

How to take infrastructure monitoring to the next level

The four imperatives for building true observability



The paradox of modern infrastructure monitoring

IT infrastructure has never been simple. But at least it all used to be in one place.

Today, the reality that DevOps and SRE teams face is a sprawling network of complex systems and changing environments. And here's the rub: This increasingly complex infrastructure is now more critical to business success than it's ever been because software itself is now more critical to business success.

The paradox of infrastructure monitoring is that the more mission critical it becomes, the more complex it becomes to monitor and manage.

So downtime in revenue-generating or customer-facing applications hurts businesses more. But it's also harder to diagnose the cause of downtime when you're dealing with a distributed architecture and large teams.

No wonder IT doesn't sleep at night. (It's not just the 2 a.m. alerts.)

What you want is faster MTTR, a shared and clear understanding of what's really going on, and why. So that instead of fighting fires, you're confidently preventing them.



This is why true observability matters so much

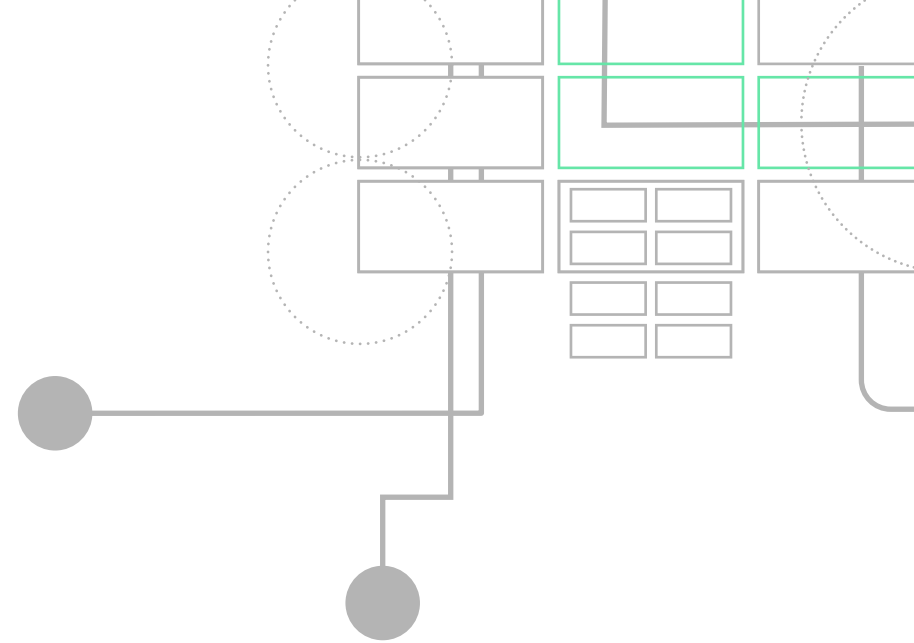
A bag of metrics from a disconnected set of tools isn't sufficient in a modern environment.

What you need is observability.

Observability means proactively collecting, visualizing, and applying intelligence to all of your metrics, events, logs, and traces in order to gain a holistic understanding of your entire software system.

How do you get there? That's what this piece is about. In it, we'll look at four imperatives for today's DevOps and SRE teams to achieve true observability:

- 01 Modern monitoring for modern environments
- 02 Customized dashboards and visualizations
- 03 Visibility in one place across the entire stack
- 04 Greater scale and efficiency





