

Modernes Testdatenmanagement in DevOps-Projekten

KUBERNETES schafft neue Möglichkeiten – auch im Umgang mit Testdaten?

Ein Expertenbeitrag zum Softwaretest aus der Trimetis Gruppe



Die Trimetis Gruppe

Die Experten für individuelle Lösungen

UNSER ANSATZ – IHR NUTZEN

Flexibilität, Geschwindigkeit, unternehmerisches Commitment, gepaart mit Professionalität und langjährigen internationalen Erfahrungen bei vielen Topunternehmen bringen dem Kunden einen besonderen Nutzen aus der Zusammenarbeit mit der Trimetis Gruppe. Wir operieren an 7 Standorten in der Region DACH und in Polen und zählen zahlreiche renommierte Unternehmen zu unseren langjährigen Kunden.

Wir operieren unter folgenden Marken in vier Geschäftsbereichen:

Trimetis

Trimetis GFB



IT-CONSULTING

Unsere erfahrenen Berater unterstützen Sie bei der IT-Transformation und der Entwicklung Ihrer IT-Strategie. Mit branchenspezifischem und technologischem Know-how entwickeln wir mit Ihnen das passende Design und die richtige Architektur. Die Ergebnisse setzen wir mit erfahrenen Projektmanagern um, damit Sie am Ende exakt die Lösung erhalten, die zu Ihnen passt.



IT-SERVICES

Wir entwickeln mit unseren Softwareentwicklern professionelle, maßgeschneiderte und kostengünstige Lösungen, übernehmen die Bearbeitung ihrer Legacy Systeme und helfen Ihnen bei der Digitalisierung Ihrer Prozesse und Daten. Mit hybriden und agilen Strukturen unterstützen wir Sie vor Ort ebenso, wie aus unserem eigenen Nearshore-Center in Polen.



SOFTWARE TESTING

Da die Qualität Ihrer Software entscheidend für Ihren Geschäftserfolg ist, beraten wir Sie rund um das Testen, bieten Ihnen Testmanagement, Testdatenmanagement, Testautomatisierung und die passenden Testressourcen an. Wir sind Ihr Partner bei funktionalem und technischem Testen sowie bei der Analyse, Konzeptionierung und Bereitstellung (anonymisiert/synthetisiert) Ihrer Daten.



EXPERT SOURCING

Unsere beiden Tochterfirmen adegna und aldoluck sind ausschließlich darauf fokussiert, Ihre komplexen IT-Projekte mit den besten Fachleuten zu versorgen. Eine schnelle, flexible, passgenaue und rechtssichere Vermittlung von externen Experten, das ist es, was Sie von uns erwarten dürfen: punktuelle Vermittlung einzelner Ressourcen oder ganzer Teams, aber auch das Management all Ihrer externen Ressourcen.



Mag. Peter F. Laggner



Gottfried Madl

Sehr geehrte Leserinnen und Leser,

in Zusammenhang mit den aktuell rasch expandierenden und agileren Anwendungslandschaften im Umfeld der Digitalisierung stellt sich die Frage nach der geeigneten Qualitätssicherung der IT-Systeme. Kurze Release- und Deployment-Zyklen erfordern eine performante Qualitätssicherung, um Anwender dauerhaft zufriedenzustellen. Dabei ist die Just-in-Time Bereitstellung passender und DSGVO-konformer Testdaten ein kritischer Ressourcen- und Zeitfaktor.

Bei Kundenprojekten in großen DAX-Unternehmen haben sich unsere Fachexperten der Trimetis GfB vielfach mit Software-Entwicklungsprozessen im Zusammenhang mit Containern, Kubernetes und DevOps auseinandergesetzt. Wir sehen, dass diese Fragestellung viele Unternehmen sehr beschäftigt. Deshalb wollen wir unsere Expertise in dem Fachartikel „Kubernetes schafft neue Möglichkeiten – auch im Umgang mit Testdaten?“ einem breiteren Fachpublikum zur Verfügung stellen.

Unsere beiden Testexperten, Christoph Knopp und Daniel Simon, beschreiben darin, warum Testdatenmanagement – gerade im Zusammenhang mit Kubernetes und DevOps – ein so wichtiger Faktor in der Wertschöpfungskette der Systemerstellung ist und auch zukünftig bleibt. Zusätzlich präsentieren die beiden Berater mit einem „Testdatengenerator als Microservice“ eine passende, in der Praxis bereits erprobte Lösung.

