new relic

Observability bei New Relic

Unser Engineering-Playbook für Hyperscale und Zuverlässigkeit

Whitepaper / Juli 2025

Überblick

Einzelhändler, Banken, große Streaminganbieter und unzählige andere Unternehmen vertrauen auf New Relic, um zuverlässige Erlebnisse in großem Maßstab zu liefern. In kritischen Momenten wie bei einer neuen Produkteinführung oder einem großen Streaming-Event nutzen sie New Relic Dashboards zur Beobachtung von Trafficspitzen und verlassen sich auf Alerts zur Meldung von abnormalem Verhalten oder Fehlern. Wir wissen, wie wichtig Observability in komplexen Umgebungen ist, denn als Engineers erleben wir diese Dinge selbst tagtäglich.

Anhand konkreter Beispiele zeigen wir in diesem Whitepaper, wie auch unsere Engineering-Teams unser eigenes Produkt täglich einsetzen, um die unterschiedlichsten kritischen Geschäftsziele zu erreichen, von erheblichen Einsparungen bei Cloudkosten und einer Erhöhung der Entwicklungsproduktivität bis hin zur kontinuierlichen Verbesserung des Kundenerlebnisses und Gewährleistung einer hohen Verfügbarkeit.

Die Engineering-Organisation bei New Relic setzt zu hundert Prozent auf unsere eigene Observability-Plattform, um eine beispiellose Uptime und geringstmögliche Latenz zu erzielen. Dieser Ansatz, den wir „New Relic on New Relic“ nennen, ist für das groß angelegte Management der Plattform wesentlich: Täglich werden über eine Billion Datenpunkte gesammelt und mehr als 20 Millionen Abfragen ausgeführt. Bei gleichzeitiger deutlicher Senkung der Betriebskosten.

Wir entwickeln die New Relic Plattform also nicht nur, wir nutzen sie auch selbst für unsere komplexen Observability-Anforderungen, und diese Erkenntnisse fließen direkt in die Produktfeatures für unsere Kundschaft ein. Dieser „Inside-Out“-Ansatz stellt sicher, dass New Relic die Anforderungen moderner verteilter Systeme erfüllt.

Wir befassen uns hier mit verschiedenen Use Cases und erläutern im Detail, wie unsere internen Teams wie SRE, Frontend, Backend, Plattform-Engineering, Netzwerke usw. durch umfassende Observability operative Exzellenz und Innovation erreichen.

Durch dieses Whitepaper möchten wir Ihnen das Engagement von New Relic für Self-Observability näherbringen, durch das wir nicht nur in den Umgebungen unserer Kundschaft für zuverlässige Performance sorgen, sondern auch die dazu genutzte Plattform kontinuierlich verbessern und validieren. Unsere Motivation sind seit jeher Innovation, technische Exzellenz und die unerschütterliche Zuversicht, die (gestützt auf unsere eigenen Erfahrungen) der Einsatz von New Relic mit sich bringt.

Wir sehen uns drei wichtige Säulen unserer Zuverlässigkeitsstrategie an.

Messen, worauf es ankommt

Analyse und Nachverfolgung der Metriken, die für zuverlässige Performance und effiziente Fehlerbehebung und letztendlich zur Erreichung der Geschäftsziele wesentlich sind.

Selbstreparierende Systeme

Weniger Engineering-Aufwand und größere Zuverlässigkeit durch Automatisierung, um möglichen Problemen zuvorzukommen.

Probleme schnell beheben

Die notwendigen Tools und Einblicke für unsere Teams, damit Probleme schnell diagnostiziert und behoben werden.

Messen, worauf es ankommt

Um ein zuverlässiges System zu betreiben, muss man das messen, worauf es ankommt. Und zwar worauf es für das Unternehmen wie auch für die Fehlerbehebung ankommt. New Relic misst dies mit New Relic, genauer gesagt mit unseren Agents und unseren Produkten für Service-Level Objectives (SLOs) und Alerting.

APM Agents bei New Relic

Für ihre Services nutzen Engineering-Teams gern den APM Agent (Application Performance Monitoring), und die gesamte Hardware wird in der Regel mit dem New Relic Infrastructure Agent überwacht. Dadurch entsteht ein einheitlicher Satz von Health-Metriken, so dass Engineers auch beim Wechsel zwischen Teams oder Services stets wissen, worum es geht.

Die APM Agents von New Relic sind präzise darauf abgestimmt, die relevanten Metriken für die Erkennung und Diagnose von Serviceproblemen zu liefern. Zu den wichtigsten Metriken gehören HTTP-Antwortzeit, Datenbankaufrufzeiten und externe HTTP-Aufrufzeiten.

Zusätzlich fügen nicht nur dedizierte Agent-Teams unseren Agents häufig neue Instrumentierung hinzu: Auch andere Engineering-Teams, die die Agents zum Monitoring ihrer eigenen Infrastruktur, Datenbanken und Streamingdienste einsetzen, leisten ihren Instrumentierungsbeitrag. Ein gutes Beispiel ist die aktuelle Kafka-Instrumentierung im Java-Agent. Diese Instrumentierung wurde ursprünglich von einem New Relic Team erstellt, das viele essenzielle Kafka-Streamingdienste betreibt, und zwar zunächst als Java-Agent-Erweiterung. Nachdem diese Instrumentierung bei vielen internen Teams Anklang fand, wurde sie schließlich in das APM-Produkt integriert. Die Kafka-UI, die diese Metriken anzeigt, wurde ebenfalls gemeinsam von einem APM-Produktteam zusammen mit unseren Kafka- und Streamingdienst-Teams erstellt.

Zusätzlich fügen nicht nur dedizierte Agent-Teams unseren Agents häufig neue Instrumentierung hinzu: Auch andere Engineering-Teams [...] leisten ihren Instrumentierungsbeitrag.

