White Paper

The Three Dimensions of Cloud-Native Observability

Jason Bloomberg
President, Intellyx

August 2020

As enterprises move to cloud-native computing, their IT infrastructure becomes increasingly dynamic, scalable, and ephemeral, raising the bar on the management of IT production environments.

Traditional IT operations management tools that offer dashboards, giving operations personnel visibility into production issues, are now insufficient. Instead, organizations require a new generation of cloud-native observability tooling.

Such tooling is able to deal with vast quantities of operational telemetry in real-time, giving operators greater insights into issues, as well as the ability to mitigate and even prevent problems.

Organizations just beginning their journey to cloud-native can take advantage of modern observability tooling. But it is the enterprises that have transformed not only their technology infrastructure, but also their processes and personnel that will deliver the most business value in the cloud-native era.

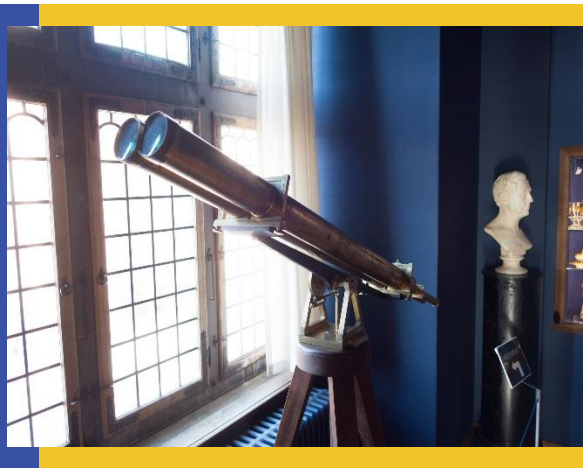
IT Infrastructure and the Cloud-Native Computing Paradigm Shift

Now well into its second decade, the advantages – and drawbacks – of the cloud are well known. Shifting the responsibility for IT infrastructure to a third party, who then exposes this underlying complexity via a relatively simple software-based abstraction, enables previously unobtainable levels of scalability and resilience.

In spite of such advantages, cloud computing presents enterprise IT leaders with a difficult choice: shift certain workloads to the cloud to achieve its benefits, while leaving others on-premises, relegating them to the limited scalability and flexibility that has always plagued IT infrastructure.

If only there were a way to extend the architectural benefits of the cloud to all of IT. What we need is a new way of thinking about software – about how to build and deploy workloads across different types of infrastructure.

Cloud-native computing goes well beyond simply leveraging Kubernetes and containers in the cloud. In reality, it means extending cloud best practices to all of IT, including on-premises tech as well as the rest of the hybrid IT landscape.



A tall order to be sure. The starting point: microservices and containers, which bring greater modularity and elasticity to application infrastructure. Containers and microservices alone, however, don't support a true cloud-like architecture. The missing piece: container orchestration, in the form of Kubernetes.

Finally, we have the building blocks of what we now call *cloud-native computing*.

Cloud-native computing, however, goes well beyond simply leveraging Kubernetes and containers in the cloud. In reality, it means extending cloud best practices to all of IT, including on-premises tech as well as the rest of the hybrid IT landscape.

Cloud-native thus comprises a broad swath of modern IT best practice. It requires a comprehensive, configuration-based software abstraction that reaches from on-premises data centers to the cloud to the edge.

Rethinking IT Operations

We must also rethink the role of IT operations. Cloud-native environments are inherently dynamic, taking advantage of ephemeral resources like containers while simultaneously supporting the continuous delivery of new software at scale.

In particular, cloud-native operations requires a revamp of the traditional dashboard-based visibility that gives traditional operators the information they need to track down and solve problems in production IT environments.

Older-generation tools and approaches are simply too slow, too incomplete, and too limited to give cloud-native operators the capabilities they require to ensure cloud-native environments operate properly.

Cloud-native operations requires a revamp of the traditional dashboard-based visibility that gives traditional operators the information they need to track down and solve problems in production IT environments.

