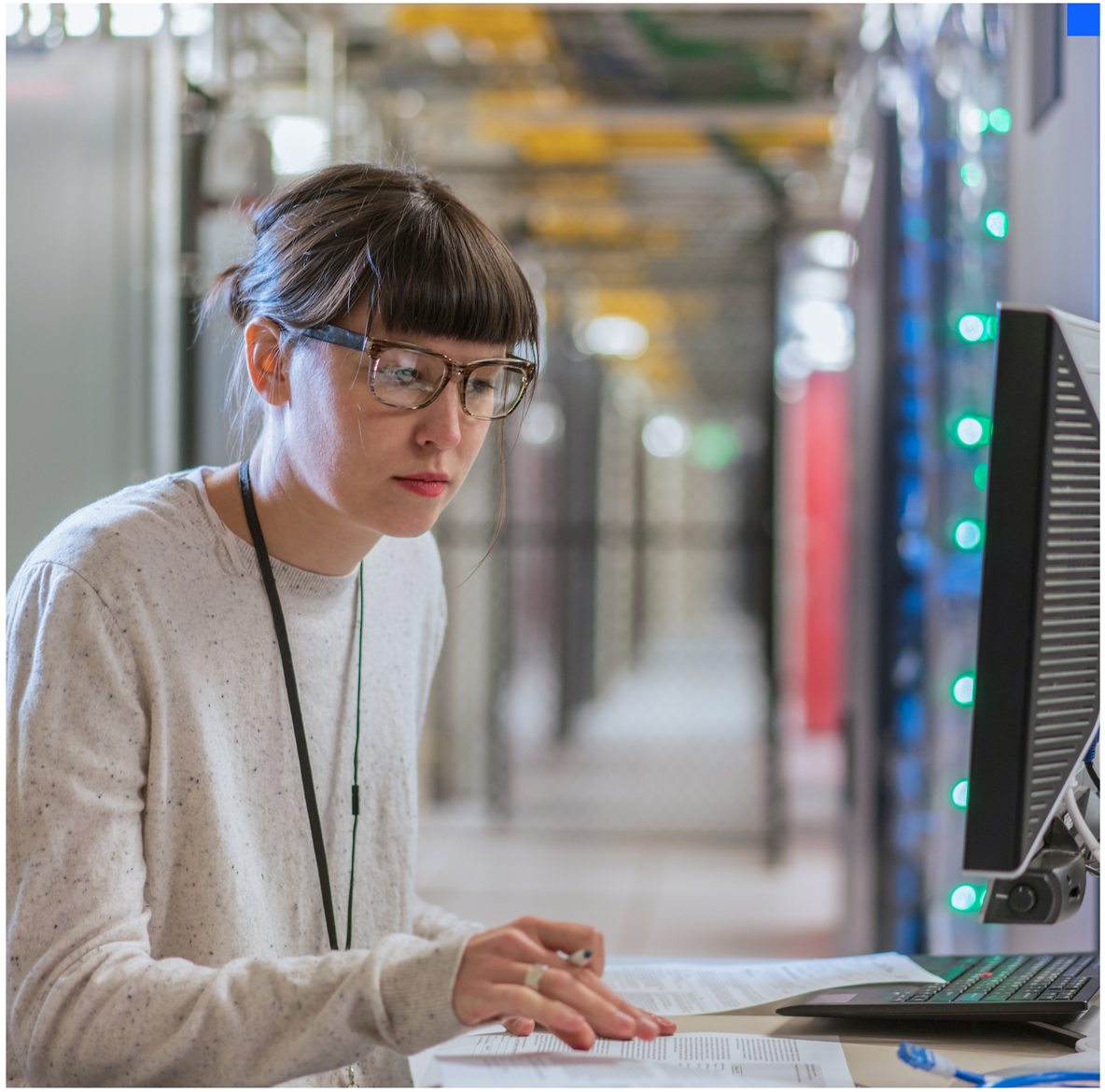
IBM Instana

Beobachtbarkeit zur Optimierung der Anwendungsmoder-nisierung steigern



Inhalt

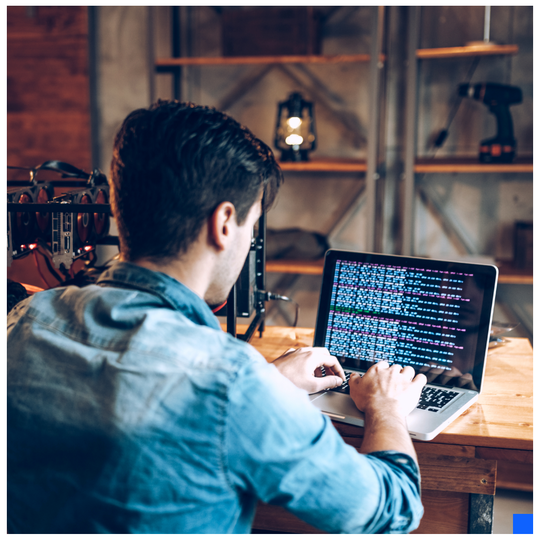
01

Übersicht

Alle sind sich einig: Anwendungsmodernisierung ist ein brandaktuelles Thema. Eines, mit dem sich die meisten Unternehmen bereits beschäftigen oder in naher Zukunft auseinandersetzen werden. Die Anwendungsmodernisierung ist ein komplexer Prozess, der eine Bewertung des aktuellen Anwendungsportfolios, die Bestimmung der wichtigsten zu modernisierenden Anwendungen und die Ausarbeitung eines kosteneffizienten Plans für die Umsetzung einbezieht. Leider ist die Anwendungsmodernisierung in der Realität meist komplexer, als von Developer Advocates in den sozialen Medien vermittelt wird.