Effektive Instrumentierung: Dashboards und Nerdpacks vermeiden Kontextwechsel

Die Teams bei New Relic werden dazu angehalten, schon während der Entwicklung Observability-Daten zu berücksichtigen. Zusätzlich zur Standardinstrumentierung in unseren Agents haben Engineering-Teams die Möglichkeit, weitere Custom-Instrumentierung zu erstellen. Einige Teams senden ihre Custom-Instrumentierung über APM Agents an New Relic. Andere haben Bibliotheken erstellt, damit die Instrumentierung direkt an unsere öffentlichen Endpunkte für Metriken, Events, Logs und Traces gesendet wird. Beispiele für Custom-Instrumentierung sind ein New Relic Event für jede Abfrage der New Relic Datenbank (NRDB) sowie ein Event für jede Erstverbindung des APM Agent mit New Relic. Diese Custom-Events werden dann in Dashboards oder Custom-Nerdpacks (benutzerdefinierte Anwendungen) angezeigt, in denen Textanweisungen mit Live-Abfrageergebnissen und Visualisierungen kombiniert werden.

Zum Beispiel können Probleme mit Kafka-Pipeline-Stalls über Ansichten auf einem Custom-Nerdlet diagnostiziert werden, das zudem automatisch den erforderlichen Befehl generiert, um die Daten-Retention zu verlängern und einen mehrstufigen manuellen Prozess in einen einzigen Copy-Paste-Vorgang zu verwandeln. Dadurch werden mühselige Kontextwechsel deutlich reduziert und die Problembehebung wird beschleunigt.

Probleme mit Kafka-Pipeline-Stalls können über Ansichten auf einem Custom-Nerdlet diagnostiziert werden, das zudem automatisch den erforderlichen Befehl generiert, um die Daten-Retention zu verlängern und einen mehrstufigen manuellen Prozess in einen einzigen Copy-Paste-Vorgang zu verwandeln.

Service-Level Objectives: Erreichen von Geschäftszielen mit SLOs

SLOs sind wichtig, da sie die Customer Experience über einen festgelegten Zeitraum definieren und messen. Um Arbeiten an der Zuverlässigkeit und neue Funktionen in Einklang zu bringen, sind sie unerlässlich. Bei New Relic wird heute von allen Teams erwartet, dass sie einen internen Satz an SLOs pflegen. Bevor dies im gesamten Unternehmen galt, war uns aufgefallen, dass unsere Telemetriedaten zwar sehr umfangreich waren, aber auf das Troubleshooting zugeschnitten waren – nicht auf die Bewertung der Customer Experience. Wir entwickelten also ein „Bar-Raiser“-Programm für SLOs, in dem Teams kundenorientierte SLOs mit Werten erstellen konnten, die die damalige Realität widerspiegelten.

Anhand dieser Messungen konnten wir die Betriebskosten, die durch Einhaltung der aktuellen SLOs anfielen, sowie die zum Erreichen höherer SLOs erforderlichen Arbeiten ermitteln und mit den Verantwortlichen im Unternehmen teilen. Teams, denen SLOs einen echten Nutzen gebracht haben, prüfen ihre eigenen SLOs täglich und ergreifen bei Bedarf Gegenmaßnahmen. Diese Teams konnten nicht nur ihre Customer Experience verbessern, sondern erhielten auch deutlich weniger Alarmrufe.

Alerting und Terraform: So automatisieren Sie proaktive Einblicke

Bei New Relic verwenden unsere Teams New Relic Alerting, um über Systemanomalien informiert zu werden. So können unsere Engineers schnell Maßnahmen ergreifen, um eine mögliche Systemverschlechterung im Rahmen zu halten. Die meisten Teams verwenden Terraform in der Versionskontrolle, um ihre Alerts zu erstellen und zu verwalten. Überwiegend werden zudem Facet-Alerts eingesetzt, um sicherzustellen, dass Alerts für jede neue Zelle oder Umgebung erstellt werden. Ein Beispiel für einen Facet-Alert für den Hostnamen sehen Sie hier:

Teams verwenden New Relic Alerting, um über Systemanomalien informiert zu werden. So können unsere Engineers schnell Maßnahmen ergreifen, um eine mögliche Systemverschlechterung im Rahmen zu halten.

```
SELECT latest(etcdServerProposalsFailedRate) FROM
K8sEtcdSample WHERE clusterName = 'my-cluster' FACET hostname
```

Um sicherzustellen, dass Teams über die richtigen Alerts verfügen, umfassen unsere New Relic Engineering-Standards eine Reihe empfohlener Alerts für Teams. Dazu gehören Alerts zu OOM-Kills (Out of Memory), Warten auf Pods, Kafka Lag, Fehlerquoten und mehr. Auf ihren Dashboards können Führungskräfte die Alerts ihres Teams aufrufen, da jeder Alert in NRDB aufgezeichnet wird, und die Trends von Woche zu Woche nachverfolgen.

Selbstreparierende Systeme

Alerting ist wichtig, damit schnell auf mögliche Serviceunterbrechungen reagiert werden kann. Die Benachrichtigung der Mitarbeitenden, die dann manuell entsprechende Maßnahmen ergreifen können, nimmt allerdings noch immer zu viel Zeit in Anspruch. Aus diesem Grund automatisiert New Relic so viel wie möglich, mit einem Schwerpunkt auf selbstreparierende Systeme.

Weniger Engineering-Aufwand dank Autoskalierung

New Relic hat erheblich in Algorithmen zur Autoskalierung investiert, mit denen sich Services schnell hoch- oder herunterskalieren lassen. Dazu nutzen diese Algorithmen Metriken wie CPU und Speicher. Störungen und Alarmierungen wurden dadurch deutlich reduziert – zuvor kam es durchaus vor, dass in unserem Loggingteam zwei- bis viermal pro Woche der Pager ging, weil jemand Hilfe bei der Skalierung eines Services brauchte. Seit der Implementierung der automatischen Skalierung erhält das Team deutlich weniger Alarmrufe.

