WHITE PAPER • APRIL 2018

ca technologies

DevOps in der Praxis mit CA Continuous Delivery Automation

Paketierung, Workflows und ein allgemeines Deployment-Modell

Inhaltsverzeichnis

SECTION 1

Komplexität verändert alles

Applikationen und Services umfassen heute mehr Komponenten und Integrationen als noch vor wenigen Jahren. In Kombination mit einer deutlich konkurrenzbetonteren Geschäftswelt hat diese Komplexität sowohl für die Entwicklung (Dev) als auch für die IT-Administration (Ops) alles verändert.

Entwickler stehen unter dem Druck, mehr Funktionen schneller liefern zu müssen, indem sie sich auf kleine inkrementelle Veränderungen konzentrieren und agile Methoden und Tools wie z. B. Continuous Integration nutzen.

Die IT-Administration ist heutzutage mit einer Unzahl von Servern und Services so umfangreich wie nie zuvor. Die Produktionsumgebung, die früher komplett physisch war, ist heute beinahe vollständig virtualisiert, sodass IT-Administratoren Server innerhalb von Augenblicken einrichten können. Zudem verwaltet die IT-Administration deutlich mehr Ressourcen als in der Vergangenheit.

Da sich die Entwicklung auf eine schnelle Bereitstellung konzentriert, wobei Services aufgeteilt, mehr Entwicklungen parallel durchführt werden und Applikationen als Ganzes funktionieren sollen, sind heute mehr Konfigurationsschritte erforderlich, und es entstehen zudem mehr Abhängigkeiten.

Darüber hinaus verbringen die Entwickler durch den Einsatz agiler Methoden weniger Zeit mit „internen" Projekten, die ihnen hätten helfen können, einen Teil der Komplexität zu überwinden. Die IT-Administration mag es nicht, Entwicklerprobleme beheben zu müssen, und die Entwickler wollen keine Zeit für Konfigurationen und Deployments aufwenden. Gegenseitige Schuldzuweisungen sind somit an der Tagesordnung: „Es liegt nicht an meinem Code, es liegt an Ihrer Umgebung" auf der einen und „Es liegt nicht an meiner Umgebung, es liegt an Ihrem Code" auf der anderen Seite. Die „DevOps"-Bewegung hingegen vertritt den Ansatz „Die Entwickler sollten denken wie die Administratoren und die Administratoren sollten denken wie die Entwickler". Es ist in der Tat wichtig, dieses Problem zu lösen und die IT-Administration in die frühen Entwicklungsphasen einzubeziehen.

Es ist zudem sehr nützlich, wenn die IT-Administration moderne Lösungen für das Konfigurationsmanagement, wie z. B. Puppet und Chef, einsetzt. Dieses Ziel lässt sich jedoch nur erreichen, wenn man auch die letzte Hürde ausräumt. Dieses White Paper befasst sich mit der Automatisierung von Applikationsdeployments für DevOps, den erforderlichen Fähigkeiten und damit, wie diese Fähigkeiten während des Lebenszyklus in den Bereichen Dev, Ops oder beiden genutzt werden.

SECTION 2

Wachsende Matrix von Applikationen und Kontrollverlust

Der Lebenszyklus von Applikationen ist ein kontinuierlicher, iterativer Prozess, bei dem Applikationen in einer Reihe von Phasen wie Entwicklung, Funktionstests, Integrationstests, Staging und Produktion konstant entwickelt, getestet, gepatched, aktualisiert und neu entwickelt werden. Während sich die Namen und Nummern der einzelnen Phasen je nach Applikation und Unternehmen unterscheiden, ist der Prozess im Allgemeinen derselbe. Beim Durchlaufen der Phasen ist es immer erforderlich, Teile oder möglicherweise auch die gesamte Applikation auf eine neue Servergruppe, auch „Umgebung" genannt, zu verschieben. Die weiteren Anforderungen umfassen in einigen Fällen die Deinstallation früherer Versionen oder in anderen Fällen die Installation auf einer parallelen Umgebung (zu Testzwecken).