Containerbasierte Microservice-Anwendungen setzen sich zwar zunehmend durch, aber viele traditionelle Anwendungen müssen noch immer migriert werden. Oft bilden On-Premises-Infrastrukturen und monolithische Anwendungen den Ausgangspunkt, während cloudnative Anwendungen das ultimative Ziel sind. Jedes Unternehmen muss jedoch auf der Grundlage seines Status quo, der jeweiligen Anforderungen und der Durchführbarkeit seinen eigenen Weg finden. Migration findet schon jetzt in erheblichem Umfang statt und wird in den kommenden Jahren voraussichtlich weiter zunehmen.

Elemente der Anwendungs-modernisierung



Code
Das Schreiben und Aktualisieren von Code für eine Anwendung ist eine mühselige Angelegenheit. Allein die Fertigstellung ist eine Herausforderung. Murphys Gesetz besagt jedoch, dass die Anwendung zu einem unbestimmten Zeitpunkt aktualisiert werden muss, an dem der Berechtigungsinhaber für den Code ausgerechnet nicht erreichbar ist. Es kann auch sein, dass der Eigner der Anwendung das Unternehmen verlässt und daher nicht mehr zur Verfügung steht.

Der Nachfolger könnte dann eine böse Überraschung erleben. Benutzerdefinierter Code, persönliche Kommentare, merk-würdige Strukturierung, das Fehlen von Kommentaren, die Verwendung von nicht akzeptierter Syntax und mangelhafte Formatierung können einfache Aufgaben in echte Detektivarbeit verwandeln. Durch veraltete Java-Sprache könnten kritische Integrationen auf Ihrer Website deaktiviert werden. Und das könnte dem Geschäft Schaden zufügen.

Anwendungskomponenten
Eine monolithische Anwendung auf Bare-Metal-Servern lässt sich nicht einfach wie durch Zauberhand in ein cloudnatives Modell verwandeln. Die Anwendung muss in Microservices zerlegt, ihre Komponenten müssen in Containern untergebracht und diese Container müssen mit einem Tool wie Kubernetes orchestriert werden.

Allein dieser Prozess ist sehr aufwendig, stellt aber zumindest einen Ausgangspunkt für Anwendungskomponenten dar. Je komplexer die Komponenten sind, desto mehr Zeit benötigen Entwickler und Ingenieure, was häufig eine gewisse Dringlichkeit für den Umstieg auf einen Cloud-Hosting-Service erzeugt. Überlassen Sie ihnen die Verwaltung der Infrastruktur, sodass sich Ihr Team um die Verwaltung der Anwendungen kümmern kann.

Infrastruktur
Erinnern Sie sich an die Aussage, dass Code-, Komponenten- und Infrastrukturänderungen parallel erfolgen können? Bei der Migration einer Anwendung in eine Cloudumgebung stehen Ihnen mehrere Optionen in Bezug auf die Anwendungskomponenten der Infrastruktur zur Verfügung.

- **Rehosting.** Durchführen eines Lift-and-Shift für die Anwendung gefolgt von ihrem erneutem Hosting in der Cloud.
- **Replatforming.** Hosten der Anwendung in der Cloud und Durchführung geringfügiger Änderungen an der Infrastruktur.
- **Repurchasing.** Erneuter Erwerb der Anwendungen, diesmal jedoch als Software-as-a-Service-Versionen (SaaS).
- **Refactoring.** Umschreiben einer Anwendung zu einem Teil oder in ihrer Gesamtheit und ggf. Bereitstellung einer neuen Anwendungsarchitektur, wie z. B. die Konvertierung einer SOA-Anwendung (serviceorientierte Architektur) in Container.

03

Antritt der Journey zur Modernisierung

Wo fängt man am besten an? Das hängt davon ab, wo Sie sich gerade befinden. Im Allgemeinen ist es erstrebenswert, sich auf der Gleitskala zum nächsten Schritt in Richtung „Cloudnativ“ zu begeben. Jeder dieser Schritte ist jedoch mit einer Reihe ganz eigener Herausforderungen verbunden.

Sie befinden sich hier:	**Ihr nächster Schritt:**
Bare Metal, vor Ort	Virtualisiert, vor Ort
Virtualisiert, vor Ort	Rechenzentren
Rechenzentren	Private Cloud
Private Cloud	Public Cloud
Public Cloud	Hybrid Cloud
Hybrid Cloud	Cloudnativ

