

Why containerization needs context

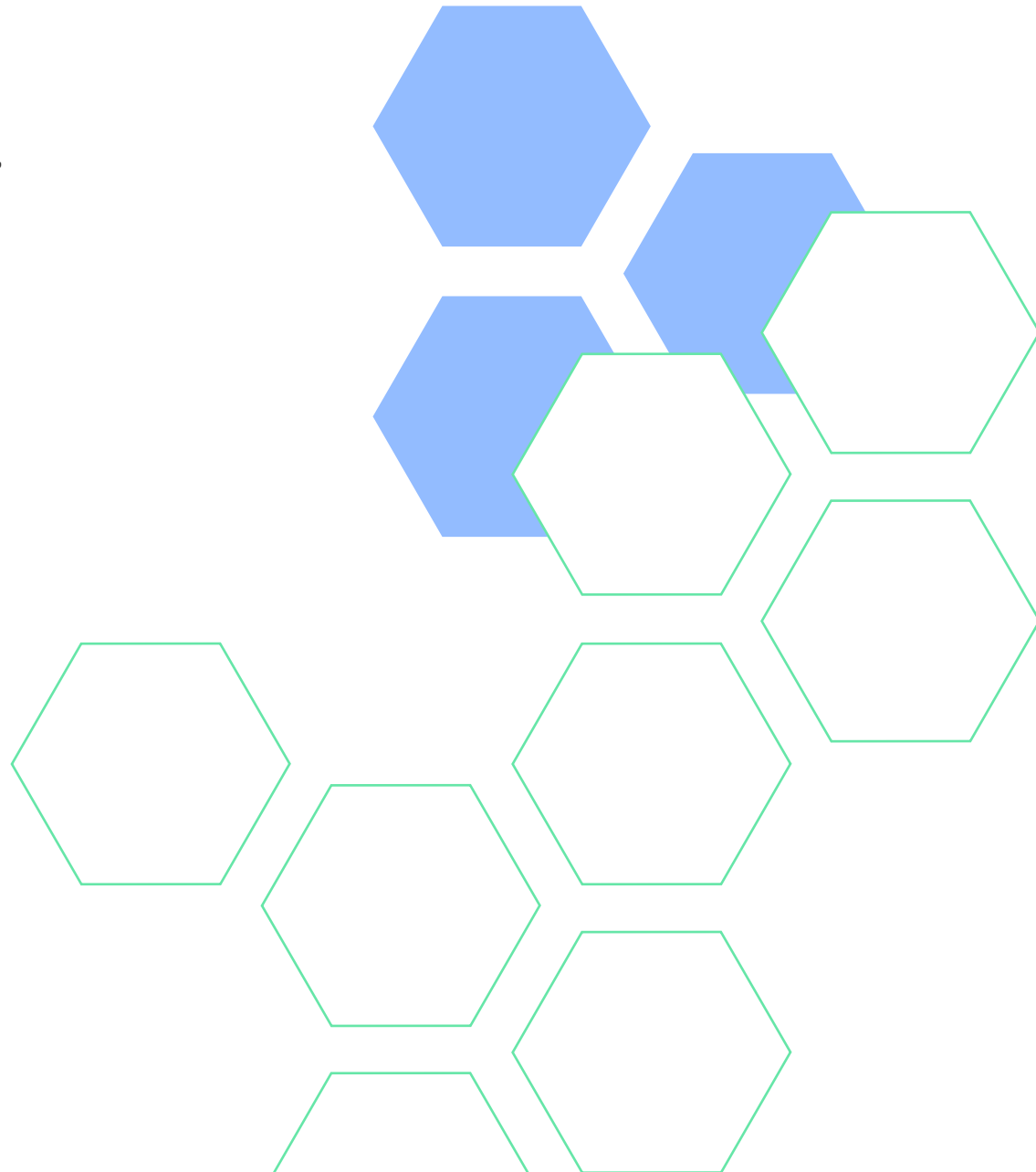
The problems with infrastructure monitoring in the age of Kubernetes



The Age of Containerization

Containers have changed how applications are designed, built, deployed, and maintained.

By creating a consistent, lightweight environment for applications, they offer velocity and improved time-to-market for software developers, and essentially eradicate the “works on my machine” problem inherent with monolithic applications. In no uncertain terms, container-based applications help companies differentiate themselves with fast, intuitive, and personalized digital experiences.



