new relic

Observability at New Relic

Our Engineering Playbook for Hyperscale and Reliability

White Paper / July 2025

Summary

Retailers, banks, streaming giants, and countless other businesses depend on New Relic as their essential “eyes and ears” to deliver reliable experiences at scale. They turn to us during their most critical moments—new product launches or major streaming events—watching traffic surge on New Relic dashboards and relying on alerts to flag abnormal behavior or errors. We understand the utmost importance of observability at scale because, as engineers, we live and breathe these challenges daily.

This paper will illustrate, through concrete examples, how our engineering organization uses our own product daily to achieve a broad set of critical business objectives, from significant cloud cost savings and boosting developer productivity to continuously improving the customer experience and maintaining high uptime.

New Relic’s engineering organization relies exclusively on its own observability platform to maintain unparalleled uptime and low latency. This self-instrumentation, or New Relic on New Relic, is crucial for managing the platform at immense scale, collecting over a trillion data points and executing more than 20 million queries daily, all while significantly reducing operational costs.

As creators and users of the New Relic platform, we develop solutions for our complex observability needs, and these learnings directly refine customer-facing product features. This “inside-out” approach ensures New Relic meets the demands of modern, distributed systems.

We will explore several use cases detailing how our internal teams—SRE, frontend, backend, platform engineering, network, and others—achieve operational excellence and innovation through comprehensive observability.

Through this overview, you’ll gain a deeper understanding of New Relic’s commitment to self-observability which not only ensures the reliable performance of our customers’ environments but also continuously refines and validates the very platform they rely on. This is our story of innovation, engineering excellence, and the unwavering confidence—backed by our own daily experience—that comes from running on New Relic.

We’ll explore three key pillars of our reliability strategy.

Measuring What Matters

Understanding and tracking the metrics crucial for reliable performance and efficient troubleshooting to achieve business objectives.

Self-Healing Systems

Reducing engineering toil and improving reliability by using automation to respond to and prevent potential issues.

Mitigating Issues Quickly

Equipping our teams with the tools and insights to rapidly diagnose and resolve problems when they do arise.

Measuring What Matters

One of the key tenets to running a reliable system is to measure what matters. This means measuring both what is important to the business and what is important for troubleshooting problems. New Relic measures what matters through New Relic. This includes using our agents, service level objective (SLO) product, and alerting product.

APM Agents at New Relic

Engineering teams like to run the application performance monitoring (APM) agent for their services and all hardware is typically monitored using the New Relic infrastructure agent. This creates a consistent set of metrics so that any engineer can move between teams or services and still understand the core health metrics.

New Relic's APM agents are highly tuned to provide the right key metrics to notice and diagnose service problems. Key metrics include HTTP response time, database call times, and external HTTP call times.

Additionally, while dedicated agent teams frequently add new instrumentation to our agents, other engineering teams—those who utilize the agents to monitor their own infrastructure, databases, and streaming services—have also contributed instrumentation for our agents. A great example is the current Kafka instrumentation in the Java agent. This instrumentation was originally created by a New Relic team that operates many core Kafka streaming services, initially as a Java agent extension. After being adopted by many internal teams, the instrumentation was eventually put in the APM product. The Kafka UI, which displays these metrics, was also jointly created by an APM product team along with our key Kafka and Streaming Services teams.

While dedicated agent teams frequently add new instrumentation to our agents, other engineering teams...have also contributed instrumentation for our agents.

Effective Instrumentation: Using Dashboards and Nerdpacks to Eliminate Context Switching

Teams at New Relic are trained to think about observability data during development. In addition to the default instrumentation provided in our agents, engineering teams have the ability to create additional custom instrumentation. Some teams send custom instrumentation to New Relic using APM agents. Others have built libraries to send instrumentation directly to our public metric, event, log, and trace endpoints. Examples of custom instrumentation include a New Relic Event for every query to the New Relic database (NRDB) and an event for every initial APM agent connection to New Relic. Teams then display these custom events in dashboards or custom Nerdpacks (custom applications). These custom applications integrate textual instructions with live query results and visualizations.

For instance, Kafka pipeline stalling issues can be diagnosed with views on a custom Nerdlet, which also automatically generates the command needed to extend data retention transforming a multi-step manual process into a single copy-paste action. This significantly reduces context switching and accelerates resolution.