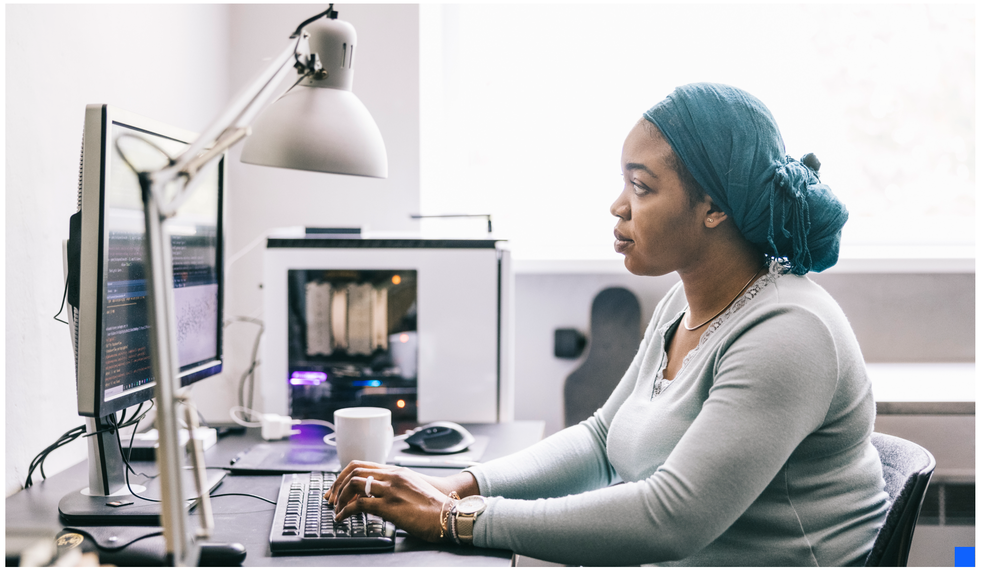
Zur Bestimmung des Erfolgs der einzelnen Prozessschritte ist sowohl eine Benchmark-Messung der Anwendungsleistung vor der Modernisierung, als auch ihre erneute Messung nach der Modernisierung zum Vergleich erforderlich. Bei manchen Tools zur Überwachung der Anwendungsleistung, d. h. APM-Tools (Application Performance Monitoring) kann es sich mitunter als schwierig erweisen, die Transparenz zu wahren und in jeder Phase dieselbe Gruppe von Metriken zu erfassen.

Begriffsdefinition von „traditionell“ für dieses E-Book
Für den Begriff „traditionelle Anwendung“ gibt es mehrere Definitionen. Für das vorliegende E-Book liefern wir eine Definition des Begriffskonzepts „traditionelle Anwendungen“, durch die sich die Verwendung von „traditionell“ und „traditionelle Anwendung“ im Gegensatz zu der längeren, vollständig qualifizierten Bezeichnung einfacher gestaltet.

Im Rahmen der vorliegenden Betrachtung ist unter „traditionelle Anwendung“ eine Form einer Java Enterprise Edition-Anwendung (Java EE) oder einer .NET-Anwendung (dot net) zu verstehen, die entweder als Monolith oder in einer SOA-Umgebung eingesetzt wird. Zu diesem Zweck werden in diesem E-Book die APM-Tools der 1. und 2. Generation gegebenenfalls als „traditionelle“ Überwachungstools oder als „traditionelle“ APM-Tools bezeichnet. Bei Erörterungen anderer traditioneller Tools wird ausdrücklich auf diese hingewiesen.

04

Vorteile der Modernisierung von Anwendungen



Die Migration von Code, Anwendungskomponenten und Infrastruktur ist mit komplexen Entscheidungen verbunden und erfordert eine sorgfältige Planung. Sind Sie trotz dieser Erschwernisse bereit, sich der Herausforderung zu stellen? Erfahren Sie hier, warum Sie diese Frage bejahen sollten.

Die Modernisierung von Anwendungen ist ein wesentlicher Schritt bei der Journey zur digitalen Transformation, der bei der Automatisierung manueller Prozesse hilft. Automatisierung kann sich sowohl auf die Serviceleistung als auch auf die Maximierung der Personalressourcen positiv auswirken. Prozesse, die häufige manuelle Eingriffe erfordern, können langsamer, fehleranfälliger und kostenintensiver im Betrieb sein. Die folgenden Gründe sprechen für die Modernisierung von Anwendungen:

- Optimierung der Entwicklerzeit.
- Beschleunigung von Innovationen und der CI/CD-Pipeline (Continuous Integration/ Continuous Delivery).
- Reduzierung von Kosten und Redundanz.
- Beschleunigung der Reaktion auf Bedürfnisse der Benutzer.
- Schaffung von mehr Homogenität im Unternehmen und in den Prozessen.
- Abschaffung veralteter Umgebungen.
- Beschleunigung bei der Lösung von Problemen.

Optimierung der Entwicklerzeit
Viele traditionelle Anwendungen basieren zwar auf Java oder .NET, sind aber nicht so ausgereift wie neuere Anwendungen. Updates und routinemäßige Wartung können problematisch für Entwicklungsteams sein. Der Mangel an Langzeitwissen kann bewirken, dass durch das Fehlen nur eines Entwicklers ein Engpass für die gesamte Arbeit an einer Anwendung entsteht.

In dem Maß, in dem ein Entwickler einen anderen ablöst, kann der jeweils neue Entwickler eine Anwendung mit Fixes und Patches im eigenen Stil zusammenstellen. Infolgedessen ist die Anzahl von Inkonsistenzen überwältigend, wie z. B. Links zu JavaScript-Funktionsdateien an manchen Stellen und unmittelbar in den Code eingefügtes JavaScript an anderen Stellen. Eigentlich einfache Korrekturen erfordern so das stundenlange Durchkämmen des Codes, um die Nadel im Heuhaufen ausfindig zu machen.