The Age of Containerization

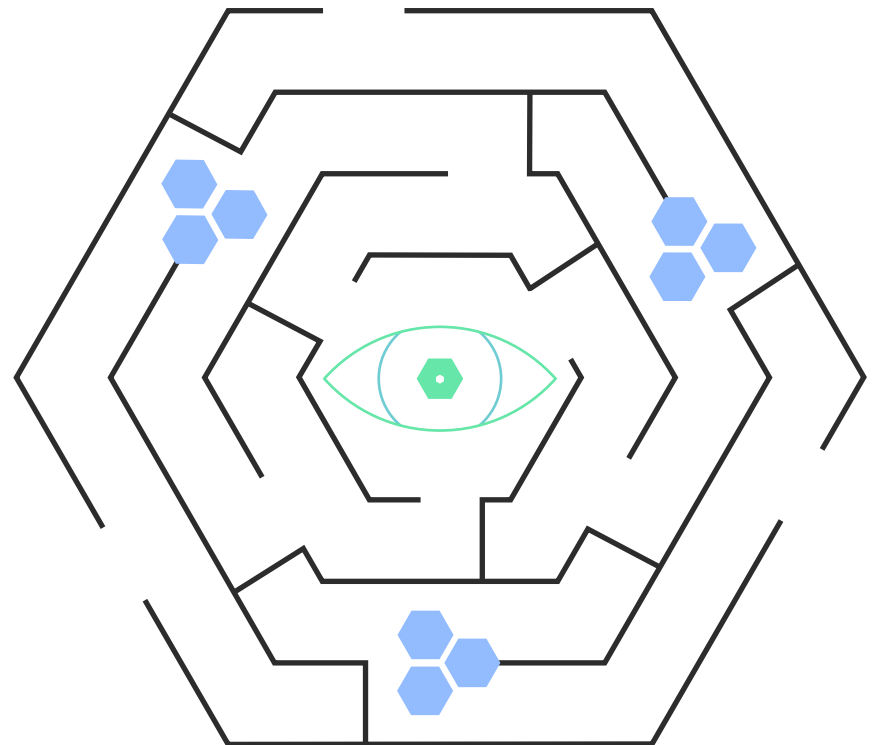
But they also introduce unprecedented complexity

Containers are ephemeral in nature and are blurring the once-solid lines between application and infrastructure, requiring a new approach to their management and orchestration.

So even though **Kubernetes** makes it easier to deploy and operate applications in a microservices architecture, it's also incredibly difficult to manage at scale. Because it doesn't address some very

basic—and very important—issues like cluster health and hierarchy, and how those affect other parts of the IT stack.

If a pod is failing, Kubernetes may be able to resolve it temporarily by spinning up a replacement pod, but it won't be able to show you all the related logs, or let you quickly figure out why it's crashing, and which other parts of your stack are impacted.



The Age of Containerization

That's the paradox of modern software environments

Organizations are leveraging modern application architectures and deployment methodologies to deliver a better experience for their customers... but how do they do it without inadvertently compromising that customer experience with downtime, bugs, or latency?

In this guide we'll take a look at how teams can successfully adopt modern software architectures like Kubernetes without jeopardizing end-user experiences.

Kubernetes monitoring 101

If you're new to monitoring Kubernetes, or interested in learning how New Relic integrates with and helps you monitor Kubernetes, **this is the guide** for you.

01

The importance of context

When a problem arises with your container-based applications, it's not enough to know that there is a problem. You need to be able to see why it happened, with the flexibility to drill into the ever-changing interdependencies between application, infrastructure, logs, and end-user experience.

Due to the distributed and ephemeral nature of containers, without the appropriate context, they're a nightmare to maintain at scale. And Kubernetes—while it makes lots of deployment functions easier—doesn't provide that context.



The importance of context

That's why observability is so important

A generally accepted definition of observability (as applied in engineering and control theory) is that it measures how well internal states of a system can be inferred from knowledge of that system's external outputs. This requires an approach that allows you to gather various types of data (whether metrics, events, logs, or traces) from across your entire stack (whether backend, frontend, or application) and to visualize them all in a single platform.

