Der Leitfaden zu **modernem APM**

splunk>
turn data into doing™

Grundlegendes zu APM für Ihren Weg zu Cloud-nativen Systemen

Um das Innovationstempo zu steigern und um ihre Investitionen in Digitalisierung und Cloud-Technologien zu schützen, brauchen Unternehmen dringend moderne Lösungen für Infrastruktur-Monitoring und Application Performance Monitoring (APM) – und zwar solche, die in Cloud-Systeme integriert sind und speziell für die Cloud entwickelt wurden.

Als Ergänzung zum Infrastruktur-Monitoring macht APM die Abläufe in der Anwendungsschicht transparent, sodass sich etwaige Probleme dort besser finden, analysieren und beheben lassen, ganz gleich, ob es sich um Probleme mit Anfragen, um Fehler oder um Latenzen handelt. Dass Sie mit der richtigen APM-Lösung den Weg einer Anforderung durch die Servicearchitektur abbilden können, ist ungeheuer wichtig. Schließlich bedeutet „langsam“ in unserer Always-on-Welt fast schon so viel wie „ausgefallen“.

Während Infrastruktur-Monitoring aufzeigen kann, ob ein Problem vorliegt, hilft Application Performance Monitoring den Teams bei der Lokalisierung des Problems. APM-Tools sind dafür ausgelegt, sicherzustellen, dass Anwendungen störungsfrei das festgelegte Service Level bieten. Die Schnelligkeit und Verfügbarkeit von Anwendungen – und zwar interner Unternehmensanwendungen ebenso wie Consumer-Apps – stehen in direktem Zusammenhang mit dem Unternehmenserfolg. Wenn bekannt ist, worauf ein Ausfall in der IT-Umgebung zurückzuführen ist, kann er deutlich schneller behoben werden, was seine Folgen erheblich verringert.

Unternehmen schaffen das jedoch nicht alleine. Sie benötigen ein Ökosystem aus zuverlässigen Partnern, die umfassende Ansätze zur effektiven Nutzung sämtlicher Daten bieten – anstatt Teillösungen, die Innovationen ausbremsen oder sogar ein falsches Bild des aktuellen Zustands geben. Das Ziel besteht darin, einen Grad von Observability zu erreichen, bei dem das ganze Bild detailgenau und immer aktuell ist. Das gelingt, wenn Monitoring und Warnmeldungen all diese Aspekte auch berücksichtigen können.

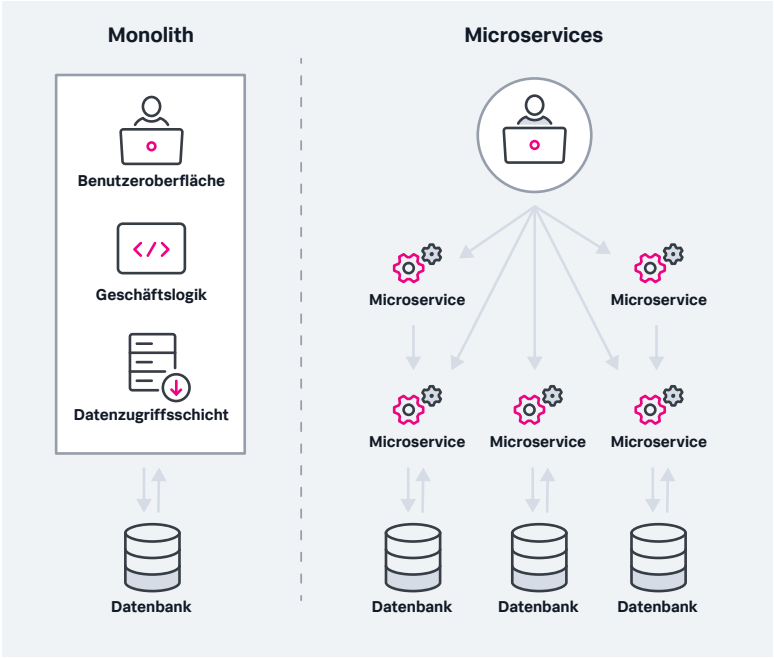
Was hat sich geändert?