Wenn Entwickler bei der Entwicklung Stringenz und die Prinzipien der Benutzerfreundlichkeit (UX-Prinzipien) auf ihren Code anwenden, ist es für alle Beteiligten einfacher, mit dem Code zu arbeiten oder – noch besser – wird dadurch die Automatisierung von Routinearbeiten sogar noch begünstigt. Kommentare definieren, worauf die einzelnen Abschnitte in der Anwendung jeweils verweisen. Scripts und Stylesheets geben die Funktionen dieser Elemente an. Und ganz gleich, ob der urhebende Entwickler Tabulatoren oder Leerzeichen verwendet, ist Konsistenz gegeben.

Das Ergebnis ist, dass Sie weniger Zeit damit verbringen, aus dem Code schlau zu werden, und mehr Zeit für Ihre eigentliche Arbeit, weniger Stress, mehr Erfolg haben und das Team als Ganzes eine bessere Leistung erbringt.

Beschleunigung von Innovationen und der CI/CD-Pipeline
In den meisten Fällen sind ältere Anwendungen weniger ausfallsicher. Demzufolge müssen Entwickler oft zu viel Zeit für Notfälle aufwenden, um den fortlaufenden Betrieb der Anwendung sicherzustellen, anstatt innovative Lösungen für Kunden zu entwickeln.

Wenn Entwickler in der Anfangsphase eines Entwicklungsprojekts mehr Arbeit erledigen, setzen sie einen positiven Kreislauf der Ressourcenzuweisung in Gang, der dazu beiträgt, dass Projekte termingerecht abgeschlossen werden und weniger Fehler auftreten. Geraten Zeitpläne hingegen ins Wanken, so wird dadurch ein Teufelskreis angestoßen, in dem sich Entwickler immer und immer wieder mit der Behebung von Notfällen herumplagen müssen. Das bedeutet, dass Entwickler in jeder Phase Tests manuell durchführen, anstatt kontinuierlich automatisierte Tests auszuführen. Sie instrumentieren den Code manuell, anstatt ihn zu automatisieren. Sie dokumentieren mühsam ihre Infrastruktur, anstatt sie unter Verwendung eines entsprechenden Tools automatisch erkennen zu lassen. Dieser Prozess kann irreführende Testergebnisse zur Folge haben. In diesem Fall könnte der Code in den Produktivbetrieb gelangen, bevor er die Reife dafür besitzt.

Unternehmen setzen bei der Anwendungsentwicklung und -bereitstellung Architekturen ein, die auf die Ausführung von Anwendungen ausgelegt sind, wie z. B. Container, Microservices, Serverless Computing und Kubernetes-Orchestrierung. Mit Automatisierungs- und Orchestrierungstools wie Jenkins, Red Hat Ansible, Chef und Puppet kann dazu beigetragen werden, dass sich die Teams jeweils auf dem gleichen Stand befinden. Jede bedeutende öffentliche Cloud-Plattform – Amazon Web Services (AWS), Microsoft Azure und Google Cloud – bietet die Verarbeitung mit durch Kubernetes verwalteten Containern als Service an.

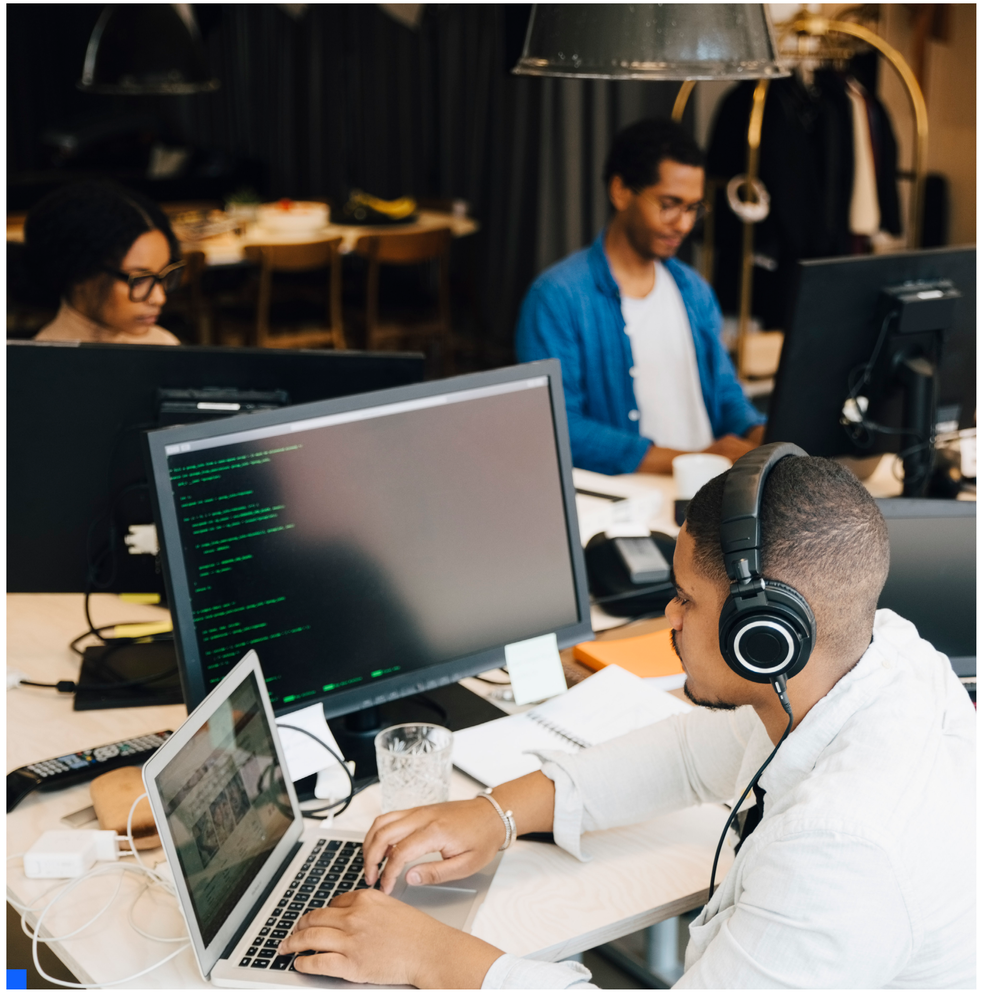
Reduzierung von Kosten und Redundanz
Einer der intelligenten Schritte, die Unternehmen vor dem Umstieg auf Cloud-Hosting unternehmen, ist der Betrieb von Rechenzentren. Globale Rechenzentren eignen sich hervorragend für Lastverteilung und Redundanz, können sich jedoch als sehr kostenintensiv erweisen.

