



DIE MIGRATION ZU CLOUDNATIVEN ANWENDUNGEN

Eine 8-Schritte-Anleitung für Ihre digitale Migration

E-BOOK

INHALTSVERZEICHNIS

1. GESCHWINDIGKEIT: DIE TOP-PRIORITÄT FÜR DIGITALE UNTERNEHMEN	3
2. WAS SIND CLOUDNATIVE ANWENDUNGEN?	3
3. TRADITIONELLE VS. CLOUDNATIVE ANWENDUNGEN	4
4. VIER PRINZIPIEN DER ENTWICKLUNG UND BEREITSTELLUNG CLOUDNATIVER ANWENDUNGEN	6
5. DIE MIGRATION ZU CLOUDNATIVEN ANWENDUNGEN: 8 SCHRITTE.....	7
SCHRITT 1: Entwicklung einer DevOps-Kultur nebst Praktiken	7
SCHRITT 2: Beschleunigung bestehender Anwendungen mithilfe schneller Monolithen	7
SCHRITT 3: Nutzung von Anwendungs-Services für eine schnellere Entwicklung	8
SCHRITT 4: Auswahl der richtigen Tools für die richtigen Aufgaben	8
SCHRITT 5: Bereitstellung einer bedarfsabhängigen Self-Service-Infrastruktur	9
SCHRITT 6: IT-Automatisierung zwecks Beschleunigung der Anwendungsbereitstellung	9
SCHRITT 7: Nutzung von CD (Continuous Delivery) -Prozessen und fortschrittlichen Bereitstellungstechniken	10
SCHRITT 8: Entwicklung einer modulareren Architektur	11
6. BUSINESS CASES FÜR CLOUDNATIVE ANWENDUNGEN	12

„Für digitale Unternehmen besteht eine achtmal höhere Wahrscheinlichkeit dafür, dass sie ein gesteigertes Wachstum verzeichnen und dabei immer noch hinter den Digital Natives hinterherhinken.“

Bain-Studie: Für traditionelle Unternehmen: der Weg in die Digitalisierung und die Rolle von Containern

*„Das Adjektiv ‚cloudnativ‘ bezeichnet Anwendungen, Plattformen/ Infrastrukturen und Prozesse, die zusammen ein wirtschaftliches **Arbeiten** bzw. ein **schnelleres Reagieren auf Wandel und die Steigerung der Vorhersehbarkeit ermöglichen.**“*

CHRISTIAN POSTA
CHIEF ARCHITECT BEI RED HAT
UND AUTOR VON *MICROSERVICES*
FOR JAVA DEVELOPERS

QUELLE: INFOQ, „DEFINING CLOUD NATIVE: A PANEL DISCUSSION“, 2017.

1. GESCHWINDIGKEIT: TOP-PRIORITÄT FÜR DIGITALE UNTERNEHMEN

Beim Begriff Digitales Unternehmen denken viele an innovative Technologien wie Mobilgeräte, intelligente Sensoren, tragbare Geräte, Virtual Reality, Chatbots, Blockchain, maschinelles Lernen u.v.m. Für manche steht er auch für die sprunghafte Zunahme der Zahl der digital-nativen Unternehmen, die eine Disruption traditioneller Geschäftsmodelle und eine Destabilisierung von Organisationen und Branchen ausgelöst haben. Für die meisten Firmen setzt das Prinzip des digitalen Unternehmens 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. Da es sich die meisten von ihnen nicht leisten können, ihre Technologiebasis komplett neu aufzubauen oder neue Praktiken und Denkweisen unmittelbar zu übernehmen, setzen sie auf schrittweise, aber grundlegende Verschiebungen in Sachen Kultur, Prozesse und Technologie, um Geschwindigkeit und Agilität zu steigern.

Software bestimmt zunehmend die Art und Weise, wie Nutzer mit Unternehmen interagieren und Organisationen ihre Innovationsentwicklung handhaben, um wettbewerbsfähig zu bleiben. Und genau deshalb ist die Geschwindigkeit der Anwendungsentwicklung und -bereitstellung die neue Priorität digitaler Unternehmen.

Der cloudnative Ansatz schreibt die **Modernisierung bestehender und Entwicklung neuer Anwendungen** basierend auf Cloud-Prinzipien vor. Dies wiederum impliziert die Nutzung von Services sowie die Umstellung auf Prozesse, die zur Steigerung der Agilität und Automatisierung des Cloud Computing optimiert wurden. Dieses E-Book beschreibt im Detail alle Schritte einer erfolgreichen Migration von aktuellen Systemen und Prozessen zu einem cloudnativen Ansatz.

2. WAS SIND CLOUDNATIVE ANWENDUNGEN?

Cloudnative Anwendungen wurden zur Nutzung von Cloud Computing-Modellen entwickelt, sollen Geschwindigkeit, Flexibilität und Qualität steigern und Bereitstellungsrisiken mindern helfen. Trotz seines Namens orientiert sich der cloudbasierte Ansatz nicht daran, **wo** genau Anwendungen genutzt, sondern **wie** sie entwickelt, bereitgestellt und verwaltet werden.

In dieser Hinsicht gleicht er einer Microservices-Architektur. Aber auch wenn Microservices bei der Entwicklung cloudnativer Anwendungen entstehen können, braucht es viele Schritte, um den benötigten Reifegrad zu ihrer Verwaltung in der Produktion zu erreichen. Microservices müssen nicht notwendigerweise von allen Vorteilen cloudnativer Apps profitieren können. Viele Organisationen realisieren diese nämlich, indem sie auf die Entwicklung besserer modularer Monolithen mit den gleichen Prinzipien fokussieren.

Die Evolution hin zu einer cloudnativen Anwendungsentwicklung und -bereitstellung basiert auf einem multidimensionalen Ansatz, der Kultur, Prozesse, Architektur und Technologie beeinflusst. So steht hier also eher der Weg als das Ziel im Vordergrund, ein Zyklus aus fortlaufenden Änderungen, der zuweilen eine große Herausforderung darstellen kann.

3. TRADITIONELLE VS. CLOUDNATIVE ANWENDUNGEN

Die Unterschiede zwischen der cloudnativen und traditionellen Anwendungsentwicklung stehen für die verschiedenen Facetten notwendigen Wandels.

TABELLE 1. TRADITIONELLE VS. CLOUDNATIVE ANWENDUNGEN

	TRADITIONELL	CLOUDNATIV
FOKUS	Langlebigkeit und Stabilität	Speed-to-Market
ENTWICKLUNGS-METHODOLOGIE	Semiagile Wasserfallentwicklung	Agile Entwicklung, DevOps
TEAMS	Isolierte DevOps, QS und Sicherheits-Teams	Kollaborative DevOps-Teams
BEREITSTELLUNGS-ZYKLEN	Lang	Kurz und kontinuierlich
ANWENDUNGS-ARCHITEKTUR	Eng gekoppelt Monolithisch	Lose gekoppelt Service-basiert API-basierte Kommunikation
INFRASTRUKTUR	Server-orientiert Für lokale Umgebungen entwickelt Infrastruktur-abhängig Vertikale Skalierung Für Spitzenkapazitäten vorkonfiguriert	Container-orientiert Für lokale und Cloud-Umgebungen entwickelt Portierbarkeit zwischen Infrastrukturen Horizontale Skalierung On-Demand-Kapazität

3.1 TRADITIONELLE ANWENDUNGSENTWICKLUNG UND -BEREITSTELLUNG

Viele Anwendungen, die für geschäftliche Operationen von Bedeutung sind, wurden nicht im Hinblick auf ein digitales Erlebnis konzipiert. Diese nahtlos gekoppelten Monolithen, die sich durch eine lange Lebensdauer auszeichnen, wurden über eine bestimmte Zeitdauer und gemäß präzisen Spezifikationen entwickelt, die meist lange vor der Bereitstellung definiert wurden.