01

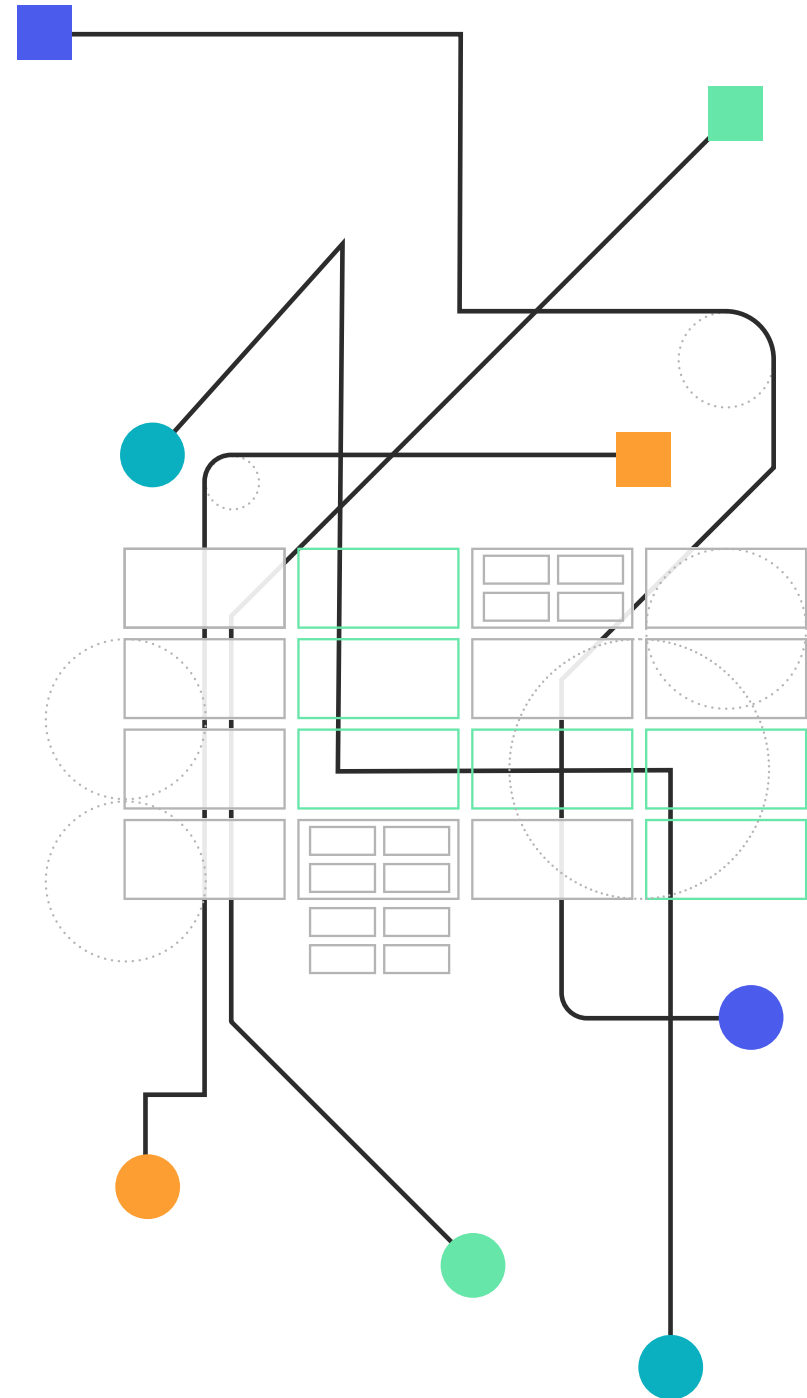
The image features a stylized background with overlapping circles in various shades of teal and blue. In the lower-left quadrant, the numbers '01' are displayed in a dark brown outline font. The '0' contains two concentric dotted lines, and the '1' contains a dotted line at its top right corner. Three small, solid light-green dots are positioned at the starting points of these dotted paths: one at the top of the '0's outer loop, one at the top of the '0's inner loop, and one at the top of the '1's dotted segment.

True observability

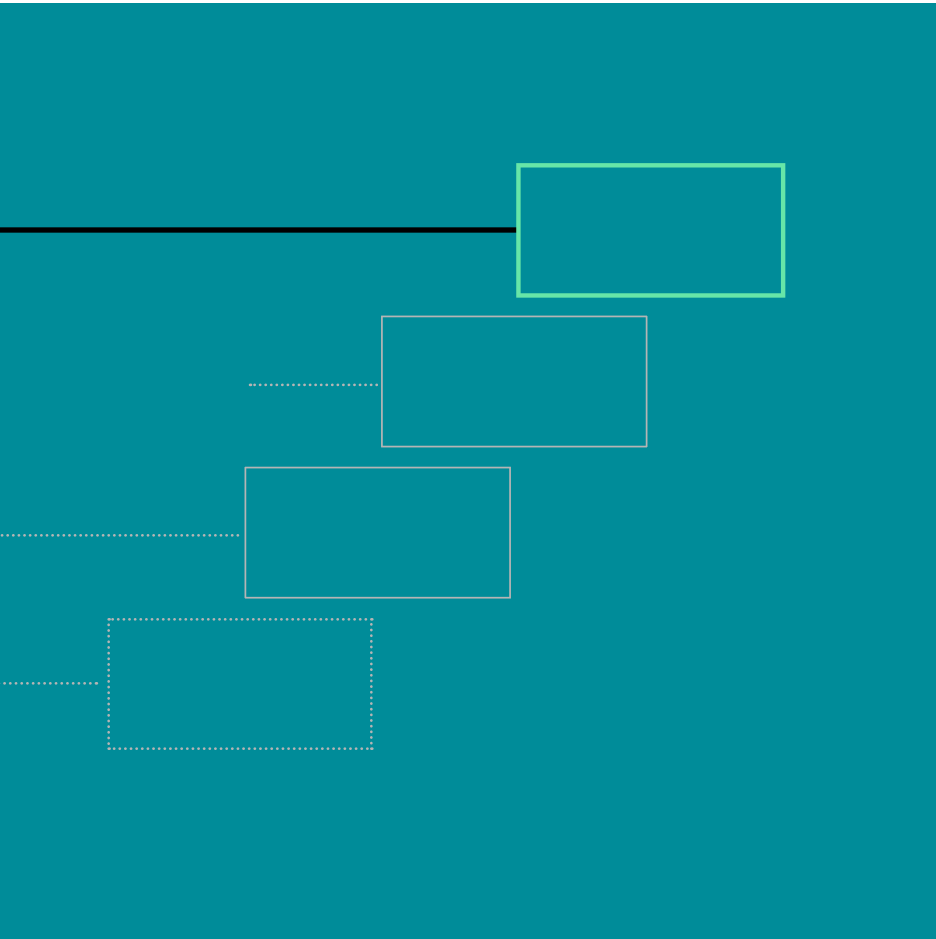
Observability is about seeing how problems in one part of your stack affect another part. So you can move from seeing that something went wrong to seeing why the issue occurred in the first place.

When you understand why problems occur, you can fix them much more quickly and prevent them from happening again.

The context you gain from true observability also helps you connect how infrastructure health and performance impacts the experience customers have—giving you more clarity about the business outcomes of software and system health.



