

LISTICLE

Effiziente CI/CD-Pipelines basieren auf fünf Best Practices

München, 20. September 2022 – Continuous Integration, Continuous Delivery und Continuous Deployment sind gängige Methoden, um die Bereitstellung von Software zu automatisieren. CI/CD-Pipelines bilden einen zentralen Bestandteil dieser komplexen Prozesse. IT-Dienstleister Consol nennt fünf Best Practices für eine höchstmögliche Effizienz des Release-Workflows.

Ob Continuous Integration, Continuous Delivery oder Continuous Deployment: Neuer Code für eine Anwendung durchläuft auf seinem Weg in Produktion bei all diesen Methoden sogenannte CI/CD-Pipelines, also automatisierte und Tool-gestützte Prozessketten. Da im Zuge der vielen Prozesse einiges schiefgehen kann, hat IT-Dienstleister Consol fünf Best Practices definiert.

1. Die Performance der Pipeline beachten

Die Anforderung an CI/CD-Pipelines sind häufige Aktualisierungen der Software bei gleichzeitig möglichst umfangreichem Testing. Zwischen diesen Gegensätzen gilt es, ein gesundes Mittelmaß zu finden. Eine verbesserte Infrastruktur und die Optimierung der Tests können bis zu einem gewissen Grad helfen. Geraten Entwicklerteams allerdings an einen Punkt, wo ein Trade-Off unvermeidlich ist, zum Beispiel durch eine Einschränkung der Tests oder die Akzeptanz verlängerter Laufzeiten, sollten sie alle Entscheidungen mit den Stakeholdern besprechen und gut dokumentieren.

2. Die CI/CD-Umgebung isolieren und absichern

Die CI/CD-Umgebung hat in der Regel Zugriff auf die Codebasis der Anwendungen sowie verschiedene andere Umgebungen – etwa für die Tests – bis hin zur Produktion. Das CI/CD-System stellt somit ein lohnendes Ziel für Angreifer dar. Daher sollten sich alle darüber im Klaren sein, dass die CI/CD-Umgebung ein Einfallstor für Hacker sein kann und sie diese entsprechend schützen müssen – etwa mit einer strengen Zugriffskontrolle.

3. Ausschließlich über die CI/CD-Pipeline deployen

Gerade zu Beginn kann die Umstellung auf eine CI/CD-Pipeline zu Stresssituationen führen. Dann kommt es vor, dass Entwicklerteams auf alte Deployment-Methoden ausweichen und Bugfixes an der Pipeline vorbei einspielen, um somit vermeintlich wertvolle Zeit bis zum Live-Gang zu sparen. Das sollten Unternehmen allerdings tunlichst vermeiden, denn auch in solchen Situationen haben CI/CD-Pipelines einen großen Wert: Sie ermöglichen es, Standardverfahren für Tests, Integration und Bereitstellung durchzusetzen und so die Entwicklungspraktiken zu verbessern sowie die Qualität des Codes zu steigern. Qualitativ minderwertigen Code erkennt die Pipeline frühzeitig und sortiert ihn aus, sodass er nicht in Produktion gelangt und dort Probleme verursacht. Das CI/CD-System nimmt eine sogenannte „Gatekeeper“-Funktion ein und schützt vor Schad-Code. Das funktioniert allerdings nur, wenn auch alle Änderungen am Code die Pipeline durchlaufen.

4. Produktions- und Testumgebung möglichst identisch halten

DevOps und CI/CD sind Methoden, denen die Idee zugrunde liegt, möglichst kleine Änderungen an der Software anzustoßen, durchzutesten und zu deployen. Die nötigen Tests sollten auf einer Umgebung stattfinden, die möglichst identisch mit der Produktionsumgebung ist, um die Integrität der Software unter realen Bedingungen zu prüfen. Sind Testumgebung und Produktion zu verschieden, kann es dazu kommen, dass problematische Änderungen im Test nicht mehr erkannt werden und erst in Produktion auftreten – und das sollte auf alle Fälle vermieden werden. Je weniger Unterschiede zwischen Test- und Produktionsumgebung bestehen, umso unwahrscheinlicher ist das Auftreten dieses Problems. Ein zentraler Punkt hierbei ist, die bestehenden Unterschiede und möglichen Auswirkungen zu kennen.