Dieser Prozess erfolgte größtenteils gemäß der Wasserfall- und/oder sequenziellen Methode über lange Zeitperioden und wurde erst in jüngster Vergangenheit mit semiagilen Praktiken integriert. Die einzelnen Phasen wie Anwendungsentwicklung, Tests, Sicherheits-Compliance, Bereitstellung und Management waren in funktionale Bereiche und separate Teams, Rollen und Zuständigkeiten unterteilt und die Kommunikation zwischen ihnen erfolgte ausschließlich linear.

Bei diesen Anwendungen, die umfangreich, multifunktional und eng gekoppelt waren, wurden Benutzerschnittstelle, verschiedene Anwendungs-Services, der Code für den Datenzugriff sowie andere Komponenten unabhängig von der jeweiligen Technologieumgebung zu einer einzelnen App vereint. So enthielt z. B. eine als eng gekoppelter Monolith gebaute e-commerce Anwendung alle Funktionen für Web-Benutzerschnittstelle, Produktkataloge, Warenkorb, Produktempfehlungen, Produkt-Ratings und -Prüfungen, Bezahlsystem sowie alle anderen Komponenten, die für den Einkauf auf einer e-commerce Website notwendig waren.

Bei der Mehrheit der traditionellen Anwendungen war die Infrastruktur auf die für die Anwendung notwendige Spitzenkapazität vorkonfiguriert. Die Skalierung erfolgte dabei vertikal durch die Steigerung der Hardware-Kapazität des Servers.

Im Jahre 2017 betrug die Zahl der von privaten Rechenzentren in die Public Cloud migrierten Modus 1* Anwendungen, die mithilfe cloudnativer Architekturprinzipien umgeschrieben wurden, noch weniger als 10 %. Bis zum Jahre 2020 wird diese Zahl auf über 50 % ansteigen.

Gartner: „Why You Must Begin Delivering Cloud-Native Offerings Today, Not Tomorrow“, Januar 2018

3.2 CLOUDNATIVE ANWENDUNGSENTWICKLUNG UND -BEREITSTELLUNG

Die Entwicklung cloudnativer Anwendungen fokussiert auf einer schnellen Markteinführung und erfordert dazu agilere, service- und API-basierte Entwicklungs- und CD (Continuous Delivery) -Ansätze. Diese Funktionen werden mithilfe der Kollaboration zwischen Entwicklungs- und Betriebs (DevOps) -Teams, einer modulareren Architektur sowie einer flexiblen Infrastruktur umgesetzt, mit der bedarfsabhängig horizontal skaliert, mehrere Umgebungen unterstützt und Anwendungs-Portierbarkeit bereitgestellt werden können.

Organisationen möchten die von modernen Cloud-Technologien gebotenen Flexibilität und Agilität nutzen, um traditionelle Anwendungen zu Cloud-Umgebungen migrieren, ihre Prozesse beschleunigen und von der bedarfsabhängigen Bereitstellung von Computing-Kapazität profitieren zu können.

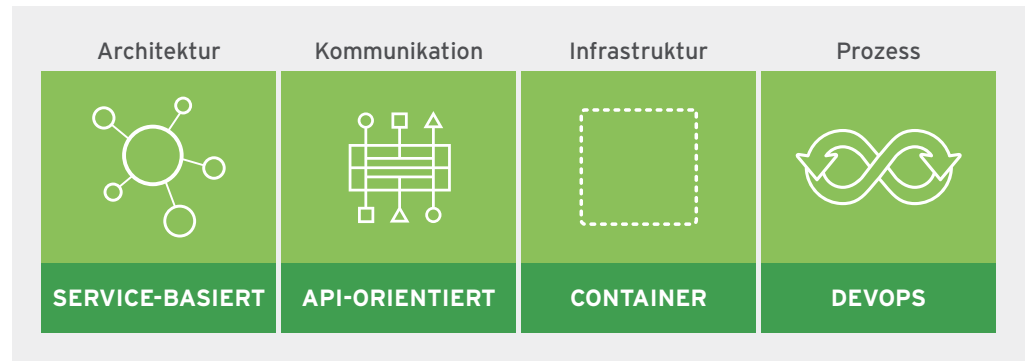
Allerdings sind viele der operationellen Kapazitäten traditioneller Plattformen entweder obsolet, werden für Cloud-Umgebungen nicht benötigt oder aber von diesen bereits bereitgestellt oder operationalisiert. Cloud-Umgebungen erleichtern das Management des Host-Lebenszyklus und unterstützen Organisationen dabei, die Prinzipien einer unveränderlichen Infrastruktur umzusetzen und Hosts auf die Anforderungen einer einzelnen Anwendungsinstanz zu konfigurieren.

Die Migration zu cloudnativen Anwendungen variiert von einer Organisation zur anderen. Die Erstellung von Microservices alleine garantiert weder eine hohe Service-Qualität noch die von digitalen Unternehmen geforderte Bereitstellungsgeschwindigkeit. In ähnlicher Weise führt die Nutzung von Tools, die die agile Entwicklung oder IT-Automatisierung unterstützen, nicht notwendigerweise zu einer Beschleunigung cloudnativer Ansätze. Ein Erfolg kann nur mit der richtigen Mischung aus Praktiken, Technologien, Prozessen und Denkweisen gewährleistet werden.

* *Gartner definiert den Begriff „Bimodalität“ als das Management zweier separater, aber kohärenter Arbeitsweisen, wobei die eine auf Vorhersehbarkeit und die andere auf Erforschung fokussiert. Modus 1 wurde für Bereiche optimiert, die vorhersehbarer und verständlicher sind. Er fokussiert auf etabliertem Wissen und versetzt Legacy-Umgebungen in einen Zustand, der für die digitale Welt geeignet ist. Beim Modus 2 geht es um die Erforschung, das Experimentieren zur Lösung neuer Probleme und die Optimierung in Bezug auf Unsicherheitsfaktoren.*

4. VIER PRINZIPIEN DER ENTWICKLUNG UND BEREITSTELLUNG CLOUDNATIVER ANWENDUNGEN

Der cloudnative Ansatz basiert auf der Entwicklung und Ausführung von Anwendungen, die eine optimale Ausschöpfung des Cloud Computing-Modells auf Basis der vier Grundprinzipien - service-basierte Architektur, API-basierte Kommunikation, container-basierte Infrastruktur und DevOps-Prozesse - vorsieht.



SERVICE-BASIERTE ARCHITEKTUR

Die service-basierte Architektur, wie z. B. Microservices, setzt auf der Entwicklung modularer, lose gekoppelter Services auf. Andere modulare Architekturansätze, wie z. B. Miniservices, die eine lose Kopplung und ein service-basiertes Design integrieren, helfen Organisationen dabei, die Anwendungserstellung zu beschleunigen, und das ohne eine Steigerung der Komplexität.



