

## LISTICLE

### **Observability ist mehr als reines Monitoring**

**München, 20. Juni 2022 – Die Nutzung von Cloud, Containern und Microservices hat die Anwendungslandschaft deutlich komplexer gemacht. Klassisches Monitoring zur Überwachung von einzelnen Systemen genügt daher nicht mehr. Besser eignet sich Observability, die Logging, Tracing und Monitoring zentral in sich vereint. Für eine erfolgreiche kontextbasierte Fehlerquellenanalyse sollten Unternehmen die fünf Best Practices von IT-Dienstleister Consol berücksichtigen.**

Tritt in einem IT-System ein Fehler auf, alarmiert klassisches Monitoring die Administratoren. Gut implementiert zeigt es, wo es brennt und was genau nicht mehr funktioniert. Für die Beantwortung der Frage nach dem Warum brauchen Administratoren allerdings einen tieferen, ganzheitlichen Einblick in die Systeme und Microservices. Monitoring stützt sich in diesem Zusammenhang insbesondere auf die Überwachung von möglichen Problemen, die der Betrieb vorhersehen muss. Darauf basierend muss die Operations-Abteilung dann ihre Dashboards konfigurieren. Bei einer Observability-Strategie hingegen erhält der Betrieb Daten aus dem gesamten System. So können Administratoren flexibel analysieren, was in all den miteinander verknüpften Umgebungen vor sich geht und wo der wahre Grund für Fehler liegt. Da die Überwachung von hochkomplexen Systemen selbst nicht ganz einfach ist, hat IT-Dienstleister Consol die folgenden fünf Best Practices definiert, nach denen Unternehmen ihre Observability-Strategie ausrichten sollten.

#### **1. Die Zielgruppen im Blick behalten**

Logs sind nur dann wirklich hilfreich, wenn ihre Inhalte auf die Zielgruppe ausgerichtet sind. Beim Logging gibt es drei relevante Zielgruppen: Betrieb, Entwickler und die Fachbereiche. Im Kontext der Observability bedeutet zielgruppengerechtes Logging also, dass die Logs genau die Informationen enthalten müssen, die für Wartung und Betrieb von Anwendungen relevant sind. Während Entwickler beispielsweise in ihren Logs haargenau sehen wollen, in welcher Code-Zeile ein Fehler auftritt, ist es für Administratoren wichtiger zu erfahren, welche Auswirkungen er auf andere Systemteile hat. Fachbereiche hingegen interessieren sich vor allem dafür, wie die geschäftlichen Use Cases laufen und ob es dort Probleme gibt.

## **2. Den Kosten-Nutzen-Faktor abwägen**

Umfangreiches Logging ist die Basis für erfolgreiche Observability. Dennoch kann weniger manchmal mehr sein, insbesondere bei der Abwägung des Kosten-Nutzen-Faktors. Das Sammeln von Daten ist gerade im Cloud-Kontext ein großer Kostenfaktor: Nicht nur die Speicherung ist teuer, zu Buche schlagen zudem Netzwerk-Traffic und Konfigurationsaufwand. Auch die Wartung und Aktualisierung der Logging-Infrastruktur verursacht durch hohen Personalaufwand Kosten. Unternehmen sollten daher nur die Daten sammeln, die für ihre Zwecke wirklich notwendig sind.

## **3. Langfristiges und holistisches Monitoring betreiben**

Gutes Monitoring als Teil einer Observability-Strategie geht weit über die Beobachtung technischer Standardmetriken wie Prozessorlast oder Arbeitsspeicherbedarf hinaus. Fachliche Metriken, etwa wie lange das Rendern von Komponenten auf einer Webseite dauert, müssen Unternehmen individuell und je nach Anwendungsfall selbst definieren. Darüber hinaus ist Monitoring erst dann wirklich effektiv, wenn es langfristig angelegt ist. Unternehmen sollten etwa nach jedem Software-Release oder der Implementierung neuer Features genau hinschauen, wie und ob sich die Performance und Gesundheit des Systems verändert haben. Voraussetzung dafür ist, entsprechende Logs in Form einer Monitoring-Historie vorrätig zu haben.

## **4. Gutes Alerting definieren**

Zur holistischen Observability-Strategie gehört auch die Definition von Alerting-Regeln. Das Monitoring versorgt Administratoren mit Informationen über das System in Echtzeit, sodass sie jederzeit nachschauen können, ob alles in Ordnung ist. Weniger zeitraubend ist es, wenn das System eigenständig Alarm schlägt, zum Beispiel sobald innerhalb von fünf Minuten ein gewisser Prozentsatz von Zugriffen auf eine Anwendung Fehler aufweist. Dann können die Verantwortlichen gezielt prüfen, was nicht stimmt und wo ein Eingreifen nötig ist. Voraussetzung dafür sind geeignete Metriken, die die Applikation bereitstellt. Dazu gehören neben technischen auch fachliche Metriken, die individuell die Business Use Cases überwachbar machen, für die das System verantwortlich ist.

## **5. Offene Standards nutzen**

Open-Source-Software (OSS) setzt sich auch im professionellen IT-Umfeld immer mehr als lukrative Alternative zu proprietären Varianten durch. Gerade im DevOps-Bereich sind Open-Source-Tools wie Prometheus (Monitoring und Alerting), Jaeger (Tracing), Logstash (Logging) und Kibana (Visualisierung) weit verbreitet. Die meisten von ihnen setzen auf offene Standards wie OpenMetrics, OpenTracing und OpenTelemetry. Die Vorteile von OSS und offenen Standards sind ihre Vielseitigkeit, die große Innovationskraft der Community sowie die hohe Kompatibilität und Anpassbarkeit.

„Observability ist heute nicht mehr Kür, sondern eindeutig Pflicht“, betont Lutz Keller, Leiter DevOps bei Consol. „Cloud, Microservices und Container-Technologie machen eine intelligente und holistische Überwachungsstrategie für IT-Systeme dringend nötig. Mittlerweile sind auch Machine Learning und künstliche Intelligenz in der DevOps-Welt angekommen, die wiederkehrende Tätigkeiten übernehmen können. Sogenanntes AIOps wird in Zukunft spannende neue Anwendungsmöglichkeiten eröffnen und Administratoren entlasten.“

Wer einen noch tieferen Einblick in das Thema Observability erhalten möchte, der sollte am 23. Juni den Webcast von Consol mit dem Thema „The Observability of Quarkus Applications“ besuchen. Weitere Informationen dazu gibt es unter <https://www.consol.de/aktuelles/webcast-observability/>.

**Dieses Listicle und Bildmaterial sind abrufbar unter [www.pr-com.de/companies/consol](http://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](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](mailto:Isabel.Baum@consol.de)

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

PR-COM GmbH

Domenik Brader

Sendlinger-Tor-Platz 6

D-80336 München

Fon: +49-89-59997-712

E-Mail: [domenik.brader@pr-com.de](mailto:domenik.brader@pr-com.de)

Web: [www.pr-com.de](http://www.pr-com.de)