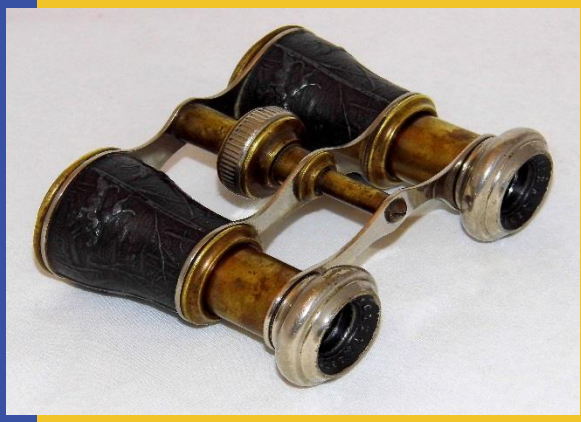
Such traditional monitoring technologies and approaches are not up to the task of providing sufficient visibility and control into inherently dynamic, ephemeral software

assets. How can you monitor something that may appear one second, scale out the next, and disappear seconds later?

The answer: operations technology must move beyond visibility to the more modern principle of *observability* that shifts the responsibility for ensuring the performance of cloud-native infrastructure to the components of that infrastructure.

This rise of observability as a core principle of cloud-native computing is far more than a terminology update. Observability brings a new context to operations in cloud-native environments.

Operations technology must move beyond visibility to the more modern principle of observability that shifts the responsibility for ensuring the performance of cloud-native infrastructure to the components of that infrastructure.



It has driven numerous open source communities to create suitable instrumentation for ensuring cloud-native tech is observable. The goal of all of these efforts is to provide insights that leverage the four pillars of observability: logging, metrics, tracing, and alerting.

These four pillars all depend upon *telemetry*: real-time data feeds from the systems and other infrastructure under observation. In fact, a core principle of cloud-native computing is that any component can only be known by the telemetry it provides. Observability, therefore, is essentially a big data challenge – and can thus take advantage of the favorable economics big data offer.

We thus have three dimensions of the challenge of operating cloud-native infrastructure. First, the shift in people, process, and technology that characterizes the cloud-native paradigm shift. Second, the rise in importance of real-time telemetry, especially as big data costs decline. Third, the shifting role of operations personnel as observability tooling gives them new levels of capability and effectiveness.

Here's a closer look.

Dimension #1: The Shift from Monoliths and Waterfall to Containers/Microservices and Parallel CI/CD

The rise of Kubernetes as a new model for ephemeral, scalable infrastructure unquestionably accounts for the lion's share of the cloud-native paradigm shift. But to say that cloud-native is all about infrastructure – or even technology in general – would be shortchanging the depth and power of the movement.

In reality, cloud-native computing represents a shift in technology, processes, and people.

The processes for building and running cloud-native, dynamic software must go through their own transformations. The entire notion of waterfall software methodologies – the idea that building software requires projects to come in phases that represent the beginning, middle, and end of each project is now woefully obsolete.

The rise of Agile and the incorporation of Lean principles were only the beginning. Today we have the new collaborative cultural transformation we call DevOps, with continuous integration (CI), continuous delivery (CD), and continuous release becoming established practices for building quality software that conforms to the cloud-native principles of ephemerality, elasticity, and scale.

Scaling the construction of software, however, is not the same thing as scaling its operation. Large organizations must implement practices for supporting multiple parallel development efforts in order to scale the development of software, while scaling its operation requires both development as well as operational best practice.

Cloud-native observability supports both such practices. For operations personnel – system reliability engineers (SREs) in particular – observability tooling provides both

insights into root causes of issues as well as the ability to mitigate and resolve such issues.

Observability tooling also serves an important role for developers and other DevOps personnel as well, a role that traditional visibility tools typically did not offer.

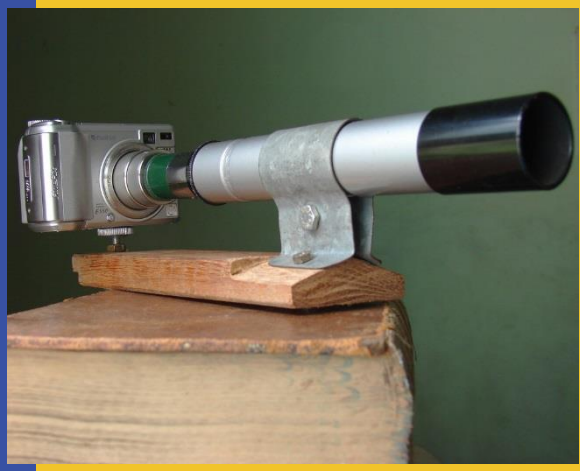
CD practices for a cloud-native infrastructure (deploying microservices into Kubernetes pods, for example) requires the developer to understand the behavior of the new software in the dynamic production environment.