Die Herausforderung für DevOps wird mit jeder neuen Komponente größer. Mit jeder Variation einer Konfiguration oder der Zielumgebung wächst die Matrix von Komponenten, Umgebungen und Deploymentanforderungen. Aufgrund der unterschiedlichen Teams an unterschiedlichsten Orten, die verschiedene Applikationen entwickeln, die letztlich direkt oder indirekt miteinander verknüpft werden müssen, verlieren viele Unternehmen die Kontrolle. In den letzten Jahren kam es zu einer Reihe öffentlich sichtbarer Ausfälle von geschäftskritischen Systemen in den verschiedensten Branchen. Banken, Telekommunikationsunternehmen und Internetgiganten wurden durch unkontrollierte oder unerwartete

Veränderungen zu Fall gebracht. „Fehlerhafte Deployments" und „Verpfuschte Upgrade-Prozesse" lauteten einige der Schlagzeilen, die darauf verwiesen, wie es zu diesen Fehlern kommen konnte.

SECTION 3

Applikationsreleases und -entwicklung mit CA Continuous Delivery Automation: Geschwindigkeit und Kontrolle

Das Ziel der DevOps-Bewegung ist es, die oben genannten Probleme zu lösen. Die Entstehung von DevOps beruht auf der Erkenntnis, dass die Schnittstelle zwischen Dev und Ops der wichtigste Grund dafür ist, dass die IT außer Kontrolle gerät. DevOps-Teams haben sich ursprünglich darauf konzentriert, die „kulturelle Brücke" und Abgrenzung zwischen Dev- und Ops-Teams zu beseitigen und beide Abteilungen zu motivieren, sich durch die Zusammenarbeit während des Lebenszyklus von Applikationen so weit wie möglich zusammenzuschließen. Diese Bemühungen werden auch heute noch fortgesetzt und auch in Zukunft ein unentbehrlicher Teil der DevOps-Welt sein.

Dennoch ist es offensichtlich, dass auch Skalierbarkeit ein großes Thema ist, insbesondere im Hinblick auf die erheblich gestiegene Zahl an Servern in der Betriebsumgebung. Es werden Tools benötigt, um Konfigurationen und Baselines im großen Maßstab zu verwalten, was zur Entwicklung von Puppet und Chef etc. geführt hat. Mit diesen Tools wird ein wichtiges Problem gelöst, denn sie ermöglichen auch in den komplexesten Umgebungen grundlegende Spezifikationen bezüglich der Infrastruktur. Obgleich sich diese Tools hervorragend dafür eignen, große Serverumgebungen zu standardisieren, eignen sie sich weniger gut für von Natur aus umgebungs- oder Tier-übergreifende Applikationsprozesse. Man könnte sagen, dass Konfigurationsmanagement-Tools einen deklarativen Ansatz verfolgen (wichtig ist der Endzustand), während CA Continuous Delivery Automation §nach dem Schema eines Benutzerhandbuchs arbeiten (machen Sie es lieber so als so). Der eine Ansatz ist umgebungsgebunden (Was genau brauche ich auf jedem Server in meiner Umgebung?), der andere bezieht sich auf den Lebenszyklus (Welches sind die erforderlichen Schritte, um diese Applikationsversion/den Patch/das Release von dieser Umgebung in die nächste zu verschieben?). Im Zusammenhang mit Continuous Delivery bieten CA Continuous Delivery Automation-Tools Kontrolle, Effizienz und eine optimale Nutzung der DevOps-Toolchain über alle Schichten des IT-Stacks und der Umgebungen hinweg, in denen das Deployment von Applikationen und ihren Abhängigkeiten erfolgt. CA Continuous Delivery Automation-Tools kombinieren Modellierungs- und Automatisierungsfunktionen, die speziell auf die Integration in jeden beliebigen Applikations-Lebenszyklus und in jede beliebige Verwaltungstechnologie ausgelegt sind.

