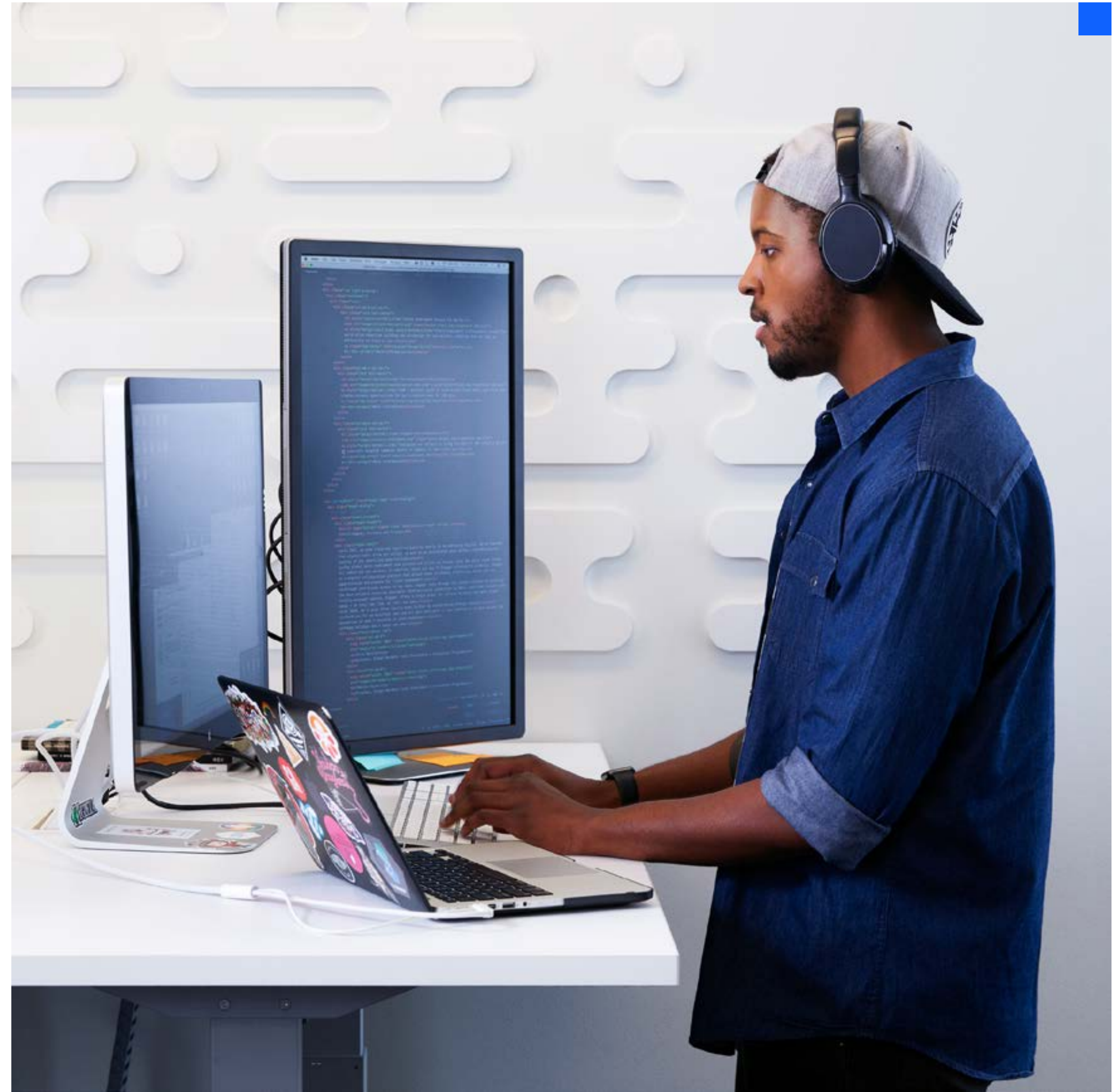


The power of enterprise observability in pre-production testing



Contents

01 →
Introduction

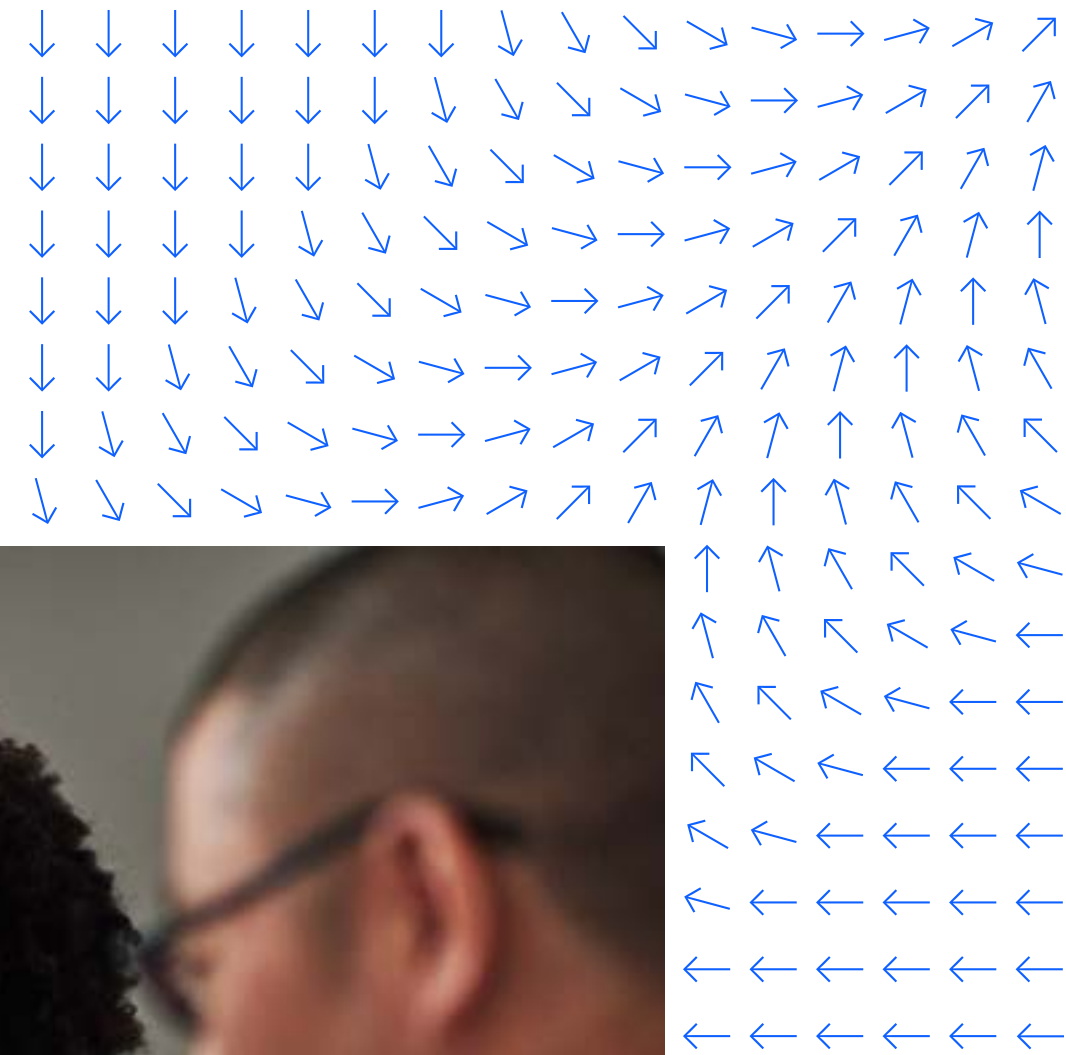
02 →
Traditional
pre-production methods

03 →
Introducing intricacy
to the testing process

04 →
Enterprise observability
is the answer

05 →
Enhance your
testing process

06 →
Why IBM Instana?



Introduction



The complexity of application testing

Application monitoring exists to prevent software issues from affecting customers. Today most applications are built using microservices to be more scalable, agile, and resilient, but despite their benefits, microservices require management and oversight. With a microservices architecture, each service operates independently from the others. Managing the entire system can be challenging.

One example: the need to ensure the availability and optimal performance of all the constituent services within a microservices application. To solve this challenge, IT teams are deploying comprehensive monitoring and management to help prevent outages and failures that could occur in deploying software.

Traditional application performance monitoring (APM)

Traditional APM tools focus on infrastructure monitoring, application dependencies, business transactions and user experience, and were designed to quickly identify, isolate and solve performance problems.

