

Der Weg zu cloudnativen Anwendungen

In acht Schritten zum Erfolg



Inhaltsverzeichnis

Einführung	3
Schritt 1: Entwickeln einer cloudorientierten Kultur und entsprechender Praktiken	3
Schritt 2: Beschleunigen bestehender Anwendungen mit Microservices	3
Schritt 3: Nutzen von Anwendungsservices für eine schnellere Entwicklung	4
Schritt 4: Auswählen der richtigen Tools für die richtigen Aufgaben	4
Schritt 5: Bereitstellen einer bedarfsorientierten Self-Service-Infrastruktur	5
Schritt 6: IT-Automatisierung, um die Anwendungsbereitstellung zu beschleunigen	6
Schritt 7: Implementieren von Continuous Delivery und modernen Deployment-Methoden	6
Schritt 8: Entwickeln einer hochmodularen Architektur	8
Wie Red Hat helfen kann	8

„Bei der cloudnativen Entwicklung sind Zusammenarbeit, Teams und Menschen ebenso wichtig wie Technologien ... Zusammenarbeit ist für die iterative, flexible Entwicklung von Anwendungen wesentlich. Ob Stakeholder oder Macher – die Beteiligten müssen beim Entwickeln, Programmieren, Testen und Bereitstellen eines Produkts ihren Beitrag leisten.“

—
[Red Hat Bericht: Cloudnative Entwicklung – ein Ausblick, Juni 2021](#)

Einleitung

Die meisten Unternehmen interagieren heute mit ihren Kundinnen und Kunden, Partnern und Beschäftigten vor allem über Anwendungen. Die sprunghafte Zunahme der neuen digital-nativen Funktionen hat eine Disruption traditioneller Geschäftsmodelle ausgelöst. Etablierte Unternehmen und Branchen sehen sich gezwungen, ihre Abläufe anzupassen und zu modernisieren.

Für die meisten Firmen setzt die Entwicklung innovativer digitaler Erlebnisse eine Kultur der organisierten Agilität voraus, in der sie der schnellen Gangart des technischen Fortschritts nur mit schnelleren und flexibleren Entwicklungs- und Bereitstellungsmethoden begegnen können. Die meisten Organisationen können sich jedoch nicht den Luxus erlauben, ihre technologische Basis von Grund auf neu zu erstellen oder neue Praktiken und Denkweisen direkt einzuführen. Stattdessen setzen sie auf graduelle, aber grundlegende Veränderungen in ihrer Unternehmenskultur, ihren Prozessen und ihren Technologien, um schneller und agiler zu werden. In diesem E-Book werden acht Schritte beschrieben, die Unternehmen vor der Einführung eines cloudnativen Ansatzes für Anwendungen berücksichtigen sollten.

Schritt 1: Entwickeln einer cloudorientierten Kultur und entsprechender Praktiken

Für die Umstellung auf cloudnative Anwendungen müssen sich Entwicklungs-, Operations- und Geschäftsbereichs-Teams in vielerlei Hinsicht weiterentwickeln, damit sie Anwendungen schneller und effizienter entwickeln und bereitstellen können. Ungeachtet ihrer Branche oder Größe müssen Unternehmen die Bandbreite an Aktivitäten, Technologien, Teams und Prozessen berücksichtigen, deren Zusammenarbeit und Koordination für das erfolgreiche Deployment von Cloud-Anwendungen erforderlich ist. Bei traditionellen Ansätzen für die Nutzung von Public oder Hybrid Cloud-Ressourcen können Teams zwar eigenständige Entscheidungen treffen und rasch handeln. Durch diese Strategien sind aber auch isolierte Prozesse für Daten und Umgebungen entstanden, die Innovationen erschweren.

Für Organisationen, die eine einheitliche Cloud-Strategie für ihre Anwendungen entwickeln wollen, können die komplexe Verwaltung mehrerer verteilter Umgebungen, hochgradig benutzerdefinierte Legacy-Anwendungen und neue Anwendungs-Workloads in einer Zeit schneller Innovationen zur Herausforderung werden. Oft schöpfen Organisationen ohne unternehmensweite Cloud-Strategie das Potenzial ihres Anwendungsportfolios nicht voll aus.