Wir wünschen Ihnen eine interessante Fachlektüre und freuen uns auf den persönlichen Austausch mit Ihnen zu diesem Thema.

Viel Spaß beim Lesen!

Mag. Peter F. Laggner
Vorstand

Gottfried Madl
Vorstand

KUBERNETES schafft neue Möglichkeiten – auch im Umgang mit Testdaten?

von Christoph Knopp und Daniel Simon, Trimetis GFB IT-Consulting GmbH

„The only constant in software development is continuity“ – so oder so ähnlich könnte man das berühmte Zitat von Heraklit auf die IT übertragen. In den vergangenen Jahren haben wir bei Trimetis GFB gesehen, dass sich viel verändert hat, immer neue Technologien entwickelt wurden und neue Konzepte entstanden sind. Alles ist mittlerweile continuous, befindet sich im Fluss und wird kontinuierlich verbessert.

Durch automatisierte CI/CD-Pipelines werden die Prozesse beschleunigt, und sogar in der sonst immer

gerne vernachlässigten Software-Qualitätssicherung hat sich Continuous Testing mittlerweile etabliert. Vollautomatisierte Prozesse, vom Einchecken des Codes bis hin zur Bereitstellung in der Produktiv-Umgebung in weniger als einer Stunde, sind heute nicht mehr nur den großen Softwareunternehmen vorbehalten, sondern sind auch im Mittelstand realisierbar. Eine Reihe der verwendeten Technologien und Ansätze existieren zwar schon länger, aber erst durch deren Kombination entstehen neue Möglichkeiten.

DATA
SEARCHING
VERIFICATION
CODING
SENDING

Agile Entwicklungsmethoden haben vieles geändert

Einen großen Anteil am Erfolg haben die agilen Prinzipien, die in der Softwareentwicklung zum Projektstandard geworden sind. Mit DevOps werden die agilen Leitsätze konsequenterweise von den Teams auf nahezu alle am Produkt beteiligten Personen übertragen. Neben den Teams aus Entwicklung und IT-Betrieb arbeiten auch die Qualitätssicherung und IT-Security Hand in Hand, sodass ein Gemeinschaftsgefühl geprägt wird (Abbildung 1). Alle sitzen gemeinsam in einem Boot und sind verantwortlich für den Produkterfolg. Hier würde sich die Überleitung zu Kubernetes (griechisch für „Steuermann“) anbieten, allerdings soll zunächst noch auf eine weitere wesentliche Komponente eingegangen werden, um den Ausflug in die Entstehungsgeschichte der kontinuierlichen Softwareentwicklung abzuschließen – die Containerisierung. Auch hierbei handelt es sich grundsätzlich um eine bereits seit längerem bekannte Technologie, die in den vergangenen Jahren mit Docker immer beliebter wurde. Damit wurde es technologisch möglich, sich von den monolithischen Architekturen zu lösen – in Richtung Microservices oder zumindest hin zu hybriden Zwischenlösungen. Die Komplexität eines Produktes wird nun frei nach divide and conquer zerlegt. Sowohl technologisch als auch fachlich gesehen findet eine Aufteilung auf agile Teams statt.

Um nun den Bogen wieder zu Kubernetes zu spannen: Die technischen und auch methodischen Möglichkeiten bieten zwar neue Chancen, erfordern aber auch neue Herangehensweisen in Bezug auf Deployment, Verwaltung und Betrieb. Kubernetes bietet hierfür die entsprechende Orchestrierung und hebt damit zusammen mit allen anderen genannten Technologien und Methoden die kontinuierliche Softwareentwicklung auf ein neues Level.



Abb. 1: Alle packen an, jeder fühlt sich zuständig – das Gemeinschaftsgefühl in DevOps-Projekten.

Die Bedeutung von Testdaten

Bis hierhin ging es in diesem Artikel noch nicht um unser Kernthema Testdaten. Es ist leider eine Tatsache, dass Testdaten generell meist nur sehr wenig Beachtung in Softwareprojekten finden. Softwaretests standen früher eher an allerletzter Stelle der To-do-Liste, wurden auch gerne nur punktuell durchgeführt oder entfielen sogar ganz. Ein strukturiertes Testdatenmanagement fehlte komplett oder wurde, wenn überhaupt, „mal nebenbei“ im Softwaretest miterledigt. Seit mit agilen Entwicklungsmethoden der Softwaretest als wesentlicher Bestandteil wahrgenommen wird, z. B. als Rolle in SCRUM-Teams, hat sich das Standing von Testdaten ein wenig verbessert. Gebremst hat diesen Trend allerdings der Datenschutz. Wo man zuvor bequem Testdaten aus dem Produktivsystem abziehen und verwenden konnte, wurde spätestens mit der EU-DSGVO die Gewinnung von Testdaten aus der Produktion wesentlich komplexer. Testdaten sind nun ein Thema, mit dem man sich zwangsläufig beschäftigen muss, sonst könnte es teuer werden – etwa durch Strafen und Imageverlust bei Datenschutzverletzungen.