The real effects of true observability



Speed is still your greatest asset in monitoring and maintaining your infrastructure. True observability gives you speed where you need it so you can put your focus where you want it.

The point is to more rapidly deploy more resilient software, quickly detect problems, reduce MTTR, and build team confidence so that when you deploy new code, you know exactly how that code will perform in production.

But it's also about having a healthy team. By working proactively to prevent future incidents, you gain more control over schedules and suffer from fewer unscheduled changes and late nights. That all makes for a happier team.

Three critical capabilities of a true observability platform

- **It needs to be open**

Viewing all telemetry data in one place—regardless of whether it's instrumented via agents or third-party sources—eliminates blindspots.

- **It needs to connect the silos**

Just dumping data into one place isn't enough. Understanding what's happening across your software and systems and quickly deriving meaning helps you pinpoint issues faster and make smarter decisions.

- **It needs to be programmable**

It isn't a platform if you can't build on it. But it's more than that; programmable means building your own unique visualizations and tailored applications that matter to your business.



We'll get into each of these points in more detail. So let's get back to it.



02

Imperative #1:

You need modern monitoring for modern environments

Modernizing your infrastructure is important if you want to maintain a competitive advantage with your software. But it means you end up using different tools to monitor hosts, the network, storage devices, logs, etc.

This prevents consolidated end-to-end visibility, and results in:

- Inconsistent and incomprehensive telemetry
- Low data resolution, meaning spikes go undetected and issues can only be dealt with too late
- Lack of visibility into the off-the-shelf applications, the SaaS applications they're responsible for, and even custom applications without an APM solution
- Problems affecting users before you notice them
- Lack of correlation between the health and performance of the different infrastructure and application components
- No visibility into unexpected or incorrect configuration changes that lead to performance issues

To make matters worse, many traditional monitoring tools run on-prem, which means they require additional resources and skills to be managed appropriately.

As a result, identifying and troubleshooting problems is slow and cumbersome, which means they take longer to resolve. And a lack of detailed data means root cause can't be identified, so issues recur, which puts a strain on your team.

This ends up having a big impact on the customer experience.

Imperative #1: You need modern monitoring for modern environments

Case in point: containers change everything

Imagine you're managing the infrastructure for a company that relies on a massive inflow of data from Internet of Things (IoT) devices. The data is crucial to the company's success and the customer experience.

Added to that is rapid growth through acquisition. Not only does acquisition mean your system is handling more data flowing through it, but it's spread out across a complex cloud architecture.

You get an alert that an application has slowed. And that's all. An alert.

But is it a code error in the application that's running inefficiently? Maybe it's a problem with the data flowing in, in which case, are you going to need to check each device individually? Could it be an infrastructure resource problem that needs to be addressed to prevent serious problems down the line?

This example is based on a real problem faced by a company that was scaling quickly. Read the case study summary to the right.

Fleet Complete uses New Relic to keep data rolling

Fleet Complete, a telematics company, uses IoT devices to collect GPS, vehicle health, and other data, and deliver meaningful insights needed to keep customer commercial vehicle forces rolling and to drive its connected commercial vehicle platform.

It needed an environment that could scale dramatically to deal with new acquisitions and increased data flow.

The solution was AWS cloud, which brought new challenges. Enter New Relic. Within 12 months, Fleet Complete's cloud migration was 60% complete, its software release cycle was three weeks shorter, and the company had complete visibility into its all-important data ingestion pipeline.

[Read the full case study here.](#)

Imperative #1:

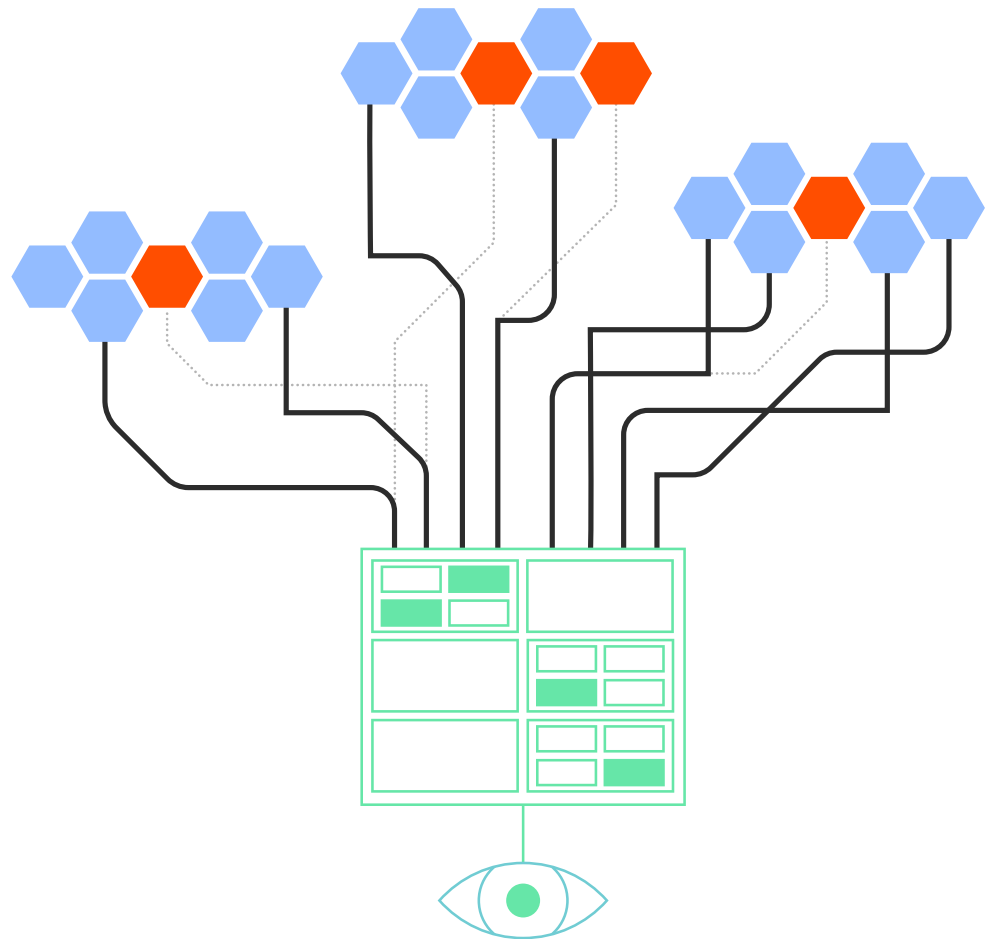
You need modern monitoring for modern environments