Die Umstellung auf eine Kultur der Zusammenarbeit hängt nicht nur von neuen Tools und Technologien ab. Unternehmen müssen auch die Bereitschaft und das Vertrauen ihrer Beschäftigten in einen integrativeren und kollaborativeren Ansatz der Anwendungsentwicklung und -bereitstellung fördern. Die Kultur von Open Source-Softwareprojekten kann bei der Entwicklung einer kohärenten, verbundenen Cloud-Strategie für Anwendungen als Orientierung dienen.

Schritt 2: Beschleunigen bestehender Anwendungen mit Microservices

Bei der Umstellung auf cloudnative Anwendungen sollten Organisationen nicht den Fehler machen, sich nur auf Neuentwicklungen zu konzentrieren. Viele Legacy-Anwendungen sind für geschäftliche Abläufe und das Generieren von Umsätzen immer noch unerlässlich und können nicht einfach ersetzt werden. Sie müssen sich in neue cloudnative Anwendungen integrieren lassen und mit diesen zusammenarbeiten können. Aber wie kann man bestehende Monolithen beschleunigen? Die Antwort: Durch Migrieren der aktuellen monolithischen Architektur zu einer modulareren Microservice-Architektur und einer API-basierten Kommunikation.

Bevor Organisationen damit beginnen, monolithische Anwendungen mühsam in Microservices umzuwandeln, sollten sie zunächst eine solide Basis für ihre Microservice-Architektur schaffen.

Wenn Sie auf einen Microservices-Ansatz umstellen, bedeutet das nicht, dass Sie überstürzt sämtliche Bereiche sofort migrieren müssen. Sie können die Monolithen in kleinere Komponenten aufbrechen und schrittweise vorgehen. Dadurch stellen Sie sicher, dass Anwendungen gemäß soliden Designprinzipien und ordnungsgemäß definierten Domain-Grenzen erstellt werden. Dieser Ansatz unterstützt, falls notwendig, einen allmählicheren und weniger risikoreichen Übergang zur Microservice-Technologie und schafft die Basis für eine erfolgreiche Microservice-Architektur.

Auch Anwendungen, die in hohem Maße von Legacy-Plattformen abhängen, können schneller aktualisiert und bereitgestellt werden, wenn der bestehende Monolith zu einer containerbasierten Plattform migriert wird. Das Ergebnis: ein schnelleres Deployment und ein höherer ROI (Return on Investment). Spätere Integrationen oder Features für diesen Monolithen können dann mithilfe von cloudnativen Techniken und Ansätzen entwickelt werden.