Automatisierung als Treiber

Der Softwaretest wird also immer wichtiger und gewinnt weiter an Bedeutung. Vor allem durch die Testautomatisierung, gerade im DevOps-Umfeld mit seinen automatischen Pipelines, kommt es zu einem Umdenken auch beim Thema Testdaten, die nun immer häufiger eine wesentliche Rolle spielen. Wo nahezu 100 % automatisiert wird, müssen Testdaten pünktlich und in ausreichender Menge verfügbar sein. Langwieriges manuelles Suchen nach Testdaten oder aufwendiges Anonymisieren von produktiven Systemen passen nicht zu einer Pipeline, die nur darauf wartet, die nächste Teststufe und das darauffolgende Deployment zu starten. Testdaten müssen zum richtigen Zeitpunkt einfach da sein, Just-in-Time und Just-in-Sequence. Zudem werden zuverlässige Testdaten benötigt, die immer einen klar definierten Zustand darstellen. Gerade bei hohen Automatisierungsgraden müssen die Testergebnisse zuverlässig sein. Grundvoraussetzung hierfür ist die Qualität der Testdaten.

Bei Projekten, die auf Microservice-Architekturen setzen, ist bereits der Trend zu erkennen, dass Testdaten als maßgeblich angesehen werden und ihr Nutzen für das Projekt erkannt wird. Durch die Reduzierung der Komplexität auf einzelne Services ist jedes Team für seine Testdaten selbst verantwortlich und hat auf Grund von klar definierten Schnittstellen die Möglichkeit, sich entsprechend um diese Testdaten zu kümmern. So bedient man sich in der Testautomatisierung bei den Schnittstellen der verschiedenen Services und erzeugt parallel zur Laufzeit der Tests die benötigten Daten. Durch den Austausch mit anderen Teams besteht der Zugriff auf viele hochwertige Testdaten. Wer schon für eine große monolithische Anwendung mit vielen Umsystemen Testdaten erstellt hat, wird hier die Vorteile von Microservices sofort erkennen: Klar definierte Verantwortlichkeiten, einfachere Komplexität pro Service und damit auch kleinere Datenhaltungssysteme erleichtern den Umgang mit den Daten im Testsystem. Schwierig ist hingegen das Tracing des Datenflusses über die Grenzen eines Microservices hin-

weg, weswegen es signifikant ist, ein Testdatenkonzept samt Testdatenmanagement zu haben.

Möglichkeiten noch nicht ausgeschöpft

Testdaten bleiben herausfordernd, gerade wenn es um höhere Teststufen oder um Last- und Performancetests geht. Wenn die Testdaten zur Testausführung über die Schnittstellen erzeugt werden, kostet das Zeit, die den gesamten Testlauf entsprechend verlängert. Für funktionale Tests, die im Bereich von mehreren Hunderten oder vielleicht auch Tausenden Datensätzen operieren, mag das vielleicht noch funktionieren. Wenn aber sehr viele Daten benötigt werden und die Testfälle zeitkritisch sind, wie bei Last- und Performancetests, kann man nicht auf die Erzeugung von mehreren Millionen Datensätzen warten.

Erzeugt man die benötigten Testdaten bereits im Vorfeld, ergeben sich Herausforderungen aus dem Testdatenmanagement, die viele sicher aus ihren eigenen Projekten kennen. Da die Testausführung in der Regel über die Eincheckvorgänge der Versionsverwaltung getriggert wird, ist der Startzeitpunkt und die Regelmäßigkeit der Ausführung unklar. Testdaten müssen aber für jede Ausführung verfügbar sein oder reserviert werden können. Solche Aspekte sind auch beim manuellen Testen bekannt, wenn zwei Tester in sich überschneidenden Bereichen einer Anwendung unterwegs sind und sich nicht gegenseitig ihre Testdaten unbrauchbar machen sollen.

