IBM Instana

Leverage automated APM to accelerate CI/CD and boost application performance

Contents

01

Executive summary

Continuous integration and continuous delivery (CI/CD) has become the goal for applications at a majority of organizations. Meanwhile, modern technologies such as Docker and Kubernetes have become preferred application environments. Applications and their infrastructure are therefore becoming increasingly dynamic, constantly changing to meet scalability requirements and fast-evolving functionality.

Unfortunately, the typical organization still relies on performance monitoring technology designed before CI/CD became the ubiquitous application delivery model. Older monitoring solutions require significant manual effort from IT staff. This not only wastes precious staff time but also increases costs without delivering meaningful visibility into applications and infrastructure.

Adopting an automated performance management strategy is the only practical way organizations can break through application challenges to capitalize on CI/CD-powered application environments.

02

Change is the new normal

Everyone knows we live in a fast-changing world. For IT organizations, however, the fast-changing nature of application environments is extreme. The overwhelming mandate is clearly a bias for speed. The faster a company can deliver custom business applications, the more value IT is delivering. This simple reality drives the adoption of new techniques and technologies, particularly CI/CD to achieve faster delivery, better business capability and improve the quality for custom applications.

The final stage in the CI/CD cycle is monitoring. Automated, seamless application monitoring helps complete the CI/CD cycle and restart the loop of improving the application. The structure of the application changes continuously at every layer. New hosts come online and disappear constantly with containers making provisioning even more dynamic.

Developers continuously build and provision completely new APIs and services without checking with operations. Even the application code can change because of improvements or bug fixes at any moment. The closer a team gets to CI/CD, the more frequently these changes occur. That’s why performance management configuration, monitoring dashboards, dependency mappings and alerting rules must evolve automatically to keep pace with the environment they’re monitoring. Otherwise, IT teams lack accurate visibility into the environments they manage, leaving the organization at great risk of failure that could impact users.

Complex dependencies
Constant changes in how and where applications run can impact the actual dependencies among different components. Any specific service depends on a unique vertical stack of software along with data or processing from other services.

Why is it important to always understand dependencies? It's the basis for faster troubleshooting. Finding the root cause of an issue in complex environments requires in-depth dependency analysis. What's causing slow or erroneous requests? Requests span many services across many frameworks and infrastructures, so you need to know the structural dependencies of every request to answer that question. But as we know, dependencies constantly change.

Manual interpretation of dependencies is simply not feasible—especially when dependencies change so quickly. Even if operators succeed in mapping dependencies at a moment in time, their mappings will quickly become outdated. What's more, manual dependency interpretation is a huge resource drain that engages your best engineers.

The rise of dynamic applications
Faster application development requires modern technology for rapid construction and delivery of new services.

In particular, the CI/CD pipeline is a primary process for supporting speed and quality. In addition, to augment that process, new architectures such as containers, microservices, serverless computing and Kubernetes orchestration are being considered and adopted.

With more workloads moving to dynamic technologies—eight billion pulls on the Docker hub in a month and 130 billion total pulls since the hub was launched in 2014[1]—there are new realities for the pace and scale of change within application environments.

The IT industry is consolidating around CI/CD terminology and methodologies. Meanwhile, public cloud platforms now offer Kubernetes-managed container processing as a service that is easily integrated into CI/CD pipelines. Clearly, there are major changes taking place in the world of application construction and delivery.

03

Why manual performance management fails

The challenges of manual performance monitoring (APM)
Because traditional monitoring tools weren't designed for dynamic applications, a new set of open-source technologies has emerged to help development teams manually set up their own monitoring through manual coding. Whether providing performance metrics, tracing paths of an application or exposing other details of code, these open-source monitoring tools create their own set of challenges when it comes to production performance monitoring.

To manually monitor the performance of an application, engineers must complete a long list of tasks.