Rechenzentren sind außerdem oft mit langfristigen Verträgen verbunden. Die Migration in die Cloud ist ein zeitaufwendiger Prozess. Um nicht in Zeitdruck zu geraten, sollten Sie mindestens ein Jahr vor Ablauf des Vertrags mit Ihrem Rechenzentrum mit der Migration beginnen.

Und falls Sie dachten, dass Rechenzentren Redundanz bieten, versuchen Sie es doch einmal mit mehreren Dutzend Containern mit Microservices in der Cloud. Das ist echte Redundanz. Einige große Unternehmen verwenden nach wie vor On-Premises-Architekturen und sorgen dafür, dass diese funktionieren. Es ist jedoch unwahrscheinlich, dass sich der Trend hin zu cloudbasierten Services umkehrt.

Beschleunigung der Reaktion auf Bedürfnisse der Benutzer
Die Anforderungen und Anliegen von Kunden und Mitarbeitern sollten die Grundlage Ihres Handelns und damit Ihr Leitfaden sein. Kunden haben heute nur noch wenig Geduld für Verzögerungen oder Unterbrechungen – und zwar sowohl online als auch bei Apps.

Wenn Ihre Webseite beispielsweise zu lange zum Laden braucht, ist die Wahrscheinlichkeit groß, dass der Kunde einfach woandershin hin klickt. Für Online-Händler kann dies zu erheblichen Umsatzeinbußen durch entgangene Verkäufe führen. Um optimale Ergebnisse zu erzielen, sollten Sie einen Zeitraum vom 2 bis 4 Sekunden für das Laden einer Seite oder den Abschluss eines App-Vorgangs anstreben.



Schaffung von mehr Homogenität im Unternehmen und in den Prozessen
Erinnern Sie sich, wie Entwickler ihre eigenen Fehlerkorrekturen und Patches erstellen, um Code zu flicken? Das hat praktische Konsequenzen. Verschiedene Anwendungen sind unterschiedlich kodiert, was es schwierig macht, sie an Mindeststandards zu messen. Wie kann man einen Standard auf eine Anwendung anwenden, wenn diese keine der Komponenten verwendet, die standardisiert werden sollen? Unternehmen modernisieren ihre Anwendungen, um ihnen eine einheitliche Ausrichtung zu verleihen, sodass sie mit Vorschriften in Einklang gebracht werden können.

Abschaffung veralteter Umgebungen
Stellen Sie sich folgendes Szenario vor: Eine Anwendung basiert auf Java 11. Die aktuellste Version ist Java 20. Wenn das Team ein Upgrade der Version durchführt und den Code dementsprechend verbessert, warum nutzt es dann nicht die Gelegenheit, um die Anwendung anderweitig zu modernisieren? Zum Beispiel durch Zerlegen in Microservices. Durch Verschieben in die Cloud. Durch Umwandeln in eine cloudnative Anwendung.

Lösung von Problemen
Verteilte Komponenten versetzen Incident Responder in die Lage, bestimmte Container oder andere Komponenten individuell zu inaktivieren, ohne sie in ihrer Gesamtheit abzuschalten. Mit anderen Worten: Verteilte, cloudbasierte Microservice-Anwendungskomponenten beseitigen den Single Point of Failure, der bei monolithischen Anwendungen einen Schwachpunkt darstellt. Dank Containern sind Entwickler außerdem in der Lage, Teile einer Anwendung zu isolieren und zu testen, ohne die gesamte Anwendung zu beeinträchtigen.

Orchestratoren wie Kubernetes vereinfachen zudem die Durchführung von Korrekturen an Anwendungen, ohne dass es hierdurch beim Kunden zu Ausfallzeiten kommt, was sowohl für den Anbieter als auch für den Benutzer ein riesiger Gewinn ist.

05

Warum Unternehmen fälschlicherweise nicht modernisieren

Unternehmen zögern die Anwendungsmodernisierung meist aus zwei Hauptgründen hinaus: kurzfristige Unannehmlichkeiten und die Schwierigkeit der Fortschrittsmessung. Die Abwägung zwischen kurzfristigen Unannehmlichkeiten und langfristigen Vorteilen kann eine Herausforderung darstellen, denn der erste Punkt bietet mehr Sicherheit, während der zweite sich als Lotteriespiel erweisen kann. Die Messung des Fortschritts im Verlauf des gesamten Modernisierungsprozesses als Journey kann ein erhebliches Hindernis darstellen.