API-BASIERTE KOMMUNIKATION

Services werden über kompakte technologie-agnostische APIs bereitgestellt, die Komplexität und Aufwand von/für Entwicklung, Skalierbarkeit und Wartung verringern helfen. Diese APIs wiederum können von Unternehmen dazu verwendet werden, intern und extern neue Funktionen und Möglichkeiten zu entwickeln.

Das API-basierte Design erlaubt nur die Kommunikation per Service-Schnittstellenaufbau über das Netzwerk, wodurch die Risiken von direkten Verknüpfungen, Shared Memory-Modellen oder direkter Lesezugriffe auf den Datenspeicher anderer Teams vermieden werden. Mit diesem Konzept lässt sich dazu die Reichweite von Anwendungen und Services bezüglich unterschiedlicher Geräte und Formen erhöhen.



CONTAINER-BASIERTE INFRASTRUKTUR

Cloudnative Anwendungen setzen auf Containern auf, die ein einheitliches Betriebsmodell für beliebige Technologieumgebungen sowie echte Portierbarkeit für alle Umgebungen und Infrastrukturen (Public, Private und Hybrid) bieten. Container-Technologie wiederum nutzt Funktionen zur Betriebssystemvirtualisierung, um die verfügbaren Computing-Ressourcen zwischen mehreren Anwendungen aufzuteilen und gleichzeitig sicherzustellen, dass diese sicher und voneinander isoliert sind.

Cloudnative Anwendungen verwenden eine horizontale Skalierung, wodurch sich Kapazitäten durch das einfache Hinzufügen von Anwendungsinstanzen - nicht selten per Automatisierung innerhalb der Container-Infrastruktur - erweitern lassen.

Der geringere Aufwand sowie die hohe Dichte der Container ermöglicht ihr Hosting auf ein und demselben virtuellen Rechner oder physischen Server und macht sie so ideal für die Bereitstellung cloudnativer Anwendungen.



DEVOPS-PROZESSE

Die Anwendungsentwicklung für cloudnative Ansätze basiert auf agilen Methoden mit Continuous Delivery sowie DevOps-Prinzipien, bei denen die kollaborative Entwicklung und Bereitstellung von Applikationen durch alle am Prozess beteiligten Teams aus Entwicklung, Qualitätssicherung, Sicherheit, IT-Betrieb usw. im Mittelpunkt steht.

5. DIE MIGRATION ZU CLOUDNATIVEN ANWENDUNGEN: 8 SCHRITTE

SCHRITT 1: ENTWICKLUNG EINER DEVOPS-KULTUR NEBST PRAKTIKEN

Die Migration zu cloudnativen Anwendungen erfordert, dass sich DevOps-Teams auf vielfältige Art weiterentwickeln, um Anwendungen schneller und effizienter entwickeln und bereitstellen zu können. So müssen Unternehmen unabhängig von Größe oder Branche die große Bandbreite an Aktivitäten, Technologien, Teams und Prozessen in Betracht ziehen, die eine DevOps-Kultur ausmachen. Um von neuen Technologien, schnelleren Ansätzen und einer nahtlosen Kollaboration profitieren zu können, ist es für solche Organisationen unerlässlich, die Prinzipien und kulturellen Werte von DevOps zu verinnerlichen und ihre Struktur entsprechend auszurichten.

Angesichts der Komplexität der Verwaltung mehrerer verteilter Umgebungen, hochgradig benutzerdefinierter Legacy-Anwendungen sowie neuer Anwendungs-Workloads in einer Zeit der rasanten digitalen Innovation kann sich DevOps für einige von ihnen als große Herausforderung darstellen. Durch die Ausweitung von DevOps-Praktiken auf das gesamte Anwendungs-Portfolio lässt sich ein bis dato noch nicht erschlossenes Potenzial ausschöpfen.

Die Umstellung auf eine DevOps-Kultur hängt nicht nur von Tools und Technologien ab, sondern auch der Bereitschaft und dem Zutrauen, einen vermehrt integrativen und kollaborativen Ansatz der Anwendungsentwicklung und -bereitstellung annehmen zu wollen. Die Kultur der Open Source-Softwareprojekte kann als Vorlage für den Aufbau einer DevOps-Kultur dienen.

Mithilfe von [Red Hat Open Innovation Labs](#) werden Organisationen behutsam in den DevOps-Prozess eingeführt und zu Experimenten, Fast Failures, einer transparenten Entscheidungsfindung sowie dem Einsatz von Prämiensystemen zur Steigerung des Vertrauens und der Verbesserung der Zusammenarbeit ermutigt. Diese Umgebung zur Förderung von Innovationen hilft Teams dabei, Open Source-Technologien zu nutzen, Prototypen schnell zu entwickeln sowie DevOps und agile Workflows anzuwenden.

Erfahren Sie, wie Sie Red Hat Open Innovation Labs auf Ihrem Weg hin zu einer DevOps-Kultur unterstützen kann

[E-BOOK HERUNTERLADEN](#)

„Wenn Sie keine Monolithen mit einer guten Struktur entwickeln können, wieso sollte Ihnen das bei Microservices gelingen?“

SIMON BROWN
[CODINGTHEARCHITECTURE.COM/PRESENTATIONS/SA2015-MODULAR-MONOLITHS](#)

SCHRITT 2: BESCHLEUNIGUNG BESTEHENDER ANWENDUNGEN MITHILFE SCHNELLER MONOLITHEN

Bei der Umstellung auf cloudnative Anwendungen sollten Organisationen nicht den Fehler machen, sich nur auf Neuentwicklungen zu konzentrieren. Viele Legacy-Anwendungen sind für geschäftliche Anwendungen und die Gewinngenerierung immer noch unerlässlich und können nicht einfach ersetzt werden. Dagegen müssen sie mit neuen cloudnativen Anwendungen integriert werden. Aber wie kann man bestehende Monolithen beschleunigen? Indem man seine aktuelle monolithische Architektur zu einer modulareren, service-basierten Architektur und einer API-basierten Kommunikation migriert.

Bevor sie die beschwerliche Aufgabe der Integration monolithischer Anwendungen in Microservices angehen, sollten Organisationen zunächst eine solide Basis für ihre monolithische Architektur schaffen. Auch wenn monolithische Anwendungen wegen ihrer mangelnden Agilität bekannt sind, rührt ihr schlechter Ruf doch in erster Linie von der Art und Weise ihrer Entwicklung her. Mit schnellen Monolithen lassen sich allerdings viele der agilen Vorteile von Microservices erzielen, und das ohne zusätzliche Komplexität oder Kosten.

Die Bewertung eines solchen Entwicklungsansatzes gewährleistet, 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. Die Entwicklung schneller Monolithen auf diese Weise schafft die notwendige Basis für eine erfolgreiche Microservices-Architektur.

Aber auch wenn Anwendungen nicht mithilfe des Ansatzes für schnelle Monolithen konzipiert werden, können sie trotzdem beschleunigt werden, indem man sie zu einer container-basierten Plattform migriert. Das Ergebnis: eine schnellere Entwicklung und ein höherer ROI. Subsequente Integrationen oder Features für diese Monolithen können mithilfe von cloudnativen Techniken und Ansätzen entwickelt werden.