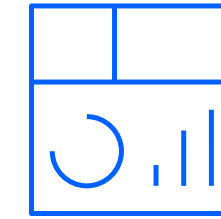
But as agile development, DevOps, multiple programming languages, new cloud-native technologies, and microservices-based applications have increased, traditional APM systems can no longer provide visibility into complex technologies.

From APM to observability

Observability is the logical evolution from traditional APM to meet the increasingly rapid, distributed and dynamic nature of cloud-native application deployments. Enterprise observability platforms perform a critical function for modern software development.

Through IBM Instana™, IBM brings a fully automated enterprise observability platform that delivers the context needed to take intelligent actions and ensure optimum application performance and deployments.

Traditional pre-production testing methods



Throughout the development and deployment process, teams have many testing options at their disposal. Here are the most common methods:

Scenario tests

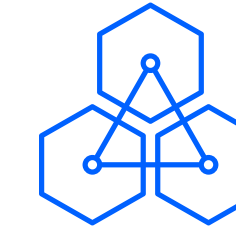
The most basic method is scenario testing, in which applications perform or receive transactions, so teams can get a performance baseline, and then test repeatedly to compare results to the baseline. This method allows for finding coding errors and agile corrections.



Validation tests

A second set of tests involves validation tests, which focus on individual application components and range in complexity approaching end-to-end tests. Here are some common tests:

- **Unit testing:** Identify the smallest application component you can test to determine whether it behaves the way should by itself, not with other components or its environment.
- **Integration testing:** Used to test basic success and error paths over a network boundary. These tests validate that the communication paths between subsystems work.
- **Component testing:** Test the behavior and resilience of one microservice. You can mimic any calls to external services, but you're not testing the microservice's interaction with the infrastructure or other services.
- **Contract testing:** An integrated contract test verifies whether APIs and other resources provided by a microservice are behaving as expected. A test suite is limited in scope to only the service that is in use. As changes are introduced to the microservice, this test reveals their impact on other services.



Manual fault injection

A third set of tests involves fault injection. Errors are added manually to a part of the application. You could add a path change in a Kubernetes cluster that causes a 503 error when it tries to orchestrate containers, or you could increase CPU usage. You can vary the duration of the errors or adjust other parameters.

Introducing intricacy to the testing process



Scriptless tests

Scripted tests, validation tests, and traditional fault injection all follow predictable patterns. Microservices-based applications use independent application components hosted in containers. With Kubernetes or another container orchestrator, containers can suddenly appear and blink out of existence automatically as conditions change.

The result is incredible complexity, leading to some more sophisticated tests.

Unscripted tests and AI/ML

Unscripted tests randomize transactions for the application to perform and upstream activities that can affect it. These tests can also be used to test the response of site reliability engineers (SREs) and their teams, which can also encourage more testable code.

Scriptless tests don't have to be purely randomized either. Using AI/ML, organizations can run new tests based on current conditions, performance against previous tests, changes introduced manually, or other parameters.

Enterprise observability is the answer

Automation and context for correlation

When monitoring tools can't deliver the meaningful results SREs need to optimize their applications, it's time to apply a new approach. Enterprise observability separates from traditional APM tools in three crucial ways: automation, context, and actionable intelligence.

- **Automatic discovery:** Immediate, automatic discovery of all components across a company's environment and extended ecosystem. Automated discovery saves time spent on manual configuration, and it greatly limits the risk of missing components or entire parts of the architecture.

- **Automated context:** When you receive an alert, it's one thing to understand the simple context. Knowing which part of your environment is failing separates enterprise observability from traditional APM, which generally only tells you whether the application is performing well or not.

- **Correlation:** Sometimes an application fails when nothing is wrong with the application. There could be an infrastructure problem or other services further upstream required by the application to run. Often, APM tools also miss which downstream services could be degraded or even fail because of the application's failure. Without correlation capabilities, your

teams could burn valuable time trying to find dependencies—or worse, miss dependencies completely. Context and correlation inform the suggested remediation actions that enterprise observability solutions produce.