Obwohl die meisten jungen und fortschrittlichen Unternehmen bei der Cloud-Migration Tempo machen, arbeiten viele Firmen immer noch mit einer monolithischen Architektur. Wie ein tausend Tonnen schwerer Felsen besteht eine monolithische Architektur aus unzähligen Sedimentschichten von Funktionen und redundanter Logik in Tausenden von Codezeilen in einer einzigen Sprache. Es kann schwierig werden, eine solche monolithische Architektur in kleinere Einheiten runterzubrechen. Dies wiederum macht das Skalieren so schwierig, weil Sie die gesamte Umgebung skalieren müssen und nicht nur einen speziellen Anforderungsbereich. Allerdings sind Code-Pushes eher selten, was das Monitoring relativ einfach macht.

Microservices dagegen sind vergleichbar mit einem Haufen winziger, aus dem Felsbrocken herausgeschnitter Würfel. Sie bestehen aus verteilten Komponenten, die jeweils durch eine Reihe spezifischer Merkmale von Geschäftsfunktionen beschrieben werden. Da man in Microservice-Architekturen häufig Container verwendet, um einzelne Funktionen zu „kapseln“, können Sie jeden einzelnen Service ganz leicht skalieren, wenn sich Ihre Anforderungen ändern.

Diese lose gekoppelten verteilten Services leisten insgesamt die Funktionalität des ursprünglichen Monolithen; sie folgen den ereignisgesteuerten, serviceorientierten Architekturprinzipien, die komplexe Anwendungen aus eigenständigen Prozessen zusammensetzen, die mittels APIs über ein Netzwerk miteinander kommunizieren.

Anwendungen, die mit Microservices erstellt werden, sind in der Regel in vielen unterschiedlichen Sprachen geschrieben, und jeder Service kann unabhängig skaliert werden. Weil die einzelnen Services voneinander unabhängig sind, können sie nach je eigenen Fahrplänen bereitgestellt werden. Unternehmen gewinnen auf diese Weise zusätzliche Agilität für häufigere Code-Pushes, sodass sie Benutzer-Feedback regelmäßiger einarbeiten können. Bei häufigeren Code-Pushes und unabhängigen Komponenten, die dauernd aktualisiert werden, benötigen Unternehmen aber ganz andere, intensivere Monitoring-Prozesse.

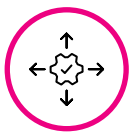


Viele namhafte Unternehmen haben schon von monolithischer Architektur auf Microservices umgestellt. Dazu gehören beispielsweise Amazon, Spotify, Uber und Groupon. Mit Microservices stellen die Entwickler bei Netflix täglich Tausende von Codeabschnitten für mehr als 139 Millionen Abonnenten und 10 Milliarden Stunden an Filmen und TV-Serien bereit.

Ein Hauptargument für den Umstieg auf Microservices ist, dass man sich aufgrund des höheren Innovationstempos besser auf die wirtschaftlichen Prioritäten des Unternehmens konzentrieren kann. Das Aufkommen von DevOps, das den Fokus auch auf Geschwindigkeit und Ergebnisse legt, hat ebenfalls das Interesse an Microservices befeuert und viel dazu beigetragen, dass sie vermehrt genutzt werden.

Was sind die Gründe für diesen Wandel?

Skalierung on Demand



Anwendungen können skaliert werden und einer stärkeren Nachfrage sofort nachkommen.

Gesteigerte Produktivität

Neue Produkte gelangen schneller zum Kunden.

Schnellere Problemerkennung

Programmierer finden Probleme in ihren Services schneller.

Höhere Verfügbarkeit

Anwendungen sind rund um die Uhr verfügbar.

Was sind die Herausforderungen?

Herkömmliche Monitoring-Lösungen sind für die neuen operativen Herausforderungen nicht gewappnet, die moderne Technologien mit sich bringen: Container, Kubernetes, serverlose Funktionen und Microservices sowie diverse DevOps-Praktiken, die mit dazugehören.

Aber warum nehmen Unternehmen und DevOps-Teams diese ganze Komplexität überhaupt auf sich?

Sie migrieren in die Cloud und zu Microservices, weil sie damit die Skalierbarkeit verbessern, die Produktivität steigern, Probleme schneller erkennen und beheben, die Anwendungsverfügbarkeit verbessern – und letztlich die bestmögliche User Experience bieten können.

