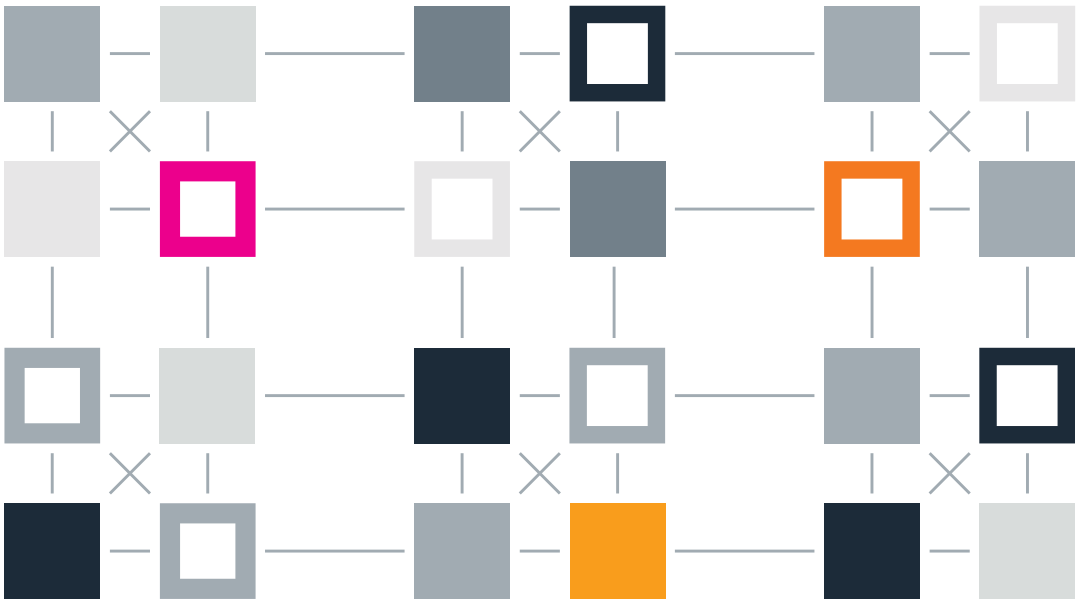
Monitoring von Microservices in Kubernetes



Laut der neuesten halbjährlichen Umfrage der Cloud Native Computing Foundation (CNCF) stellt das Monitoring eine der größten Herausforderungen für Organisationen dar, die Kubernetes einführen.

Dieses Whitepaper gibt Einblicke dazu, wie das Monitoring sowohl von Kubernetes-Clustern als auch von auf Kubernetes bereitgestellten Microservices bestmöglich gestaltet werden kann, um Performance-Probleme zu erkennen und die MTTR (Mean Time-To-Resolution) zu minimieren.

Das Management von Performance-Probleme in Echtzeit stellt eine praktische und kontrollierbare Möglichkeit dar, um eine fehlerfreie Endbenutzererfahrung zu ermöglichen. Unternehmen müssen Best Practices für den Betrieb von Kubernetes-Clustern sowie für die Beurteilung der Performance des gesamten Stacks entwickeln, einschließlich verteiltem Tracing über alle Microservices hinweg.

Das Gehirn hinter der Orchestrierung von containerisierten Anwendungen

Kubernetes ist eine führende Container-Orchestrierungslösung für den Einsatz containerisierter Anwendungen. Unternehmen aus allen Branchen führen Microservices in Containern ein, um folgende Vorteile zu nutzen:

- **Geschwindigkeit** – um das Innovationstempo zu beschleunigen
- **Skalierung** – um optimierte Anwendungserfahrungen über geografische Grenzen und über verschiedene Cloud-Anbieter hinweg zu bieten
- **Effizienz** – um durch eine optimierte Ressourcenauslastung mehr Wert bei geringeren Kosten zu bieten

Kubernetes verfügt über mehrere logische Schichten, um verschiedene Anwendungsfälle zu unterstützen und den Lebenszyklus von containerisierten Microservices zu verwalten.

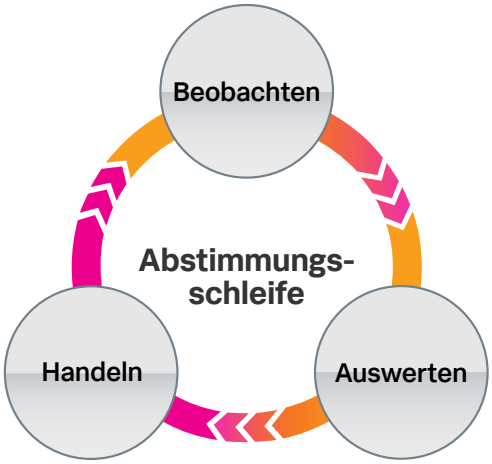
Die Kubernetes-Deployment-Philosophie stützt sich auf drei grundlegende Säulen

Infrastrukturabstraktion	Deklarative Konfiguration	Unveränderlichkeit
Anwendungsorchestrierung aus den Infrastrukturressourcen wie Compute-Servern, Speichervolumen und Netzwerken	Kubernetes überwacht kontinuierlich den Zustand der bereitgestellten Anwendungen und der Kubernetes-Objekte. Es implementiert darüber hinaus den gewünschten Systemzustand, wie er von den Operatoren ausgedrückt wird, etwa bei der Frage, wie viele Replikate eines Services ausgeführt werden sollen	Die von Docker festgelegten Anforderungen an die Unveränderlichkeit werden fortgesetzt; wobei Container-Images, ebenso wie Kubernetes-Objekte, unveränderlich sind, d. h., verschiedene Versionen von Kubernetes-Services sind vollständig getrennt und untereinander nicht austauschbar

Warum Sie Kubernetes nutzen sollten?