Sie können alternativ aber auch die Monolithen in kleinere Komponenten aufbrechen und einen schrittweisen Ansatz verwenden.

SCHRITT 3: NUTZUNG VON ANWENDUNGS-SERVICES FÜR EINE SCHNELLERE ENTWICKLUNG

Wiederverwendbarkeit war schon immer der Schlüssel zu einer beschleunigten Softwareentwicklung. Das Gleiche gilt auch für cloudnative Anwendungen. Allerdings müssen wiederverwendbare Komponenten für cloudnative Anwendungen, um all ihre Vorteile ausschöpfen zu können, optimiert und in die zugrundeliegende Infrastruktur integriert werden.

Warum Caching Services, Regeln oder Workflow Engines, Integrationskonnektoren, Mobil- und API-Managementfunktionen, Datenvirtualisierungs-Services, Messaging Broker oder serverlose Frameworks neu erstellen, wenn Sie aktuelle Versionen verwenden können, die optimiert und in die zugrundeliegende container-basierte Infrastruktur eingegliedert werden können? Bei diesen Anwendungs-Services, egal ob Software-as-a-Service (SaaS), Plattform-as-a-Service (PaaS) oder iPaaS, handelt es sich in Wahrheit um einsatzbereite Entwickler-Tools.

Cloudnative Anwendungen benötigen ggf. mehrere dieser Service-Typen, um eine beschleunigte Entwicklung und Markteinführung neuer Applikationen gewährleisten zu können. DevOps und Container beschleunigen die Bereitstellung cloudnativer Anwendungen und Anwendungs-Services ihre Entwicklung.

So können Entwickler cloudnativer Anwendungen beispielsweise eigens erstellte Anwendungs-Services nutzen, um nicht nur eine gute Leistung container-basierter Infrastrukturen zu gewährleisten, sondern auch von Plattformfunktionen wie CI/CD Pipelines, rollender und Blue-Green-Bereitstellung, automatische Skalierbarkeit, Fehlertoleranz und mehr zu profitieren.

SCHRITT 4: AUSWAHL DER RICHTIGEN TOOLS FÜR DIE RICHTIGEN AUFGABEN

Neuentwicklungen im Bereich Software, darunter das Internet der Dinge (IoT), maschinelles Lernen, künstliche Intelligenz (KI), Data Mining, Bilderkennung, selbstfahrende Autos und mehr, haben zu einer Steigerung der Zahl der Frameworks, Sprachen und Ansätze der Softwareentwicklung geführt.

Die Entwicklung cloudnativer Anwendungen wird vielfältiger, weil die Wahl der Sprache/des Frameworks immer mehr auf die Anforderungen der jeweiligen Geschäftsanwendung zugeschnitten ist. Die daraus resultierende gesteigerte Komplexität erfordert den Einsatz einer container-basierten Anwendungsplattform, die die richtige Mischung aus Frameworks, Sprachen und Architekturen für die cloudnative Entwicklung unterstützt.

Die cloudnative Entwicklung hängt dazu von der Wahl der richtigen Tools für die richtigen Aufgaben ab. Unabhängig davon, mit welchem Konzept cloudnative Anwendungen bereitgestellt werden - 12-Faktor-Ansatz, domain-basiertes Design, test-basierte Konzipierung und Entwicklung, MonolithFirst, schnelle Monolithen, Miniservices oder Microservices - muss die cloudnative Plattform die richtige Kombination aus Frameworks, Sprachen und Architekturen zur Unterstützung der gewählten

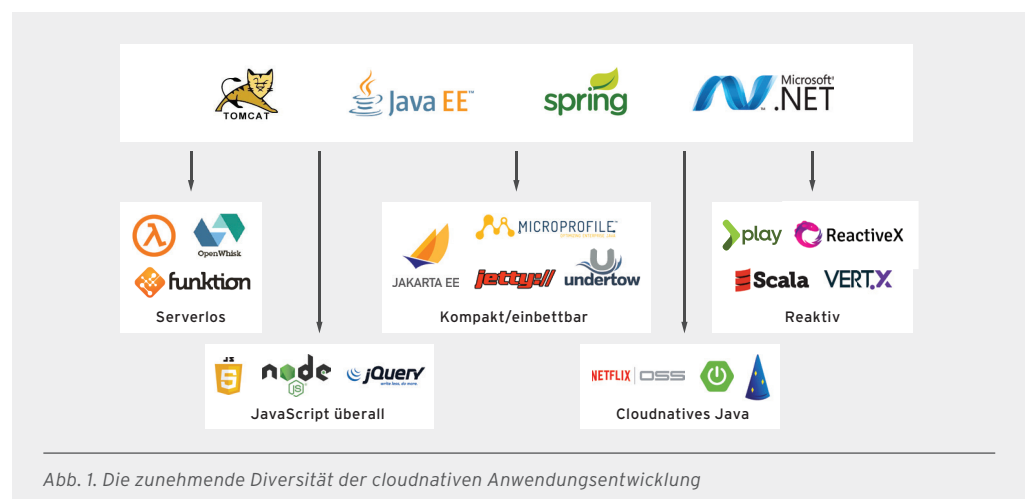


Abb. 1. Die zunehmende Diversität der cloudnativen Anwendungsentwicklung

Entwicklungsanforderungen bieten. Dazu sollte die zugrundeliegende container-basierte Plattform Support für einen Satz ausgereifter Runtimes und Frameworks bereitstellen, der kontinuierlich mit technologischen Änderungen aktualisiert wird.

SCHRITT 5: BEREITSTELLUNG EINER BEDARFSORIENTIERTEN SELF-SERVICE-INFRASTRUKTUR

Agile Methoden unterstützen Entwickler dabei, Software schnell zu erstellen und zu aktualisieren, bieten aber keinen effizienten Mechanismus für einen zeitnahen Infrastrukturzugriff, und zwar egal wo oder wann dieser notwendig wird. Durch die Freigabe von Anwendungen zur Produktion wird die Speed-to-Market entscheidend beeinflusst. Das Erstellen eines Tickets und das wochenlange Warten auf die Freigabe von Ressourcen durch das Ops-Team sind in einer Zeit, in der Infrastrukturen günstig und Ingenieurstalente teuer sind, keine nachhaltigen Geschäftspraktiken mehr.

Eine Infrastrukturbereitstellung, die dem Prinzip des Self-Service und der Bedarfsabhängigkeit folgt, bietet eine überzeugende Alternative zur nicht autorisierten Schatten-IT. Dieses Modell kann aber nur dann effektiv sein, wenn Ops-Teams die notwendige Kontrolle und Transparenz über/für eine oftmals dynamische und komplexe Umgebung zur Verfügung stehen.

Mithilfe von Containern und Container-Orchestrierungstechnologie lassen sich der Zugriff auf die zugrundeliegende Infrastruktur vereinfachen und ein robustes Lifecycle-Management für Anwendungen über verschiedene Umgebungen (darunter Rechenzentren, Private und Public Clouds) hinweg bereitstellen. 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 Entwickler endlich auf die Anwendungsentwicklung konzentrieren können, ohne von den Hindernissen und Verzögerungen der Infrastrukturbereitstellung ausgebremst zu werden.