SECTION 4

Applikations-, Pipeline- und Umgebungsmodelle

Eine CA Continuous Delivery Automation-Lösung ermöglicht Ihnen die Modellierung eines automatisierten Prozesses für die Softwareverteilung, mit denen unterschiedliche Teams den Fortschritt von Builds aus Continuous-Integration-Prozessen über zahlreiche Test- und Entwicklungsphasen hinweg nachverfolgen können. Mit CA Continuous Delivery Automation-Lösungen können Sie physische Artefakte, Umgebungen und Deployment-Schritte in Modellen abstrahieren, die sich einfacher an neue Änderungen, Kriterien und Situationen anpassen lassen.

Applikationsmodelle

Die Modellierung von Applikationen ist eine grundlegende Fähigkeit einer jeden ernsthaften CA Continuous Delivery Automation-Lösung. Applikationen bestehen aus zahlreichen binären Artefakten, Konfigurationsdateien, Skripten und abhängigen Services. Die Artefakte stammen aus unterschiedlichen Quellen (z. B. Repositorys, Build-Server, File Shares und FTP-Server), für die jeweils genaue Versionen erforderlich sind, damit sie dem aktuellen Deployment entsprechen. Durch die Erstellung eines logischen Modells Ihrer Applikation verfügen Sie über eine Abstraktionsschicht zwischen all den physischen Bits und

Bytes Ihrer Deployment-Pipeline, sodass auf jede Umgebung die gleichen Automatisierungsmechanismen angewendet werden können. Bei einigen CA Continuous Delivery Automation-Tools oder Lösungen für die Deployment-Automatisierung bezieht sich das Mapping Ihrer modellierten Applikationskomponenten und/oder Deployment-Ziele direkt auf physische Artefakte. Für jede neue App-Version müssen somit sowohl Ihr logisches Modell als auch die Deployment-Pipeline neu erstellt werden. Dieses 1:1-Mapping (kein Modellieren) bietet nicht die signifikanten Vorteile, die man sich für die Störungsbehebung erhofft, da die Automatisierungsmechanismen selbst als Teil Ihres Aufwands für die Störungsbehebung bewertet werden müssen.

Ein effektives Applikationsmodell hingegen sorgt für Konsistenz und Wiederholbarkeit über alle Umgebungen und Release-Phasen hinweg. Dies reduziert Überwachungsprobleme und den Aufwand für die Störungsbehebung, sollte diese einmal erforderlich sein. Zudem ermöglicht ein logisches Applikationsmodell die Ausführung von automatisierten Deploymentmnechanismen auf einem zentralen Deployment-Server/-Container oder über verteilte Produktionsserver/-container hinweg.

Applikationsmodelle sollten so konstruiert sein, dass sie wie eine Stückliste (BOM) für jede Deployment-Pipeline funktionieren und alle unterschiedlichen Versionen von Komponenten für das Deployment berücksichtigen. Zudem sollten Statusänderungen der Pakete möglich sein, damit sie für die Verbreitung freigegeben oder zurückgewiesen werden können. So erhalten Sie einen optimalen Einblick in die Produktivität und Effizienz Ihres Continuous-Delivery-Ansatzes und schaffen eine Grundlage für die Verbesserung der Prozesse.

Pipeline-Modelle

CA Continuous Delivery Automation-Lösungen verwenden zwei unterschiedliche Automatisierungsmechanismen – deklarativ (gewünschter Endzustand) und Workflow-basiert (wie kann der Endzustand erreicht werden). Deklarative Modelle schreiben vor, wie der Deployment-Prozess ausgeführt werden muss, ohne dass der Softwareanbieter den DevOps-Teams die Möglichkeit lässt, selbst die von ihnen bevorzugte Methode auszuwählen oder sich an technologische Neuerungen anzupassen. Workflow-basierte Modelle hingegen verschaffen DevOps-Teams vollkommene Flexibilität und eine erstklassige Übersicht darüber, wie die Software bereitgestellt wird.