Schritt 3: Nutzung von Anwendungsservices für eine schnellere Entwicklung

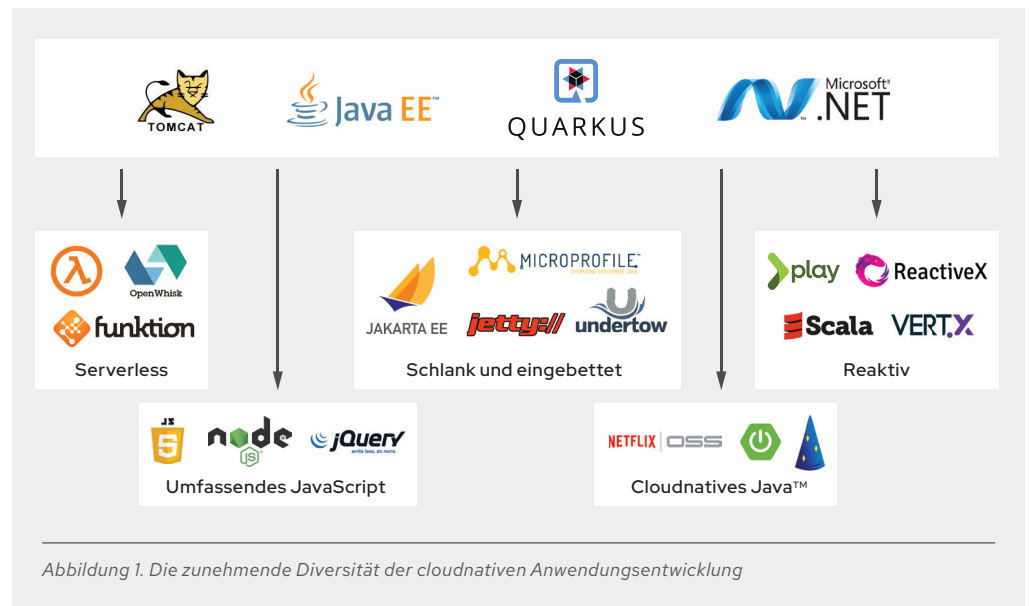
Wiederverwendbarkeit war schon immer der Schlüssel zu einer beschleunigten Softwareentwicklung. Dieses Prinzip gilt auch für cloudnative Anwendungen. Allerdings müssen wiederverwendbare Komponenten für cloudnative Anwendungen optimiert und in die zugrunde liegende Infrastruktur integriert werden, damit sie die Geschwindigkeit und Skalierbarkeit der Cloud bieten.

Warum Caching-Services, Regel- oder Workflow-Engines, Integrationskonnektoren, Mobil- und API-Managementfunktionen, Datenvirtualisierungsservices, Messaging-Broker oder serverlose Frameworks neu erstellen, wenn dies gar nicht erforderlich ist? Sie können bereits vorhandene Komponenten verwenden, die optimiert und in die zugrunde liegende containerbasierte Infrastruktur integriert wurden. Bei diesen Anwendungsservices – unabhängig davon, ob SaaS (Software-as-a-Service), PaaS (Platform-as-a-Service) oder iPaaS (Integration PaaS) – handelt es sich um einsatzbereite Entwicklungstools.

Mit DevOps-Prozessen und Containerisierung lassen sich Anwendungsbereitstellung und -Deployment zwar beschleunigen. Cloudnative Anwendungen benötigen aber unter Umständen mehrere der zuvor genannten Service-Typen, um die Entwicklung und Markteinführung neuer Anwendungen zu beschleunigen. Entwicklungsteams können beispielsweise Anwendungsservices nutzen, die speziell für containerbasierte Infrastrukturen entwickelt wurden. Diese Services sind für die Nutzung von Plattformfunktionen wie CI/CD-Pipelines (Continuous Integration/Continuous Delivery), Rolling Deployment und Blue/Green Deployment, automatisierte Skalierbarkeit oder Fehlertoleranz konzipiert.

Schritt 4: Auswählen der richtigen Tools für die richtigen Aufgaben

Mit cloudnativen Anwendungen werden die jeweiligen Entwicklungssprachen oder -Frameworks zunehmend auf die speziellen Anforderungen einer Geschäftsanwendung zugeschnitten. Für das Management der daraus resultierenden gesteigerten Komplexität und Anwendungsdiversität benötigen Unternehmen eine containerbasierte Anwendungsplattform, die die richtige Mischung aus Frameworks, Sprachen und Architekturen für die cloudnative Entwicklung unterstützt.



Die cloudnative Entwicklung hängt zusätzlich von der Wahl der richtigen Tools für die richtigen Aufgaben ab. Die Implementierung von cloudnativen Anwendungen kann nach verschiedenen Methoden erfolgen: 12-Faktor-Ansatz, domainbasiertes Design, testbasierte Konzipierung und Entwicklung, monolithorientierte Strategie, Strategie der schnellen Monolithen, Miniservices oder Microservices. Unabhängig davon, für welchen Ansatz Sie sich entscheiden, sollte Ihre cloudnative Plattform die richtige Kombination aus Frameworks, Sprachen und Architekturen bieten, um die entsprechenden Entwicklungsanforderungen zu unterstützen. Zusätzlich sollte die zugrunde liegende containerbasierte Plattform Support für eine kuratierte Reihe von Runtimes und Frameworks bereitstellen, die kontinuierlich und abgestimmt auf die technologischen Änderungen aktualisiert wird.