Kurzfristige Unannehmlichkeiten
Nachfolgend finden Sie eine kurze Liste der Gründe dafür, warum Unternehmen Projekte zur Anwendungsmodernisierung herauszögern oder abbrechen.

- **Bedenken der Kunden.** Der Modernisierungsprozess ist in der Regel mit gewissen Unterbrechungen verbunden und in Branchen wie dem Finanz- und Gesundheitswesen geben Datenschutzbedenken und Unterbrechungen Anlass zur Sorge.
- **Zeit.** Erinnern Sie sich an die Migration? Jeder dieser Schritte kann gut und gerne bis zu einem Jahr in Anspruch nehmen. Allein die Zerlegung einer monolithischen Architektur in Microservices erfordert ein Verständnis dafür, welche Funktionalität Sie separat bereitstellen möchten, und macht eine Aufgliederung des Prozesses in einzelne Schritte notwendig, damit Sie den Fortschritt messen können.
- **Geld.** Sowohl die Upgrades von Code als auch die Änderungen an der Architektur können kurzfristig betrachtet kostenintensiv sein. Wenn Sie die Kosten für die Entwicklerzeit, die für Notfälle, Neueinstellungen, Fluktuation, Schulungen und geplante Code-Freezes aufgewendet wird, zusammenrechnen, können sich die versteckten Kosten schnell summieren.

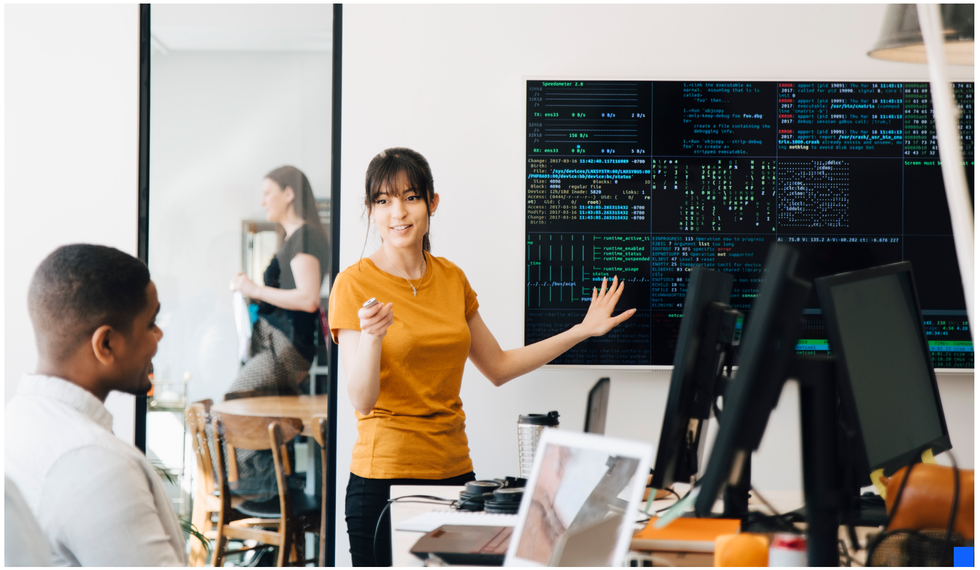
Unternehmen, die darauf hoffen, keine weiteren Millionenbeträge für den Betrieb von Rechenzentren ausgeben zu müssen, zahlen am Ende oft noch zusätzlich für den Hosting-Service.

Unfähigkeit des Benchmarking und des Leistungsvergleichs
Der berühmte Managementberater Peter Drucker hat einmal geschrieben: „Was man nicht messen kann, kann man nicht verbessern.“

Wie Sie feststellen werden, ist Modernisierung nicht einfach. Unternehmen, die monolithische Anwendungen verwenden, können traditionelle APM-Tools nutzen, um die entsprechenden Metriken zu messen. Sobald die Anwendungskomponenten jedoch zerlegt worden sind und in Containern ausgeführt werden, sind manche traditionelle APM-Tools nicht mehr in der Lage, den Inhalt von Containern zu erkennen. Beim Cloud-Hosting ist sogar eine noch geringere Anzahl von traditionellen APM-Tools imstande, diese Aufgabe der Beobachtbarkeit zu erfüllen.

06

Die Rolle der Beobachtbarkeit bei der Anwendungsmodernisierung



Bei der Planung und Durchführung der Anwendungsmodernisierung und der digitalen Transformation ist es von entscheidender Bedeutung, Beobachtbarkeit in Ihr Portfolio miteinzubeziehen. Ein Kontrollmechanismus für die Anwendungsqualität ist eine Grundvoraussetzung für die Durchführung von Modernisierungen.

- Führen Sie ein Benchmarking durch und ziehen Sie Vergleiche mit denselben Messwerten.
- Bei einer containerisierten Docker-Lösung sind Messungen vor Ort einfacher.
- Ein traditionelles APM-Tool kann keine Messungen für cloudnative Apps durchführen.