Workflows sind die „Arbeitstiere" von skalierbaren und professionellen CA Continuous Delivery Automation-Lösungen. Sie ersetzen die manuellen Arbeiten, die erforderlich sind, um eine Applikation zu installieren, zu aktualisieren, zu patchen oder zu entfernen, und sie können wieder und wieder in jeder Umgebung verwendet werden, in der sich die Applikation befindet. Ausgereifte CA Continuous Delivery Automation-Lösungen ermöglichen eine visuelle Modellierung von Workflows auf einer Leinwand, ähnlich wie mit einem Microsoft Visio®-Blockdiagramm. Die Workflows werden per Drag-and-drop aus einer großen Bibliothek vorhandener Verhaltensweisen und Integrationen für die meisten gängigen Hosts von Applikationen zusammengestellt und sorgen für ein optimales Zusammenspiel der Continuous-Integration-/Continuous-Delivery-Toolchain (CI/CD) und der damit verknüpften Systeme.

Die nachstehenden Beispiele demonstrieren die Vielzahl unterschiedlicher Infrastrukturen und Tools, für die vorgefertigte Workflows zusammengestellt werden können:

- Applikationsserver wie Oracle WebLogic, IBM WebSphere® und JBoss/IIS
- Datenbank-Server wie Oracle und Microsoft SQL Server®
- Integrationsserver wie BizTalk
- Container-Technologien wie Docker, Kubernetes und LXC
- Applikationsserver wie WebLogic, WebSphere und JBoss/IIS
- Cloud-Anbieter wie AWS, Microsoft AZURE™ etc.
- Betriebssystem-Befehle, Paketmanager und Quellkontrollsysteme wie GitHub und Bitbucket
- CI-Tools wie Jenkins, Bamboo und Travis CI

Vorkonfigurierte Integrationen ermöglichen ein standardisiertes und konsistentes Deployment von neuen Änderungen auf unterschiedlichsten Servern bei gesicherter Qualität. Vorkonfigurierte Integrationen können zudem – ohne Beeinträchtigung der Sicherheit – mit unterschiedlichen Zugangsdaten ausgeführt werden, die während des Prozesses erforderlich sind und während der gesamten Zeit automatisch geprüft werden.

Andere wichtige Eigenschaften von Workflows sind:

Allgemeingültigkeit: Workflows sollten vollständig abgekoppelt sein von Umgebungsbedingungen, Zugangs- und Berechtigungsdaten sowie den ausgeführten Inhalten. So kann derselbe Workflow in unterschiedlichsten Umgebungen ausgeführt werden, von Test- bis zu Produktionsumgebungen.

Versionskontrolle: Genauso wie Applikationskomponenten und -services von mehreren Teams parallel entwickelt werden, was zu unterschiedlichen Versionen führt, kann dies auch bei Workflows der Fall sein. Tatsächlich kann dasselbe Team, das den Code entwickelt, auch Workflows entwickeln. Ein paralleles Arbeiten in mehreren Teams ist daher nur mit einem Versionsverwaltungssystem möglich. Deshalb ist es wichtig, dass Ihre CA Continuous Delivery Automation-Plattform für die Zusammenarbeit von mehreren Teams und die effektive Arbeit an unterschiedlichen Umgebungsworkflows und unterschiedlichen Versionen geeignet ist.

Ausgereiftheit und Einfachheit: Deployment-Workflows müssen sowohl einfach als auch ausgereift sein. Einfache, serielle Automatisierungsprozesse sind zu optimistisch und zu starr für die Verwaltung der komplexen Zusammenhänge, Nuancen und der Fehlertoleranz, die das Technologieportfolio moderner Unternehmen erfordern. Workflows müssen in der Lage sein, mit Neuversuchen, If-Then-Else-Verzweigungen, teilweise erfolgreichen/fehlgeschlagenen Prozessen, Runtime-Entscheidungen, Workload-Verteilungen und vielem mehr umzugehen. Unternehmen benötigen nicht nur ausgereifte und flexible Workflows, diese müssen sich auch einfach nutzen lassen.