Zusammengefasst kann man sagen, dass in aktuellen Entwicklungsprojekten Testdaten endlich den richtigen Stellenwert haben und auch einen direkt messbaren Nutzen für das Projekt produzieren. Dennoch werden selten alle technischen und fachlichen Möglichkeiten ausgeschöpft. Wir wollen daher hier einen Ansatz aufzeigen, wie wir ihn als Trimetis GFB in unseren Kundenprojekten erfolgreich umgesetzt haben. Mit den bekannten Technologien und Methoden, allen voran Kubernetes, kann das Thema Testdaten zu einem entscheidenden Baustein in aktuellen Entwicklungsprojekten werden.

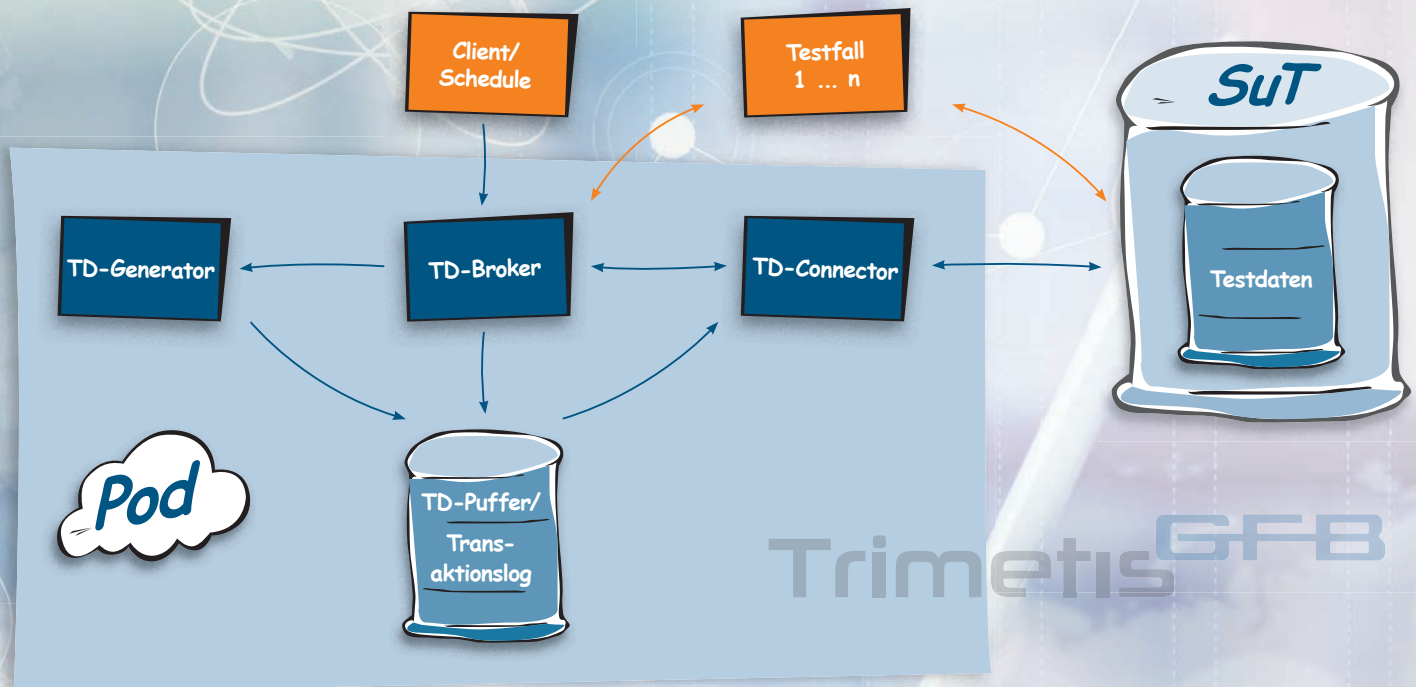


Abb. 2: Microservice-Architektur des Testdatengenerators

Testdatengenerator als Microservice

Die grundsätzliche Idee ist recht einfach: Um für eine Microservice-Architektur Testdaten zu erstellen, sollte der Testdatengenerator ebenfalls dieselbe Architektur aufweisen. Als Container kann dieser direkt in der Kubernetes-Umgebung bereitgestellt werden und versorgt dort die einzelnen Services mit Testdaten. Ein direkter Zugriff auf die Schnittstellen der Services sowie auf deren Datenhaltungssysteme ist damit grundsätzlich möglich. Das Deployment des Testdatengenerators kann angeglichen werden an die Methoden, die ohnehin für die eigentlichen Services verwendet werden. Installation und Setup lassen sich somit auch problemlos in die Pipeline integrieren.