Über die Vorzüge hinaus, welche die Containerisierung für Anwendungsentwickler bereithält, macht Kubernetes das Leben von Operations in vielerlei Hinsicht einfacher:

Automatisiertes Management des Anwendungslebenszyklus-	Cloud-Interoperabilität	Selbstheilendes System
Kubernetes automatisiert das Management des Anwendungslebenszyklus als End-to-End-Lösung einschließlich Bereitstellung von Anwendungen, automatischer Skalierung und Replikation ebenso wie CI/CD	Kubernetes bietet Abstraktion von der Infrastruktur, sodass Anwendungen wirklich über verschiedene Clouds hinweg portierbar werden	Kubernetes schützt Anwendungen vor Ausfällen oder unzuverlässigem Verhalten, welches das System destabilisiert. Es ergreift kontinuierlich Maßnahmen, um sicherzustellen, dass der aktuelle Zustand des Systems dem gewünschten Zustand entspricht, wie dieser in der deklarativen Konfiguration ausgedrückt ist

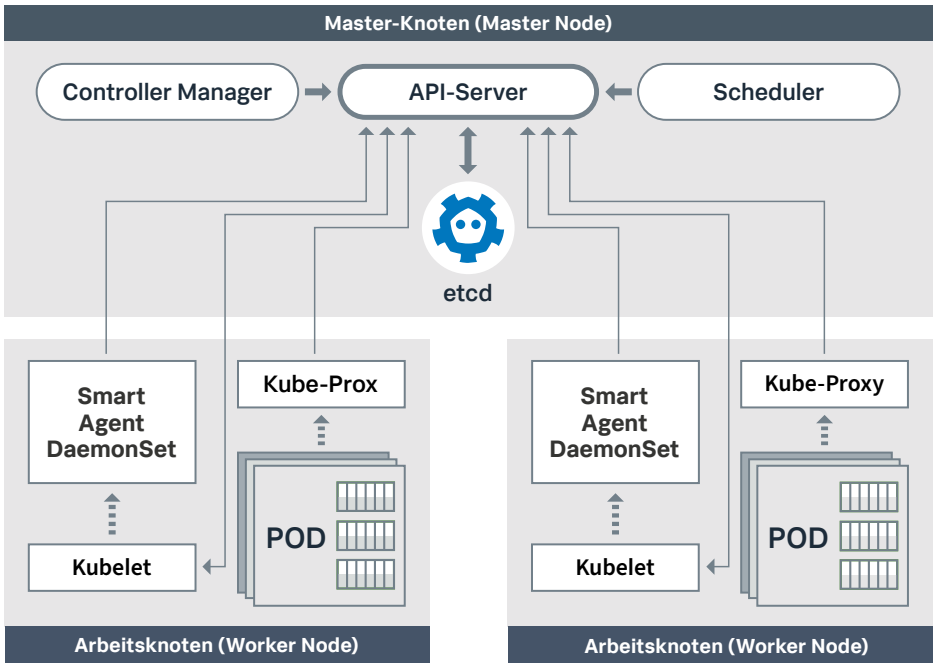


Der Kubernetes Controller-Manager vergleicht fortlaufend den Soll-Zustand mit dem Ist-Zustand, bewertet die Unterschiede zwischen ihnen und gleicht sie mithilfe entsprechender Maßnahmen aus.

Kubernetes-Komponenten verstehen

Kubernetes ist ein System, das eine Gruppe von Computern in einer einzelnen Einheit zusammenfasst, welche über eine API konsumiert werden kann. Kubernetes teilt die Compute-Ressourcen weiter in zwei Gruppen auf:

Arbeitsknoten und Master-Knoten



Im Folgenden finden Sie die Hauptkomponenten, die auf Master- und Arbeitsknoten ausgeführt werden:

Master-Knoten
API-Server • Gateway zum Kubernetes-Cluster • Vermittelt alle Anforderungen von Clients und API-Objekten, die in etcd gespeichert sind • Führt Authentifizierung und rollenbasierteZugriffssteuerung (RBAC) durch • Fordert Validierung und Zulassungssteuerung an • Weitere Informationen: http://bit.ly/k8s-apiserver
Steuerungs-Manager (Control Manager) • Daemon-Prozess, der die in Kubernetes integrierten Steuerungs-schleifen implementiert – rollierende Bereitstellungen, Replica Sets, Anzahl der Arbeitsknoten usw. • Weitere Informationen: http://bit.ly/k8s-ctrl-mgr
Scheduler • Entscheidet auf der Grundlage mehrerer Faktoren – Affinität, verfügbare Ressourcen, Bezeichnungen, QoS usw. –, wo Pods ausgeführt werden sollen • Mehr dazu: http://bit.ly/k8s-scheduler
etcd • Herzstück des Kubernetes-Clusters; bewahrt Schlüssel-Wert-Informationen zu allen Kubernetes-Objekten dauerhaft auf Mehr dazu: http://bit.ly/2COkMx9