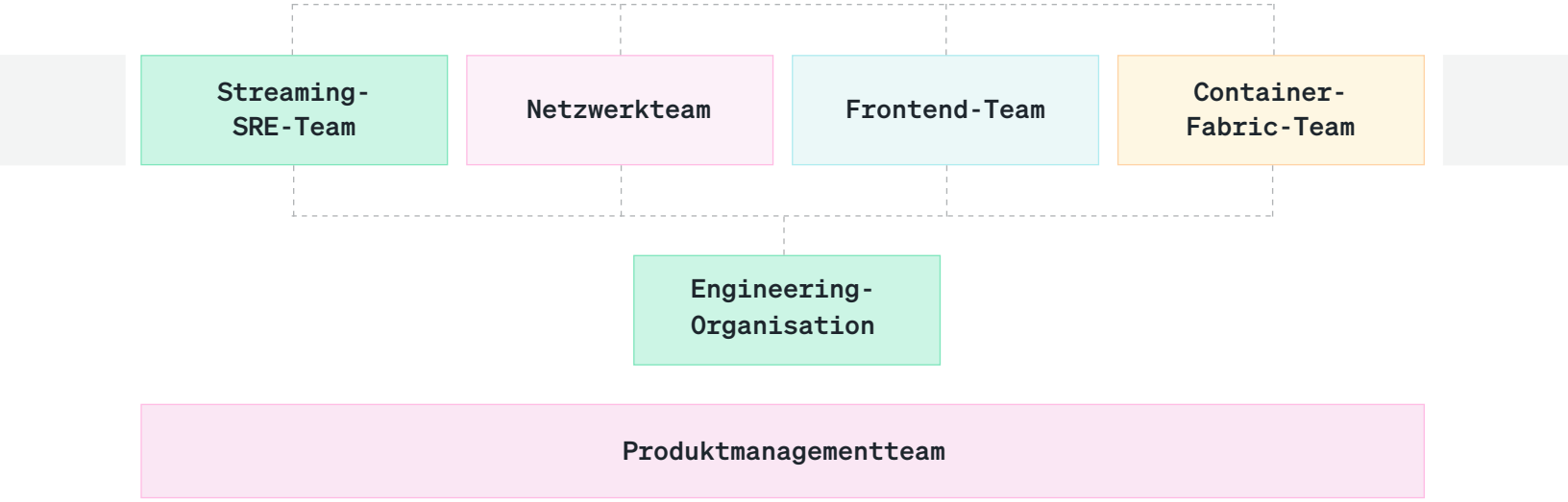
Größere Ausfallsicherheit dank Auto-Rollback

Obgleich New Relic Services vor der Produktion eine Reihe von Tests durchlaufen, schaffen es ab und zu doch ein paar Bugs in die Produktion. In diesen Fällen führt New Relic ein automatisches Service-Rollback durch. Wird eine Änderung über die Continuous-Deployment-Pipeline von New Relic durchgeführt, startet ein Workflow, der die Integrität der Entity prüft. Verschlechtert sich die Service-Health, löst der Workflow die Continuous-Deployment-Pipeline aus, um die betroffenen Instanzen zurückzusetzen.

Probleme schnell beheben

Zwar unterstützt New Relic die Selbstreparatur durch Automatisierung, aber ab und zu wird für Fehlererkennung und -behebung weiterhin menschliches Fachwissen benötigt. In diesem Szenario dient New Relic selbst als Hauptplattform für unsere Incident-Responder. Unsere Teams nutzen Change Tracking, um potenziell problematische Bereitstellungen zu lokalisieren. Produktseiten, kuratierte Dashboards, spezialisierte Erlebnisse und Ad-hoc-Abfragen werden dann mit anderen geteilt, um die Grundursache zu ermitteln und die jeweils erforderlichen Gegenmaßnahmen zu unterstützen.

Folgende Beispiele aus unseren internen Teams zeigen, wie uns New Relic hilft, Probleme entweder zu verhindern oder rasch zu beheben. Und so können wir eine kontinuierlich hohe Zuverlässigkeit gewährleisten – sowohl für unsere Kundschaft als auch für uns selbst.



OBSERVABILITY-INNOVATION MIT KAFKA

Zuverlässigkeit im Fokus für das Streaming-SRE-Team

New Relic betreibt eine der weltweit führenden Kafka-Implementierungen sowie Hunderte von Services, die Kafka als Producer und/oder Consumer nutzen. Schon früh legte das New Relic Streaming-SRE-Team den Grundstein für den Aufbau einer zuverlässigen Kafka-Umgebung im Dienste unserer Kundschaft. Zum Beispiel kann Kafka Lag direkt mit Verzögerungen oder Unterbrechungen bei der Telemetrieerfassung einhergehen. Dies wiederum kann dazu führen, dass Kunden-Alerts verpasst oder verzögert werden. Die Zuverlässigkeit von Kafka ist also nach wie vor von entscheidender Bedeutung.

Das Team ist speziell für Kafka-Operationen zuständig, und so konzentrierte man sich auf die Erstellung eines Custom-Nerdpacks für Kafka-Observability. Dieses speziell auf New Relic zugeschnittene Nerdpack mit zahlreichen Custom-Metriken wurde zu einem unverzichtbaren Tool für über 50 interne Teams, die Kafka-Services nutzen, und die operativen Erkenntnisse erwiesen sich als so wertvoll, dass sie direkt zur Entwicklung kundenorientierter Kafka- und Observability-Features beitrugen.

Dieses speziell auf New Relic zugeschnittene Nerdpack mit zahlreichen Custom-Metriken wurde zu einem unverzichtbaren Tool für über 50 interne Teams, die Kafka-Services nutzen.

Die Hauptmotivation für den Aufbau einer derart umfangreichen Kafka-Observability war, dass man die blinden Flecken abschaffen wollte, die bei Incidents oft ein Problem darstellen. Ohne granulare Daten war es nicht leicht, die zugrunde liegenden Fehlerursachen zu ermitteln und wiederkehrende Probleme schnell zu finden. Das Ziel war „stapelweise Observability“, um das Verhalten von Kafka vor, während und nach Incidents wirklich vollständig zu verstehen.