Schritt 5: Bereitstellen einer bedarfsorientierten Self-Service-Infrastruktur

Agile Methoden unterstützen Entwicklungsteams bei der schnellen Entwicklung und Aktualisierung von Software, bieten aber keinen effizienten Mechanismus, der einen zeitnahen Infrastrukturzugriff ermöglicht – und zwar unabhängig davon, wann oder wo dieser erforderlich ist. Die Freigabe von Anwendungen zur Produktion wirkt sich auf die Markteinführungszeit aus. In einer Zeit, in der Infrastrukturen günstig und technische Fachkräfte teuer sind, ist das Erstellen von Tickets und das wochenlange Warten auf die Freigabe von Ressourcen durch IT-Operations-Teams nicht mehr tragfähig.

Eine Infrastruktur-Provisionierung, die dem Prinzip des Self-Service und der Bedarfsabhängigkeit folgt, bietet eine überzeugende Alternative zur nicht autorisierten Schatten-IT. Ein solches Modell kann aber nur dann effektiv sein, wenn IT-Operations-Teams über ausreichende Kontrolle und Transparenz der oft dynamischen und komplexen Umgebung verfügen.

Container und Container-Orchestrierungstechnologien vereinfachen den Zugriff auf die zugrunde liegende Infrastruktur und ermöglichen ein robustes Anwendungs-Lifecycle-Management für verschiedene Infrastruktur-Umgebungen wie Rechenzentren oder Private und Public Clouds. Eine Container-Plattform bietet zusätzliche Funktionen für Self-Service, Automatisierung und das Management von Anwendungs-Lifecycles. Mithilfe dieses Modells können DevOps-Teams schnell einheitliche Umgebungen hochfahren, wodurch sich die Entwicklerinnen und Entwickler ganz auf die Anwendungsentwicklung konzentrieren können, ohne sich mit den Hindernissen und Verzögerungen der Infrastrukturprovisionierung befassen zu müssen.

Container unterstützen außerdem die Portierbarkeit von Anwendungen und ermöglichen so auch das Erstellen cloudnativer Anwendungen, die von beliebigen Cloud-Anbietern bereitgestellt und ausgeführt werden können. Durch diese Portierbarkeit haben Sie die freie Wahl unter den Cloud-Anbietern und können von einer Cloud zur anderen migrieren, um Ihre Cloud-Kosten zu optimieren und Multi-Cloud-Anwendungen zu entwickeln, ohne diese für die APIs bestimmter Anbieter programmieren zu müssen.

Schritt 6: IT-Automatisierung zur Beschleunigung der Anwendungsbereitstellung

Die IT- oder Infrastrukturautomatisierung ist für eine beschleunigte Bereitstellung cloudnativer Anwendungen und die Eliminierung manueller IT-Aufgaben unerlässlich. Automatisierung kann in beliebige Aufgaben oder Komponenten integriert werden – ob Provisionierung von Netzwerken und Infrastrukturen, Deployment von Anwendungen oder Konfigurationsmanagement.

Mit IT-Management- und -Automatisierungstools lassen sich wiederholbare Prozesse, Regeln und Frameworks erstellen, die arbeitsintensive menschliche Eingriffe ersetzen oder mindern können, die oft zu Verzögerungen bei der Markteinführung führen. Die Tools können auch auf bestimmte Technologien ausgeweitet werden, wie etwa [Container](#), Methoden wie [DevOps](#) sowie auf größere Bereiche wie [Cloud Computing](#), Sicherheit, Tests, Monitoring und Alerts. Automatisierung beschleunigt die Wertschöpfung insgesamt und ist damit für die IT-Optimierung und digitale Transformation wesentlich.