Arbeitsknoten
Kubelet – Agent auf jedem Arbeitsknoten • Initiiert Pods (eine Gruppe aus einem oder mehreren Containern) mithilfe von PodSpec und stellt sicher, dass alle Pods laufen und fehlerfrei sind • Interagiert mit Containern – z. B. Docker • Mehr dazu: http://bit.ly/k8s-kubelet
Kube Proxy – Agent auf jedem Arbeitsknoten • Netzwerkproxy und Load Balancer für Kubernetes-Services • Mehr dazu: http://bit.ly/k8s-proxy

Hier die Kubernetes-Add-Ons, die für die meisten Anwendungen erforderlich sind:

kube-dns	kubectl
• In Kubernetes als Pod und als Service bereitgestellt • Jeder Service enthält einen DNS-Eintrag in Kubernetes • kube-dns löst die DNS aller Services im Cluster auf	• Die offizielle Befehlszeile für Kubernetes • „Behind the Scenes“ – Verwendet REST-basierte API-Aufrufe an den Kubernetes-API-Server

Wichtige Kubernetes-Konstrukte und -Objekte:

Namespaces	Pods
Virtuelle Segmentierung einzelner Cluster	Eine logische Gruppierung eines oder mehrerer Container, die von Kubernetes verwaltet werden
Knoten (Nodes)	**Replikatsatz (Replicaset)**
Infrastruktur-Fabric von Kubernetes (Host von Arbeits- und Masterkomponenten)	Fortlaufende Schleife, die sicherstellt, dass eine bestimmte Anzahl Pods ausgeführt wird
Rollen	**Zugänge (Ingresses)**
Rollenbasierte Zugriffssteuerung für Kubernetes-Cluster	Verwaltet externen HTTP-Datenverkehr zum gehosteten Service
Deployments	**Services**
Verwaltet ein ReplicaSet, Pod-Definitionen, Updates und weitere Konzepte	Eine logische Schicht, die Persistenz von IP, DNS usw. für dynamische Pods bereitstellt

Herausforderungen an das Monitoring in Kubernetes-Umgebungen

Gemäß einer Umfrage der CNCF zählen Komplexität, Monitoring und Sicherheit zu den größten Herausforderungen für Unternehmen, die Kubernetes einführen. Die Monitoringstrategien von gestern funktionieren in der Cloud-nativen Ära nicht mehr – hauptsächlich aus folgenden Gründen:

- Das Monitoring muss sich auf viel mehr Komponenten erstrecken
- Container sind vergänglich und dynamisch
- Kubernetes plant Pods automatisch, ausgehend von der besten Ressourcenauslastung. Dies steigert zwar die Effizienz, zugleich wird es aber unvorhersehbar, wo die Pods bereitgestellt und ausgeführt werden, sofern nicht mithilfe von Affinität oder Taints bestimmte Absichten ausgedrückt werden

Dadurch ergeben sich die folgenden Bedingungen für Kubernetes-Umgebungen:

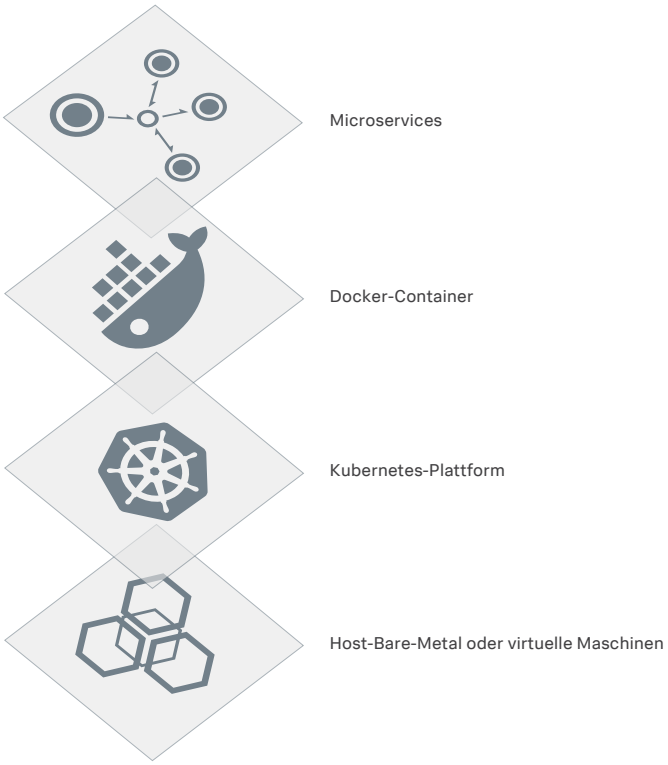
- Das Monitoring muss sich auf viel mehr Komponenten erstrecken
- Container sind vergänglich und ihre Platzierung in Pods lässt sich nicht vorhersehen
- Die Problembehandlung containerisierter Microservices ist komplexer

Mehr Komponenten, die Monitoring benötigen

In der monolithischen Welt gab es nur zwei Komponenten, die überwacht werden mussten – Anwendungen und die Hosts, auf denen die Anwendungen bereitgestellt waren.