In other words, the development iterations familiar from Agile now require ongoing deployments in production environments in order to understand the impact the new software has on other deployed software in a real-world context. Observability tooling is essential for this purpose.

Dimension #2: The Changing Economics of Telemetry: From Scarcity to Plenty

With the move from monolithic to distributed to fully cloud-native architectures, the ability to manage various software assets has evolved dramatically.

Where monolithic software was stingy with telemetry, today's applications and infrastructure have the luxury of being able to produce copious quantities of it. The reason: all of the costs associated with telemetry continue to fall, including storage, bandwidth, compute processing, and memory.



In the early days of software applications, no one thought to instrument those programs to support a management infrastructure. Over time, outputting logging information became common, but mostly as an afterthought. We still think of such log data as 'data exhaust' – little more than a waste product.

Today, in contrast, vendors, as well as open source projects and bespoke software efforts are building observability directly into their software by generating telemetry – the raw data that feed observability tooling.

Where monolithic software was stingy with such telemetry – assuming it generated any at all – today's applications and infrastructure have the luxury of being able to produce copious quantities of it. The reason: all of the costs associated with telemetry continue to fall, including storage, bandwidth, compute processing, and memory.

Where before we lived in a world of scarcity, where we expected and desired minimal telemetry, today we enjoy the economics of plenty. Not only does more telemetry mean better insights into the behavior of software, but such large data sets also drive better machine learning (ML), the technology at the heart of modern anomaly detection, mitigation, and resolution.

In fact, we are currently in a race to zero, as the bottom falls out of telemetry costs. Vendors like New Relic are applying substantial downward pressure on the various telemetry-associated costs, thus forcing its competitors to do likewise.

For enterprises, the plummeting costs of observability do not simply mean companies spend less. More likely than not, this trend means that organizations can take advantage of increasing amounts of telemetry, thus improving their observability overall.

Given that cloud-native operational environments are increasingly complicated as they grow more dynamic and ephemeral, this changing economics of observability couldn't have come at a more opportune time.

In addition, telemetry standards and open source technologies like OpenTelemetry also shifting the economics of telemetry, as they form an essential part of the cloud-native infrastructure story.

Finally, there isn't simply *more* telemetry than ever before. The telemetry we have is also *faster*. In fact, today's modern applications and infrastructure generate telemetry in real-time, and observability tooling works with such information in near real-time as well.

This real-time, streaming aspect of the telemetry-driven observability story transforms the work of both SREs and developers, as it leads to more iterative behavior and in the end, better customer centricity.

Dimension #3: From Reactive to Proactive and Predictive

Real-time telemetry and ML-driven insights form a powerful combination that transforms how SREs perform their daily work.

Traditional monitoring is reactive. It leads to delays in problem resolution, incomplete analysis of the problem landscape, and as a result, the inevitable war-room fingerpointing.

Modern observability, in contrast, is largely proactive. True, unexpected issues can always crop up – but improved automation enables planning ahead for problem resolution, without the war room.

ML in particular brings better predictive capabilities that can prevent many issues, especially when SREs already understand the context for such issues – what we might call the 'known unknowns.'

For example, running low on memory is a familiar situation, so even though we can't always predict when it will happen with certainty, modern observability tools can provide greater predictability into low-memory problems than older, threshold-based monitoring approaches ever could.

Regardless of an organization's ability to gain insight into the known unknowns, it must still be proactive, because ML can't predict and mitigate everything. There is always the possibility of an 'unknown unknown.'

An unknown unknown is a problem that doesn't fall into a preexisting pattern, and thus predicting it based upon past behavior is difficult, regardless of the quantity or quality of available telemetry.

We could argue that as we get better at managing the operational environment, our experience will lower the chance of such unknown unknowns – and it will to some extent, to be sure.