Für den Aufbau des Testdatengenerators als Microservice haben wir die wesentlichen Bestandteile identifiziert:

- **Broker:** Stellt die Schnittstelle des Testdatengenerators dar und verwaltet alle Operationen rund um die Testdaten.
- **Generator:** Erzeugt Testdaten beispielsweise per Zufallsgenerator.
- **Connector:** Hat Informationen über die Schnittstellen der Zielsysteme und kann diese aufrufen.

- **Transaktionslog:** Dient zur Protokollierung der Operationen der Broker und der internen Kommunikation zwischen den Services des Testdatengenerators.
- **Puffer:** Hält die vom Generator erstellten Testdaten vor.

In Abbildung 2 sind die verschiedenen Bestandteile in einem beispielhaften Anwendungsfall dargestellt. Über den Broker kann eine Anfrage zur Erstellung von Testdaten erfolgen. Der Generator bearbeitet diese Anfrage und übergibt die erstellten Testdaten in den Puffer. Von dort wiederum kann der Connector auf diese erstellten Daten zugreifen und über die Schnittstelle der Software unter Test bereitstellen. Die verschiedenen Schritte werden entsprechend im Transaktionslog protokolliert, sodass der Broker jederzeit weiß, in welchem Zustand sich die Testdatenanfrage befindet. Nach solch einer erfolgreichen Erzeugung und Bereitstellung von Testdaten kann eine Testautomatisierungslösung beim Broker die aktuellen Testdaten abfragen und diese für die Ausführung der Testfälle nutzen. Über die Kopplung von Puffer und Transaktionslog hat der Broker immer den aktuellen Zustand der Testdaten verfügbar und kann die Testautomatisierung mit einer ausreichenden Anzahl geeigneter Testdaten versorgen.

Man kann an dieser Stelle schon erahnen, welche Möglichkeiten sich aus dieser Konstellation ergeben. So können beispielsweise der tatsächliche Bedarf an Testdaten getrackt und die erforderlichen Testdaten immer bereitgehalten werden. Ebenfalls kann der Broker über einen Scheduler getriggert werden, um etwa für eine Nachverarbeitung über Nacht rechtzeitig bestimmte Testdaten bereitzustellen. Mit einer solchen Architektur lassen sich alle im Testdatenmanagement notwendigen Prozesse umsetzen. Die eigentlichen Prozesse im Testdatenmanagement sollen nicht Thema dieses Artikels sein. Wir als Trimetis GFB bieten hierzu gerne unsere individuelle Beratung an und empfehlen zudem das Buch vom ASQF zum Thema Testdatenmanagement [ASQF-TDM].

Features von Kubernetes

Das hier vorgestellte Konzept kann prinzipiell auch ohne Kubernetes umgesetzt werden. Die Vorteile, die auch sonst von Microservice-Architekturen geboten werden, gelten auch für den Testdatengenerator. Wie in Abbildung 3 gezeigt, kann das Schema auch in mehrere Pods zerlegt werden. Für Transaktionslog und Datenhaltung spricht beispielsweise die Möglichkeit zur Auslagerung an Datenbank-Dienste in der Cloud. Ausfallsicherheit und Skalierungsfähigkeit sprechen hierfür ebenso wie die Tatsache, dass die Administration des Datenbanksystems in die Verantwortlichkeit des Cloud-Providers übergeht. Grundsätzlich ist von einer Datenhaltung innerhalb von Pods abzuraten, da der (Daten-)Zustand der Anwendung beispielsweise durch das Entfernen eines Pods beeinträchtigt werden könnte.