Kafka pipeline stalling issues can be diagnosed with views on a custom Nerdlet, which also automatically generates the command needed to extend data retention transforming a multi-step manual process into a single copy-paste action.

Service Level Objectives: Achieving Business Objectives With SLOs

SLOs are important as they define and measure the customer experience over a set time window. They are critical for balancing reliability work with new functionality. At New Relic, we require all teams to maintain an internal set of SLOs. Prior to enforcing SLOs across the company, we found that our telemetry data was extremely rich, but was tuned for troubleshooting and not for measuring customer experience. So we created an SLO bar raiser program that helped teams create customer focused SLOs with values that reflected the reality at the time.

Using these measurements, we were able to share with the business the cost to run at the current SLO and the work required to increase the SLO. Teams who have benefited the most from SLOs watch their SLOs daily and take corrective action when necessary. These teams have not only improved customer experience, but also significantly reduced their pager load.

Alerting and Terraform: How to Automate Proactive Insight

At New Relic, our teams use New Relic Alerting to get notified of any system anomalies. This allows New Relic engineers to take action quickly to mitigate any potential system degradation. Most teams use Terraform to create and maintain their alerts in version control. Most teams also use facet alerts to ensure alerts are created for any new cell or environment. An example facet alert on hostname is shown below:

Teams use New Relic Alerting to get notified of any system anomalies. This allows New Relic engineers to take action quickly to mitigate any potential system degradation.

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

To ensure teams have the right alerts, New Relic provides a set of recommended alerts to teams in our Engineering Standards. These include alerts for Out of Memory (OOM) kills, pods waiting, Kafka lag, error rates, and more. Leaders then dashboard their team's alerts (since each alert is recorded in NRDB) and watch trends weekly.

Self-Healing Systems

Alerting is important to respond rapidly to potential service interruptions. However, alerting an incident responder to take manual action is still time consuming. That is why New Relic automates as much as possible, focusing on self-healing systems.

Auto-Scaling Reduces Engineering Toil

New Relic has invested significantly in auto-scaling algorithms which can rapidly scale up or down services. These algorithms use metrics such as cpu and memory to perform the scale ups and downs. This has significantly reduced interruptions and team pager notifications. For example, it was not unusual for our Logging Team to be paged 2-4 times per a week to help scale a service. After implementing autoscaling, the team receives significantly fewer pages.