Monitoring ist ein wichtiger Bestandteil jeder Microservice-Architektur. Das Aufsplitten von Anwendungen in einzelne Microservices bringt zwar viele Vorteile, erhöht aber auch die Komplexität. Microservices müssen miteinander kommunizieren, und jede einzeln erstellte und aktualisierte Komponente muss mit einem Minimum an Latenz mit anderen Komponenten zusammenarbeiten. Wenn Sie also eine aus Microservices bestehende Anwendung verwalten, dann verwalten Sie ein Netz von Komponenten, die zueinander in Beziehung stehen. Ein effektives Monitoring dieser zahllosen Kommunikationsinteraktionen ist für die Zuverlässigkeit des Gesamtsystems unerlässlich. Das Stichwort hierzu lautet Observability.

Observability trägt wesentlich dazu bei, diese Gesamtzuverlässigkeit zu erreichen. Observability ist ein Zielkonzept, das neue Ansätze nutzt, um Erkenntnisse zur Performance von Anwendungen zu gewinnen. Dies geschieht meist durch die Bereitstellung sehr detaillierter Daten in drei Kategorien: Metriken, Traces und Logs. Durch diese detaillierten Daten macht es Observability möglich, auch die „unknown unknows“ bzw. „unbekannten Unbekannten“ in Anwendungen zu untersuchen und so Antworten auf Fragen zu finden, an die Sie vermutlich noch gar nicht gedacht haben, und Probleme aufzudecken, sobald sie auftreten.

Observability muss unbedingt mit dem Monitoring kombiniert werden. Entwicklern, die bereits ein DevOps-Mindset verinnerlicht haben, fallen Monitoring und Observability leicht. Genau wie bei DevOps setzen auch Microservices auf Automatisierung und Zusammenarbeit über alle Abschnitte und Aspekte des Softwareentwicklungszyklus (SDLC) hinweg. Konfigurationsmanagement, CI/CD-Server, APM, Dashboards, automatische Warnmeldungen und Incident Management gehören für Teams, die Microservices laufen lassen, zu den Grundlagen.

Klassisches APM schafft es einfach nicht mehr

Es gibt etliche Probleme, die herkömmliches APM nicht lösen kann:

- Unvollständige Trace-Erfassung führt dazu, dass Anomalien nicht erkannt werden, sodass diese Daten ungenutzt bleiben und die Produktivität bei den Entwicklern geringer ausfällt.
- Batch-basierte Analytik bedeutet langsame Problemerkennung.
- Komplizierte Preisstrukturen und Performance-Probleme behindern notwendige Skalierungen und damit das weitere Wachstum.
- Schwerfällige, proprietäre Tools verhindern Innovationen.
- Der Einsatz mehrerer separater Observability-Tools führt dazu, dass viele Schritte mehrfach durchgeführt werden müssen.

Full-Fidelity Tracing, Metriken und Logs

Es gibt zwei wichtige Fragen bezüglich unserer Anwendungen, die wir beantworten können müssen. Erstens: Funktioniert die Anwendung richtig? Zweitens: Hat die Anwendung jemals wie erwartet funktioniert? An diesem Feedback sollten sich fast alle grundlegenden Maßnahmen orientieren.

Ohne konsistentes und vollständiges Feedback lässt sich gar nicht feststellen, ob eine Änderung gegriffen hat oder wann bei einem bestimmten System Probleme auftreten. Ganz gleich, ob Sie Code entwickeln oder eine Infrastruktur bereitstellen, Feedback ist ein zentraler Bestandteil des Erfolgs.

Wie können Sie unabhängig von der Größe Ihres Unternehmens durch das richtige Maß an Feedback konsequent zum Kern eines Problems vordringen? Und ebenso wichtig: Wie können Sie Probleme in Echtzeit erkennen und beheben?

Trace Sampling vs. Full-Fidelity Tracing

Die Umstellung auf Public Clouds und Cloud-native Anwendungen hat neue Möglichkeiten eröffnet, die Unternehmen viele Vorteile bringen: einfachere Skalierung, schnellere Anpassung der Rechenkapazitäten und geringeren Bedarf an Rechenzentren, um nur einige zu nennen. Doch sie hat auch eine Reihe neuer Herausforderungen mit sich gebracht. Mit APM können Sie diese Herausforderungen meistern, aber nur, wenn Sie es richtig anstellen.