Einblicke

Das maßgeschneiderte Kafka-Nerdpack liefert tiefe Einblicke in eine Vielzahl von Metriken. Diese Einblicke nutzt das Streaming-SRE-Team für folgende Zwecke:

1. **Auslösen von Alerts zu Kafka Lag, damit die Datenerfassungsintegrität aufrechterhalten wird:** Der wichtigste Zweck, für den das Team New Relic einsetzt, ist die Integrität der Telemetrieerfassung, und diese wird mithilfe von Alerts zu Kafka-Metriken wie Kafka Lag gewährleistet. Kafka Lag korreliert direkt mit Verzögerungen oder Unterbrechungen bei der Telemetrieerfassung, was dazu führen kann, dass kritische Kunden-Alerts verpasst oder verzögert werden. Dies stellt ein erhebliches Geschäftsrisiko dar, da die Kundschaft auf zeitnahe Alerts angewiesen ist. Umfassende Alerts zu Kafka Lag ermöglichen die Erfassungsskalierung und Performance-Optimierung.
2. **Maximieren der Reaktionsfähigkeit:** New Relic versetzt das Team in die Lage, rasch auf Kafka-Verarbeitungsprobleme zu reagieren, sodass sie schnell behoben und eventuelle Auswirkungen auf Kundenseite minimiert werden.
3. **Verständnis des Kafka-Clientverhaltens:** Dazu zählt die Erkennung von Fehlkonfigurationen, überlasteten Puffern und ins Stocken geratenen Clients.
4. **Überwachen der serverseitigen Integrität:** Monitoring von Brokerleistung und Ressourcennutzung.
5. **Beobachten von Anfragenmustern:** Analyse von Änderungen in Client-Anfragenmustern, um potenziellen Problemen zuvorzukommen.

Die Implementierung von Kafka-Observability hatte tiefgreifende Auswirkungen auf die betriebliche Effizienz und Zuverlässigkeit von New Relic.

Deutlich schnellere Fehlerbehebung

Da das Streaming-SRE-Team jetzt umfassende Observability-Daten an der Hand hat, kann es Kafka-Client-Incidents innerhalb von Minuten diagnostizieren und den gesamten Zeitaufwand bis zur Behebung ebenfalls oft in Minuten und Sekunden zählen. Ohne solch detaillierte Einblicke dauerte das Ganze früher manchmal eine Stunde oder länger.

Nerdpacks als dynamische Runbooks

Eine wichtige Innovation, für die sich das SRE-Team einsetzt, ist die Verwendung von New Relic Nerdpacks als dynamische Runbooks. In diesen benutzerdefinierten Anwendungen werden Textanweisungen mit Live-Abfrageergebnissen und Visualisierungen kombiniert. Zum Beispiel können Probleme mit Kafka-Pipeline-Stalls über Ansichten auf dem Nerdlet diagnostiziert werden, das zudem automatisch den erforderlichen Befehl generiert, um die Daten-Retention zu verlängern und einen mehrstufigen manuellen Prozess in einen einzigen Copy-Paste-Vorgang zu verwandeln. Dadurch werden mühselige Kontextwechsel deutlich reduziert und die Problembehebung wird beschleunigt.

Einblicke für Führungskräfte

C-Suite- und andere Führungskräfte bei New Relic nutzen das Kafka-Observability-Nerdlet, um den Gesamtstatus von Verzögerungen über ganze Kafka-Cluster oder Umgebungen hinweg schnell zu bewerten und sich einen Überblick über die Erfassungsleistung und Skalierbarkeit zu verschaffen.

Performance- und Kostenoptimierung durch intelligente Autoskalierung

Das Streaming-SRE-Team hat durchdachte Autoskalierungstools entwickelt, die sowohl New Relic Telemetrie als auch benutzerdefinierte Metriken verwenden. So zum Beispiel New Relic CPU-Metriken, um Kubernetes-Ressourcen je nach Traffic dynamisch hoch- oder herunterzuskalieren. So kann das Team Trafficspitzen bei der Erfassung effektiv managen und Verzögerungen durch Hochskalieren aus der Welt räumen. Bei geringem Traffic wird dann wieder herunterskaliert. Die dynamische Autoskalierung verhindert eine Überbereitstellung von Ressourcen und gewährleistet Kosteneffizienz – aber ohne die Kapazität zur Bewältigung schwankender Workloads aufs Spiel zu setzen.

ÜBERRAGENDE NETZWERKZUVERLÄSSIGKEIT

Netzwerkteam erhöht Zuverlässigkeit mit New Relic

Das Netzwerkteam von New Relic überwacht die eigene globale Netzwerkumgebung. Diese umfasst Hunderte von Zellen mit Kubernetes-Clustern, die wiederum mit mehreren Cloudumgebungen verknüpft sind. Für umfassende Visibility entwickelten SREs und Netzwerk-Engineers Code mithilfe von New Relic Bibliotheken, die in jedem Cluster bereitgestellt werden, um wichtige Netzwerktelemetrie zu sammeln.

Die Hauptmotivation für den Aufbau dieser umfassenden Netzwerk-Observability bestand darin, der Kundschaft einen besseren Überblick über das Netzwerk zu vermitteln und Vertrauen aufzubauen. Letztendlich sollen andere Teams in die Lage versetzt werden, sich selbst zu helfen und beim Troubleshooting als Erstes Netzwerkprobleme auszuschließen.

Einblicke
Die benutzerdefinierten Netzwerk-Dashboards bieten tiefe Einblicke in zahlreiche Metriken, z. B.:

1. **Netzwerkleistung:** Monitoring von Bandbreite, Paketverlust, Jitter, Latenz und Pfadnutzung.
2. **Infrastruktur-Health:** Verwenden des Infrastructure Agent mit Amazon- und Azure-Konnektoren, um Informationen von diesen Plattformen abzurufen und in New Relic aufzunehmen.
3. **Konnektivitätsvalidierung:** Verwenden eines benutzerdefinierten Skripts, mit dem die Konnektivität zwischen zwei Orten durch Pingen geprüft wird.
4. **Kostenoptimierung:** Monitoring eines NAT-Dienstes (Egress Network Address Translation), um das Netzwerk eines Cloudanbieters zu einem deutlich niedrigeren Preis zu verlassen, und Überwachung auf unerwartete Kostenspitzen.