In der Cloud-nativen Welt bestehen containerisierte und mithilfe von Kubernetes orchestrierte Anwendungen aus mehreren Komponenten, für die ein Monitoring erforderlich ist:

- Hosts
- Die eigentliche Kubernetes-Plattform
- Docker-Container
- Containerisierte Microservices



Vergänglichkeit von Containern und unvorhersehbare Platzierungen

Im Gegensatz zum herkömmlichen Modell langlebiger Hosts werden moderne, auf Microservices basierende Anwendungen normalerweise in Containern bereitgestellt, die dynamisch und vergänglich sind. Kubernetes stellt sicher, dass die gewünschte Anzahl von Anwendungsreplikaten ausgeführt wird. Kubernetes platziert die Pods auf den Knoten, die es für geeignet hält, es sei denn, es wird über Knotenaffinität oder Taints spezifisch angewiesen, dies nicht zu tun. Tatsächlich bildet die Fähigkeit von Kubernetes, Pods zu planen, das wichtigste Entwurfsziel für dieses selbstregulierende System.

Herkömmliche Monitoringansätze funktionieren in diesen hochdynamischen Umgebungen nicht, da sie üblicherweise langlebigen Hosts mithilfe von Hostnamen oder IP-Adressen folgen. In containerisierten Umgebungen müssen Monitoring-Tools die sofortige Erkennung von Services und die automatische Ermittlung von Lebenszyklusereignissen von Containern bieten. Gleichzeitig müssen sie die Metrikerfassung anpassen können, wenn Container innerhalb von Sekunden hochgefahren oder neu gestartet werden.

Performance-Monitoring von Microservices

Das genaue Bestimmen von Problemen in einer Microservices-Umgebung stellt sich schwieriger dar als in einer monolithischen Umgebung, da Anfragen sowohl verschiedene Schichten des Stacks als auch verschiedene Services durchqueren. Moderne Monitoring-Tools müssen diese miteinander verbundenen Schichten überwachen und gleichzeitig das Verhalten von Anwendung und Infrastruktur effizient korrelieren, um die Problembehandlung zu optimieren.

Wichtige Leistungsmetriken für das Monitoring

POD-Metriken

Es ist von grundlegender Wichtigkeit, alle Kubernetes-Objekte zu überwachen, um sicherzustellen, dass jeder Cluster fehlerfrei arbeitet und die Ressourcennutzung optimiert wird. Das Monitoring von Kubernetes-Pod-Metriken sowie von Bereitstellungen und Services hilft Ihnen festzustellen, ob Kubernetes in Ihrer Umgebung wie vorgesehen funktioniert.

PODS	Anzahl der gewünschten Pods
	Anzahl der verfügbaren Pods
	Pods nach Phase (fehlgeschlagen, ausstehend, aktuell ausgeführt)
	Gewünschte Pods pro Bereitstellung
	Gewünschte Pods pro Service
	Verfügbare Pods nach Bereitstellung
	Verfügbare Pods nach Service
	Gewünschte Pods nach ReplicaSet
	Ausgeführte Pods pro Knoten

Metriken zur Ressourcennutzung

Es ist wichtig, auch die Ressourcennutzung nachzuverfolgen, um sicherzustellen, dass Ihre Anwendungen und Kubernetes-Cluster fehlerfrei bleiben.

- Docker Socket bietet Container-Metriken und Metriken zur Ressourcennutzung auf Knotenebene, wie etwa CPU- und Arbeitsspeicherauslastung. Kubernetes bietet ebenso collectd-Metriken sowie von Kubernetes veröffentlichte Metriken.
- Das Korrelieren von CPU-, Arbeitsspeicher-, Datenträger-E/A- und Netzwerkleistungs-Metriken mit Kubernetes-Ereignissen hilft dabei, die Grundursache von Performance-Problemen schneller zu ermitteln.
- Mithilfe des Monitorings von Docker- und Kubernetes-Events, wie etwa Container- oder Pod-Lebenszyklus-Ereignissen, können Fehlkonfigurationen oder Ressourcenmängel leichter aufgedeckt werden.

Monitoring von Kubernetes mit Splunk

Kubernetes stellt detaillierte Informationen zu seinen Komponenten und der Nutzung von Anwendungsressourcen auf der Cluster-, Pod- und Serviceebene bereit, die leicht durch eine Vielzahl von Monitoring-Lösungen erfasst werden können. Die Aufgabe, diese Daten konkret verwertbar zu machen, bleibt den Endbenutzern überlassen.

Splunk ermöglicht es Ihnen, Kubernetes mit flexibler, offener Instrumentierung und vordefinierten Dashboards zu überwachen, welche sofortige Transparenz für die verschiedenen Ebenen ihrer Umgebung bieten. Durch die Kombination von Streaming-Analysen und verteiltem Tracing können Splunk-Benutzer über eine einfache Datenerfassung hinausgehen und stattdessen Benachrichtigungen und Performance-Analysen in Echtzeit nutzen, um Probleme innerhalb von Sekunden zu finden und zu beheben.