Erfahren Sie mehr über die wichtige Rolle der IT-Automatisierung im E-Book „Das automatisierte Unternehmen“.

[E-Book herunterladen](#)

CD (Continuous Delivery) ist ein Verfahren zur Softwareentwicklung, bei dem neuer Code mithilfe von Automatisierung schneller veröffentlicht werden kann. Dabei wird ein Prozess eingerichtet, mit dem die Änderungen eines Entwicklungsteams an einer Anwendung durch [Automatisierung](#) in ein Code-Repository oder eine [Container Registry](#) verschoben werden können.

Guide zur IT-Automatisierung

1. Führen Sie einen unternehmensweiten programmatischen Automatisierungsansatz für Ihre IT-Abläufe ein. Fördern Sie im gesamten Unternehmen einen kollaborativen Dialog zur Definition von Serviceanforderungen.
2. Eine Automatisierungs-Sandbox kann für das Erlernen von Automatisierungssprachen und -prozessen eine gute Basis darstellen.
3. Setzen Sie auf eine umfassende Automatisierung. Stellen Sie sicher, dass unnötige manuelle Schritte eliminiert werden, auch wenn es verlockend sein kann, vorsichtshalber einige manuelle Kontrollen beizubehalten.
4. Gehen Sie die Automatisierung in kleinen, umsetzbaren Schritten und mithilfe systematischer Methoden an. Auf diese Weise schaffen Sie schrittweise eine umfassende Automatisierungspraxis.
5. Beginnen Sie mit der Automatisierung einzelner Aufgaben oder Services wie Computing, Netzwerk, Storage oder Provisionierung. Teilen Sie die Automatisierungsprozesse mit anderen und setzen Sie sie systematisch fort.
6. Implementieren Sie Self-Service-Kataloge mit den notwendigen Funktionen für Nutzende, um die Bereitstellung zu beschleunigen.
7. Führen Sie Richtlinien und Prozesse für Metering, Monitoring und Chargeback ein.

Nach einiger Zeit erreichen Sie nicht nur eine integrierte, vollständige Automatisierung, sondern erzielen dadurch auch mehr Effizienz, schnellere DevOps-Pipelines und beschleunigte Innovation.

Schritt 7: Implementieren von Continuous Delivery und modernen Deployment-Methoden

Bei langen Release-Zyklen vergeht zwischen dem Entdecken und Beheben von Software-Bugs oft sehr viel Zeit. Außerdem sind zeitnahe Reaktionen auf geänderte Kunden- und Marktanforderungen fast unmöglich. Bei datenintensiven Anwendungen – wie etwa Mobil-, Web-, Edge-Computing- oder IoT-Anwendungen (Internet of Things) – können ungelöste Bugs viele Nutzende betreffen und zu einem negativen Kundenerlebnis, Sicherheitsproblemen,

„Fortschrittliche Bereitstellungs-techniken verleihen Innovationen Struktur und Klarheit. Mit reifen Technologien lässt sich eine Umgebung schaffen, die echte Experimente, direktes Feedback und klare Analysen ermöglicht. Und bessere Experimente führen zu besseren Innovationen.“

Burr Sutter,
Director of Developer Experience,
Red Hat

einer geringeren Produktivität oder Gewinnverlusten führen. Selbst bei anderen internen Geschäftsanwendungen können durch Software-Bugs verursachte Ausfälle oder Verzögerungen hohe geschäftliche Kosten nach sich ziehen.

