herausforderungen dar. Jede Cloud hat ihre eigenen Tools, APIs und Managementprozesse, was die Standardisierung und den Überblick erschwert.

Lösungsansätze:

- **Standardisierung und Automatisierung:** Einsatz von Infrastructure as Code (IaC) und Automatisierungstools, um Konsistenz über verschiedene Umgebungen hinweg zu gewährleisten und manuelle Fehler zu reduzieren.
- **Multi-Cloud-Management-Plattformen (CMP):** Implementierung einer zentralen Management-Plattform, die eine einheitliche Sicht und Kontrolle über alle Cloud-Ressourcen bietet.
- **Containerisierung und Microservices:** Nutzung von Containerisierung (Docker) und Orchestrierung (Kubernetes) zur Abstraktion von Anwendungen von der zugrunde liegenden Infrastruktur, was die Portabilität und Skalierbarkeit in Multi-Cloud-Umgebungen verbessert.
- **Phasenweise Migration:** Eine schrittweise Migration von Legacy-Systemen in die Cloud, beginnend mit weniger kritischen Anwendungen, minimiert Störungen und ermöglicht eine schrittweise Anpassung.

Herausforderungen bei der Talentgewinnung und -bindung

Trotz der Vorteile der Cloud-Migration bleibt der Fachkräftemangel eine Herausforderung, insbesondere bei Spezialisten für Cloud-native Technologien – genau dafür stellen wir [Senior-IT-Experten](#) bereit.

Der personelle Aufwand wiegt schwerer als der technische. Der Bitkom Cloud Report 2026 fragt deutsche Unternehmen ab 20 Beschäftigten nach den Gründen gegen einen Anbieterwechsel. Der personelle

Aufwand steht mit 45 Prozent gleichauf mit der Komplexität der Anwendungen – und deutlich vor dem technischen Aufwand mit 32 Prozent (S. 20).

Die Frage zielt auf den Anbieterwechsel, nicht auf DevOps, und die Werte sind Mehrfachnennungen. Das Verhältnis ist trotzdem aufschlussreich: Auch dort, wo eine Aufgabe technisch beschrieben ist, entscheidet die Personaldecke über die Machbarkeit.

Lösungsansätze:

- Flexible Arbeitsmodelle: Das Anbieten von Homeoffice, Remote Work und flexiblen Arbeitszeiten ist entscheidend, um Talente anzuziehen und zu halten.
- Globaler Talentpool und Outsourcing: Nutzung von Staff Augmentation, Dedicated Teams oder Outsourcing, um auf globale Talentpools zuzugreifen, insbesondere in Regionen mit starken IT-Fachkräften wie Osteuropa.
- Mitarbeiterbindung durch Entwicklung: Klare Karrierepfade, Weiterbildungsmöglichkeiten und Mentoring-Programme sind wichtig, um Mitarbeiter langfristig zu binden und ihre Motivation zu steigern.

Zukünftige Trends: DevOps und Cloud Computing im Wandel

Die Landschaft von DevOps und Cloud Computing entwickelt sich rasant weiter. Mehrere Trends werden die Softwareentwicklung in den kommenden Jahren prägen.

KI-gesteuerte Cloud-Migration und Kostenoptimierung

Künstliche Intelligenz (KI) und Maschinelles Lernen (ML) werden zunehmend in Cloud-Migrationsprozesse integriert, um diese intelligenter, schneller und zuverlässiger zu gestalten. KI-gesteuerte Tools können Anwendungsbewertungen automatisieren, Kosten vorhersagen, effiziente Migrationsstrategien erstellen und potenzielle Probleme frühzeitig identifizieren. Dies reduziert Risiken, minimiert Ausfallzeiten und optimiert die Ressourcenzuweisung.

Der Engpass verschiebt sich dabei vom Schreiben zum Prüfen. Wenn 90 Prozent der Befragten KI-Werkzeuge einsetzen und 30 Prozent dem erzeugten Code wenig oder nicht vertrauen (DORA, State-of-DevOps-Report 2025), dann wächst die Menge des zu prüfenden Codes schneller als die Fähigkeit, ihn zu prüfen.