Metriken von einem Cloud-Anbieter erfassen

Für das grundlegende Monitoring eines verwalteten Kubernetes-Services wie AWS Elastic Container Service for Kubernetes (EKS), Google Container Engine (GKE) oder Azure Kubernetes Service bietet sich als einfachste Möglichkeit zum Erfassen von Metriken die Integration von Splunk in Services wie AWS CloudWatch, Google Stackdriver und Azure Monitor an.

Dies ist der unkomplizierteste Ansatz. Er ermöglicht die Erfassung von Kubernetes-Metriken mithilfe von Splunk, ohne dass ein Agent installiert oder Anwendungscode geändert werden müsste. Diese Services sind jedoch standardmäßig so konfiguriert, dass sie Metriken in relativ großen Abständen zusammenfassen und melden (AWS CloudWatch wird standardmäßig alle 5 Minuten aktualisiert) und bieten keinen Einblick in die spezifischen Dienste, die auf Ihren Kubernetes-Clustern bereitgestellt sind.

Vollautomatisches Kubernetes-Monitoring

Um einen besseren Einblick in die Services zu erhalten und ein detaillierteres Monitoring der Containermetriken zu erreichen, empfehlen wir, den Smart Agent durchgängig auf Ihren Kubernetes-Clustern zu installieren.

Als Open-Source-Agent zur Metrikerfassung, der auf collectd aufbaut, bietet der Smart Agent die automatische Ermittlung von Services und die Konfiguration für Monitoring-Inhalte. Ein Vorteil bei diesem Ansatz besteht in der Fähigkeit des Smart Agent, Metriken mit einer Auflösung von einer Sekunde an Splunk zu übermitteln, wodurch er sich besonders für die grundsätzlich vergänglichen Kubernetes-Pods eignet.

Der Smart Agent wird in Kubernetes als DaemonSet ausgeführt, um sicherzustellen, dass dieser auf jedem Knoten im Cluster installiert ist. Er erfasst Daten aus Kubernetes und verwendet cAdvisor, um die Ressourcen- und Leistungsmerkmale bei den ausgeführten Docker-Containern abzurufen. Dank Zero-Touch-Konfiguration mit automatischer Erkennung von Kubernetes-Komponenten und containerisierten Services wird sofort der gesamte Stack überwacht. Der Smart Agent wird in einem einfachen Schritt installiert:

```
helm install splunk --set splunkAccessToken=<
IHR_ZUGRIFFSTOKEN> --set
clusterName=<IHR_CLUSTERNAME> splunk/
splunk-agent
```

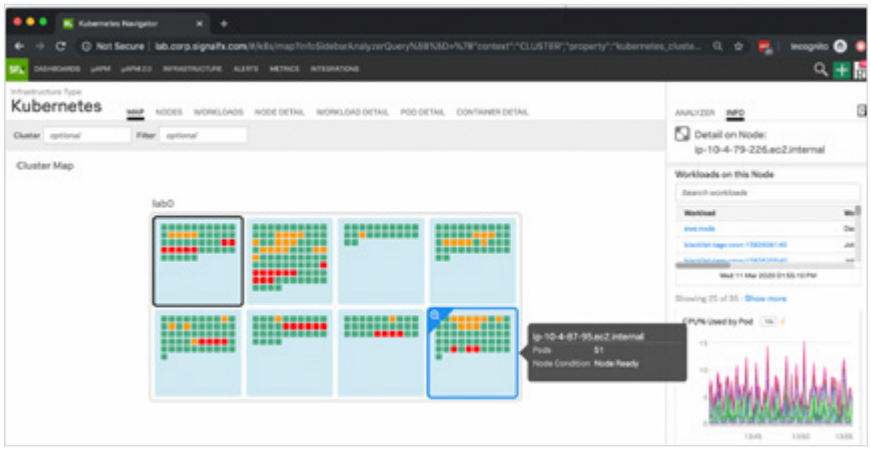
Der Smart Agent kann nicht nur ein Upstream-Monitoring für Kubernetes ausführen, er ist auch imstande, Metriken von verwalteten Kubernetes-Services wie AWS Elastic Container Service for Kubernetes (EKS) und Google Container Engine (GKE) sowie von Plattformen wie Openshift und Pivotal Kubernetes Service (PKS) zu erfassen.

Der Smart Agent stellt darüber hinaus Monitore bereit, die Daten aus anderen Metrikprotokollen wie Prometheus und StatsD erfassen können, sodass Teams mit vorhandenen Metrik-Pipelines in der Lage sind, die Vorteile von Splunk problemlos zu nutzen.