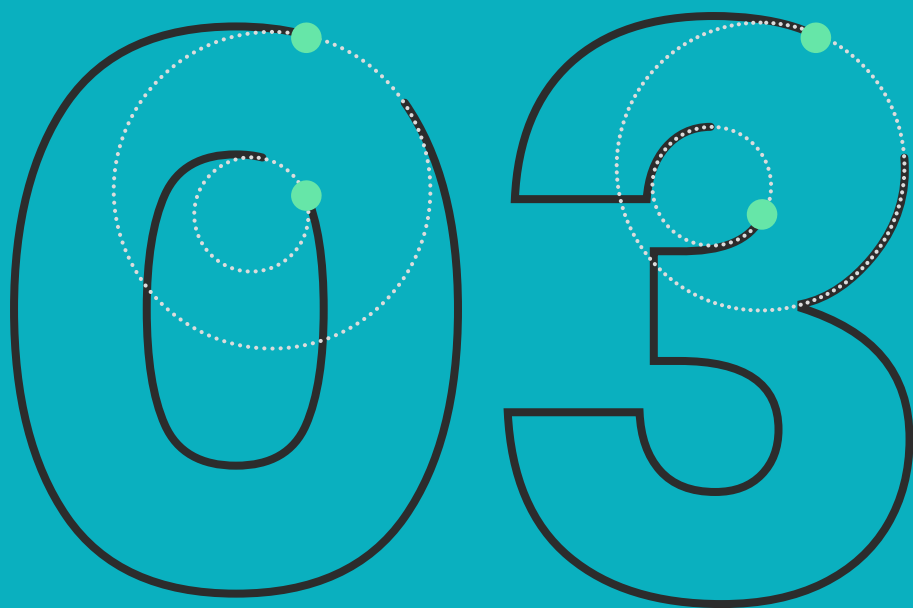
To observe modern environments, you want to assess the health of the elements in a cluster; check the status, metrics, and logs for a specific container; and see specific Kubernetes events that affected the container. Not only that, but you want to see the application metrics and traces for a service running in that container.

By bringing your monitoring tools in-line with the challenges of distributed, cloud-based infrastructure, you get better insight into the performance of distributed apps. And, of course, a better overview of your entire stack.

With an observability platform built to handle containerization and Kubernetes' environments, you can achieve faster deployment of changes, fixes, and upgrades.

All this facilitates more resilient systems and decreases downtime. Spending less time being reactive means more time future-proofing your systems. Which allows you to embrace automation and build self-service tooling, so development teams can build and deploy applications faster and more frequently.





Imperative #2: Customized dashboards and visualizations



Your business, software, and infrastructure systems are not, and never will be, exactly like anyone else's. Each one is critical to delivering specific goals for your organization. Not only that, the deployment of that software and utilization of your infrastructure is unique to your ops team.

That's why modern monitoring solutions provide curated experiences, out-of-the-box, to surface key telemetry and insights. But true observability goes a step further. Your teams need the ability to build tailored visualizations and applications that surface the data and insights that matter to them and the business.

With customizable dashboards and visualizations, you can choose to monitor the parts of your stack most relevant to particular business outcomes. As these shift, and your monitoring requirements change, so can your dashboards.

Imperative #2: Customized dashboards and visualizations

For instance, a retailer that relies on fulfillment and distribution centers across the country needs to summarize the health of the business, according to specific distribution KPIs. They have data from multiple accounts across many different centers. The C-suite doesn't need to see, or even understand, all of it. But they need to see performance.