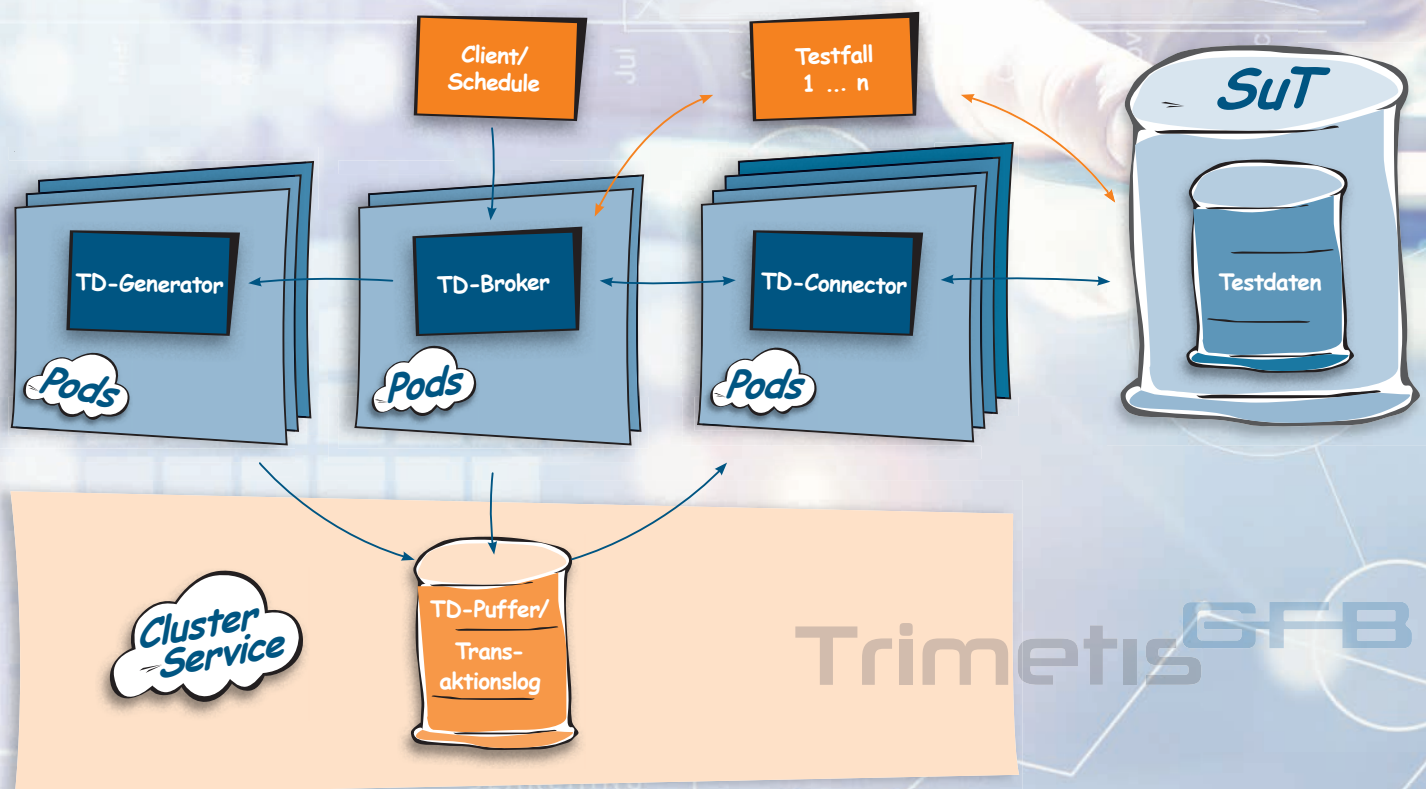


Abb. 3: Aufteilung des Testdatengenerators in multiple Pods

Auch bei den anderen Bestandteilen spricht einiges dafür, die jeweiligen Services in eigene Pods zu verlagern. Grundsätzlich sind Ausfallsicherheit und Skalierung hierfür gute Argumente, aber ein großer Vorteil ergibt sich im Performancegewinn, wenn Testdaten-Anfragen auf mehreren Instanzen des Generators parallelisiert ausgeführt werden können. Gleiches gilt auch für den Connector, der ebenfalls mit mehreren Instanzen auf die Schnittstelle des Zielsystems zugreifen kann.

Vorteile dieser Aufteilung in mehrere Pods bestehen auch bei global verteilten Clustern. Bestimmte Teile des Testdatengenerators können im jeweiligen Cluster platziert werden, wodurch die Netzwerkstrecken und damit auch die Latenzen minimiert werden. Jedes Cluster kann seinen eigenen Generator und Connector bekommen. Der Broker wird entweder als zentrale Steuerzentrale eingesetzt oder aber dezentral pro Cluster konfiguriert. Die Anzahl der einzelnen Broker, Generatoren oder Connector Pods kann beliebig an die Zielarchitektur der Anwendung unter Test bzw. an die jeweilige Testumgebung angepasst werden.