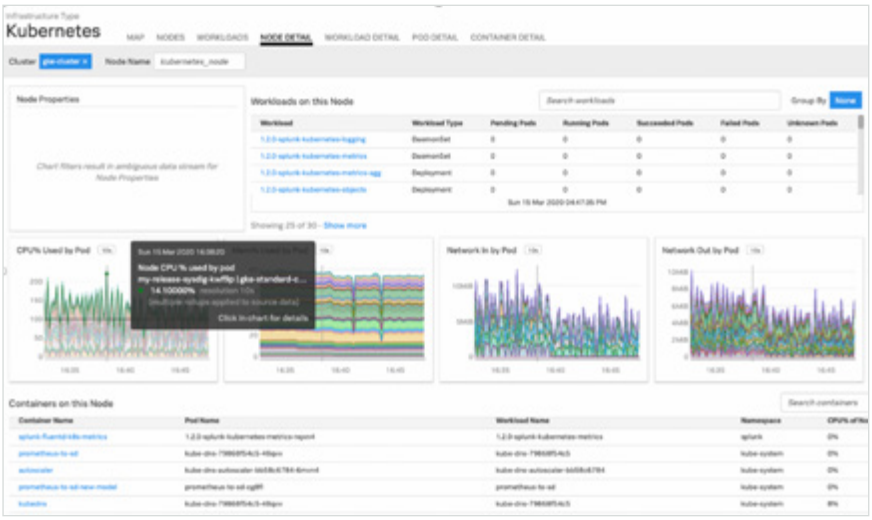
Navigieren in Ihrer Kubernetes-Umgebung

Vordefinierte Kubernetes-Dashboards

Aus der Vogelperspektive können Teams mithilfe von Kubernetes Navigator die Leistung der gesamten Kubernetes-Umgebung dank einer intuitiven und hierarchischen Steuerung schnell verstehen. In Sekundenschnelle können beliebige Kubernetes-Entitäten wie etwa Knoten, Pods und Container-Ebenen ausgewählt, gesucht oder gefiltert und Drilldowns für die Detailanalyse ausgeführt werden. Beziehungen zwischen dynamischen Kubernetes-Komponenten lassen sich besser verstehen und Performance-Probleme, die auf störende Nachbar-Komponenten zurückzuführen sind, schneller beheben.



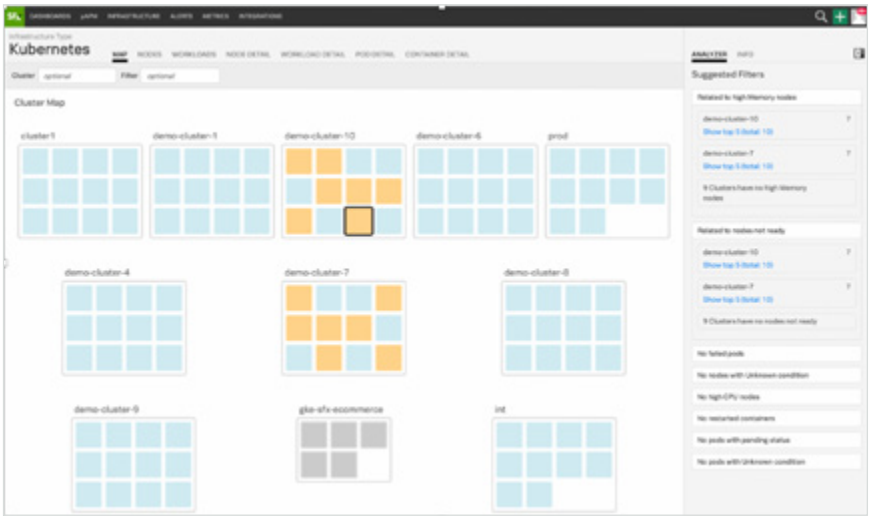
Beim Drilldown zu einem Einzelknoten werden Systemmetriken für diesen bestimmten Knoten sowie Dashboards für die spezifischen Services angezeigt, die auf diesem Knoten ausgeführt werden.



Sie können außerdem alle Pods anzeigen, die auf Ihrem Kubernetes-Cluster ausgeführt werden, und die Aktivitäten für einen bestimmten Pod oder übergreifend für alle Pods in Ihrem Cluster nachverfolgen. Von hier aus kann ein Drilldown zu den einzelnen Docker-Containern in jedem Pod ausgeführt werden.