Die meisten APM-Lösungen verwenden lediglich Stichproben, um den Ursachen von Performance-Einbrüchen oder Fehlern auf die Spur zu kommen, und zwar einfach deshalb, weil sie so die Datenmenge reduzieren, die sie analysieren müssen. Wenn aber eine Lösung gar nicht alle Daten erfasst und analysiert, kann sie zwangsläufig keine geeigneten Erkenntnisse über das fehlerhafte Verhalten liefern.

Nur APM-Lösungen, die sämtliche Transaktionsdaten in Echtzeit erfassen, können Unternehmen dabei helfen, die User Experience zu verbessern, den guten Ruf der Marke zu wahren und die Markteinführung zu beschleunigen. Dank offener Standards wie OpenTelemetry befreit Splunk APM Ihren Code von den Fesseln der Anbieterabhängigkeit. Mit Splunk können Sie genau die Sprachen und Frameworks nutzen, die sich am besten für Ihre Zwecke eignen. OpenTelemetry unterstützt zudem alle drei Kategorien von Observability-Daten und ist ein Schritt in Richtung digitale Demokratisierung, heraus aus dem Vendor Lock-in.

Metriken

Diese Zahlen geben Einblick in einen Prozess, einen Vorgang oder in den Status eines Systems, Netzwerks oder Speichers. Im Allgemeinen werden Metriken im zeitlichen Verlauf gemessen und daher oft auch als Zeitreihen bezeichnet.

- **Systemmetriken** (Auslastung von CPU und Arbeitsspeicher, Datenträger-E/A etc.)
- **App-Metriken** (Rate, Fehler, Latenz)
- **Geschäftsmetriken** (Umsatz, Kundengewinnung, Absprungrate, verlassene Warenkörbe)

Traces

Als Traces bezeichnet man die Aufzeichnung des Wegs, den eine Anforderung durch eine Anwendung nimmt, inklusive all der unzähligen Services.

Eine Trace-Aufzeichnung umfasst normalerweise Daten zu den folgenden Aspekten:

- **Spans** (Servicename, Vorgangsname, Dauer und weitere Metadaten)
- **Fehler**
- **Dauer wichtiger Vorgänge innerhalb der einzelnen Services**
- **Benutzerdefinierte Attribute**

Logs

Logs sind Ereignisprotokolle, unveränderliche Aufzeichnungen definierter Events in ihrem zeitlichen Ablauf. Event-Logs liegen in Form von unformatiertem Text, strukturiert oder als Binärdaten vor. Gängige Logfiles sind z. B. die folgenden:

- **System- und Serverlogs** (syslog, journald)
- **Logs von Firewalls und Intrusion-Detection-Systemen**
- **Feeds aus sozialen Medien** (Twitter usw.)
- **Anwendungs-, Plattform- und Serverlogs** (log4j, log4net, Apache, MySQL, AWS)

Metriken

Metriken sind die Zahlen, die uns Einblick in einen Prozess, eine Aktivität oder den Status von grundlegenden Systemen, Netzwerken oder Speichern geben. Im Allgemeinen werden Metriken im zeitlichen Verlauf gemessen, weshalb sie oft auch als Zeitreihen (Time Series) bezeichnet werden. Sie werden über Zeiträume hinweg erfasst, weil Systeme eben nicht stillstehen, sondern sich ständig verändern.

In puncto Metriken liefert ein Service einen Metrikschlüssel (das „Was“) und einen zugehörigen Wert. Dies wird noch mit einem Zeitstempel (dem „Wann“) versehen, sodass sich Zeitreihendaten ergeben und die Werte über ein Intervall als Reihe von Datenpunkten dargestellt werden können.

Sowohl beim Logging als auch bei Metriken liefert aber nicht nur die Anwendung selbst Erkenntnisse: Auch Strukturen (z. B. Cloud-Konstruktionen), Datenbanken, Cache-Speicher, Warteschlangen, Server und alle möglichen anderen Dinge generieren Telemetriedaten (Ausgabedaten von Anwendungen und Infrastruktur), die in unterschiedlichem Maß neue Erkenntnisse ermöglichen.

Das richtige APM-Tool berücksichtigt all dies und liefert Ihnen ein präzises und umfassendes Bild, mit weniger Fehlalarmen und ausführlichen Hintergrundinformationen zu jeder Meldung.