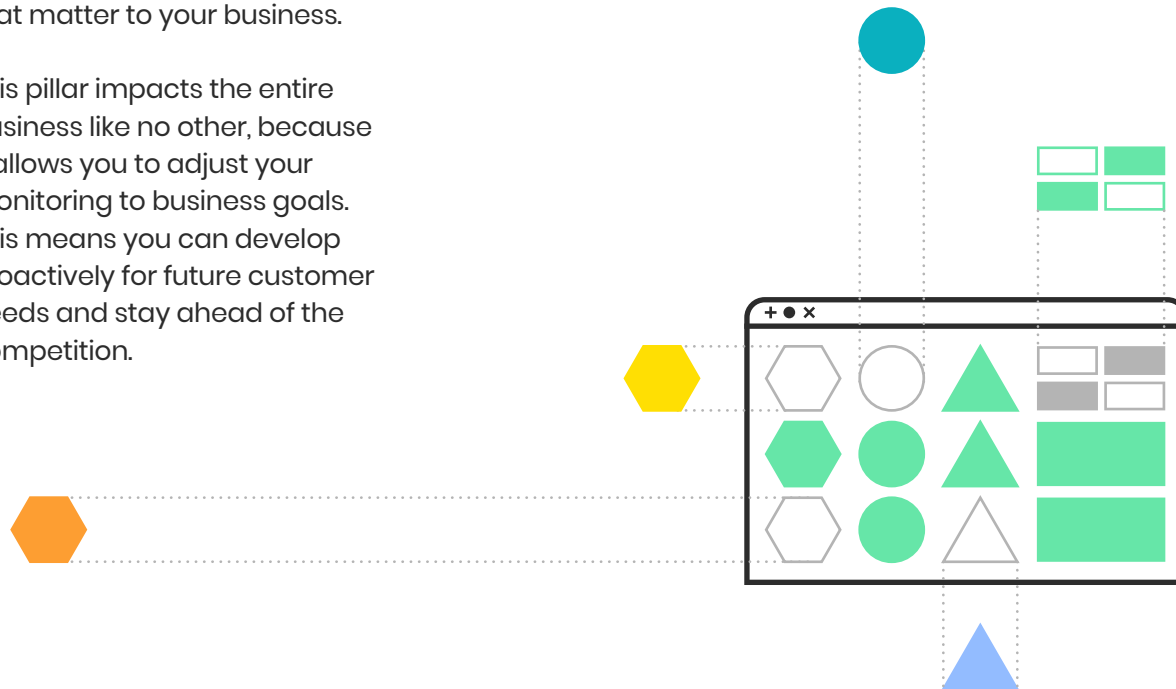
With a customizable dashboard, they can build a view that lets them see incidents at specific centers against certain business functions. For this example, a grid format that cross-references centers with functions would give the best view. With a single click they can dive into center-specific functions and see a list of incidents.

Customized insights, for specific business outcomes, is really the crux of true observability. It's how you make your monitoring work for you and proactively contribute to business goals.

It's easier to find and fix problems when you can tailor your telemetry data to use cases that matter to your business.

This pillar impacts the entire business like no other, because it allows you to adjust your monitoring to business goals. This means you can develop proactively for future customer needs and stay ahead of the competition.

You also get the benefit of bespoke solutions, without costly implementation by an external team.



Imperative #2:

Customized dashboards and visualizations

Start with open source, customizable solutions

Open source apps allow you to customize existing solutions or use parts of the code to build your own applications for your specific needs. Here are three examples of applications developed by us for the New Relic One platform.

Cloud optimize

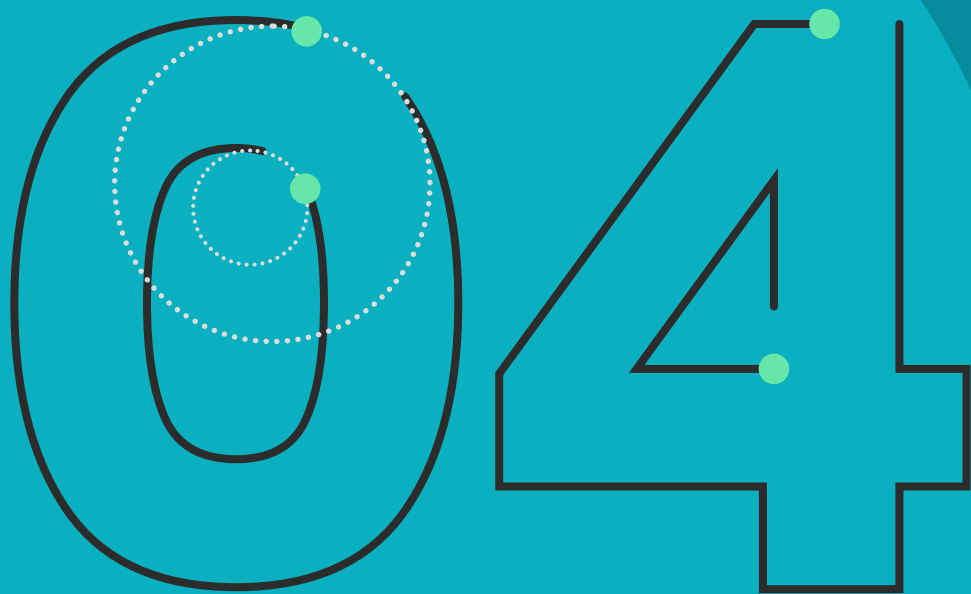
Combat over-resourcing by comparing the size of instances to their utilization, and estimate your savings by optimizing resource size. Select the hosts, regions, and other configurations to specify your unique business use cases. Cloud optimize supports AWS, Azure, and GCP.

Browser analyzer

Optimize web page performance with Browser Analyzer, which displays an analysis of performance and forecasts how performance improvements can impact KPIs like bounce rate or traffic. You can identify which pages have the worst performance to target high-impact fixes.