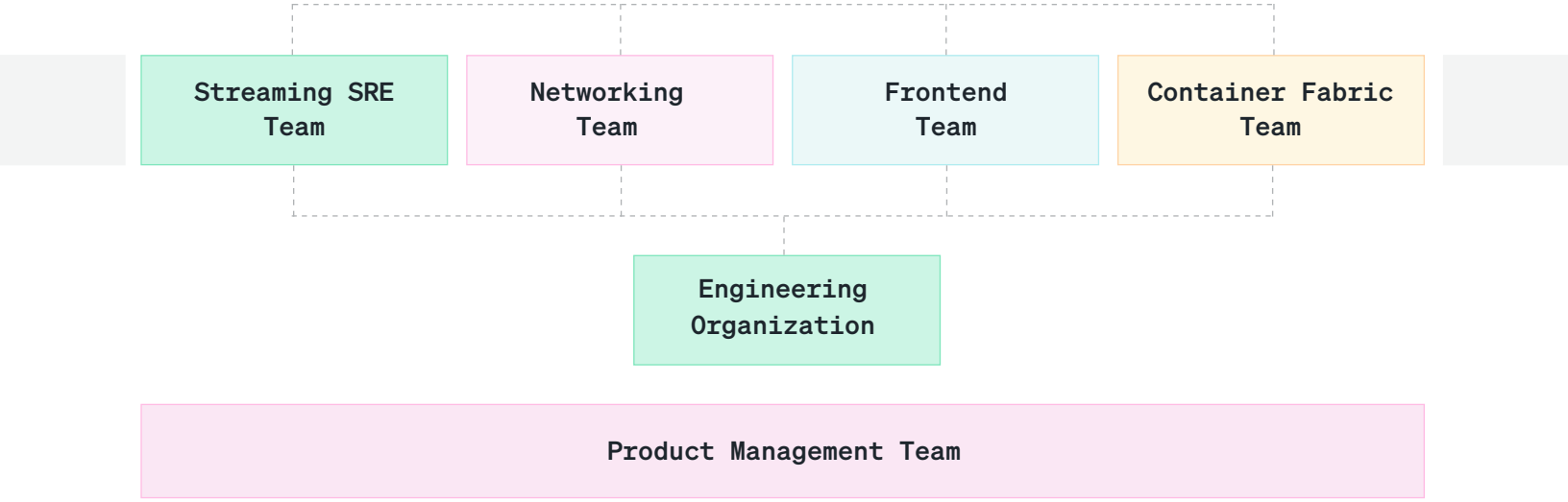
Auto-Rollback Improves Reliability

While New Relic services go through a series of checks before being deployed to production, sometimes bugs do reach production. In these cases, New Relic employs automatic service rollback. When a change is deployed through New Relic's continuous deployment pipeline, a workflow is started that listens for the entity's health. If the service becomes unhealthy, then the workflow will trigger the continuous deployment pipeline to rollback the unhealthy instances.

Mitigating Issues Quickly

While New Relic supports self-healing through automation, there are times when human expertise is indispensable for troubleshooting and mitigation. In these moments, New Relic itself serves as the primary platform for our incident responders. Our teams use change tracking to pinpoint potentially problematic deployments. Product pages, curated dashboards, specialized experiences, and ad-hoc queries are then shared to identify the root cause and guide the precise mitigation actions needed.

The following examples from our internal teams demonstrate how New Relic empowers us to prevent and rapidly mitigate issues, ensuring continuous reliability for our customers and ourselves.



KAFKA OBSERVABILITY INNOVATION

How the Streaming SRE Team Focuses on Reliability

New Relic runs one of the top Kafka implementations in the world and operates hundreds of services that either produce and/or consume from Kafka. Early on, the Streaming SRE Team invested in the foundations of a reliable Kafka environment for New Relic serving our customers. For instance, Kafka lag can directly correlate to delays or drops in telemetry ingest, which can lead to customer alerts being missed or delayed, evidencing that Kafka's reliability remains essential.

The charter of this team is dedicated to Kafka operations, and they focused on the creation of a custom Nerdpack for Kafka observability. This highly customized New Relic Nerdpack, brimming with custom metrics, became an indispensable tool, shared internally with over 50 teams relying on Kafka services. The value derived from these operational insights was so profound that it directly spurred the development of customer-facing Kafka and observability functionalities.

This highly customized New Relic Nerdpack, brimming with custom metrics, became an indispensable tool, shared internally with over 50 teams relying on Kafka services.

The primary motivation for building this extensive Kafka observability was to overcome the blind spots experienced during incidents. Without granular data, diagnosing root causes and quickly identifying recurring problems was a significant challenge. The objective was to "layer on lots and lots of observability" to understand Kafka behavior comprehensively—before, during, and after incidents.

Insights

The custom Kafka Nerdpack provides deep insights into a wide array of metrics. The streaming SRE Team uses these insights to:

1. **Alert on Kafka Lag to Maintain Ingest Integrity:** The team's most critical use of New Relic is for assuring telemetry ingest integrity using alerting on Kafka metrics, such as Kafka Lag. Kafka lag directly correlates to delays or drops in telemetry ingest, which can lead to critical customer alerts being missed or delayed. This poses a significant business risk, as customers rely on timely alerts for their own operational awareness. Comprehensive Kafka lag alerts allow for scaling of ingest and optimizing performance.
2. **Maximize Responsiveness:** New Relic enables the team to be highly responsive to Kafka processing issues, facilitating rapid remediation and minimizing customer impact.
3. **Understand Kafka Client Behavior:** Identifying misconfigurations, overloaded buffers, and stalled clients.
4. **Monitor Server-Side Health:** Monitoring broker performance and resource utilization.
5. **Observe Request Patterns:** Analyzing changes in client request patterns to anticipate and mitigate potential issues.

The implementation of Kafka observability has had a profound impact on New Relic operational efficiency and reliability.

Dramatic Reduction in Troubleshooting Time

With comprehensive observability data at their fingertips, the streaming SRE Team can diagnose Kafka client incidents in minutes, often resolving issues within a total impact time in minutes and seconds. This contrasts sharply with the hour or so that may have been necessary without such detailed insights.