- Write data collectors
- Perform manual code tracing to track distributed requests
- Configure a data repository
- Identify and designate dependencies, usually through reverse engineering
- Select data to correlate
- Build dashboards to visualize correlation
- Configure alerting rules and threshold

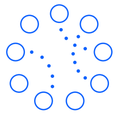
The biggest problem is that the people capable of performing these tasks are usually the highest skilled—and highest paid—technicians and engineers in the company.

In short, although open source implies simplified performance management, too many manual tasks are required, resulting in deployment slowdowns, cost increases and additional labor requirements. And that's before you look at the opportunity costs of relegating your precious developers to writing monitoring code instead of creating business application code.

The true effort of building your own
Historically, manually setting up a monitoring system wasn't a problem because none of the components—application code nor, application infrastructure middleware, app servers and so on—changed very often. IT would provision a box, set its IP address, load some software, set up the monitoring and then never touch it again for years.

A similar situation applied to application environments built on traditional technologies. The number of application instances and host servers running at any given moment typically didn't change. As a result, manual configuration of the tech stack in static environments worked fine with little impact on the organization's ability to monitor and manage performance.

But when workloads move into dynamic environments based on technologies such as containers and on methodologies such as CI/CD, manual performance management strategies and build-it-yourself solutions simply can't keep up. This is true for several reasons.



Rule deterioration
When your environment changes constantly, the rules that your monitoring tools use to determine the health of applications and services need to change continuously as well. If they don't—which is likely to happen if you depend on manual intervention to update rules—the rules will quickly deteriorate.

For example, when a new service is deployed or an orchestrator moves workloads, health check rules will cease to accurately interpret environment dependencies until the rules are updated. If rules aren't updated manually, monitoring alerts and insights will be based on outdated configurations. This deterioration undercuts visibility and increases the risk of an infrastructure or service failure.

Manual monitoring impedes speed
Tasks such as creating tracing and monitoring code are too time consuming. So is interpreting monitoring information by hand and manually updating performance management rules when application code or environment architectures change.

Simply put, humans can't support rapidly changing environments without automation. Attempting to manage performance manually will significantly slow down application release cycles. And for the business, it can mean incorrect use of the expensive expertise that your IT staff represents.

Implementing automated APM

If manual monitoring is so laborious, then why hasn't every organization automated monitoring yet? There are actually three reasons for this.

1. Until recently, fully automatic monitoring technology did not exist.
2. Software engineers tend to code their own solutions to problems, leading to the monitoring-as-code approach, which can be functional in the short term but is not maintainable or scalable.
3. Teams try to use previous investments in existing monitoring tools, hoping they'll work in their new high-speed environments. But unfortunately, they don't.

With these reasons in mind, let's take a look at the capabilities that should be part of a fully automated APM solution.

- **Automatic discovery and monitoring** of the full infrastructure and application stack. This is a key element of the CI/CD process that is missing today and slowing down the release process.
- **Real-time complex dependency mapping,** with automatic updates for any change. This is critical for root cause analysis to address performance issues quickly.
- **Automatic, rapid identification of performance problems.** Quickly identifying performance issues based on automatic rule configuration and monitoring is the only way to minimize false positives and avoid application delivery delays.
- **Rule configuration and alerting that use machine learning and AI** to establish dynamic baselines for healthy application behavior, then identify anomalies based on those baselines. This eliminates the need for tedious configuration, monitoring and data interpretation by humans while minimizing false-positive alerts.
- **Automatic monitoring setup** which includes agent deployment, code instrumentation, infrastructure discovery and more. Maximize the data collected while minimizing the effort required from human administrators to collect it.

Both existing performance management tools and modern open source monitoring tools lack this automation functionality. Furthermore, both sets of tools require too much manual configuration, programming, setup and administration to optimize monitoring.

05

Conclusion: Maximizing CI/CD agility

Is your CI/CD as agile as it could be? Successful management of dynamic application environments doesn't end with automated monitoring.