Die Implementierung dieser Netzwerk-Observability hatte tiefgreifende Auswirkungen auf die betriebliche Effizienz und Zuverlässigkeit von New Relic.

Deutlich schnellere Fehlerbehebung

Dank der Implementierung erhalten die Netzwerkteams jetzt deutlich weniger Alarmrufe. Durch Netzwerk-Observability wurde beispielsweise ein Routing-Problem festgestellt, bei dem der Traffic aufgrund einer fehlenden statischen Route per Failover auf eine unterdimensionierte Backup-Lösung umgeschaltet wurde. Die New Relic Netzwerkteams waren daher in der Lage, das Problem schnell zu beheben und später ein Active/Active-Setup für die Routen der Cloudanbieter zu implementieren, um den Traffic auszugleichen und eine Überlastung zu verhindern.

Proaktive Erkennung von Fehlkonfigurationen

Durch die Erkennung von Problemen wie fehlenden statischen Routen optimiert New Relic die Ressourcennutzung und erhöht die Systemzuverlässigkeit erheblich. Und das spart Geld.

Dynamische Runbooks

Letztendlich sollen andere Teams in die Lage versetzt werden, sich selbst zu helfen und beim Troubleshooting als Erstes Netzwerkprobleme auszuschließen.

Einblicke für Führungskräfte

Das Team verwendet New Relic auch zur Kostenoptimierung, und zwar durch Monitoring eines Egress-NAT-Dienstes. Zudem achtet das Team auf unerwartete Kostenspitzen und hilft anderen Teams, Probleme zu identifizieren und zu beheben, die die Datentraffic-Kosten unnötig erhöhen.

EXZELLENZ IM FRONTEND

New Relic Frontend-Teams entwickeln branchenführende UI-Plattformen

Für die New Relic Frontend-Organisation schafft die Monitoring-Umgebung in der Produktion eine kontinuierliche Feedbackschleife und liefert sofortige Einblicke in mögliche Probleme. Dieses Self-Monitoring erstreckt sich über alle kritischen Einsatzbereiche von New Relic. Dazu gehören sogar komplexe Micro-Frontend-Architekturen, für die das Team detaillierte Custom-Dashboards und Alerts entwickelt hat.

Features

Auf folgende New Relic Features verlassen sich die Frontend-Teams vor allem.

Synthetics: Proaktives Monitoring kritischer Features und Benutzerabläufe.

Dashboards: Benutzerdefinierte Dashboards bieten eine ganzheitliche Sicht auf System-Health, Performance-Trends und kritische Alerts für verschiedene Komponenten und Micro-Frontends.

New Relic Query Language: Engineers nutzen den intuitiven Query Builder und die New Relic Query Language intensiv für Ad-hoc-Datenexploration, Hypothesentests und rasche Incident-Untersuchungen.

Service-Level und Alerts: Proaktive Alerts anhand zuvor festgelegter SLOs sorgen dafür, dass Sie sofort über Serviceverschlechterungen oder potenzielle Ausfälle benachrichtigt werden, oft sogar bevor Benutzer:innen etwas mitbekommen.

Metriken

Folgende Schlüsselmetriken werden kontinuierlich beobachtet:

1. **Ladezeit:** Performance-Metrik für das Frontend-Erlebnis.
2. **Verfügbarkeit:** Stellt sicher, dass Services zugänglich und betriebsbereit sind.
3. **Latenz:** Nachverfolgung der Antwortzeiten, um Bottlenecks zu identifizieren und ein angenehmes Nutzungserlebnis zu gewährleisten.
4. **Throughput:** Monitoring von Datenvolumen und Verarbeitungsraten zur Bewertung der Systemkapazität.
5. **Fehlerquoten:** Identifizieren und Quantifizieren von Fehlern, insbesondere von JavaScript-Fehlern für das Frontend, um Bereiche mit sofortigem Handlungsbedarf zu ermitteln.

Für die Frontend-Engineering-Teams bringt „New Relic on New Relic“ erhebliche strategische Vorteile.

Proaktive Problembehebung

Da Probleme bereits in Staging-Umgebungen aufgedeckt werden, und durch rigoroses SLO-Alerting können New Relic SRE-Teams Probleme beheben, bevor sie sich auf Kundenseite bemerkbar machen, und so eine qualitativ hochwertigere Software liefern.

Schnellere Untersuchung und Behebung von Incidents

New Relic Diagramme und NRQL sind die Haupttools zur Analyse von Problemen, Ermittlung der Ursachen und Verkürzung der Zeit bis zur Behebung, selbst für komplexe Grenzfälle in der Produktion. Der Kontext, der von New Relic Dashboards und Runbooks gemeinsam bereitgestellt wird, verringert frustrierende Kontextwechsel und beschleunigt die kollaborative Fehlerbehebung. Darüber hinaus sorgt die granulare Visibility mit New Relic und die damit einhergehende Zuweisung von Verantwortlichkeiten für Services und Komponenten dafür, dass der Zeitaufwand bei der Suche nach dem verantwortlichen Team im Ernstfall deutlich sinkt.

Kontinuierliche Produktvalidierung und -verbesserung

Monitoring der New Relic eigenen Staging- und Produktionsumgebungen liefert unschätzbares Feedback aus der Praxis und ermöglicht schnelle Verbesserungen an der UX.