Nerdpacks as Dynamic Runbooks

A key innovation championed by this SRE Team is the use of New Relic Nerdpacks, as dynamic runbooks. These custom applications integrate textual instructions with live query results and visualizations. For instance, Kafka pipeline stalling issues can be diagnosed with views on the Nerdlet, which also automatically generates the command needed to extend data retention transforming a multi-step manual process into a single copy-paste action. This significantly reduces context switching and accelerates resolution.

Executive-Level Insight

Directors and executives at New Relic utilize the Kafka observability nerdlet to quickly assess the overall status of lag across entire Kafka clusters or environments, providing a high-level view of ingestion performance and scalability.

Intelligent Auto-scaling for Optimized Performance and Cost

The Streaming SRE Team has developed sophisticated auto-scaling tools that use both New Relic telemetry and custom metrics. For instance, they use New Relic CPU metrics to dynamically scale Kubernetes resources up or down based on traffic demands. This allows the team to effectively manage surges in ingested traffic by scaling up to burn down lag, and then downscaling during low-traffic periods. This dynamic auto-scaling prevents over-provisioning of resources, ensuring cost efficiency while maintaining the capacity to handle fluctuating workloads.

SUPERIOR NETWORK RELIABILITY

Networking Team Increases Reliability with New Relic

New Relic's Networking Team monitors its global network environment, which includes hundreds of cells with Kubernetes clusters that connect to multiple cloud environments. To achieve comprehensive visibility, SREs and network engineers developed code using New Relic libraries that are deployed in every cluster to collect key network telemetry.

The primary motivation for building this extensive network observability was to provide customers with a better understanding of the network and to build trust. The goal is to empower other teams to self-serve and eliminate network issues as the first suspect when troubleshooting.

Insights

The custom network dashboards provide deep insights into a wide array of metrics, including:

1. **Network Performance:** Monitoring bandwidth, packet loss, jitter, latency, and path utilization.
2. **Infrastructure Health:** Using the infrastructure agent with Amazon and Azure connectors to get information from those platforms and ingest it into New Relic.
3. **Connectivity Validation:** Utilizing a custom script that pings from one location to another to confirm connectivity.
4. **Cost Optimization:** Monitoring an egress network address translation (NAT) service to exit a cloud provider's network at a significantly lower price point, and monitoring for unexpected cost spikes.

The implementation of this network observability has had a profound impact on New Relic's operational efficiency and reliability.

Dramatic Reduction in Troubleshooting Time

The implementation has reduced the number of pages the Networking Team receives. An example of network observability was identifying a routing issue where traffic was failing over to an undersized backup solution due to a missing static route. This allowed New Relic networking teams to quickly remediate the issue and later implement an active-active setup for cloud providers' routes to balance traffic and prevent saturation.

Proactive Identification of Misconfigurations

By identifying issues like missing static routes, New Relic optimizes resource usage and significantly enhances system reliability, leading to cost efficiencies.

Dynamic Runbooks

The goal is to empower other teams to self-serve and eliminate network issues as the first suspect when troubleshooting.

Executive-Level Insight

The team also uses New Relic to optimize costs by monitoring an egress NAT service. They also monitor for unexpected cost spikes and help other teams identify and resolve issues that lead to unnecessary increased data traffic charges.

FRONTEND EXCELLENCE

How New Relic Frontend Teams Build Industry-Leading UI Platforms

For the New Relic frontend organization, the production monitoring environment creates a continuous feedback loop and ensures immediate insights into potential issues. This self-monitoring extends across all critical facets of New Relic's operations, even for complex micro-frontend architectures where the team has developed detailed custom dashboards and alerts.

Functionalities

The frontend teams rely heavily on these New Relic functionalities.

Synthetics: Proactive monitoring of critical functionalities and user flows.

Dashboards: Customized dashboards provide a holistic view of system health, performance trends, and critical alerts across various components and micro-frontends.

New Relic Query Language: Engineers extensively use the intuitive query builder and New Relic Query Language for on-the-fly data exploration, hypothesis testing, and rapid incident investigations.

Service Levels and Alerts: Proactive alerting based on defined SLOs ensures immediate notification of service degradations or potential outages, often before customers are impacted.

Metrics

Key metrics continuously observed include:

1. **Loading Time:** Performance metric for frontend experiences.
2. **Availability:** Ensuring services are accessible and operational.
3. **Latency:** Tracking response times to identify bottlenecks and ensure a smooth user experience.
4. **Throughput:** Monitoring data volume and processing rates to assess system capacity.
5. **Error Rates:** Identifying and quantifying errors, particularly JavaScript errors for front-end, to pinpoint areas requiring immediate attention and improvement.