Die Methoden der agilen Entwicklung haben sich ständig weiterentwickelt und ermöglichen es Unternehmen heute, Releases frühzeitig und häufig zu veröffentlichen. DevOps- und CD-Ansätze (Continuous Delivery) bieten eine Erweiterung dieser Methoden, indem sie DevOps-, Qualitätssicherungs- und Sicherheitsteams enger zusammenbringen und so die Softwarebereitstellung verbessern. Dadurch lassen sich Code-Änderungen schneller und zuverlässiger in die Produktion bringen, was wiederum ein schnelles Feedback für die Entwicklungsteams bedeutet. Ermöglicht wird diese iterative, schnelle Feedback-Schleife durch den CI/CD-Ansatz. Dabei entsteht durch die Infrastrukturautomatisierung ein durchgängiges, automatisiertes Bereitstellungssystem, das sämtliche Aspekte der Anwendungsbereitstellung abdeckt, darunter automatisierte Tests, Schwachstellen-Scanning, Sicherheits-Compliance und Richtlinienkontrollen. Durch die automatisierten Bereitstellungs-Pipelines sollen Updates zur Verfügung gestellt werden, ohne dass dabei die betriebliche Kapazität beeinträchtigt wird. Das wiederum sorgt für geringere Risiken bei der Bereitstellung.

Bevor CD ([Continuous Delivery](#)) erzielt werden kann, müssen Unternehmen zunächst CI (Continuous Integration) ermöglichen. CI-Lösungen sind Build-Systeme, die verschiedene Quellkontroll-Repositories auf Änderungen überwachen, relevante Tests durchführen und die aktuelle Version der Anwendung anhand der einzelnen Quellkontrolländerungen automatisch entwickeln.

Mit fortschrittlichen Bereitstellungsmustern sollen das Risiko von Software-Releases gemindert werden und eine Umgebung entstehen, die ein Experimentieren mit kontrollierten Ergebnissen ermöglicht – und zwar ohne unbeabsichtigte negative Konsequenzen für Kunden. Dieses Ziel ist für eine verbesserte Innovationsfähigkeit in der gesamten Organisation unerlässlich.

Mit modernen Techniken lässt sich der Bereitstellungsprozess von einer Aktivität außerhalb der normalen Geschäftszeiten mit Service-Zeitfenstern und Ausfallzeiten in eine Routineaufgabe umwandeln, die während der normalen Arbeitszeit und ohne Ausfallzeiten in der Produktion durchgeführt werden kann. So bleibt die Anwendung für Kunden kontinuierlich verfügbar.

Mithilfe dieser Techniken können Organisationen für die Kunden unbequeme neue Deployments eliminieren und dadurch Updates und Releases so oft bereitstellen, wie es die Geschäftslage erfordert. Im Folgenden werden einige der gängigen Deployment-Techniken beschrieben, mit denen sich Ausfallzeiten je nach Use Case der Anwendung komplett vermeiden lassen:

Rolling Deployment ist ein Ansatz, bei dem die Instanzen einer Anwendung nicht auf einmal, sondern einzeln und außerhalb des Load Balancers aktualisiert werden, um keinen Datenverkehr zu empfangen. Sobald das Update abgeschlossen ist, wird die Instanz wieder in den Load Balancer integriert. Der Prozess dauert so lange an, bis sämtliche Instanzen aktualisiert wurden.

Blue/Green Deployment bezieht sich auf die Praktik, zwei identische Umgebungen auszuführen, wobei eine aktiv ist und die andere ruht. Änderungen werden für die ruhende Umgebung ausgeführt und dann in der Produktion verifiziert. Anschließend wird der Live-Datenverkehr auf die aktualisierte Umgebung umgeschaltet. Ein Rollback zur vorherigen Version lässt sich genauso leicht vornehmen wie ein erneutes Umschalten des Datenverkehrs (je nach Zeitaufwand der Datenübertragung).

Auch beim Canary Deployment werden, ähnlich dem Blue/Green Deployment, zwei identische Umgebungen verwendet. Der Unterschied liegt in der Kontrolle des Rollouts. Nachdem ein neues Release bereitgestellt wurde, wird eine kleine Teilmenge der Kunden zum neuen Release gesendet, um es in der Produktion zu testen. Ist dieser Vorgang erfolgreich, wird der Datenverkehr schrittweise zur neuen Version umgeleitet und der Prozess so lange überwacht und geprüft, bis sämtliche Nutzende zur neuen Version migriert worden sind.