Umgebungsmodelle

Viele Fehler werden durch unterschiedliche Umgebungskonfigurationen eines bestimmten Deployments verursacht, nicht durch Workflow-Bugs. Das Problem liegt darin, dass Software und Applikationen auch auf die kleinsten Veränderungen in Bezug auf Umgebung und Konfigurationen sehr sensibel reagieren. Ein einfaches Beispiel sind unterschiedliche Verzeichnisstrukturen auf dem QS- und dem Produktionsserver. Das kann ein winziger Unterschied in der Portkonfiguration eines Services, in den Zugangsdaten für die Ausführung oder in der Umgebungskonfiguration sein, wie beispielsweise Cluster-Größen. Es ist ein mühseliger und fehleranfälliger Prozess, diese wichtigen Variablen während des Deployments zu berücksichtigen. Die Anwender müssen sie alle verifizieren und jeden Zielserver ebenso wie das System als Ganzes anpassen. Wenn man bedenkt, dass zehn oder Hunderte Server aktualisiert werden müssen, wird die Matrix bald manuell nicht mehr handhabbar sein. Zudem erfordert das Skripting Programmierkenntnisse und erfolgt oft einmalig.

CA Continuous Delivery Automation verfügt über ein integriertes Modell, mit dem die IT-Administration und Entwicklung Runtime-Konfigurationen in einer einzigen Umgebung kontrollieren können. Das System übermittelt die richtigen Einstellungen für die richtigen Ausführungen im richtigen Moment, und das automatisch. Das Modell berücksichtigt die Varianten der Umgebungen (z. B. Art und Anzahl der Server), Servereinstellungen (z. B. Verzeichnis- und Versionsunterschiede) und Konfigurationsdaten (z. B. Variablenwerte). Es beinhaltet Deployment-Ziele, Profile, Logins und mehr und stellt nicht nur sicher, dass die richtigen Einstellungen zum richtigen Zeitpunkt zur richtigen Durchführung vorgenommen werden, sondern spielt auch eine wichtige Rolle dabei, Workflows einfach und handhabbar zu halten. Tatsächlich führt das Nichtvorhandensein eines integrierten Modells zu sehr komplexen Workflows und der Notwendigkeit zum Speichern von Runtime-Einstellungen in externen Quellen, z. B. XML oder Datenbanken. Dies kann sowohl Sicherheitsrisiken als auch Komplexität mit sich bringen, denn der Anwender muss die Ausführungsabläufe als Teil des Workflows lesen, parsen und verzweigen.

SECTION 5

Fazit

Die Automatisierung von Application Releases ist der einzige Weg, um echte Kontrolle über die komplexe Matrix von Applikationen, Releases und Umgebungsvarianten zu erlangen. Die Unfähigkeit, Änderungs-, Deployment- und Release-Prozesse vollständig zu kontrollieren und zu prüfen, führt zu IT-Ausfällen, die das Unternehmen Millionen kosten können und DevOps unerreichbar machen. CA Continuous Delivery Automation ist die letzte Hürde auf dem Weg zur Continuous Delivery in Unternehmen.

Um diese Hürde zu überwinden, muss eine CA Continuous Delivery Automation-Lösung die folgenden drei Merkmale bieten:

• Applikationspakete und Promotion Paths, damit Sie sich der Richtigkeit Ihrer Deployments sicher sein können

• Workflows für ein standardisiertes und konsistentes Deployment von neuen Änderungen bei gesicherter Qualität

• Umgebungsmodelle, damit auf jede Umgebung die richtigen Einstellungen und Konfigurationen angewendet werden

Um sicherzugehen, dass Ihr Unternehmen für alle zukünftigen Anforderungen gewappnet ist, sollten Sie bei der Auswahl einer geeigneten Lösung mindestens die oben genannten Punkte beachten. Nur so erzielen Sie schnellere Prozesse, senken die Kosten und behalten die Kontrolle.

For more information, please visit **ca.com/automation**

Kontaktieren Sie CA technologies

CA Technologies (NASDAQ: CA) entwickelt Software, die die Transformation von Unternehmen vorantreibt und ihnen ermöglicht, die Chancen der Application Economy zu nutzen. Software ist das Herz eines jeden Unternehmens, in jeder Branche. Von der Planung über die Entwicklung bis zu Management und Sicherheit arbeitet CA weltweit mit Unternehmen zusammen, um unsere Lebensweise zu verändern in Bezug auf unsere Kommunikation und Interaktion und das unterwegs, zuhause und in öffentlichen Clouds sowie in Mainframe-Umgebungen. Erfahren Sie mehr unter: ca.com/automation.

CS200-374361_0618