Die Standardisierung ist ebenfalls ein wichtiger Bestandteil des Self-Service-Modells. Mit ihr können Sie eine einheitliche Automatisierung und Bereitstellung zur Erfüllung Ihrer geschäftlichen Anforderungen gewährleisten. Die Prozessstandardisierung integriert ein Mapping der exakten Reihenfolge der benötigten Ereignisse und Aktivitäten zur Aufgabenerfüllung, darunter die Bereitstellung einer Anwendung in einer neuen Umgebung.

Container bieten dazu Anwendungs-Portierbarkeit, wie die Erstellung cloudnativer Anwendungen, die von jedem Cloud-Anbieter bereitgestellt und ausgeführt werden können. Durch diese Portierbarkeit wiederum haben Sie die freie Wahl unter den Cloud-Anbietern und/oder können von einem zum anderen migrieren, dadurch Kosten optimieren und Multi-Cloud-Anwendungen entwickeln, ohne sie für die APIs spezifischer Anbieter programmieren zu müssen.

Erfahren Sie mehr über die verschiedenen Praktiken und Techniken zur Unterstützung Ihrer Migration zu cloudnativen Anwendungen.

[ENTDECKEN SIE DIE OPEN PRACTICE LIBRARY](#)

SCHRITT 6: IT-Automatisierung zwecks 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, von der Netzwerk-, Infrastruktur-Anwendungsbereitstellung bis zum Konfigurationsmanagement.

Mit IT-Verwaltungs- und -Automatisierungs-Tools können wiederholbare Prozesse, Regeln und Frameworks erstellt werden, die arbeitsintensive menschliche Eingriffe ersetzen oder mindern können, die die Markteinführung ausbremsen. Dazu lassen sie sich auf bestimmte Technologien wie [Container](#) oder Methoden wie [DevOps](#) und größere Bereiche wie [Cloud Computing](#), Sicherheit, Tests, Überwachung und Alarmierung ausweiten. Im Ergebnis ist Automatisierung der Schlüssel zur IT-Optimierung und eine beschleunigte Wertschöpfung.

Anleitungen zur IT-Automatisierung

1. Setzen Sie für die Automatisierung des IT-Betriebs einen unternehmensweiten programmatischen Ansatz ein. Integrieren Sie einen unternehmensübergreifenden, kollaborativen Dialog zur Definition von Serviceanforderungen.
2. Ziehen Sie Sandboxes für das Erlernen von Automatisierungssprachen und -prozessen in Betracht.
3. Denken Sie ernsthaft über die Automatisierung nach. Stellen Sie sicher, dass alle unnötigen manuellen Schritte eliminiert werden, auch wenn es ratsam erscheint, für alle Fälle bestimmte manuelle Kontrollen beizubehalten.
4. Gehen Sie die Automatisierung in kleinen, umsetzbaren Schritten 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 Bereitstellung. Teilen Sie die Ergebnisse mit anderen und setzen Sie den Prozess systematisch fort.
6. Integrieren Sie Self-Service-Kataloge für eine effiziente Nutzung und beschleunigte Bereitstellung.
7. Stellen Sie Richtlinien und Prozesse für Messung, Überwachung und Chargeback bereit.

Irgendwann verfügen Sie dann über eine integrierte und komplette Automatisierung, die Ihrer Organisation eine gesteigerte Effizienz, schnellere DevOps-Integration und beschleunigte Innovation beschert.

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

[E-BOOK HERUNTERLADEN](#)

SCHRITT 7: Nutzung von CD (Continuous Delivery) -Prozessen und fortschrittlichen Bereitstellungstechniken

Zeitintensive Release-Zyklen bedeuten umfassende Verzögerungen zwischen der Entdeckung und Behebung von Software Bugs sowie ein inhärentes Hindernis für eine zeitnahe Reaktion auf geänderte Kunden- und Marktanforderungen. Bei datenintensiven Anwendungen wie z. B. Mobil-, Web oder IoT Apps können solche ungelösten Bugs viele Nutzer beeinträchtigen und zu einem minderwertigen Kundenerlebnis, Sicherheitsproblemen, 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 mittlerweile die häufige Veröffentlichung von Releases in großer Zahl. DevOps- und CD-Ansätze bieten eine Erweiterung dieser Methoden, indem sie die Teams von DevOps, Qualitätssicherung und Sicherheit vereinen und so die Softwarebereitstellung verbessern. Im Ergebnis lassen sich Code-Änderungen dadurch schneller und zuverlässiger zur Produktion verschieben, was wiederum ein schnelles Feedback für die Entwickler ermöglicht. Diese iterative, schnelle Feedback-Schleife wird durch den CI/CD-Ansatz gewährleistet, der die Infrastrukturautomatisierung in ein automatisches End-to-End-Bereitstellungssystem verwandelt, mit dem alle Aspekte des Prozesses abgedeckt werden, darunter automatische Tests, Schwachstellen-Scanning, Sicherheits-Compliance und Richtlinieneinhaltung. Das Ziel der automatisierten Pipelines ist die Bereitstellung von Updates ohne Beeinträchtigung der betrieblichen Kapazität sowie die Reduzierung von Risiken.

Um eine Continuous Delivery (CD) zu erreichen, muss zunächst die Continuous Integration (CI) umgesetzt werden. CI-Lösungen sind Build-Systeme, wie z. B. Jenkins, die verschiedene Quellkontroll-Repositories auf Änderungen überwachen, beliebige relevante Tests durchführen und automatisch Builds der aktuellen Anwendungsversion bedingt durch jede Quellkontrolländerung erstellen.

Erfahren Sie, wie manche moderne Automatisierungstechnologien wie Red Hat Ansible® Automation CI/CD-Prozesse unterstützen

[WHITEPAPER HERUNTERLADEN](#)

Continuous Delivery (CD) ist ein Ansatz der Softwareentwicklung, bei dem Teams wertvolle Software in kurzen Zyklen entwickeln und sicherstellen, dass jederzeit zuverlässige und risikoarme Releases bereitgestellt werden können. Auf diese Weise können Anwendungen fortlaufend mit Nutzer-Feedback sowie durch Märkte und Geschäftsstrategie bedingten Änderungen aktualisiert werden.

Definition von Gartner

„Fortschrittliche Bereitstellungstechniken verleihen Innovationen Struktur und Klarheit. Mit reifen Technologien lässt sich eine Umgebung schaffen, die ein authentisches Experimentieren, direktes Feedback und klare Analysen ermöglicht. Und bessere Experimente führen zu besseren Innovationen.“

BURR SUTTER
DIRECTOR OF DEVELOPER
EXPERIENCE, RED HAT
[REDHAT.COM/DE/ENGAGE/
TEACHING-AN-ELEPHANT-TO-DANCE](https://redhat.com/de/engage/teaching-an-elephant-to-dance)

Mit fortschrittlichen Bereitstellungsstrukturen sollen das Risiko von Software Releases gemindert und die Entwicklung einer Umgebung gefördert werden, die ein Experimentieren mit kontrollierten Ergebnissen ermöglicht - und zwar unter Vermeidung unbeabsichtigter negativer Konsequenzen für den Kunden. Dieses Ziel ist für eine verbesserte Innovationsfähigkeit in der gesamten Organisation unerlässlich.