Customer journeys

Create an interactive funnel so you can customize the steps that are relevant to your customers' workflow. Get displays of standard data for each step—such as page views, error rate, and error count—and access deeper metrics with a click



Imperative #3:

Unified end-to-end visibility

Modern microservices architectures provide abstractions that blur the line between infrastructure and applications.

This simplifies deployment but adds complexity for monitoring. Your tools need to provide unified, end-to-end visibility across your entire estate, and through your full stack.

Time spent switching between different tools that monitor different parts of your stack is time wasted. It creates data silos that increase toil and the risk of blind spots. Interpreting performance metrics from multiple tools can also lead to human error. Having those metrics in one place reduces the chance of error so you can act decisively and quickly.

Tool consolidation puts all your infrastructure and application performance, customer experience, and log data in one place, so you can detect, diagnose, and resolve problems faster.

Imagine you're 90% of the way through diagnosing an app problem but the final part is in a log somewhere and you need to switch tools to find and fix it. You're losing precious seconds you don't have each time you need to switch contexts.

Or consider that the health of the services you rely on is just as important as the health of your own system. Each service has its own status page, but checking those requires visiting 14 different pages. Most of those pages publish their APIs, so why shouldn't you have a consolidated, single view of their status?



Imperative #3: Unified end-to-end visibility

When you run a combination of legacy and DIY monitoring tools, you lose the overall view of your system's health. And this problem is amplified by teams working in silos with siloed data. This puts a strain on IT and makes resource allocation difficult.

Seeing your system through an integrated, single screen removes blind spots and puts into view the full map, from infrastructure health to customer experience. Improved MTTR means less downtime, and that means less lost revenue and improved profitability. And, of course, the consolidated view and single tool means better allocation of resources.



05

Imperative #4: Greater scale and efficiency

Infrastructure needs to scale. And as it does, you need monitoring tools that can scale, too.

But traditional, self-hosted monitoring tools take time to scale, maintain, and upgrade as your surface area expands.

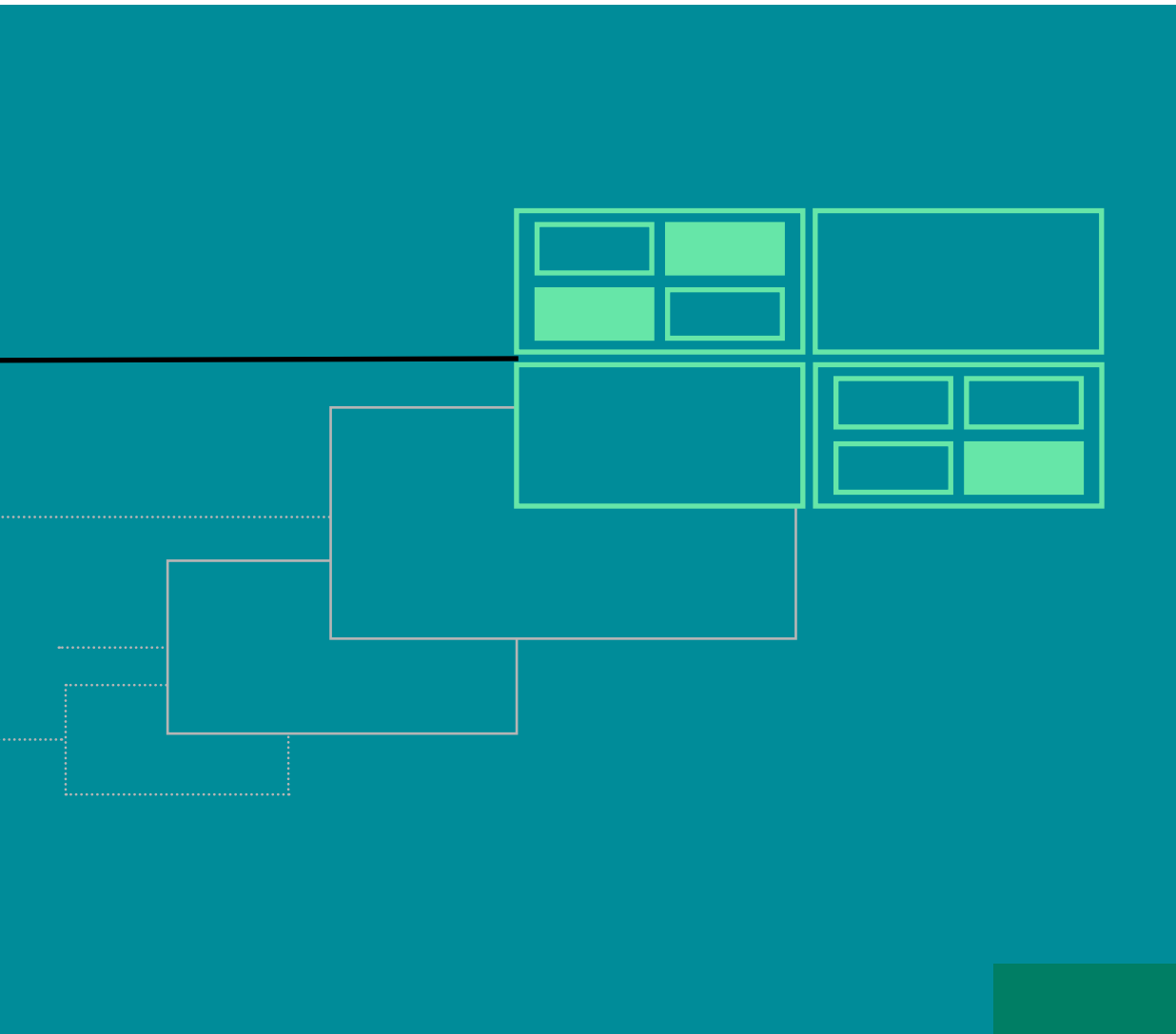
A modern infrastructure monitoring solution, delivered as a SaaS offering, should feel invisible. It should make it easier to see the reality of your environment as it becomes more complex, instead of making it harder.