Für eine Auslieferungskette heißt das: Automatisierte Tests, statische Analyse und Reviews werden zum begrenzenden Faktor. Eine Pipeline, die auf menschliche Prüfschritte in der bisherigen Menge ausgelegt ist, wird zum Nadelöhr, sobald der Zufluss steigt. Dieselbe Freibefragung stellt sich beim [Betrieb von ML-Modellen](#), nur dort für Modelle statt für Code.

Im Bereich der Kostensteuerung wird KI die Analyse von Ausgabenmustern und die Automatisierung von Einsparmaßnahmen grundlegend verändern. KI-gestützte Überwachungstools wie AWS Compute Optimizer oder Google Cloud Recommender geben intelligente Empfehlungen zur Kostensenkung. Die Automatisierung der Ressourcen-Skalierung und die Erkennung von Anomalien in Echtzeit werden dazu beitragen, unerwartete Kosten zu vermeiden und die Effizienz zu steigern.

Multi-Cloud- und Hybrid-Cloud-Strategien

Wie verbreitet Multi-Cloud tatsächlich ist, hängt daran, wen man fragt. Die viel zitierten 89 Prozent stammen aus dem Flexera State of the Cloud Report 2024 – befragt wurden im Winter 2023 753 Cloud-Fachleute, zu 68 Prozent in Organisationen mit über 2.000 Beschäftigten und zu 62 Prozent in Amerika. Für Deutschland nennt der Bitkom Cloud Report 2026 einen anderen Wert: 38 Prozent der cloudnutzenden Unternehmen ab 20 Beschäftigten setzen auf [Multi-Cloud](#) (S. 13; telefonische Befragung von 603 Unternehmen, KW 14 bis KW 20 2026). Beide Zahlen stimmen – sie beschreiben nur nicht dieselben Unternehmen.

Haupttreiber sind die Vermeidung von Vendor Lock-in, die Erhöhung der Zuverlässigkeit und Redundanz sowie der Zugang zu Best-of-Breed-Services von verschiedenen Anbietern. Multi-Cloud ermöglicht es Unternehmen, die Stärken jedes Cloud-Anbieters zu nutzen, z. B. Google Cloud für Datenanalysen oder AWS für rechenintensive Workloads.

Die Integration von Hybrid-Cloud-Modellen, die On-Premise-Infrastrukturen mit öffentlichen Clouds kombinieren, wird ebenfalls zunehmen, insbesondere für sensible Daten oder Anwendungen mit speziellen Compliance-Anforderungen. Die Komplexität der Verwaltung mehrerer Cloud-Umgebungen erfordert jedoch einheitliche Management-Plattformen und Automatisierung.

Green Cloud Computing und Nachhaltigkeit

Nachhaltigkeit ist für Unternehmen längst kein Nebenthema mehr. Green Cloud Computing, das den Einsatz von Cloud-Technologie auf umweltfreundliche Weise fördert, gewinnt an Bedeutung. Dies beinhaltet energieeffiziente Systeme, die Reduzierung von Kohlenstoffemissionen und die Nutzung erneuerbarer Energien. Unternehmen, die auf Green Cloud Computing umsteigen, können ihre Energiekosten deutlich senken und ihren CO₂-Fußabdruck erheblich reduzieren. Zukünftige Trends umfassen KI-gesteuerte Effizienzoptimierung, globale grüne Standards und dezentrale Infrastrukturen.

Weiterentwicklung von Cloud-nativen Technologien

Die Cloud-native Landschaft wird sich weiterentwickeln, mit einem verstärkten Fokus auf:

- **WebAssembly (Wasm):** Wasm wird eine größere Rolle bei der Erstellung von Hochleistungsanwendungen für Web und Server spielen, einschließlich Edge Computing und IoT-Geräten. Die Integration von Wasm in die Open Container Initiative (OCI) und die Unterstützung durch die Cloud Native Computing Foundation (CNCF) positionieren es als eine wichtige Technologie für zukünftige Cloud-native Anwendungen.
- **FinOps-Integration:** FinOps wird sich als Standard für das Cloud-Kostenmanagement etablieren, um Transparenz, Verantwortlichkeit und Optimierung in dynamischen Cloud-Umgebungen zu gewährleisten.
- **KI-Native Cloud-Infrastruktur:** Cloud-Anbieter werden KI/ML direkt in ihre Plattformen integrieren, um KI-Workloads in großem Maßstab zu ermöglichen und die Infrastruktur durch KI-gesteuerte Optimierung zu automatisieren.