Mit modernen Techniken ist es möglich, den Bereitstellungsprozess von einer Aktivität außerhalb der normalen Geschäftszeiten mit Service-Zeitfenstern und Ausfallzeiten in eine routinemäßige Werktagsaufgabe umzuwandeln, mit dem sich jegliche Ausfallzeiten in der Produktion vermeiden und eine kontinuierlicher Verfügbarkeit der Anwendungen für den Kunden gewährleisten lassen.

Indem sie mithilfe dieser Techniken unerwünschte Neuentwicklungen eliminieren, können Organisationen Updates und Releases so oft bereitstellen, wie es die Geschäftslage erfordert. Im Folgenden werden einige der gängigen Bereitstellungstechniken beschrieben, mit denen sich Ausfallzeiten je nach Use Case der Anwendung komplett vermeiden lassen:

Rollende Bereitstellung: ein Ansatz, bei dem die Instanzen einer Anwendung nicht auf einmal, sondern separat außerhalb des Load Balancers und/oder ohne Datenverkehr aktualisiert und dann wieder in den Prozess integriert werden. Der Prozess dauert so lange an, bis alle Instanzen aktualisiert wurden.

Blue-/Green-Bereitstellung: die Praktik der Ausführung zweier Umgebungen, von denen eine aktiv ist und sich die andere im Standby befindet. Änderungen werden für die im Standby befindliche Umgebung vorgenommen und in der Produktion verifiziert, wonach der Live-Datenverkehr zur aktualisierten Umgebung umgeschaltet wird. Das Rollback zur vorherigen Version geht genauso leicht vonstatten wie die erneute Umschaltung des Datenverkehrs (je nach Zeitaufwand der Datenübertragung).

Canary-Bereitstellung: ähnelt der Blue-/Green-Bereitstellung in Bezug auf die Nutzung zweier identischer Umgebungen. Allerdings wird der Rollout hier auf unterschiedliche Art und Weise kontrolliert. Nach der Installation einer neuen Release wird ein kleines Kunden-Subset an die neue Release gesendet, um sie 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 alle Nutzer zur neuen Version migriert worden sind.

Einem Elefanten das Tanzen beibringen

[E-BOOK HERUNTERLADEN](#)

SCHRITT 8: ENTWICKLUNG EINER MODULAREREN ARCHITEKTUR

Bei einem microservice-basierten Architekturansatz der Softwareprogrammierung werden Anwendungen in ihre kleinsten Komponenten zerlegt, die voneinander unabhängig sind. Im Gegensatz zur traditionellen, monolithischen Herangehensweise, bei der alle Komponenten miteinander integriert sind, bestehen Microservices aus voneinander getrennten Elementen, die gemeinsam zur Erfüllung der gleichen Aufgaben verwendet werden. Dieses Prinzip der Softwareentwicklung zeichnet sich durch Granularität, Kompaktheit sowie die gemeinsame Verwendung der gleichen Prozesse für mehrere Apps aus. Auch wenn Microservice-Architekturen keine spezifische zugrundeliegende Infrastruktur vorschreiben, wird als optimale Grundlage eine container-basierte Plattform empfohlen.

Die Entwicklung einer microservice-basierten Architektur zeitigt für große Teams oder Produktionsbereitstellungen ggf. einen zusätzlichen Vorteil, und das mehrmals am Tag. Aus Sicht der Architektur gesehen müssen mit Microservices alle Services in eigene Bereitstellungseinheiten aufgeschlüsselt werden. Jeder Microservice wird dann unabhängig verwaltet und bereitgestellt, wobei für ihren Lifecycle unterschiedliche Teams zuständig sein können.

Allerdings erfordert die Bereitstellung einer Microservice-Architektur gewisse Investitionen und Fertigkeiten und kann sich für manche Organisationen als zu disruptiv erweisen. Analysten und Fachleute empfehlen für Microservices einen **MonolithFirst**-Ansatz. Dabei wird die Anwendung, auch wenn Sie als Endprodukt eine Microservice-Architektur beabsichtigen, zuerst als Monolith entwickelt. Ziel hierbei ist es, zunächst ein gutes Verständnis der Domain Ihrer Anwendung zu erlangen, um

dann Kontextgrenzen besser erkennen zu können, die für die Umwandlung in Microservices geeignet sind. Auf diese Weise können Sie technische Schulden vermeiden, wie z. B. Reparaturkosten bei der Entwicklung eines Satzes an Microservices, bevor Sie Domain und Kontextgrenzen der Anwendung kennen.

Eine weitere Alternative zu Microservices sind Miniservices. Es handelt sich hierbei um eine Ansammlung von Services, die nach Domain unterteilt sind und üblicherweise auf einem Anwendungsserver ausgeführt werden. Mit Miniservices lassen sich Agilität und Skalierung verbessern, ohne dass sich dadurch die Komplexität von microservice-basiertem Design oder Infrastruktur steigert. Trotzdem erfordern sie agile, DevOps- und CI/CD-Ansätze, um einen modernen Anwendungsserver oder eine Lösung aus mehreren Frameworks, Architekturen und Sprachen in Kombination mit dem Ideal einer container-basierten Infrastruktur zu entwickeln.

Eine Plattform, die unterschiedliche Frameworks, Sprachen und Ansätze der cloudnativen Anwendungsentwicklung unterstützt - Microservices, Miniservices oder MonolithFirst - ist der Schlüssel für erfolgreiche cloudnative Anwendungen.

6. BUSINESS CASES FÜR CLOUDNATIVE ANWENDUNGEN

Nicht alle Unternehmen haben bezüglich der digitalen Transformation die gleichen Prioritäten. Manche modernisieren ihre bestehende Anwendungsarchitektur und -infrastruktur auf dem Weg zu einer modernen, cloudnativen Umgebung, während andere Innovationen mithilfe neuer Geschäftsmodelle und -anwendungen entwickeln. Unabhängig von Strategie und Business Case aber haben sie alle die gleichen Ziele: mehr Geschwindigkeit, Flexibilität und digitale Bereitschaft. Die gängigsten Use Cases für cloudnative Anwendungen lassen sich in die vier folgenden Kategorien einteilen:



„ Am ersten Tag nach der Akquisition einer neuen Bank konnten wir mit 10 Änderungen in Produktion gehen, und das ohne jegliche Fehler.“

JOHN RZESZOTARSKI
DIRECTOR OF DEVOPS, KEYBANK

Beschleunigung der
Bereitstellung:
von 3 Monaten auf 1 Woche

Bis zum Jahre 2023 werden 90 % aller aktuellen Anwendungen immer noch im Einsatz, allerdings die meisten von ihnen unzureichend modernisiert sein.

Gartner: „Application Modernization should be business-centric, continuous, and multiplatform“, Januar 2018.

„ Dank des Expertensupports für Red Hat JBoss Enterprise Application Platform müssen wir uns um tägliche Geschäftsabläufe keine Gedanken mehr machen.“

FRÜHERER GESCHÄFTSFÜHRER
CORPORATE TECHNOLOGY,
AUSTRALIAN SECURITIES EXCHANGE

60 Mal schnellere
Anwendungs- Neustarts

Geringerer Zeit- und Kostenaufwand für Plattformen, wodurch Ressourcen für eine innovative Service-Entwicklung freiwerden

GESCHÄFTLICHE HERAUSFORDERUNG NR. 1: BESCHLEUNIGUNG DER ANWENDUNGSBEREITSTELLUNG

Ziel:

Beschleunigte Bereitstellung bestehender und neuer Anwendungen an Kunden.