For the frontend engineering teams, the New Relic on New Relic approach yields significant strategic advantages.

Proactive Problem Solving

By catching issues in staging environments and through aggressive alerting on SLOs, New Relic SRE teams can address problems before they reach and impact customers, leading to higher quality software.

Faster Incident Investigation and Resolution

New Relic charts and NRQL are primary tools for dissecting problems, identifying root causes, and accelerating time to resolution, even for complex edge cases in production. The shared context provided by New Relic dashboards and runbooks reduces context switching and enables faster collaborative troubleshooting. Further, granular visibility with New Relic that assigns ownership to services and components, drastically reducing time spent identifying the responsible team during incidents.

Continuous Product Validation and Enhancement

Monitoring New Relic's own staging and production environments provides invaluable real-world feedback, allowing rapid UX improvements.

KUBERNETES AT SCALE

How the New Relic Container Fabric Team Achieves Massive Scale and Cost Savings

The Container Fabric Team, responsible for providing a self-service Kubernetes platform for internal engineering teams, also uses New Relic to monitor and optimize an expansive multi-cloud environment. With an impressive scale of hundreds of Kubernetes clusters and tens of thousands of nodes across major public cloud providers, the team relies on New Relic for end-to-end visibility, proactive problem-solving, cost optimization, and fostering cross-team collaboration. The Container Fabric Team also uses New Relic for observability needs, integrating it deeply into their Kubernetes and multi-cloud operations.

Functionalities

`Infrastructure Agent:` Deployed across all Kubernetes nodes to collect host-level and container-level metrics.

`Custom Instrumentation:` Extensively used to expose specific metrics from Kubernetes controllers, automation, CoreDNS, and even Linux OS details for deep insights.

`Cloud Integrations:` Used to pull metrics from major public cloud providers APIs, providing a holistic view of cloud provider services alongside internal telemetry.

`Dashboards and Query Builder:` Critical for visualizing platform health, performance trends, and for ad-hoc data exploration during incident investigations.

`Alerting:` Proactive alerts based on key platform health indicators.

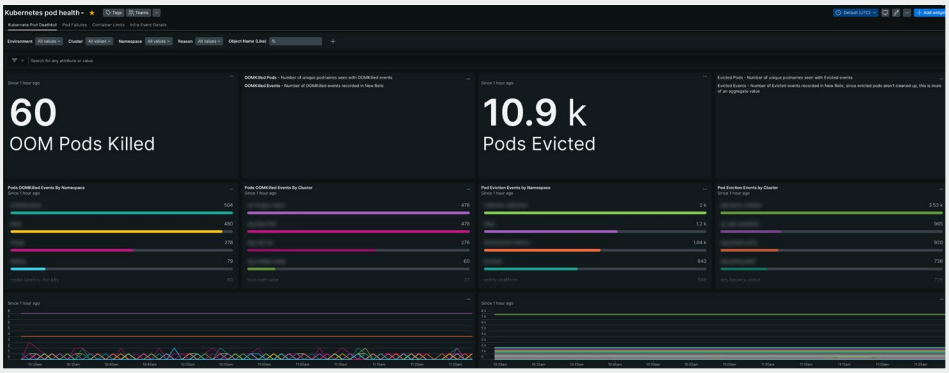
`Centralized Data Platform:` New Relic provides a shared data context that breaks down silos between the container fabric team and application and developer teams they support.

Metrics

The team focuses on platform-level health and efficiency and is using the following key performance indicators (KPIs):

1 **Kubernetes Health**

- → Number of unscheduled pods
- → Issues related to worker node scaling
- → Pod states (for example, “CrashLoopBackOff”)
- → Kubernetes API server, Scheduler, and CoreDNS metrics