The more tools you use, the more difficult this can become. Many teams today rely on a handful of different point solutions and diffuse

dashboards to analyze and troubleshoot system health. Individually, these may provide insight into what's happening within one part of the stack, but together they provide at best a vague patchwork of overall service health, application performance, and the resulting customer experience.

What's more, these disparate solutions further widen the chasm between application developers and infrastructure operators, promoting different "sources of truth" and reference vocabulary across teams. What starts out as a convenient way to solve singular problems quickly devolves into a technical debt.¹

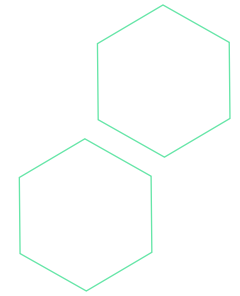
According to the **DORA 2019 State of DevOps report**, teams with high technical debt are on average 1.6 times less productive, while the highest performing DevOps teams were 1.4 times more likely to have low technical debt.

A shared observability framework helps these teams understand the underlying contextual relationships between containerized applications and the infrastructure that supports them. Engineers become equipped to deliver excellent customer experiences with software despite the complexity of the modern digital enterprise.

¹ services.google.com

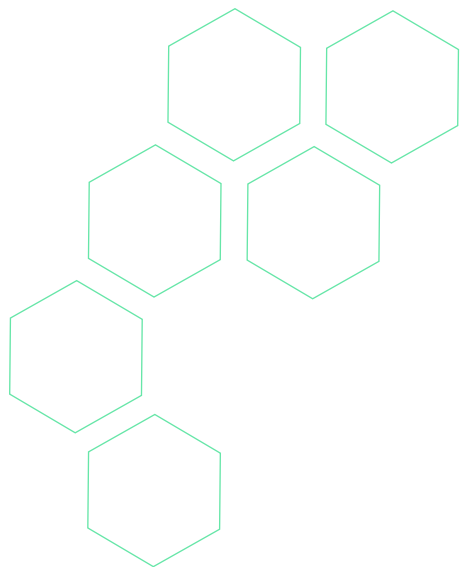
02

Observability in action



What does observability look like in the New Relic One platform?

Let's take a look, with the [New Relic Kubernetes cluster explorer](#).



What does true observability look like with New Relic One?

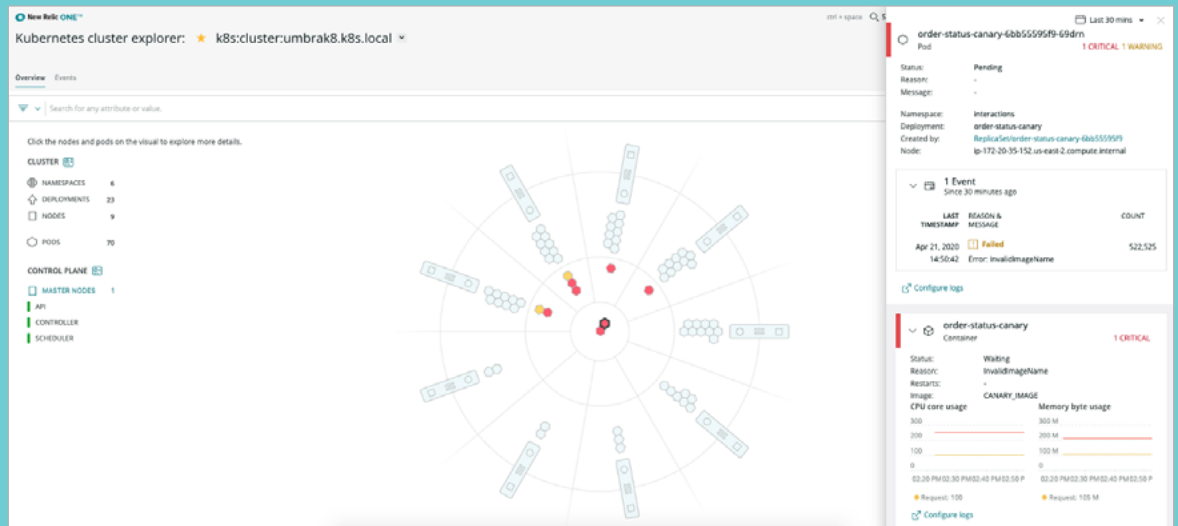
- **End-to-end**
Integrates APM, Infrastructure, and Logs in one intuitive user experience that also connects with Mobile, Browser, and Synthetics
- **Open**
Bring in data from any source, whether agent-based, open source, or third party
- **Connected**
Enables you to intuitively understand every important connection and interdependency in your stack
- **Programmable**
Lets you build custom applications and visualizations based on your exact needs and your specific environment
- **AI-enabled**
Proactively detects anomalies in your stack and empowers teams to improve mean time to detection (MTTD) and mean time to resolution (MTTR)
- **Unlimited scale**
Leverages the [world's most powerful telemetry database](#) so you can scale with confidence