Ansatz:

Container bieten eine gemeinsame Plattform, mit der sich Teams aus Entwicklung, Betrieb, QS und anderen Abteilungen vereinigen lassen, um DevOps unabhängig von Infrastruktur- und Anwendungstechnologie zu nutzen. Bei einem DevOps-Ansatz werden Automatisierung und CI/CD-Praktiken eingesetzt, um Software-Releases schnell und zuverlässig veröffentlichen zu können. Dadurch, dass Bereitstellungsprobleme mithilfe der container-basierten Automatisierung ausgeräumt werden, kann die Anwendungsbereitstellung beschleunigt und an die Geschäftsentwicklung angepasst werden, anstatt umgekehrt.



Kunden-Spotlight:

KeyBank, eine der Top 15 Banken der USA, hat eine Initiative zur digitalen Modernisierung auf den Weg gebracht und konnte so ihre Weberfahrung aktualisieren und eine neue mobile Web App entwickeln. Red Hat OpenShift® hat das Unternehmen dazu bei der Migration seiner monolithischen Anwendungen zu Microservices unterstützt, was die Entwicklung einer automatischen CD Pipeline bzw. die Reduzierung der Bereitstellungszeit von Monaten auf Wochen ermöglicht hat.

GESCHÄFTLICHE HERAUSFORDERUNG NR. 2: MODERNISIERUNG BESTEHENDER ANWENDUNGEN

Ziel:

Beschleunigung der Änderungsgeschwindigkeit durch die **Modernisierung bestehender Anwendungen** zwecks Anpassung an Märkte und Kunden.

Ansatz:

Viele wertvolle Geschäftsapplikationen sind Legacy-Anwendungen, die nicht für die digitale Ära konzipiert wurden. Ein radikaler Komplettaustausch aber ist selten durchführbar oder wirtschaftlich tragbar. Dazu eignen sich nicht alle Legacy-Anwendungen für eine Modernisierung.

In Fällen, in denen die Migration traditioneller Anwendungen zur Cloud möglich ist, wird der Einsatz von Containern empfohlen, mit denen Abhängigkeiten der zugrundeliegenden Infrastruktur entfernt werden können. Im Ergebnis können Anwendungen von der lokalen zur Cloud-Infrastruktur portiert und dort ggf. zu cloudnativen Applikationen refaktorisiert oder umgestaltet werden. Dazu lassen sich mit einer Container-Plattform die Vorteile von Automatisierungsfunktionen sowie DevOps-Praktiken für eine vereinfachte Migration bestehender Anwendungen ausschöpfen.



Kunden-Spotlight:

Die ASX öffnet jeden Tag als erste der großen Börsen ihre Pforten und spielt deshalb eine wichtige Rolle auf dem weltweiten Finanzsektor. Für einen reibungslosen Börsenablauf sind dabei höchste Stabilität, Sicherheit, Leistung und Kosteneffizienz unabdingbar. Dies aber konnte die Legacy-Plattform der Anwendungsserver nicht mehr leisten. Die ASX hat eine Initiative zur Modernisierung ihrer digitalen Plattform mit neuen Technologien gestartet und entschieden, mit Red Hat JBoss® Enterprise Application Platform eine robuste Basis für seinen Anwendungsserver zu schaffen. Als erstes wurde eine der wichtigsten B2B-Webapplikationen des Unternehmens, ASX Online, installiert, die Preise, Firmenankündigungen sowie wichtige Berichte für den Markt bereitstellt und die Erfüllung regulatorischer Richtlinien gewährleistet.

„Red Hat OpenShift Container Platform hat mich wirklich überzeugt. Die Lösung ist innovativ und ermöglicht eine schnelle Bereitstellung und mühelose Kontrolle unserer Container.“

MICHAEL AALBERS
SENIOR TECHNICAL
APPLICATION COORDINATOR,
AMSTERDAM SCHIPHOL

Um 50 % schnellere
Entwicklung neuer APIs

GESCHÄFTLICHE HERAUSFORDERUNG NR. 3: ENTWICKLUNG CLOUDNATIVER ANWENDUNGEN

Ziel:

Beschleunigung der Entwicklung **neuer Anwendungen** für neue Geschäftsmöglichkeiten.

Ansatz:

Änderungen bei geschäftlichen Anforderungen sowie dem Kundenbedarf bieten denjenigen Organisationen neue Möglichkeiten, die in der Lage sind, Ideen umgehend in Services oder Produkte umzuwandeln, die Ergebnisse in der neuen Umgebung zu evaluieren und sie dann entsprechend anzupassen. Der cloudnative Ansatz zur Entwicklung neuer Anwendungen ist ein Schritt hin zur Beschleunigung des Prozesses von der Idee zur innovativen App, der durch Faktoren wie die service-basierte Architektur, API-Integrationen, containerisierte Services und Orchestrierung sowie DevOps-Praktiken, -Automatisierung und Tools unterstützt wird.



Kunden-Spotlight:

Amsterdam Schiphol ist mit 64 Millionen Passagieren pro Jahr der drittgrößte Flughafen Europas. Die Betreiber haben das Ziel ausgegeben, Schiphol bis zum Jahre 2018 zum besten digitalen Airport zu machen. Dazu aber braucht es eine beschleunigte Anwendungsentwicklung auf einer cloudagnostischen Plattform. Ein wichtiger Teil der Strategie der Betreiber ist die Bereitstellung von Services über APIs, darunter ihre Flight API, die Passagieren Informationen wie z. B. Flugsteig, Terminal und Check-in-Zeit zur Verfügung stellt. Das Unternehmen ist dabei, mit Red Hat OpenShift Container Platform eine Self-Service Multi-Cloud-Plattform für sein internes IT-Team sowie Geschäftspartner zu entwickeln, um die Entwicklungszeit für neue Services zu verkürzen.

„Das Aufregendste an dieser Zusammenarbeit ist, dass die Neuentwicklung auf die Natur und Funktionsweise der IT abgestimmt ist. Wir sind dabei, unsere Art des Arbeitens radikal zu verändern, [und] werden auch den Bankbetrieb komplett modernisieren.“

WAYNE MARCHANT
CHIEF INFORMATION OFFICER,
HERITAGE BANK

GESCHÄFTLICHE HERAUSFORDERUNG NR. 4: DIE BESCHLEUNIGUNG DER GESCHÄFTLICHEN INNOVATION

Ziel:

Steigerung der **Innovationsgeschwindigkeit** in der gesamten Organisation für eine kontinuierliche Erfüllung von Geschäftsanforderungen.

Ansatz:

In einer Welt des schnellen Wandels ist Stillstand tödlich. So sind IT-Teams ständig damit befasst, in kurzen Intervallen neue Features und Services bereitzustellen, um Kundenanforderungen zu erfüllen und Mitarbeitern ein intelligenteres Arbeiten zu ermöglichen. Der Erfolg hängt dabei hauptsächlich von einer kontinuierlichen Innovation und daher mehr als nur Tools und Technologien ab. So werden eine neue Kultur, Tools und Prozesse benötigt, die für Innovationen sowie das Experimentieren im gesamten Unternehmen einen revolutionären Ansatz verwenden.