Schritt 8: Entwickeln einer hochmodularen Architektur

Microservices stehen für eine Architektur und ein Softwarekonzept, bei dem Anwendungen in ihre kleinsten Komponenten zerlegt werden, die voneinander unabhängig sind. Im Gegensatz zur traditionellen, monolithischen Herangehensweise, bei der die Komponenten in eine einzige Einheit integriert sind, bestehen Microservices aus voneinander getrennten Elementen, die gemeinsam zur Erfüllung der gleichen Aufgaben verwendet werden. Bei diesem granularen, schlanken Konzept der Softwareentwicklung werden ähnliche Prozesse auf mehrere Apps verteilt. Auch wenn Microservice-Architekturen keine spezifische zugrunde liegende Infrastruktur vorschreiben, stellt eine containerbasierte Plattform eine optimale Basis dar.

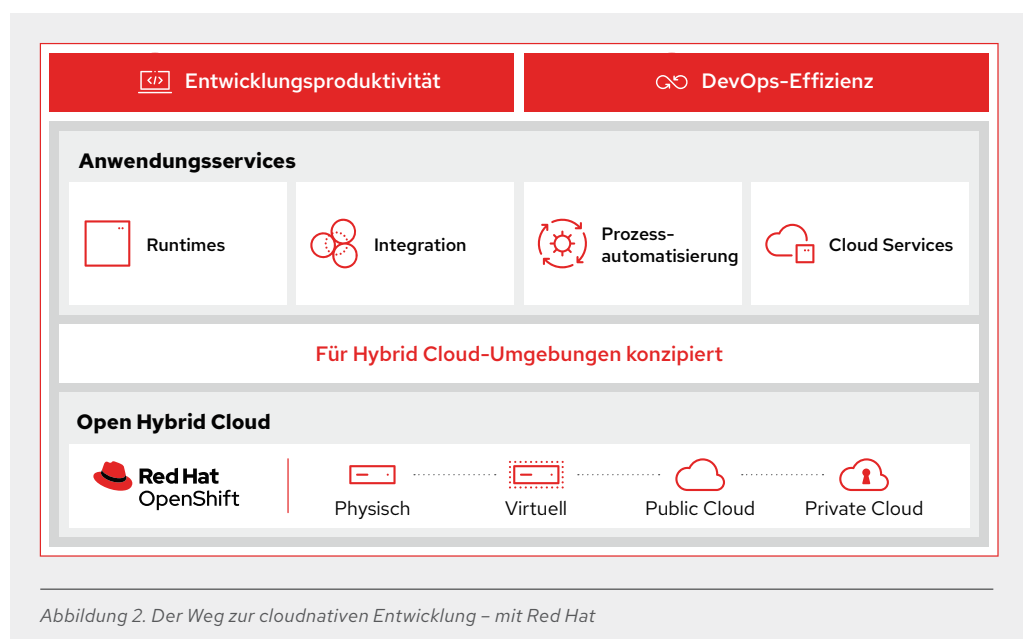
Für große Teams oder Unternehmen, die mehrmals am Tag Produktiv-Deployments durchführen, kann die Entwicklung einer Microservice-Architektur zusätzlich von Vorteil sein. Im Hinblick auf die Architektur müssen bei der Verwendung von Microservices sämtliche Services in eigene Bereitstellungseinheiten aufgeschlüsselt werden. Die Microservices werden dann unabhängig gemanagt und bereitgestellt, wobei für die jeweiligen Lifecycles unterschiedliche Teams zuständig sein können.

Eine weitere Alternative zu Microservices sind Miniservices. Es handelt sich hierbei um eine Sammlung von Services, die nach Domains unterteilt sind und üblicherweise auf einem Anwendungsserver ausgeführt werden. Miniservices bieten eine bessere Agilität und Skalierung – ohne die Komplexität der auf Microservices basierenden Designs oder Infrastrukturen. Dennoch erfordern sie den Einsatz von agilen sowie DevOps- und CI/CD-Ansätzen. Ein moderner Anwendungsserver oder eine Lösung mit mehreren Frameworks, Architekturen und Sprachen in Kombination mit einer containerbasierten Infrastruktur sind daher ideal.