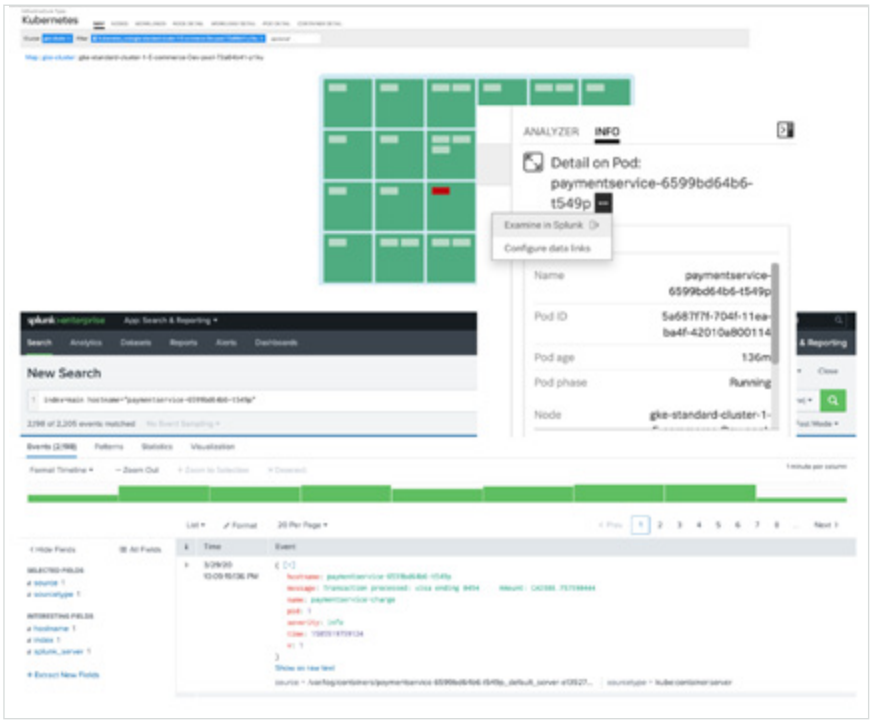
Kubernetes Analyzer

KI-gestützte Analysen liefern automatisch Erkenntnisse und Empfehlungen, mit denen sich präzise und in Echtzeit beantworten lässt, was Anomalien im gesamten Kubernetes-Cluster verursacht – in Knoten, Containern und Workloads. Hochentwickelte Algorithmen (einschließlich Algorithmen für historische Performance-Basiswerte und plötzliche Änderungen) erkennen Probleme auf Systemebene, wie z. B. einen plötzlichen Anstieg der Goroutinen oder Container-Neustarts, und setzen innerhalb von Sekunden Benachrichtigungen ab.



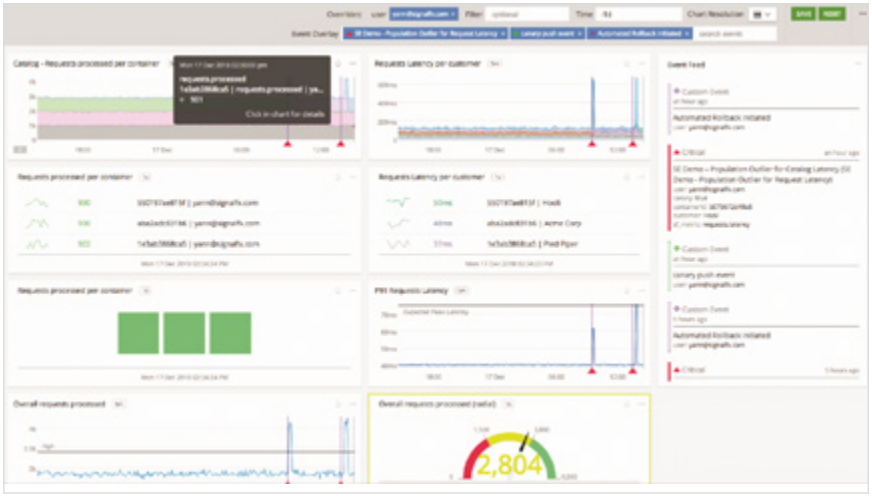
Logs im Kontext

Sie können nahtlos Pivots zu Logs erstellen und für Transparenz bis ins kleinste Detail bei Anwendungs-, Kubernetes- und Container-Logs sorgen, um die Performance ohne Kontextwechsel über den gesamten Stack hinweg zu korrelieren. Mit Transparenz bei Lebenszyklus-Events von Kubernetes- und API-Server-Audit-Logs können Sie Ihren Sicherheits- und Compliance-Status besser verstehen und aufrechterhalten.



Benutzerdefinierte Service-Dashboards erstellen

Mithilfe von Splunk können Sie Containermetriken parallel zu anderen Leistungsindikatoren einsehen und erhalten so Transparenz für jede Schicht Ihrer Umgebung. Beispielsweise kann ein Service Owner, der Canary-Bereitsstellungen überwachen möchte, das folgende Dashboard erstellen, bei dem Container-Metriken neben Diagrammen zur Messung der Anforderungslatenz und einem Ereignisfeed angezeigt werden, der Code-Pushes, Benachrichtigungen und alle eventuell ergriffenen Abhilfemaßnahmen verfolgt.



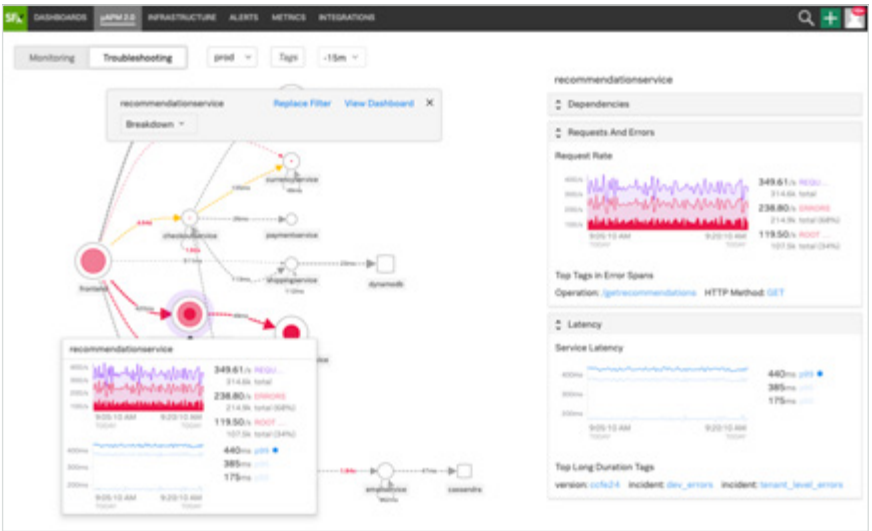
Probleme in Echtzeit erkennen

Mit Splunk bedeutet das Einrichten von Benachrichtigungen keinen Kompromiss mehr zwischen Genauigkeit und Aktualität eingehen zu müssen. Erstellen Sie bei Problemen proaktiv Warnmeldungen, indem Sie hochauflösende Metriken aus Ihren Kubernetes-Clustern mit einer Bibliothek statistischer Funktionen und Algorithmen in Splunk kombinieren. Führen Sie eine Vorschau von Warnmeldungen im Vergleich mit historischen Verlaufsdaten durch, um sie vor der Bereitsstellung zu optimieren, und wenden Sie Data Science auf die Benachrichtigungen an, um False Positives und eine Überflutung an Meldungen zu vermeiden, die für hochgradig vergängliche Umgebungen typisch sind.