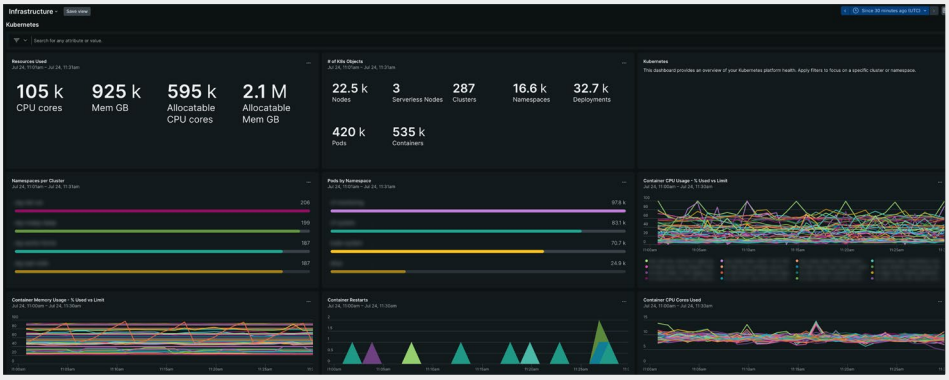
Kubernetes Pod Health

2 **Resource Utilization and Cost Optimization**

- → Significant CPU usage on workers
- → Idle CPU and memory for nodes (to improve bin packing and reduce waste)

3 **Cloud Infrastructure**

- → Virtual machine (VM) instance metrics (CPU, memory, disk I/O, network)
- → Kafka broker metrics (for example, replication factor, networking drops)
- → Monitoring of underlying cloud provider services and their performance



Kubernetes Resource Utilization

Below are some of the results that the Container Fabric Team achieves using New Relic.

Enhanced Availability and Reliability

Proactive Problem Solving: By continuously observing the platform, the team can identify and address potential issues before they impact customers.

Faster Incident Investigation and Resolution: New Relic dashboards, custom instrumentation, and the ability to correlate data across various layers—from application and services to Kubernetes layers (pods, nodes) to underlying cloud infrastructure—significantly reduce mean time to resolution (MTTR). For instance, when the browser team reported an issue with the frontend, the Container Fabric Team tied it to unscheduled pods, and quickly traced the issue to an Istio control plane alert before resolving it by scaling Istio pods.

Identifying External Dependencies: The detailed telemetry allowed the team to pinpoint a networking issue on a cloud provider's storage servers as the root cause of certain performance spikes, even when initial investigations pointed elsewhere. This deep visibility into third-party cloud services is crucial for maintaining platform reliability.

Significant Cost Optimization

Data-Driven Instance Selection: Through performance benchmarking using New Relic data, the team can compare the cost-effectiveness and performance of different instance types and cloud providers. This enables them to select the most financially optimal infrastructure for their workloads.

Improved Resource Utilization: By monitoring idle CPU and memory, the team can proactively identify opportunities to improve "bin packing" of services on nodes, leading to better resource utilization and reduced cloud expenditure. This also allows them to force scaling down of underutilized nodes.

Smooth Cross-Team Collaboration

Shared Observability Context: New Relic acts as a common language and data source across internal teams. Sharing dashboards and NRQL queries facilitates quick context sharing and reduces friction during incident troubleshooting, allowing teams to collaborate effectively to pinpoint and resolve issues.

Bidirectional Knowledge Transfer: The shared data and the process of joint investigation makes it easy for teams to learn about other teams' workloads and functions, leading to improved overall engineering practices.

Empowered Self-Service for Internal Customers

The Container Fabric Team provides internal developer teams with the tools and data within New Relic to monitor their own services at the application level. While the platform team focuses on infrastructure health, application teams are empowered to self-serve their observability needs, reducing reliance on the platform team for day-to-day monitoring of their services.

Informed Strategic Decision Making

Beyond incident resolution, New Relic provides the granular data necessary for long-term strategic decisions, such as expanding cloud footprints, comparing cloud provider offerings, and optimizing their multi-cloud strategy based on real-world performance and cost data.

HYPERSCALE LOG MANAGEMENT

Powering Quality of Service at Petabyte Scale for New Relic Engineering Organization

The New Relic global engineering organization responsible for log management extensively uses their own products to deliver exceptional service to internal and external customers. When it comes to log management, the New Relic engineering organization achieves significant scale, handling tens of petabytes of logs, along with billions of log-focused queries, monthly. This organization operates in a true continuous deployment mode, deploying frequently—often dozens of times per day—using New Relic for observability. Because of these frequent changes, reliable validation of deployments is paramount, and New Relic provides the necessary insight.

By using New Relic on New Relic and its logging product, we're achieving significant results.

Quality of Service

The primary value is delivering a certain quality of service to customers by helping to ensure the logging product works effectively internally.