In addition, a modern approach to observability needs to incorporate AIOps and intelligence capabilities to enable faster incident response.

This gives you the ability to proactively detect anomalies and automatically correlate incident events to reduce alert noise. Metadata and enrichment help you diagnose incidents and get to root cause faster. So you can take action to remediate more quickly.



Imperative #4: Greater scale and efficiency



This means you get notified about problems before the customer notices and you can diagnose them more efficiently. Smarter alerting ensures the teams best equipped to respond are notified first.

The result is a focus on better customer experiences, more essential tasks, and proactively preventing incidents. Your team is free to focus on their real job: shipping new products, software, and features more quickly to the market.

A modern approach to infrastructure monitoring lets you spend less time maintaining your monitoring and more time focusing on scaling and optimizing your infrastructure.

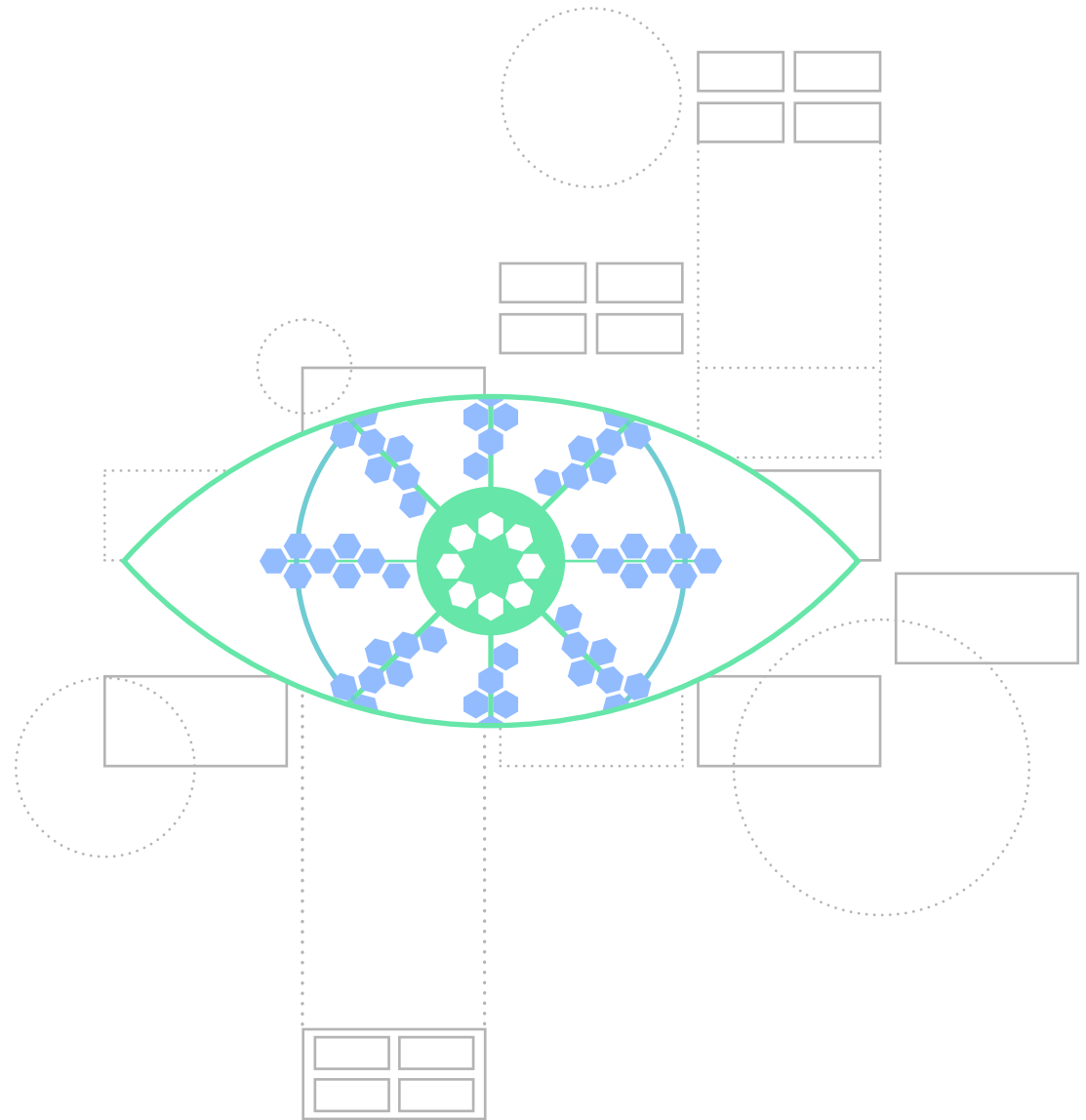
The background features a composition of overlapping circles in various shades of teal and blue. A vertical line divides the image into two halves. On the left, a large, bright teal circle is prominent, with a darker blue circle partially overlapping it. On the right, a medium-sized teal circle is visible, and a larger, lighter blue circle is partially seen in the upper right corner. The word "Conclusion" is written in white, bold, sans-serif font on the left side, positioned within the bright teal circle.