Traces

Als Traces bezeichnet man die Aufzeichnung des Wegs, den eine Anforderung durch eine Anwendung nimmt, inklusive all der unzähligen Services. Denn Anwendungen sind komplex. Es ist heute nicht mehr so, dass jeder den kompletten Flow eines Systems überschauen könnte. Das heißt, dass wir auf Traces angewiesen sind, wenn wir wissen möchten, welchen Weg eine Anforderung durch die Anwendung nimmt.

Traces liefern uns ebenfalls Metriken. Im Allgemeinen werden sie mit einem modernen Monitoring-Verfahren namens RED (Rate, Errors, Duration) dargestellt. Diese Kombination von Rate, Fehlern und Dauer bietet einen schnellen, aussagekräftigen Überblick über die separaten Services und die Performance der Anwendungen. Mit RED können wir auch KI-/ML-Techniken nutzen (künstliche Intelligenz/maschinelles Lernen), um Probleme zu identifizieren und optimal darauf zu reagieren.

Logs

Ein Log-Eintrag ist die Aufzeichnung eines definierten Ereignisses in einem System, einer Anwendung, einer Umgebung oder einem anderen Element. Logs können in verschiedenen Formaten vorliegen, z. B. als Rohdaten oder in strukturierter Form, und aus vielerlei Quellen stammen, etwa von Systemsoftware, Software von Drittanbietern oder proprietären Lösungen.

Um für Observability zu sorgen, benötigen Sie Input aus allen drei Datenkategorien, also Metriken, Traces und Logs. Dies sind die drei Säulen der Observability. Metriken zeigen Ihnen, ob ein Problem vorliegt. Traces helfen Ihnen, das Problem zu beheben. Und mit Logs finden Sie heraus, was das Problem verursacht hat.

Mit einer Observability-Lösung, die alle drei Kategorien umfasst, können Sie Probleme rasch erkennen, schnell beheben und die Ursachen herausfinden, sodass Sie Ihre MTTR (Mean Time To Repair) reduzieren und die Systeme stets verfügbar und am Laufen halten.

Mehr als Monitoring – der Umstieg auf Observability

Je weiter Sie auf Ihrem Weg zu Cloud-nativen Systemen vorankommen, desto eher stößt das klassische Monitoring an seine Grenzen. Sie erfassen dann mehr Daten aus mehr Quellen, es gibt häufige Änderungen und die Services verhalten sich elastischer. Verteilte Microservices mit komplexen Abhängigkeiten, einer schnelllebigen Infrastruktur und regelmäßigen Code-Pushes resultieren in komplexeren Monitoring-Aufgaben.

Nur wenn Sie eine Observability-Kultur schaffen und eine moderne Observability-Plattform nutzen, werden Sie auf Ihrem Weg zu Cloud-nativen Systemen vorankommen und das für Ihren dauerhaften Geschäftserfolg notwendige Tempo erreichen. Die richtige Lösung sollte vollständige Ende-zu-Ende-Transparenz leisten, Ergebnisse in Echtzeit streamen, betriebswirtschaftliche Erkenntnisse liefern und dabei KI-/ML-Techniken nutzen, damit der Aufwand bei der Fehlerbehebung minimal bleibt.

Bessere Customer Experience

Probleme abfangen, bevor sie sich auf Kunden auswirken; mit minimaler MTTR (Mean Time To Repair).

Besser vorhersagbare Prozesse

Alerts in Sekunden-schnelle ergeben kürzere Entdeckungs- und Reaktionszeiten.

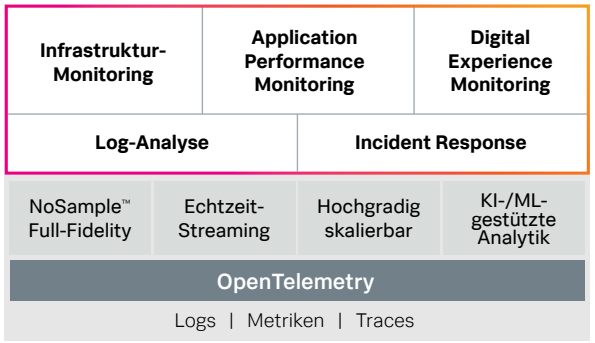
Höhere Ressourcen-effizienz

Transparenz verbessern und Metriken aus IT-System und Betriebs-wirtschaft zueinander in Beziehung setzen.