Durch die Verwendung eines Datenbankservices für den Puffer und das Transaktionslog – statt der Nutzung eigener Pods – können sich weitere Vorteile ergeben. So können Testdatenaufträge überall im Cluster oder, falls notwendig, auch global verteilt verwaltet werden. Jede Broker-Instanz kann eigenständig Aufträge aus der Queue bearbeiten und die den Datenzielen entsprechenden Connector-Pods anbinden. Der Lebenszyklus der Testdaten, inklusive deren Bearbeitungsstatus, können verfolgt und abgebrochene oder verwaiste Aufträge wiederaufgenommen werden. Darüber hinaus eliminiert die Auslagerung der Testdaten in Cloud-Services die Datenhaltung innerhalb der Pods. Das Beenden oder ein ungeplanter Verlust von Pods (zer-)stören somit keine Testdatenbestände, und die generelle Verfügbarkeit der Testdaten-Services bleibt unbeeinträchtigt.

Ob die technische Umsetzung dabei mittels REST-APIs, Message-Queues oder Service-Meshes geschieht, hängt letztlich vom jeweiligen Projektumfeld ab. Die Verwendung bereits genutzter Technologien und Toolsets kann hier von Vorteil sein, ist aber auf Grund des Microservice-Ansatzes nicht zwingend.

Letztendlich ermöglicht die Einbindung der Testdaten-Services in die Clusterinfrastruktur auch die Nutzung vorhandener Services wie z. B. Monitoring. Der Status der Testdaten-Aufträge kann somit auch in den typischen DevOps-Dashboards verfolgt werden, und die Feed-Back-Loops der CI/CD-Pipelines können um die spezifischen Aspekte der Erzeugung und Bereitstellung der Testdaten erweitert werden.

Fazit

Testdaten und vor allem Testdatenmanagement sind Themen, die immer wichtiger werden – nicht nur auf Grund von Datenschutzvorgaben. Längst ist klar, welchen großen Nutzen ein Testdatenkonzept mit einem soliden Testdatenmanagement vor allem für den hohen Bedarf an Automation in DevOps-Projekten hat. Kubernetes bietet viele Möglichkeiten für das Testdatenmanagement, die aber in den meisten Projekten noch nicht erkannt, geschweige denn umgesetzt werden. Mit unserem Testdatengenerator als Microservice haben wir ein Konzept entwickelt, welches sich optimal in eine Kubernetes-Umgebung integrieren lässt und dessen technologische und organisatorische Möglichkeiten nutzt. Durch das Einbinden in das Monitoring und in entsprechende DevOps-Dashboards können die Ergebnisse des Testdatenmanagements in einer Feedback-Loop eingesetzt oder als KPI genutzt werden.

Wir bei Trimetis GFB haben hierzu dieses Konzept entwickelt und erproben es derzeit in Kundenprojekten, für die wir den Testdatengenerator um weitere Anforderungen aus dem Testdatenmanagement erweitern.

Über die Autoren

Christoph Knopp

ist bei der Trimetis GFB IT-Consulting GmbH Teamlead für den Bereich Test Data Management. Wenn er nicht gerade seine Kunden berät, deren Testvorgehen analysiert und optimiert, ist er immer auf der Suche nach neuen Trends und Entwicklungen im Softwaretest, mit dem Fokus auf Testdaten und agile Testprozesse.



E-Mail: christoph.knopp@trimetis.com

Daniel Simon

ist Teamlead für den Bereich Test Automation und Test Consulting bei der Trimetis GFB IT-Consulting GmbH. Neben der Optimierung von Testprozessen und der Automatisierung im agilen Testumfeld interessiert er sich insbesondere für die Automatisierung im Continuous Testing und die neuen Möglichkeiten, die DevOps und Kubernetes in diesem Bereich bieten.



E-Mail: daniel.simon@trimetis.com

Links & Literatur

[ASQF-TDM] Klaus Franz, Tanja Tremmel, Eckehard Kruse, dpunkt.verlag, 2018:
<https://dpunkt.de/produkt/basiswissen-testdatenmanagement/>

Bildrechte

utah778/istockphoto.com; Gordon Johnson/pixabay;
Ferdichack/pngtree.com; Trimetis GFB

DAMIT IHNEN DER SOFTWARETEST IN
DEVOPS NICHT UM DIE OHREN FLIEGT!