Conclusion

Getting to true observability

Observability is about making the job of managing IT infrastructure easier at the exact same time that it's getting more and more complex. Being able to see where a problem is, why it's happening, what to do about it, and how it's affecting the rest of your infrastructure is the difference between traditional monitoring and true observability.

It's also the difference between seconds and minutes, and between scaling for tomorrow's opportunities and firefighting today's problems.



An observable stack is an adaptable stack

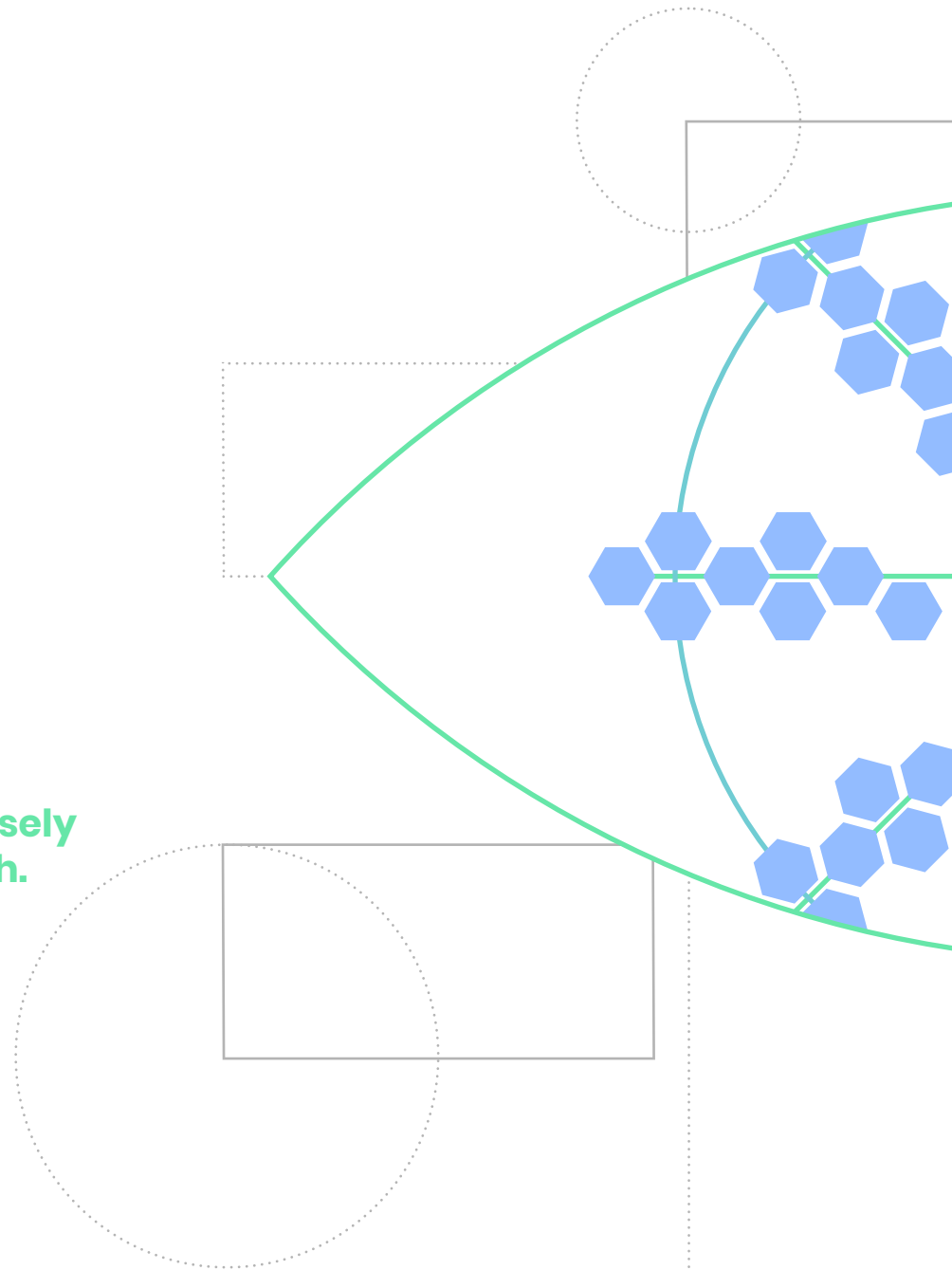
As important as observability is to effective infrastructure management, it's also important to remember what all of this is really about. Observability isn't an end in and of itself.

The point is to empower you and your infrastructure team to more rapidly understand what systems components need to adapt and how. In some cases, that might be to prevent downtime. In others it might be to provision an appropriate amount of resources. In others still it might be to accommodate a new innovation.

Because at the end of the day, perhaps the only thing that is certain about modern infrastructure management is that there will be changes. And those changes will have a ripple effect across an increasingly complex surface area.

The teams that have the most positive impact on their businesses will be the ones that can navigate all this change and adapt in the ways the business needs them to.

**The ones that know precisely
what they're working with.**



If this all sounds like something your team needs, we should probably talk

New Relic 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 quickly find and fix problems.

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