Fazit: DevOps in der Cloud als strategische Notwendigkeit

DevOps in der Cloud ist nicht nur ein Trend, sondern eine strategische Notwendigkeit für Unternehmen, die in der modernen digitalen Wirtschaft bestehen und erfolgreich sein wollen. Die Synergie zwischen DevOps-Prinzipien und Cloud-nativen Architekturen ermöglicht eine beispiellose Beschleunigung der Softwareentwicklung, indem sie Prozesse automatisiert, die Skalierbarkeit und Resilienz verbessert, Kosten optimiert und die Sicherheit erhöht.

Die konsequente Umsetzung von Kollaboration, Automatisierung, kontinuierlicher Verbesserung, Kundenzentrierung und End-to-End-Verantwortung innerhalb von Cloud-Umgebungen ist der Schlüssel zur Freisetzung dieses Potenzials. Obwohl Herausforderungen wie kultureller Widerstand, technische Komplexität und der Fachkräftemangel bestehen bleiben, bieten proaktive Strategien – von umfassenden Schulungen über die Nutzung globaler Talentpools bis hin zur Implementierung fortschrittlicher Automatisierungs- und KI-Tools – praktikable Wege zur Überwindung dieser Hürden.

Unternehmen, die eine Cloud-native Denkweise annehmen und in die kontinuierliche Weiterentwicklung ihrer Teams und Technologien investieren, werden nicht nur ihre Softwareentwicklung beschleunigen, sondern auch eine agile, resiliente und wettbewerbsfähige Position für die Zukunft aufbauen. Die Zukunft der Softwareentwicklung ist untrennbar mit der Cloud und DevOps verbunden, und diejenigen, die diese Symbiose meistern, werden die nächste Welle der Innovation anführen.

Ihr nächster Schritt

Die Toolchain ist in den meisten Häusern das kleinere Problem. Schwieriger ist der Schnitt der Verantwortung: Wer betreibt, was er gebaut hat, und wer gibt frei, was eine Maschine geschrieben hat? Wir bringen Auslieferungskette, Prüfschritte und Zuständigkeiten in eine Form, die im Alltag trägt – mit deutschsprachigen Senior-Teams und Steuerung in Deutschland.

Häufige Fragen

Was macht die Cloud für DevOps so wirksam?

Die Cloud liefert on-demand die Infrastruktur, die DevOps voraussetzt: elastische Ressourcen für automatisierte Tests und Deployments, Infrastructure as Code für reproduzierbare Umgebungen und Skalierung nach Bedarf. Sie hostet DevOps nicht nur, sie hebt dessen Möglichkeiten auf ein anderes Niveau.

Bringt DevOps mehr Tempo auf Kosten der Stabilität?

Nicht, wenn es richtig gemacht ist. Die DORA-Forschung zeigt, dass Spitzenteams häufiger ausliefern und trotzdem eine niedrigere Änderungs-Fehlerrate haben. Tempo und Stabilität sind kein Widerspruch, sondern Ergebnis von Automatisierung, Tests und kurzen Feedback-Schleifen.

Ist DevOps vor allem eine Werkzeugfrage?

Nein. Werkzeuge wie CI/CD, Container und Kubernetes sind notwendig, aber nicht hinreichend. Der häufigste Grund für gescheiterte DevOps-Vorhaben ist kultureller Widerstand: Ohne Abbau der Silos zwischen Entwicklung und Betrieb und ohne geteilte Verantwortung bleibt die Toolchain wirkungslos.

Was bedeutet CI/CD?

CI/CD steht für Continuous Integration und Continuous Delivery/Deployment: Codeänderungen werden laufend automatisiert zusammengeführt, getestet und ausgeliefert. Das ersetzt seltene, riskante Groß-Releases durch viele kleine, sichere Schritte.

Wie hilft Cloud-DevOps gegen den Fachkräftemangel?