Keep in mind, however, that cloud-native environments are inherently dynamic and ephemeral – characteristics that can always potentially lead to unknown unknowns, regardless of the experience and expertise of the SREs.

Cloud-native observability thus shifts the focus of operations. All the known unknowns are now well-trodden ground, and observability tools like New Relic One can mitigate and even predict such issues with relative ease.

The primary focus of the SREs, therefore, becomes the unknown unknowns. Once again, cloud-native observability tooling like New Relic's is essential for addressing such issues, even though they are difficult or impossible to predict with certainty.

The Intellyx Take

Cloud-native computing may represent a paradigm shift in how enterprises leverage IT, but paradigm shifts take time. Most enterprises are somewhere along the road to cloud-native to be sure, but most still have a ways to go.

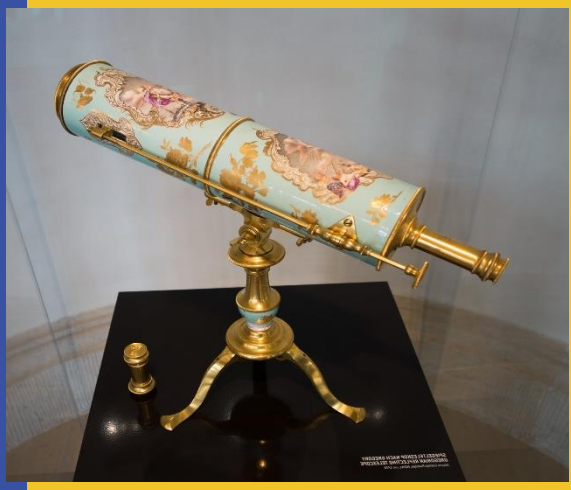
Nevertheless, there's no reason why organizations can't take advantage of modern observability tools today. To do so, enterprises must move beyond familiar dashboard-based visibility to leverage the full breadth of observability, even for systems that are not intrinsically observable.

Instrumenting the existing full stack of today's existing IT assets for observability as part of an overall modernization strategy is therefore an essential stepping stone to cloud-native computing.

New Relic can be an important part of this instrumentation. It offers a well-differentiated telemetry data platform that it has optimized for both time-series and event data, enabling real-time investigation of operational issues with both speed and flexibility at scale.

In the final analysis, remember that resolving issues in the operational environment isn't simply a technology challenge. Today's businesses (and public sector organizations) rely on modern IT to provide value to their customers and citizens. s

The move to cloud-native is subject to transformational business pressures that drive the dynamic and ephemeral nature of the underlying infrastructure. The only way organizations can ever keep up – let alone continue to focus on the business need – is by taking advantage of cloud-native observability.



Digital transformation, after all, is about aligning the organization with customer priorities. Such alignment breaks down organizational silos across the entire business, including the IT organization.

The move to cloud-native, therefore, is subject to transformational business pressures that drive the dynamic and ephemeral nature of the underlying infrastructure. The only way organizations can ever keep up – let alone continue to focus on the business need – is by taking advantage of cloud-native observability.

My final advice to operations personnel who are struggling to make sense of their organizations' road to cloud-native: look beyond the dashboard. While visibility is a one-way street, observability is interactive – and the more complex and automated production environments are, the more important such interactivity will become.

About the Author: Jason Bloomberg

Jason Bloomberg is a leading IT industry analyst, author, keynote speaker, and globally recognized expert on multiple disruptive trends in enterprise technology and digital transformation.

He is founder and president of Digital Transformation analyst firm Intellyx. He is ranked #5 on Thinkers360's Top 50 Global Thought Leaders and Influencers on Cloud Computing for 2020, among the top low-code analysts on the Influencer50 Low-Code50 Study for 2019, #5 on Onalytica's list of top Digital Transformation influencers for 2018, and #15 on Jax's list of top DevOps influencers for 2017.

Mr. Bloomberg is the author or coauthor of five books, including *Low-Code for Dummies*, published in October 2019.

About New Relic

The world's best engineering teams rely on New Relic to visualize, analyze and troubleshoot their software. New Relic One is the most powerful cloud-based observability platform built to help companies create more perfect software. Learn why customers trust New Relic for improved uptime and performance, greater scale and efficiency, and accelerated time to market at newrelic.com.