SKALIERUNG MIT KUBERNETES

New Relic Container-Fabric-Team erzielt enorme Skalierung und deutliche Kosteneinsparungen

Das Container-Fabric-Team, das für die Bereitstellung einer Self-Service-Kubernetes-Plattform für interne Engineering-Teams verantwortlich ist, nutzt New Relic ebenfalls zum Monitoring und Optimieren einer umfassenden Multicloud-Umgebung. Mit seinen Hunderten von Kubernetes-Clustern und Zehntausenden von Nodes bei großen Public-Cloud-Anbietern vertraut das Team auf New Relic für End-to-End-Transparenz, proaktive Problemlösung, Kostenoptimierung und Förderung der teamübergreifenden Zusammenarbeit. Zudem nutzt das Container-Fabric-Team New Relic für Observability – tief in die Kubernetes- und Multicloud-Operationen integriert.

Features

`Infrastructure Agent:` Wird auf allen Kubernetes-Nodes bereitgestellt, um Metriken auf Host- und Containerebene zu erfassen.

`Custom-Instrumentierung:` Wird häufig verwendet, um bestimmte Metriken von Kubernetes-Controllern, Automatisierung, CoreDNS und sogar Linux-Betriebssystemdetails für detaillierte Einblicke zu erfassen.

`Cloud-Integrationen:` Werden verwendet, um Metriken von den APIs der wichtigsten Public-Cloud-Anbieter abzurufen, liefern eine umfassende Übersicht der Cloudanbieterdienste sowie der internen Telemetrie.

`Dashboards und Query Builder:` Unverzichtbar für die grafische Darstellung von Plattform-Health und Performance-Trends sowie für Ad-hoc-Datenexploration bei Incident-Untersuchungen.

`Alerting:` Proaktive Alerts auf Basis der wichtigsten Health-Indikatoren der Plattform.

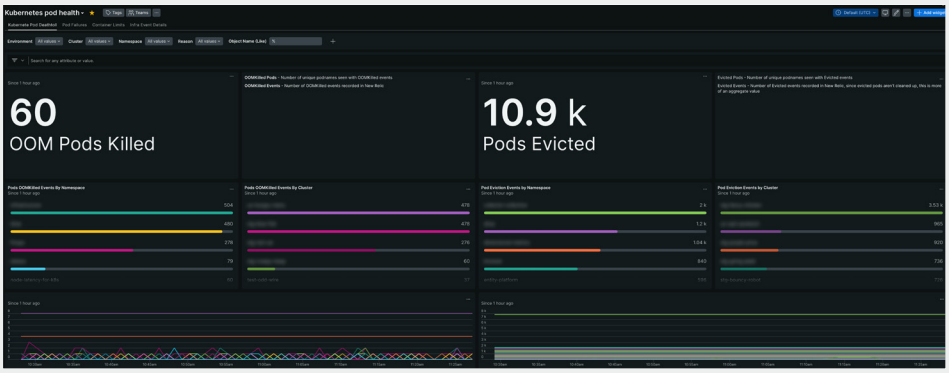
`Zentrale Data Platform:` New Relic liefert einen gemeinsamen Datenkontext, sodass Datensilos beim Container-Fabric-Team und den von diesem Team unterstützten Anwendungs- und Entwicklungsteams kein Thema mehr sind.

Metriken

Das Team konzentriert sich auf Health und Effizienz auf Plattformebene und verwendet daher folgende Leistungsindikatoren (KPIs):

1 Kubernetes-Health

- → Anzahl ungeplanter Pods
- → Probleme im Zusammenhang mit der Workerknoten-Skalierung
- → Pod-Zustände (z. B. „CrashLoopBackOff“)
- → Kubernetes-API-Server-, Scheduler- und CoreDNS-Metriken



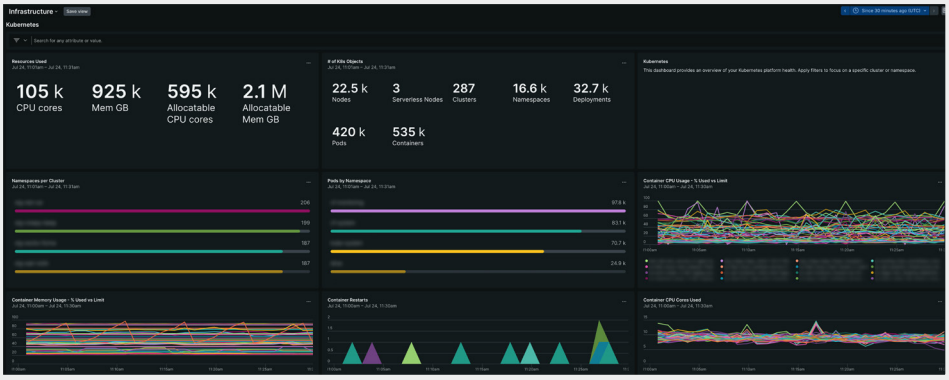
Kubernetes-Pod-Health

2 Ressourcennutzung und Kostenoptimierung

- → Erhebliche CPU-Auslastung bei Workern
- → Leerlauf-CPU und Speicher für Nodes (für besseres Bin Packing und effizientere Ressourcennutzung)

3 Cloud-Infrastruktur

- → Virtual-Machine(VM)-Instanzmetriken (CPU, Arbeitsspeicher, Festplatten-E/A, Netzwerk)
- → Kafka-Broker-Metriken (z. B. Replikationsfaktor, Netzwerkunterbrechungen)
- → Monitoring der zugrunde liegenden Cloudanbieter-Services und ihrer Performance



Kubernetes-Ressourcennutzung

Nachstehend sehen Sie einige der Resultate, die das Container-Fabric-Team dank New Relic erzielt.

Erhöhte Verfügbarkeit und Zuverlässigkeit

Proaktive Problembehebung: Durch kontinuierliche Beobachtung der Plattform kann das Team potenzielle Probleme identifizieren und angehen, bevor sie sich auf die Kundschaft auswirken.