Verteiltes Tracing zur gezielten Problembehandlung nutzen

Splunk APM™ gibt Benutzern Funktionen zum verteilten Tracing an die Hand, mit denen sich Ereignisse, die die Performance beeinträchtigen, im Detail untersuchen lassen. Unabhängig davon, wo in Ihrer Kubernetes-Umgebung ein Problem auftritt, können Sie von Echtzeit-Benachrichtigungen direkt zu Anwendungs-Traces navigieren und Performance-Trends zwischen Infrastruktur, Kubernetes und Ihren Microservices korrelieren.
Splunk APM baut auf einer **NoSample™** Full-Fidelity-Architektur auf, die alle Traces einbezieht und analysiert, sodass keine Ausreißer oder Anomalien unentdeckt bleiben.

Der **Outlier Analyzer™** bietet zielgerichtetes Troubleshooting, indem er die besonders häufig auftretenden Muster in Traces von Ausreißern aufdeckt und es Ihnen so ermöglicht, das Problem schnell auf anomal funktionierende Kubernetes-Knoten, Cluster, Cloud-Regionen oder anwendungsspezifische Bezeichnungen oder Tags einzugrenzen.



Die DevOps-Geschwindigkeit steigern

Sobald Daten aus Kubernetes an Splunk fließen, können DevOps-Teams leichter bewährte Monitoring-Best-Practices und Lösungen für häufige Probleme in der gesamten Organisation nutzen. Splunk bietet sofort einsatzbereite Transparenz für Hosts, Container, Pods und Kubernetes-Objekte mit integrierten, konfigurationsfreien Dashboards. Splunk erlaubt es Teams aber trotzdem, Dashboards, Warnungen und Benachrichtigungen an ihre spezifischen Bedürfnisse anzupassen.

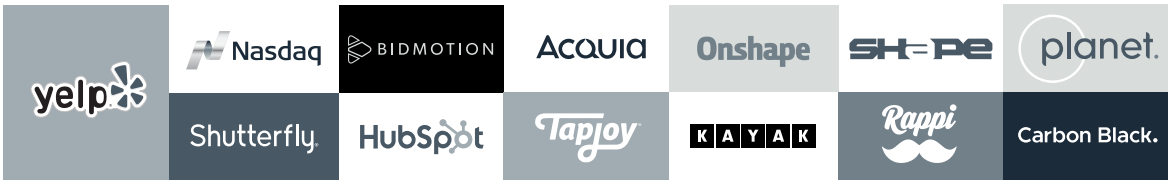
Zentrale Teams können Monitoring in Form von Code bereitstellen, indem sie die Splunk API nutzen, um programmatisch Monitoring-Inhalte zu erstellen und Analysen zu definieren. Sie können außerdem das Service Bureau einsetzen, um Nutzungs-, Zugriffs- und Bearbeitungsberechtigungen für Benutzer und Teams zu steuern.

```
curl \  --request post \ --header "x-sf-token: your_access_
token" \ --header "content-type: application/json" \ --data \'{
"name": "cpu load", "programtext": "data('cpu.load').publish()"
}' \ https://api.splunk.com/v2/chart
```

Führende Unternehmen vertrauen auf Splunk

Unternehmen aus jeder Branche weltweit nutzen Splunk, um ihren Weg hin zu Cloud-nativen Anwendungen zu beschleunigen, und implementieren Kubernetes in ihren Organisationen. Die folgende Liste nennt nur ein paar unserer Kunden.

Hier finden Sie weitere Informationen und Erfahrungsberichte von Kunden aus **Deutschland** und dem **Rest der Welt**



Steigen Sie noch heute in Splunk Infrastructure Monitoring ein

Möchten Sie erfahren, wie Splunk die Einführung von Kubernetes in Ihrer Organisation beschleunigen kann?

Unsere Customer Success Manager und Solutions Engineers arbeiten eng mit Ihren Teams zusammen, um Ihre Einführung von Kubernetes voranzutreiben und Kubernetes in Echtzeit zu überwachen.

Registrieren Sie sich für eine kostenlose Testversionen

Erste Schritte

Stellen Sie Smart Agent als DaemonSet bereit

Weitere Informationen

Beginnen Sie mit dem Monitoring von Kubernetes-Umgebungen

Weitere Informationen

Wenn Sie mehr über Kubernetes Navigator für das Echtzeit-Monitoring und die Problembehandlung von Kubernetes-Umgebungen jeder Größe erfahren möchten, laden Sie das kostenlose **Kubernetes Navigator-Datenblatt** herunter.

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

20-signalfx-monitoring-microservices-kubernetes-WP-111-A4