Proactive Issue Identification

Using New Relic daily allows teams to proactively identify and address issues before they reach production, minimizing customer impact.

Safer Releases

The ability to identify and address issues early enables safer releases of new features and updates.

Faster Incident Response

As part of the New Relic Emergency Response Force (NERF), the teams rely on New Relic logging product for effective incident response. PagerDuty alerts linked to New Relic alerts provide charts of key metrics and runbook links for quick diagnostic steps and resolution eliminating context switching.

Metrics

While many metrics are observed, here are some of the most useful to maintain high reliability.

1. **Service Level Indicators (SLIs):** Top-level SLIs are regularly reviewed for key experiences, such as endpoint latency for log ingestion and compliance across various integrations (for example, AWS Kinesis Firehose, TCP, syslog).
2. **Service Level Objectives:** There is a high target of availability of the New Relic platform to its customers. This metric reflects New Relics commitment to data integrity and reliability.
3. **JavaScript Errors:** Monitored by environment, browser, user, and product component to track user experience and identify potential issues.
4. **Data Lag:** Monitoring lag increase and decrease is crucial for incident response in particular because New Relic customers depend on high availability of the platform.

Functionalities

The log management engineering organization uses many New Relic functionalities including:

Service Levels, APM, Infrastructure Observability, and Logs:
These core platform capabilities and insights are used to ensure that key services operate within designated error budgets, and to proactively troubleshoot and resolve issues.

Proactive Alerting:
On-call engineers rely on alerting as a crucial component of their incident response, in particular when getting paged for potentially high-severity issues. These alerts link directly to New Relic alerts that provide charts for immediate diagnosis. This integrated alerting process, coupled with established runbooks, significantly cuts down on their response time and helps them proactively identify and address issues.

Comprehensive Integrations:
Integration with major cloud providers' services, open source tooling, coupled with New Relic agents allows for ingestion of data and correlation of logs across tooling powering comprehensive observability.

EMPOWERING PRODUCT MANAGERS

How New Relic Fuels Deeper Customer Understanding

Product managers at New Relic use the New Relic platform for aggregated analytics to support customer adoption and satisfaction. The NRDB, which lies at the foundation of New Relic, supports a variety of data structures for all data (logs, events, metrics, traces and more), allowing product managers to better understand customer adoption, spikes, and drops in an aggregated fashion. This allows New Relic product managers to get critical insight into business analytics to better serve and deliver the services our customers use. This includes understanding trends in the adoption of new features to better identify product-led growth (PLG) opportunities.

Product managers actively rely on the New Relic observability platform daily to track the health and impact of feature releases. Many product managers create dashboards to track holistic adoption of new features and identify important functionality based on that adoption. They can use custom events to track trends in funnels of known user journeys, providing holistic insight into product adoption. The flexibility of NRQL makes it an ideal tool for this purpose, providing product managers with real-time insights into overall product adoption. This enables them to quickly analyze performance, iterate effectively, and optimize the platform to drive maximum adoption.

Many product managers create dashboards to track holistic adoption of new features and identify important functionality based on that adoption.

Using New Relic internally for product management offers several benefits.

Faster Feedback Loop

It provides real-time feedback, reducing the need to wait for customer interviews to gain insights.

Better Quality Product Development

The ability to quickly understand trends in customer adoption leads to improved product quality.

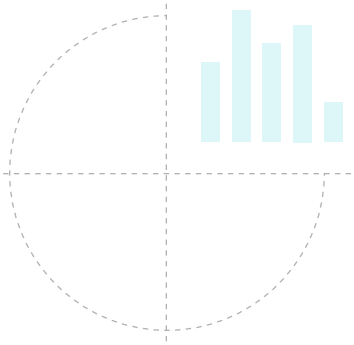
Optimized Pricing

With key visibility into cloud-resource usage, product managers (PMs) are empowered to give better prices to New Relic customers.

Conclusion

The engineering culture that has been built around the New Relic on New Relic strategy is a powerful testament to the efficacy and reliability of the New Relic observability platform. By consistently using our own product suite to monitor, understand and improve our critical systems, we not only ensure the high availability and performance of our customer environments but also continually refine and validate our platform.

New Relic's "inside-out" approach, where our learnings from operating at hyperscale directly influence product development, is a cornerstone of operational excellence and a key differentiator in the observability market.



new relic.