5. Nach Trunk-based Development vorgehen

In der Prozesskette sind menschliche Interaktionen ein großer Zeitfresser. Beispiele dafür sind das Zusammenführen von Code-Änderungen (Mergen), manuelle Schritte bei der Testausführung oder der Installation sowie manuelle Konfigurationsänderungen. Zusätzlich können menschliche Fehler unnötige Reparaturaufwände zur Folge haben.

Eine mögliche Methode, um diesem Risiko zu begegnen, ist das „Trunk-based Development“. Es ermöglicht, dass zum Zeitpunkt der Release-Entscheidung alle

Änderungen schon an richtiger Stelle, dem sogenannten „Trunk“, vorliegen. Die Entwicklerteams müssen sie somit nicht noch manuell in einen Release-Branch übertragen und prüfen. Voraussetzung dafür ist allerdings, dass der Trunk auch tatsächlich jederzeit Release-fähig ist. Änderungen, die dazu führen, dass er diese Eigenschaft verliert, sollten sofort zurückgerollt werden. Idealerweise erfolgt dies automatisch durch die CI/CD-Pipeline, die somit als Sicherheitsnetz fungiert.

„Welchen Grad der Automatisierung Unternehmen wirklich implementieren wollen, ist individuell verschieden“, betont Lutz Keller, Leiter DevOps bei Consol. „Die Best Practices, wie die Sicherheit zu bedenken und nicht an der Pipeline vorbei zu arbeiten, sollten jedoch in jedem Fall berücksichtigt werden.“

Weitere Informationen unter: <https://www.consol.de/software-engineering/ci-cd/>

Dieses Listicle und Bildmaterial sind abrufbar unter www.pr-com.de/companies/consol.

Über Consol

Die Consol Consulting & Solutions Software GmbH mit Hauptsitz in München begleitet seit mehr als 35 Jahren lokale und internationale Unternehmen mit passgenauen IT-Lösungen durch den gesamten Software-Lifecycle. High-End-IT-Beratung, agile Software-Entwicklung sowie Betrieb und Support sind die Eckpfeiler des Portfolios, das Consol unter Anwendung von modernsten Technologien ständig erweitert. Dazu zählen Open Source-Projekte wie Quarkus, OpenShift oder Tekton. Das Unternehmen entwickelt und vertreibt auch die Software Consol CM, eine Plattform zur Digitalisierung von Geschäftsprozessen. Bei der Umsetzung der Digitalisierungsstrategien seiner Kunden macht Consol IT-Umgebungen und Geschäftsprozesse fit für die Herausforderungen von morgen. Mit den Leitmotiven Exzellenz und höchste Qualität folgt Consol dem Ziel, Businesses weiter voranzubringen. Dabei fokussiert Consol Bereiche wie Cloud-native, Container, Microservice-Architekturen oder IT Automation.

Consol ist Red Hat Premier Partner und NGINX Preferred Partner. Strategische Partnerschaften bestehen außerdem zu AWS und Microsoft Azure. Zu den Kunden zählen Großunternehmen wie Haribo, Daimler oder Vodafone. Aktuell beschäftigt Consol rund 260 Mitarbeiter an seinen Standorten München, Düsseldorf, Wien, Krakau, Dubai sowie San Francisco.

Weitere Informationen unter <https://www.consol.de>, <https://cm.consol.de> und <https://labs.consol.de> sowie auf Twitter unter https://twitter.com/consol_de.

Pressekontakt

ConSol Consulting & Solutions Software GmbH
Isabel Baum
St.-Cajetan-Straße 43
D-81669 München

Fon: +49-89-45841-101

E-Mail: Isabel.Baum@consol.de

Web: <https://www.consol.de> und <https://cm.consol.de>

PR-COM GmbH

Nicole Oehl

Sendlinger-Tor-Platz 6

D-80336 München

Fon: +49-89-59997-758

E-Mail: nicole.oehl@pr-com.de

Web: www.pr-com.de