Your IT teams should continually assess how well they have automated all performance management tasks by asking themselves the following questions:

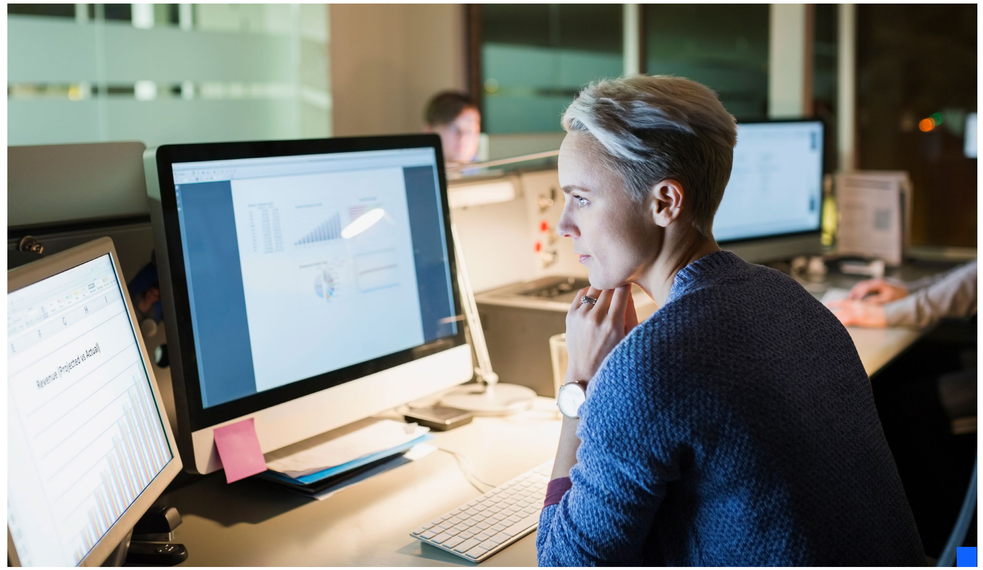
- How long does it take to achieve sufficient visibility into application performance after we push out a new release?
- How long does it take to update monitoring rules when a new application or service deployment occurs?
- How much time and effort do our developers expend writing tracing code?
- How many performance or availability incidents are we missing per month or quarter?
- How are we handling alert storms—those rapid streams of alerts in a short period? Are we able to respond to each alert effectively without suffering from alert fatigue? Can we trace alerts quickly to root causes so that we know when multiple alerts are stemming from the same underlying issue?
- Is our monitoring and performance management process as automated as the rest of the application delivery pipeline? If not, how can we automate it further?

Regardless of your IT team size, automation is critical for an effective performance management strategy so that you can achieve high-speed delivery of new business services.

IBM Instana™, the automated performance management solution born in the age of microservices, cloud computing and containers can help you truly deliver on the promise of CI/CD. Designed specifically to manage the performance of dynamic CI/CD-driven application environments, IBM Instana uses AI and automation to deliver comprehensive, actionable insight with no manual effort.

06

Is IBM Instana right for you?



IBM Instana™ is an enterprise observability platform that includes automated application performance monitoring capabilities. It's designed for businesses operating complex, modern, cloud-native applications no matter where they reside—on premises, in public and private clouds, on mobile devices or in an IBM zSystems™ environment.

IBM Instana helps you control modern hybrid applications with AI-powered discovery of deep contextual dependencies inside hybrid applications. IBM Instana also provides visibility into development pipelines to help enable closed-loop DevOps automation.

These capabilities provide actionable feedback needed for clients as they optimize application performance, enable innovation and mitigate risk. These features help DevOps increase efficiency and add value to software delivery pipelines so they can meet their service and business-level objectives.

See the power of IBM Instana for yourself. Sign up today for a free 14-day trial of the full version of the product. No credit card required.

IBM Instana free trial →

Explore IBM Instana →

1. Team DevClass, DevClass.com, “Docker knits together Hub stats, says Pulls over 8 billion”, 5 February 2020.

IBM Corporation
New Orchard Road
Armonk, NY 10504

Produced in the United States of America
June 2023