Observability in action

The cluster explorer allows teams to manage Kubernetes clusters in a new and intuitive way. By leveraging observability telemetry and connecting the data in a compelling user experience, users are empowered to move beyond infrastructure and investigate deeper into applications, traces, logs, and events—with a single click—while staying grounded in a centralized UI.

One of the biggest difficulties with monitoring Kubernetes is the lag in knowledge: by the time a problem is identified, your infrastructure could have already made dozens of changes, and suddenly, without full observability, the information you need is almost impossible to find.

The cluster explorer makes finding that information easy.



The New Relic Kubernetes cluster explorer

Observability in action

You can see the logs, containers, and traces for any pod—in one click, without toggling between tools to find the data that matters.

The default data visualizations of your cluster provide a fast and intuitive path to getting answers and understanding Kubernetes environments, so you can gain context into the complexity associated with running Kubernetes at scale.

Teams that have this kind of context are empowered to resolve errors more rapidly. Because you can quickly detect cluster performance issues—even before they have a noticeable impact on the customer experience.

Observing a Kubernetes environment in this way lets you address issues proactively, reducing MTTD and MTTR. So you can find issues and their underlying causes and address them before a malfunction becomes critical.

And you can set up smart alerts, so instead of getting overwhelmed by hundreds of different event alerts—some critical, others minor—you can limit alerts to the ones you actually need to act on, such as when hosts stop reporting or if a node's CPU or memory usage exceeds a desired threshold.

“By being able to trace across all of those systems and then very quickly tie down where our performance bottlenecks are with New Relic, we can get to the root of what is actually involved in serving a user request and fix it.”

Andrew Duncan, Principal Engineer, ITV

“Since deploying the full New Relic platform, AB InBev has seen its MTTR for incidents drop by 80% and its MTBO improve—again, proof-positive that customers are having a better experience with the mobile app.”

Yang Tang, Global Director of Engineering, AB InBev

03

The future of infrastructure monitoring

Today, most enterprises' observability data is found in silos, leading to a lack of correlation between the health and performance of infrastructure, the applications that run on it, and the customer experiences those applications support and enable.

In the age of containerization, that means there's greater potential for every issue, however big or small, to lead to downtime.

The best solution is one that gives you full observability: both a bird's-eye view and granular detail of your whole complex infrastructure and how different parts of your stack are interacting, even if they may seem to be working independently.

When it comes to complex hybrid infrastructure in a large enterprise, you have a lot to worry about: system resilience, tool consolidation, resolving errors quickly and efficiently, team alignment, the potential for IT innovation, and a reduction in direct and indirect costs.

But at the end of the day, what matters most is the health of the infrastructure propping up your business.

To maintain that health, you need to make sure you can see the whole picture. You need to make sure you have context.

According to Gartner, the average cost of IT downtime is \$5,600 per minute, which is why infrastructure visibility is so key. For more, see Uptime is everything.

“With New Relic, we moved from needing to monitor deployments, to having enough insight and trust in the platform that we could literally push the button and start working on the next feature. Visibility into the deep workings of the platform enables us to reconsider how things are architected, make decisions based on real data, and focus on what will deliver value.”

Mike Robinson, Chapter Lead, IT Applications and Development, [Lightbox](#) / Spark New Zealand

Find out how we can help you maintain availability and uptime.

New Relic One is an open, connected, and programmable platform that gives you end-to-end, contextual observability across your entire tech stack. It gives you a consolidated view of all your data, from your customers' browser and mobile device experiences to your applications and infrastructure, wherever it runs.

This reduces blindspots, provides context, and gives you insights across artificial organizational boundaries—so you can focus on building more perfect software for your customers.