Bevor Sie die Modernisierung einer Anwendung planen, ist die Durchführung eines Benchmarkings von zentraler Bedeutung, damit Sie bestimmen können, wo die Leistung Ihrer Anwendung schwach ist, wo sie stark ist und wo sie genau richtig ist. In diesem Zusammenhang sollten einige grundlegende Best Practices berücksichtigt werden.

Benchmarking für die Anwendungsqualitätskontrolle

Beim Benchmarking und Leistungsvergleich in jeder Phase ist es wichtig, die gleichen Messungen vorzunehmen, damit die gezogenen Vergleiche auch Gültigkeit haben. Das Benchmarking ist ein wesentliches Feature, das besondere Aufmerksamkeit verdient. Es stellt die einzige vertretbare Methode für die Verfolgung der Anwendungsqualitätskontrolle dar.

Der erste Schritt bei der Modernisierung einer monolithischen Anwendung besteht darin, ihre Funktionalität im Rahmen einer Entkopplung in Komponenten und Microservices zu zerlegen. Der effektive Betrieb von Microservices kann jedoch mit Risiken einhergehen. Um diese Risiken zu mindern und möglichst umfassende Metriken und Anforderungsverfolgungen bereitstellen zu können, sollten Sie daher mit der Entkopplung einfacher Edge-Services beginnen.

Der nächste Schritt besteht darin, Microservices in Container innerhalb einer virtualisierten Umgebung zu platzieren, bevor die Komponenten anschließend in Rechenzentren und letztendlich in die Cloud verschoben werden. Dafür sind neue Authentifizierungsservices, Sensoren für Cloudarchitekturen und Integrationen mit APIs erforderlich. Eine veraltete APM-Lösung wird diesen Aufgaben möglicherweise nicht gerecht.

Beobachtbarkeit: Mehr als nur Metriken, Traces und Protokolle

Beobachtungstools, die durch Metriken, Traces und Protokolle definiert sind, liefern möglicherweise nicht genügend Details zu traditionellen Anwendungen. Dies ist einer von mehreren Gründen, warum Sie eine Observability-Plattform in Betracht ziehen sollten, die in der Lage ist, die besonderen Anforderungen von Unternehmen zu erfüllen – vor allem von solchen Unternehmen, die traditionelle Anwendungen sowohl in traditionellen als auch in hybriden Umgebungen ausführen.

Traditionelle APM-Tools können zwar monolithische Anwendungen überwachen, sie sind aber nicht für Microservices, Container und Cloud-Architekturen geeignet. Wenn sie bei containerisierten Anwendungsinfrastrukturen eingesetzt werden, fungieren sie als „Blackbox“-Überwachung und sind nicht in der Lage, die internen Abläufe der Anwendung zu überprüfen. Dasselbe Problem potenziert sich bei cloudbasierten oder cloudnativen Anwendungen. Ein Enterprise Observability Tool ist ganz eindeutig die Lösung, um diese Probleme anzusprechen. Die folgenden Vorzüge sprechen für den Einsatz eines solchen Tools.

Erkennung. Sie können jede Anwendungs- und Infrastrukturkomponente sehen. Dadurch behalten Sie den Überblick und alles steht im richtigen Kontext zueinander.

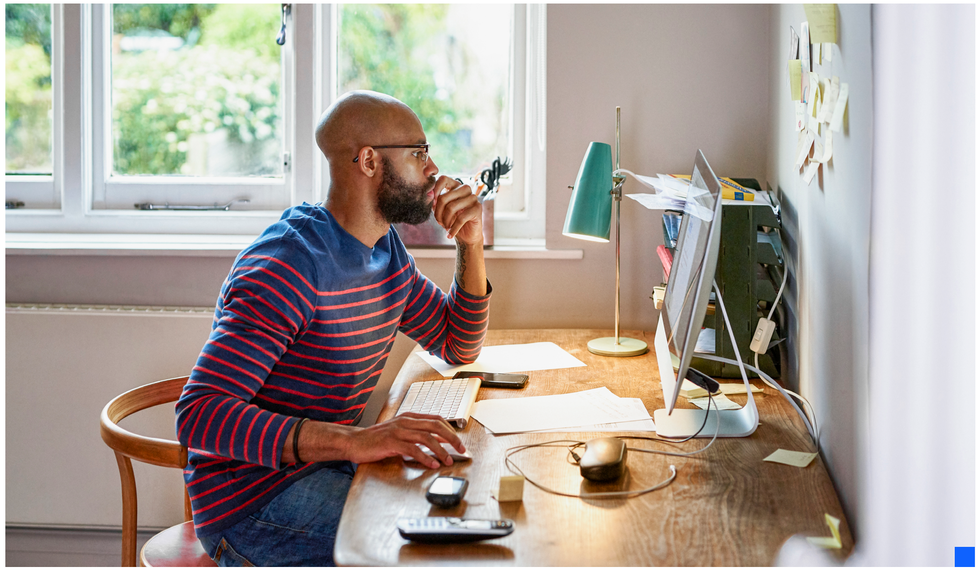
Anforderungen verfolgen. Verfolgen Sie Anforderungen in traditionellen Anwendungen und erstellen Sie für jede Anforderung unabhängig von der Umgebung einen Trace, selbst in cloudnativen Anwendungen.

Hohe Granularität. Zwischen den einzelnen Überwachungsintervallen vergeht nicht mehr als jeweils eine Sekunde.