Schnellere Untersuchung und Behebung von Incidents: New Relic Dashboards, Custom-Instrumentierung und die Möglichkeit, Daten über verschiedene Ebenen hinweg zu korrelieren, von Anwendungen und Services über Kubernetes-Ebenen (Pods, Nodes) bis hin zur grundlegenden Cloud-Infrastruktur, sorgen für eine deutlich verkürzte mittlere Zeit bis zur Behebung (MTTR). Als das Browserteam beispielsweise ein Problem mit dem Frontend meldete, konnte das Container-Fabric-Team das Problem mit ungeplanten Pods in Verbindung bringen und es schnell auf einen Istio-Kontrollebenen-Alert zurückführen. Durch Skalieren von Istio-Pods ließ sich das Problem dann beheben.

Ermitteln externer Abhängigkeiten: Dank der detaillierten Telemetrie konnte das Team ein Netzwerkproblem auf den Speicherservern eines Cloudanbieters als Grundursache für bestimmte Leistungsspitzen identifizieren, obgleich erste Untersuchungen auf etwas anderes hinzudeuten schienen. Diese umfassende Visibility in Cloudservices von Drittanbietern ist entscheidend für die Aufrechterhaltung der Plattformzuverlässigkeit.

Deutliche Kostenoptimierung

Datengestützte Instanzauswahl: Durch Performance-Benchmarking anhand von New Relic Daten kann das Team die Kosteneffizienz und Performance verschiedener Instanztypen und Cloudanbieter vergleichen und so die für ihre Workloads finanziell optimale Infrastruktur auswählen.

Verbesserte Ressourcennutzung: Durch Überwachung von CPU und Arbeitsspeicher im Leerlauf kann das Team proaktiv Möglichkeiten zur Optimierung des Bin Packing von Services auf Nodes finden. Das verbessert die Ressourcennutzung und senkt die Cloudkosten und ermöglicht auch die Herunterskalierung von nicht ausgelasteten Nodes.

Effektive teamübergreifende Zusammenarbeit

Frei verfügbarer Observability-Kontext: New Relic ist die zentrale Sprache und Datenquelle für interne Teams. Dank der gemeinsamen Nutzung von Dashboards und NRQL-Abfragen kann Kontext leichter geteilt werden, was wiederum Reibungsverluste beim Troubleshooting von Incidents verringert. Teams arbeiten effektiv zusammen und lokalisieren sowie beheben Probleme gemeinsam.

Bidirektionaler Wissenstransfer: Da die Daten gemeinsam genutzt und Probleme ebenfalls kollaborativ angegangen werden, können sich Teams leichter über Arbeitslast und Funktionen anderer Teams informieren. Und das kommt den Best Practices im Engineering zugute.

Self-Service für interne Kund:innen

Das Container-Fabric-Team stellt internen Entwicklungsteams die notwendigen Tools und Daten in New Relic zur Verfügung, damit sie ihre eigenen Services auf Anwendungsebene überwachen können. Während sich das Plattformteam auf die Infrastruktur-Health konzentriert, können die Anwendungsteams sich selbstständig um ihre Observability-Anforderungen kümmern. Dadurch sinkt die Abhängigkeit vom Plattformteam beim Routine-Monitoring der Services.

Fundierte strategische Entscheidungsfindung

New Relic unterstützt nicht nur die Incident-Behebung, sondern liefert auch die granularen Performance- und Kostendaten aus der Praxis, die für langfristige strategische Entscheidungen erforderlich sind – ganz gleich, ob es darum geht, den Cloud-Footprint zu erweitern, Cloudanbieter zu vergleichen oder die Multicloud-Strategie zu optimieren.

LOGMANAGEMENT AUF HYPERSCALE-NIVEAU

Stärkung der Servicequalität im Petabyte-Bereich für New Relic Engineering

Die globale für das Logmanagement verantwortliche Engineering-Organisation von New Relic verwendet ebenfalls die eigenen Produkte, um internen und externen Kund:innen einen herausragenden Service zu bieten. Das Logmanagement nimmt bei der New Relic Engineering-Organisation erheblichen Raum ein: Monat für Monat werden diverse Petabytes an Logs sowie Milliarden von Log-Abfragen verarbeitet. Die Organisation setzt echtes Continuous Deployment ein, mit häufigen, manchmal mehreren Dutzend Deployments pro Tag, und nutzt New Relic für Observability. Bei so häufigen Änderungen ist die zuverlässige Validierung von Deployments unverzichtbar, und New Relic liefert die dafür erforderlichen Einblicke.

„New Relic on New Relic“ und unser Loggingprodukt bringen uns echte Resultate.

Servicequalität

Der Hauptnutzen liegt darin, dass wir der Kundschaft eine bestimmte Servicequalität bieten können, da sichergestellt wird, dass das Loggingprodukt intern effektiv funktioniert.

Größere Release-Sicherheit

Die Möglichkeit, Probleme frühzeitig zu erkennen und zu beheben, sorgt für mehr Sicherheit bei Releases neuer Funktionen und Updates.

Proaktive Fehlererkennung

Durch tägliche Verwendung von New Relic können Teams Probleme noch vor der Produktion ermitteln und beheben, wodurch die Auswirkungen auf die Kundschaft minimiert werden.

Schnellere Incident Response

Die Teams, die zur New Relic Emergency Response Force (NERF) gehören, vertrauen auf das Loggingprodukt von New Relic zur effektiven Incident Response. Mit New Relic Alerts verknüpfte PagerDuty-Alerts liefern Diagramme mit Schlüsselmetriken und Runbook-Links, sodass die Diagnose in wenigen Schritten und die Behebung ohne Kontextwechsel durchgeführt werden kann.

Metriken

Aus den vielen Metriken, die nachverfolgt werden, sind nachstehend diejenigen aufgeführt, die für einen zuverlässigen Betrieb am nützlichsten sind.