Eine Plattform, die unterschiedliche Frameworks, Sprachen und Ansätze der cloudnativen Anwendungsentwicklung unterstützt – also etwa Microservices, Miniservices oder monolithorientierte Konzepte – ist der Schlüssel für erfolgreiche cloudnative Anwendungen.

Wie Red Hat helfen kann

Unabhängig davon, in welcher Phase der cloudnativen Migration Sie sich befinden und welche Prioritäten Ihr Unternehmen hat – Red Hat kann Ihnen mit seinen Technologien und Services wertvolle Unterstützung bieten.



Manche Organisationen konzentrieren sich nur auf cloudnative Use Cases, während andere mehrere unterschiedliche Use Cases gleichzeitig angehen. Ob Sie nun den evolutionären oder revolutionären Ansatz wählen, eines steht fest: Ihr Weg ist höchst individuell und nicht unbedingt linear. Eine schnelle Markteinführung von Anwendungen erfordert jedoch die richtigen Technologien, DevOps-Praktiken und eine entsprechende Unternehmenskultur.

Red Hat unterstützt Sie auf Ihrem Weg – mit [Red Hat® OpenShift®](#), einer Plattform für die Entwicklung und Orchestrierung cloudnativer Container. Und mit [Red Hat Application Services](#) erhalten Sie ein umfassendes Portfolio für die Anwendungsentwicklung, das auf Red Hat OpenShift ausgeführt wird. Dieses Portfolio beinhaltet Frameworks, Tools und Lösungen für die Entwicklung, Bereitstellung und Skalierung cloudnativer Anwendungen – ein großer Vorteil, wenn Ihr Unternehmen auf schnellere Markteinführungszeiten angewiesen ist. Bei den [Anwendungsservices von Red Hat OpenShift](#) handelt es sich nämlich um gehostete und gemanagte Cloud Services für Red Hat OpenShift, die Teil von [Red Hat Cloud Services](#) sind und ein optimiertes IT-Erlebnis für die Entwicklung, Bereitstellung und Skalierung cloudnativer Anwendungen bieten.

Mit den strategischen Ratschlägen und der umfassenden technischen Expertise von [Red Hat Consulting](#) können Sie außerdem die Komplexität der cloudnativen Entwicklung besser bewältigen. Von [Red Hat Open Innovation Labs](#) bis hin zu Discovery Sessions und Plänen zur Projektimplementierung beraten Sie unsere Consultants in den einzelnen Phasen der Migration zu cloudnativen Anwendungen.



Über Red Hat

Red Hat, weltweit führender Anbieter von Open-Source-Software-Lösungen für Unternehmen, folgt einem community-basierten Ansatz, um zuverlässige und leistungsstarke Linux-, Hybrid Cloud-, Container- und Kubernetes-Technologien bereitzustellen. Red Hat unterstützt Kunden bei der Entwicklung cloudnativer Applikationen, der Integration neuer und bestehender IT-Anwendungen sowie der Automatisierung, Sicherung und Verwaltung komplexer Umgebungen. [Als bewährter Partner der Fortune 500](#)-Unternehmen stellt Red Hat [vielfach ausgezeichnete](#) Support-, Trainings- und Consulting-Services bereit, die jeder Branche die Vorteile der Innovation mit Open Source erschließen können. Als Mittelpunkt eines globalen Netzwerks aus Unternehmen, Partnern und Communities unterstützt Red Hat Unternehmen bei der Steigerung ihres Wachstums und auf ihrem Weg in die digitale Zukunft.

f facebook.com/redhatinc
t [@RedHatDACH](https://twitter.com/RedHatDACH)
in linkedin.com/company/red-hat

**EUROPA, NAHST
UND AFRIKA (EMEA)**

00800 7334 2835
de.redhat.com
europe@redhat.com

TÜRKEI

00800 448820640

ISRAEL

1 809 449548

VAE

8000-4449549