DEV
OPS!

TEST DATA AAS

MICROSERVICES

CLOUD
NATIVE

KUBERNETES

[GFB.TRIMETIS.COM/DEVOPS/](https://gfb.trimetis.com/devops/)

AUTOMATION

CONTINUOUS
TESTING

CONTAINER

CI/CD PIPELINE

Impressum

Trimetis **GFB**

Trimetis GFB IT-Consulting GmbH
Obere Zeil 2 (Atrium)
61440 Oberursel

Kontakt:

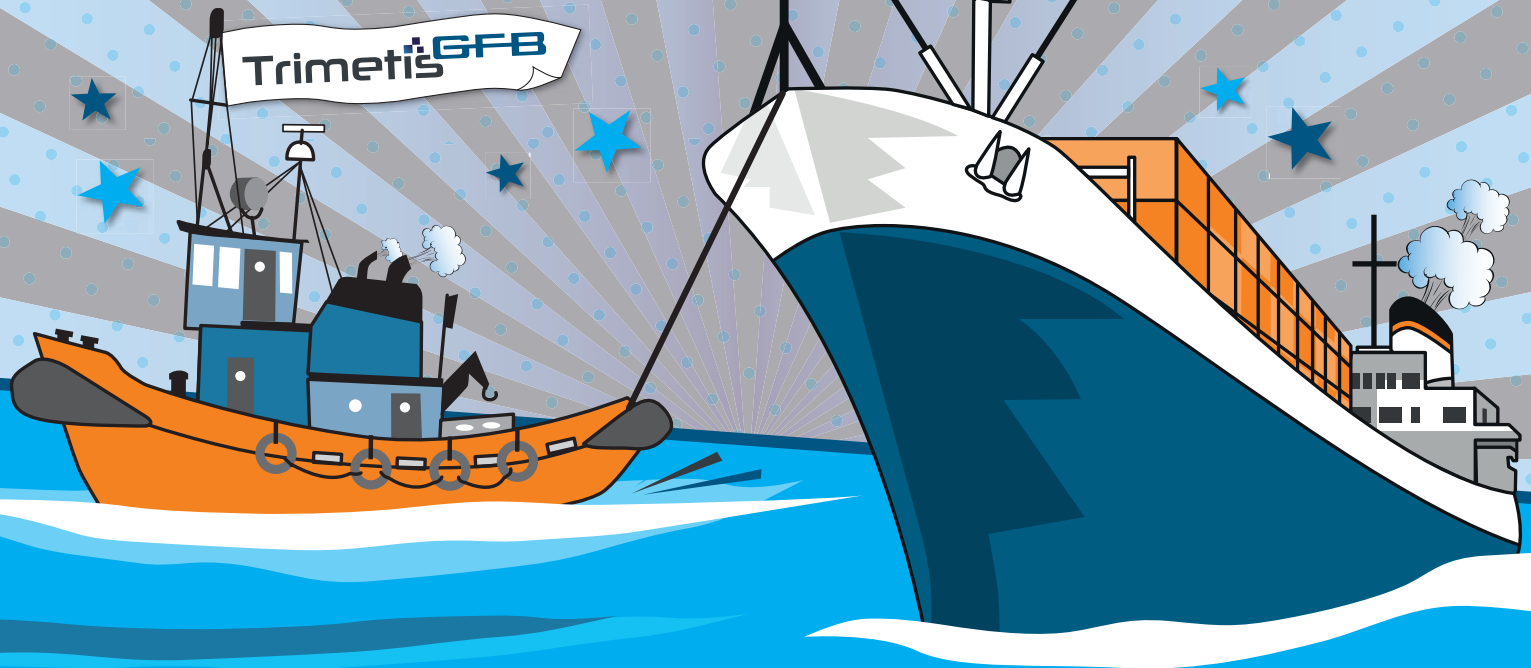
Internet: <https://gfb.trimetis.com>
E-Mail: Office-GFB@trimetis.com

Geschäftsführer: Mag. Peter Laggner, Gottfried Madl

HRB: 5878 Amtsgericht Bad Homburg

Copyright © 2021 Trimetis GFB IT-Consulting GmbH, Oberursel

DAMIT SIE MIT KUBERNETES IMMER AUF KURS BLEIBEN



Trimetis[®] GFB

Trimetis GFB IT-Consulting GmbH
Telefon +49 6171 5060-60
gfb@trimetis.com

<https://gfb.trimetis.com>