Höhere Produktivität der Entwickler

Mehr Agilität und Effektivität in der Software-Entwicklung führen schneller zu besserem Code.

Die Splunk Observability Suite



Umfassende, integrierte Lösung

Nahtloser Workflow aus Monitoring, Fehlersuche, Analyse und Behebung

APM-Modernisierung zu Ihrem Vorteil

Observability

NoSample™ Full-Fidelity Tracing spürt sämtliche Anomalien auf, und dank unserer Streaming-Architektur passiert dies in Echtzeit, also bevor sich Auswirkungen auf die Anwender ergeben. Zudem werden Trace-Daten für mehrere Programmiersprachen und Open-Source-Frameworks erfasst, automatisch oder manuell instrumentiert. Und falls Sie damit nicht gleich klarkommen: Splunk ist aktives Mitglied und Hauptmitwirkender im OpenTelemetry-Projekt – es stehen also erstklassige Experten zu Ihrer Unterstützung bereit. Und da Splunk Infrastructure Monitoring auch für das Handling und die Analyse in größeren Dimensionen ausgelegt ist, kann das System mit Abhängigkeitsanalysen sofort aufzeigen, wie sich ein neues Release auf andere Services auswirkt, sodass Ihr Team leicht erkennt, welche Folgen für einzelne Benutzer oder Komponenten zu erwarten sind.

Kontrolle

Ihre Daten, Ihre Entscheidungen. Sie entgehen der Abhängigkeit von einzelnen Anbietern und verkürzen die Time-to-Value durch Open Source, schlanke Agents und eine Instrumentierung mit offenen Standards. Sie bekommen Hunderte einsatzfertiger Integrationen für gängige OSS, Cloud-Infrastruktur und Services an die Hand; damit können Sie automatisch Standardmetriken Ihrer Services abrufen, in vorbereitete Dashboards einfließen lassen und sich so schnell ein klares Bild verschaffen. Alerts richten Sie dynamisch per Point-and-Click ein, und ein Smart Agent zur automatischen Service-Erkennung sorgt dafür, dass sich Ihr System einfach anpasst. Ebenfalls mit dabei: vollautomatisches Kubernetes-Monitoring, sodass Ihr Team bei Fehlersuche und Ursachenanalyse schneller vorankommt.

Geschwindigkeit

Probleme beheben Sie schnell mit KI-gestützter Fehlersuche und Sofortwarnungen, die innerhalb von Sekunden anspringen – so schnell wie das Monitoring serverloser Funktionen nur möglich ist. Hochauflösende, benutzerfreundliche Dashboards und Diagramme aktualisieren sich in Echtzeit mit den für Sie wichtigsten Metriken; Sie können sich sogar eine konsolidierte Live-Heatmap Ihrer gesamten Infrastruktur anzeigen lassen. Das Ergebnis: Die Leistungseinbußen bleiben minimal, und Benutzer erleben Ihre Anwendungen in Bestform.

Erste **Schritte.**

Die hier angesprochenen Punkte machen den Unterschied zwischen einem APM-Tool aus, das einigermaßen akzeptabel läuft, und einem, das richtig gut zu Ihren Bedürfnissen passt – jetzt und in Zukunft. Nur wenn Sie sicherstellen, dass Ihre Lösung alle genannten Punkte abdeckt, lassen Sie das alte Monitoring mit viel manueller Arbeit und Wartezeiten wirklich hinter sich.

Erfahren Sie, wie Sie und Ihr Unternehmen maximale Business Performance sicherstellen, Ausfallzeiten minimieren und erstklassige digitale Erfahrungen bieten können.

Jetzt mehr erfahren

splunk>

Splunk, Splunk>, Data-to-Everything, D2E und Turn Data Into Doing sind Marken und eingetragene Marken von Splunk Inc. in den Vereinigten Staaten und anderen Ländern. Alle anderen Markennamen, Produktnamen oder Marken gehören den entsprechenden Inhabern. © 2020 Splunk Inc. Alle Rechte vorbehalten.

20-16346-SPLK-The Guide to Modern APM-106