Indem Cloud-DevOps Routine automatisiert und Infrastruktur per Code beherrschbar macht, kommt ein Team mit den vorhandenen Fachkräften weiter und ist weniger von seltenen Spezialisten abhängig. Standardisierung und Automatisierung entlasten dort, wo Personal knapp ist.

Was unterscheidet Multi-Cloud von Hybrid-Cloud?

Multi-Cloud nutzt bewusst mehrere Cloud-Anbieter parallel, etwa um Abhängigkeiten zu vermeiden; Hybrid-Cloud verbindet Cloud mit eigener On-Premise-Infrastruktur. Beide erhöhen die Flexibilität, stellen aber auch höhere Anforderungen an Steuerung und Sicherheit.

Was sind die DORA-Metriken?

Die DORA-Metriken sind vier etablierte Kennzahlen für die Leistungsfähigkeit der Softwareauslieferung: wie häufig ausgeliefert wird, wie lange Änderungen bis zur Produktion brauchen, wie oft Änderungen fehlschlagen und wie schnell wieder hergestellt wird. Sie zeigen, dass Tempo und Stabilität zusammen steigen können — nicht gegeneinander.

Thematische Vertiefungen

[Cloud, DevOps & FinOps \(Expertise\)](#)

[Cloud-Native-Entwicklung: Was den Unterschied macht](#)

[Multi-Cloud-Strategie](#)

Quellen

- DORA / Google Cloud: State of AI-assisted Software Development (State-of-DevOps-Report 2025), 142 Seiten. Befragung von 4.867 IT-Fachleuten weltweit. Belegt: 90 % KI-Einsatz bei der Arbeit; mehr als 80 % berichten höhere Produktivität; Median rund zwei Stunden tägliche KI-Interaktion; 71 % Schreiben von neuem Code als häufigster Einsatzzweck; 30 % mit wenig oder keinem Vertrauen in KI-erzeugten Code. — <https://dora.dev/research/2025/dora-report/>
- Flexera: 2024 State of the Cloud Report — Herkunft der 89 Prozent. Befragung von 753 Cloud-Fachleuten im Winter 2023; 68 % in Organisationen mit über 2.000 Beschäftigten, 62 % aus Amerika. Erhebungsdesign hier mitgeführt, weil die Zahl im Netz durchgehend ohne Kontext zitiert wird. — <https://resources.flexera.com/web/pdf/Flexera-State-of-the-Cloud-Report-2024.pdf>

- CNCF: Annual Cloud Native Survey — Ankündigung vom 20. Januar 2026, Bezugsjahr 2025. Belegt: Hürden bei der Cloud-native-Einführung — 47 % Kulturwandel, 36 % Schulung, 36 % Sicherheit, 34 % Komplexität; GitOps 58 % („innovators“) gegenüber 23 % („adopters“). Einschränkung: Die CNCF veröffentlicht auf ihren öffentlichen Seiten kein Erhebungsdesign. — <https://www.cncf.io/announcements/2026/01/20/kubernetes-established-as-the-de-facto-operating-system-for-ai-as-production-use-hits-82-in-2025-cncf-annual-cloud-native-survey/>
- Bitkom e. V.: IT-Fachkräftemangel in den letzten drei Jahren halbiert. Arbeitsmarktstudie 2026. Belegt: 79.000 offene IT-Stellen; 97 % der Unternehmen können IT-Stellen nur schwer besetzen. — <https://www.bitkom.org/Presse/Presseinformation/IT-Fachkraeftemangel-in-den-letzten-drei-Jahren-halbiert>
- Bitkom e. V.: Cloud Report 2026. Studienbericht vom 17. Juni 2026, CATI-Befragung von 603 Unternehmen in Deutschland ab 20 Beschäftigten, KW 14 bis KW 20 2026. Belegt: 38 % Multi-Cloud (S. 13); Wechselbarrieren 45 % personeller Aufwand, 45 % Komplexität der Anwendungen, 32 % technischer Aufwand (S. 20). — <https://www.bitkom.org/sites/main/files/2026-06/17062026-bitkom-studienbericht-cloud-report-2026.pdf>