1. **Service-Level-Indikatoren (SLIs):** Top-Level-SLIs werden regelmäßig auf wichtige Faktoren überprüft, z. B. Endpoint-Latenz für die Logerfassung und Compliance über verschiedene Integrationen hinweg (z. B. AWS Kinesis Firehose, TCP, syslog).
2. **Service-Level Objectives:** Zuverlässige Verfügbarkeit steht auf der Prioritätenliste von New Relic ganz oben, und diese Metrik spiegelt die Verpflichtung von New Relic zur Einhaltung von Datenintegrität und Zuverlässigkeit wider.
3. **JavaScript-Fehler:** Diese werden nach Umgebung, Browser, Benutzer:innen und Produktkomponenten überwacht, um die User Experience nachverfolgen und potenzielle Probleme zu ermitteln.
4. **Datenverzögerungen:** Den Anstieg und Rückgang von Verzögerungen zu überwachen, ist besonders für die Incident Response wichtig, denn New Relic Kund:innen sind auf eine hohe Plattformverfügbarkeit angewiesen.

Features

Für das Logmanagement nutzt die Engineering-Organisation zahlreiche New Relic Features, z. B.:

Service-Level, APM, Infrastruktur-Observability und Logs:
Dies sind Schlüsselelemente der Plattform. Sie dienen dazu sicherzustellen, dass die Hauptservices im Rahmen festgelegter Fehlerbudgets betrieben werden, und dass Probleme proaktiv erkannt und behoben werden.

Proaktives Alerting:
Für On-Call-Engineers ist das Alerting ein wesentlicher Bestandteil ihrer Incident Response, vor allem, wenn ihr Pager wegen potenziell schwerwiegender Probleme losgeht. Diese Alerts sind direkt mit New Relic Alerts verknüpft, die hilfreiche Diagramme für eine schnelle Diagnose bereitstellen. Durch diesen integrierten Alerting-Prozess zusammen mit etablierten Runbooks wird die Antwortzeit erheblich verkürzt und Probleme können proaktiv identifiziert und behoben werden.

Umfassende Integrationen:
Integrationen für die Services der großen Cloudanbieter, Open-Source-Tools sowie New Relic Agents ermöglichen eine toolübergreifende Datenerfassung und Logkorrelation, und das wiederum sorgt für flächendeckende Observability.

MEHR POWER FÜR DAS PRODUKTMANAGEMENT

So versteht New Relic seine Kundschaft besser

New Relic Produktmanager:innen nutzen die New Relic Plattform für aggregierte Analysen mit dem Ziel, die Kundenakzeptanz und -zufriedenheit zu erhöhen. Die NRDB ist das Rückgrat der New Relic Plattform und unterstützt eine Vielzahl von Datenstrukturen für alle möglichen Daten (Logs, Events, Metriken, Traces und mehr). Sie liefert aggregierte Informationen, die zu einem besseren Verständnis von Kundenakzeptanz, Spitzen und Unterbrechungen beitragen. So erhalten Produktmanager:innen von New Relic kritische Einblicke in die Geschäftsanalysen und können die von unserer Kundschaft genutzten Services besser pflegen und bereitstellen. Dazu gehört auch, Trends bei der Nutzung neuer Features zu erkennen, denn das hilft bei der Identifizierung von PLG-Opportunities (Product-led Growth).

Die New Relic Observability-Plattform wird tagtäglich dazu eingesetzt, den Zustand und die Auswirkungen von Feature Releases zu verfolgen. Viele Produktmanager:innen erstellen Dashboards, um zu sehen, wie neue Features insgesamt angenommen werden, und um anhand dieser Erkenntnisse weitere wichtige Funktionen zu identifizieren. Mithilfe von Custom-Events können sie Trends in bekannten User-Journey-Funnels im Blick behalten und sich einen umfassenden Überblick über die Produktakzeptanz verschaffen. NRQL ist dank ihrer enormen Flexibilität das ideale Tool für diesen Zweck und sie liefert Produktmanager:innen Live-Einblicke in die Produktakzeptanz insgesamt. Das ermöglicht eine rasche Performance-Analyse und effektive Iterationen. Eine so optimierte Plattform kann dem Produkt zu maximaler Nutzung verhelfen.

Viele Produktmanager:innen erstellen Dashboards, um zu sehen, wie neue Features insgesamt angenommen werden, und um anhand dieser Erkenntnisse weitere wichtige Funktionen zu identifizieren.

Die Verwendung von New Relic bei New Relic für das Produktmanagement hat mehrere Vorteile.

Schnellere Feedbackschleife

Es bietet sofortiges Feedback, sodass man weniger auf Kundenumfragen angewiesen ist.

Bessere Produktentwicklung

Da Trends hinsichtlich der Kundenakzeptanz schneller zutage treten, kann die Produktqualität schneller verbessert werden.

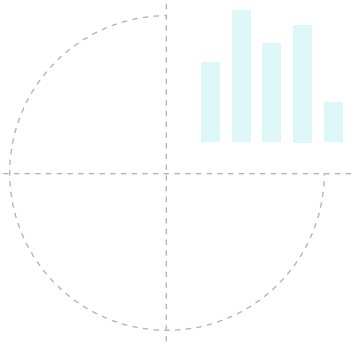
Optimierte Preisgestaltung

Umfassende Einsicht in die Cloud-Ressourcennutzung erlaubt Produktmanager:innen (PMs), die Preisgestaltung für New Relic Kund:innen zu verbessern.

Fazit

Die Tatsache, dass die Strategie „New Relic on New Relic“ die gesamte Engineering-Kultur im Unternehmen prägt, ist ein Zeugnis der Effektivität und Zuverlässigkeit von New Relic. Durch den konsequenten Einsatz unserer eigenen Produktsuite zum Monitoring, zum Verständnis und zur Verbesserung unserer kritischen Systeme stellen wir nicht nur die hohe Verfügbarkeit und Performance unserer Kundenumgebungen sicher, sondern können unsere Plattform selbst fortlaufend optimieren und validieren.

Dieser „Inside-Out“-Ansatz ist ein Grundpfeiler unserer operativen Exzellenz und ein wichtiges Alleinstellungsmerkmal auf dem Observability-Markt, denn die Erkenntnisse aus unserem eigenen Hyperscale-Betrieb haben direkten Einfluss auf die Produktentwicklung.



new relic.