Sofortige Benachrichtigungen. Verwandeln Sie Informationen in intelligente Maßnahmen.

07

Enterprise Observability mit IBM Instana

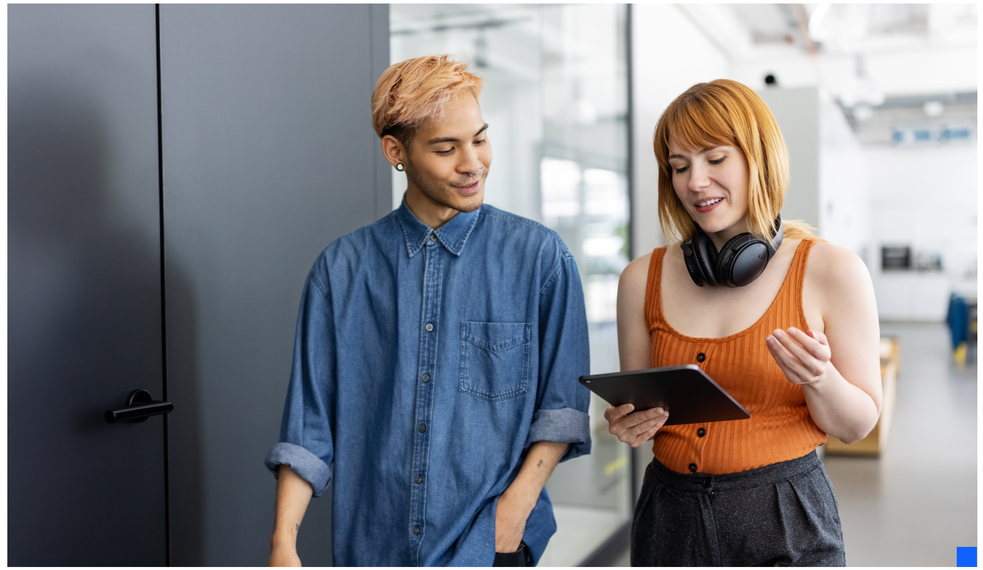


IBM® Instana ist Marktführer auf dem Gebiet der Enterprise Observability. Die folgenden Beispiele zeigen, wie IBM Instana den Funktionsumfang erweitert, um die Anwendungsmodernisierung effizienter zu gestalten.

- **Automatisierte Erkennung.** IBM Instana erkennt jede Anwendungs- und jede Infrastrukturkomponente automatisch direkt bei ihrer Installation.
- **Keine Stichprobenentnahme.** Traditionelle Anwendungen führen stichprobenartige Prüfungen von Transaktionen und manchen Elementen von Traces durch. Instana entnimmt grundsätzlich keine Stichproben und liefert daher eine funktional erweiterte Version der Metriken für traditionelle Anwendungen zusammen mit allen Transaktionen und Ereignissen von Anwendungen, die auf Microservices basieren.
- **Tracing für Anforderungen.** IBM Instana erstellt für jede Anforderung systemübergreifend einen Trace. Diese Tracefunktion ist automatisiert und spart den Entwicklern so Zeit.
- **Granularität von einer Sekunde.** Jede Sekunde wird ein neuer Snapshot der Infrastruktur erstellt, was die Aktualität der Messungen sicherstellt.
- **Drei-Sekunden-Benachrichtigungen.** Wenn es zu Vorfällen kommt – und das wird unausweichlich passieren –, werden die zuständigen Personen benachrichtigt und es wird sofort ein Korrekturprozess eingeleitet.

08

Ist IBM Instana die richtige Lösung für Sie?



IBM Instana bietet eine branchenführende, automatisierte, echtzeitorientierte Observability-Plattform für Unternehmen. Seine Funktionen zur Überwachung der Anwendungsleistung eignet sich ideal für Unternehmen, die komplexe, moderne, cloudnative Anwendungen betreiben. IBM Instana ist überall dort einsatzbereit, wo Sie Ihre Workloads ausführen – in Public Clouds, Private Clouds, Hybrid Clouds, auf mobilen Geräten, vor Ort oder in einer IBM z Systems-Umgebung.

Dank präziser Metriken, vollständiger End-to-End-Traces für alle Transaktionen und KI-gestützter Erkennung kontextbezogener Abhängigkeiten innerhalb hybrider Anwendungen verleiht Ihnen IBM Instana die erweiterte Kontrolle über moderne Hybridanwendungen. IBM Instana hilft Systems Reliability Engineers, die Zuverlässigkeit und Ausfallsicherheit von cloudnativen Anwendungen zu verbessern, indem es verhindert, dass sich Probleme zu Vorfällen ausweiten. Und sollte es doch einmal zu Vorfällen kommen, können diese binnen kürzester Zeit behoben werden.

Überzeugen Sie sich selbst von der Leistungsfähigkeit von IBM Instana. Melden Sie sich noch heute für eine kostenlose 14-tägige Testversion der Vollversion des Produkts an. Keine Kreditkarte erforderlich.

Kostenlose Testversion von IBM Instana →

IBM Instana erkunden →

IBM Deutschland GmbH
IBM-Allee 1
71139 Ehningen
ibm.com/de

IBM Österreich
Obere Donaustraße 95
1020 Wien
ibm.com/at

IBM Schweiz
Vulkanstrasse 106
8010 Zürich
ibm.com/ch

Hergestellt in den Vereinigten Staaten von Amerika
Mai 2023