Kunden-Spotlight:

Heritage Bank ist mit 142 Jahren die älteste Finanzinstitution Australiens. Angesichts zunehmender Konkurrenz in der Branche sowie neuen Marktanforderungen musste Heritage Bank einen Weg finden, ihre Softwarebereitstellung zu beschleunigen. Dank eines intensiven Programms im Rahmen der Red Hat Open Innovation Labs konnte die Bank zusammen mit einem hochleistungsfähigen Partner eine innovative Banking-Lösung entwickeln. Und so ist ihr Inhouse Team jetzt in der Lage, kontinuierlich und schnell bessere Software zu entwickeln, und zwar sowohl heute als auch in der Zukunft.

[Video der Heritage Bank](#)

SO KANN RED HAT HELFEN

Je nach der Phase der digitalen und cloudnativen Migration, in der Sie sich befinden sowie Ihren Prioritäten, kann Red Hat mit seinen Technologien und Services wertvolle Unterstützung bieten.

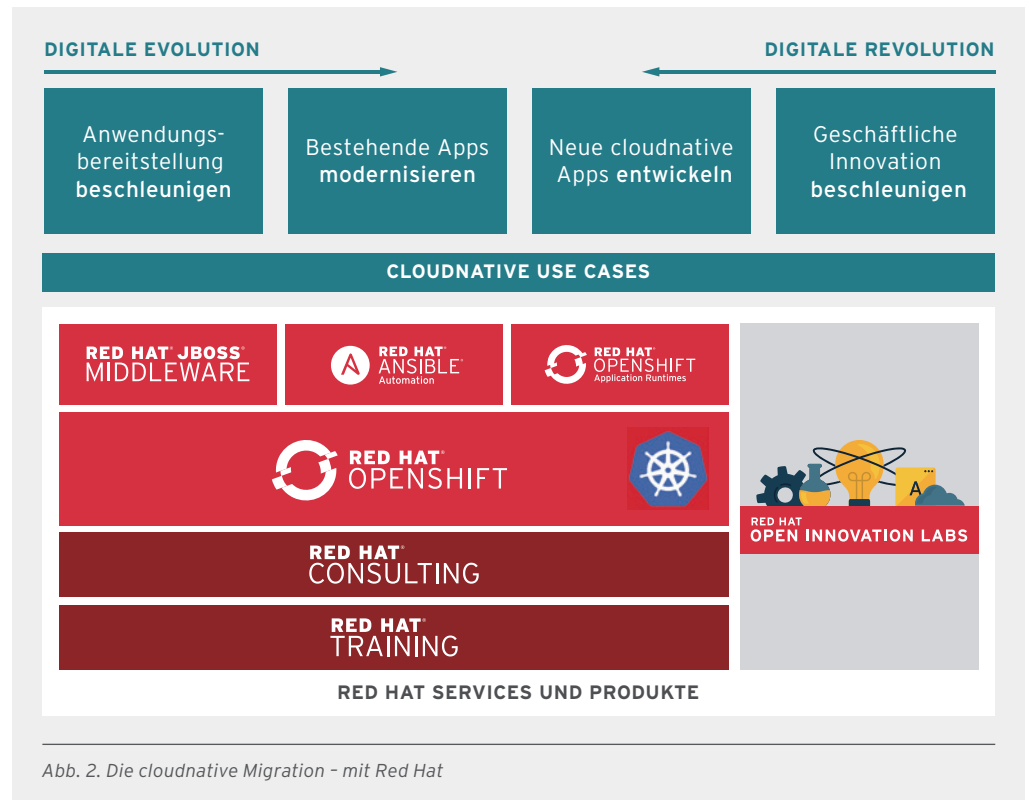


Abb. 2. Die cloudnative Migration - mit Red Hat

Manche Organisationen fokussieren vielleicht nur auf cloudnative Use Cases, während andere gleichzeitig unterschiedliche Szenarien angehen. Ob Sie nun den evolutionären oder revolutionären Ansatz wählen, eines steht fest: Ihr Weg ist höchst individuell und nicht notwendigerweise linear. In jedem Fall gilt, dass eine schnelle Markteinführung von Anwendungen die richtigen Technologien, DevOps-Praktiken bzw. -Kultur erfordert.

Red Hat unterstützt Sie auf Ihrem Weg mit der cloudnativen Container-Entwicklungsplattform [Red Hat OpenShift](#). [Red Hat OpenShift Application Runtimes](#) bietet Open Source-Laufzeiten und Frameworks zur Entwicklung cloudnativer Anwendungen und/oder die Beschleunigung der Entwicklungszeit über containerisierte Runtimes auf OpenShift. Auf OpenShift kann eine Reihe von Red Hat JBoss Middleware Technologien bereitgestellt werden, darunter Ansible [Automatisierungs- und Managementtechnologien](#).

Um die Komplexität der digitalen Transformation auf ein Mindestmaß zu beschränken, bietet [Red Hat Consulting](#) strategische Ratschläge sowie umfassende technische Expertise. Von [Red Hat Open Innovation Labs](#) bis hin zu Discovery Sessions und Projektumsetzungsplänen unterstützen Sie unserer Berater in jeder Phase Ihrer Migration zu cloudnativen Anwendungen.

SIE BEFINDEN SICH MITTEN IN DER MIGRATION ZU CLOUDNATIVEN ANWENDUNGEN?

Erfahren Sie, wie Red Hat Sie auf Ihrem Weg unterstützen kann:

- Finden Sie heraus, wie Red Hat Consulting helfen kann: Erhalten Sie Best Practice- und Planungsanleitungen in Ihrer ganz persönlichen [Discovery Session](#).
- Entdecken Sie unseren [Services Speak Blog](#) mit Einblicken, Tipps und vielem mehr.
- Welchen DevOps-Reifegrad besitzen Sie aktuell? Sind Sie bereit für die Migration zu cloudnativen Anwendungen? Finden Sie es heraus: mit einer [Ready To Innovate](#)-Beurteilung.



ÜBER RED HAT

Red Hat, weltweit führender Anbieter von Open Source Software-Lösungen für Unternehmen, folgt einem community-basierten Ansatz, um verlässliche und leistungsstarke Technologien in den Bereichen Linux, Hybrid Cloud, Container und Kubernetes bereitzustellen. Wir unterstützen Kunden bei der Integration neuer und bestehender IT-Anwendungen, der Entwicklung cloudnativer Anwendungen, der Standardisierung auf unserem branchenführenden Betriebssystem sowie der Automatisierung, Sicherung und Verwaltung komplexer Umgebungen. Dank unserer vielfach ausgezeichneten Support-, Training- und Consulting-Services ist Red Hat ein bewährter Partner der Fortune 500 Unternehmen. Als strategischer Partner für Cloud-Anbieter, Systemintegratoren, Anwendungsanbieter, Kunden und Open Source Communities hilft Red Hat Organisationen auf ihrem Weg in die digitale Zukunft.



facebook.com/redhatinc
@RedHatDACH
linkedin.com/company/red-hat

de.redhat.com
#F12255_0518

EUROPA, NAHOST,
UND AFRIKA (EMEA)
00800 7334 2835
de.redhat.com
europe@redhat.com

TÜRKEI
00800 448820640

ISRAEL
1 809 449548

VAE
8000-4449549