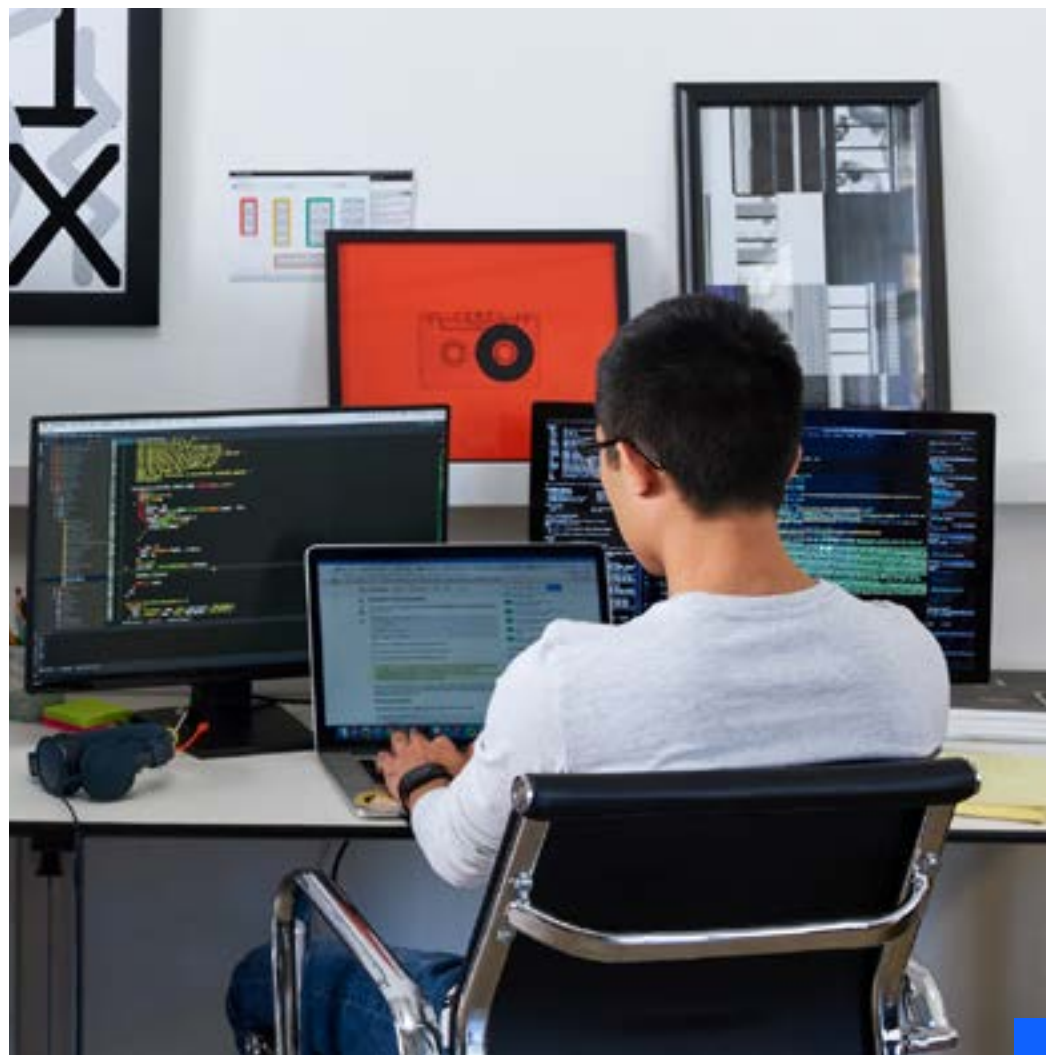
Enterprise observability in pre-production

Automated testing during the development cycle is standard, but scripted fault injection tests are no longer good enough. Organizations need to do something different to effectively test microservices-based applications to prepare for application deployment. Artificial Intelligence (AI) can provide automated unscripted tests that help to ensure greater resilience for applications before they hit production.

Enhance your testing process

Setting up a series of automated tests in which each test repeatedly calls a microservice enables a baseline of expected performance to compare results. The narrow scope limits the value of the results. Here are some ideas to improve testing to produce more valuable content:

- **Test different components of the application:** Run new test scripts each time a microservice refreshes and compare the outputs of the new code with previous outputs.



- **Test in cloud-based environments:** Cloud platforms free up resources by allocating resources dynamically only as tests need them. For many organizations, cloud-based testing in various environments also creates a more realistic result since they are using software as a service (SaaS) on a platform on a service (PaaS).
- **Vary your tests:** Apply different scripts and environments, as well as web browsers for web applications, to test code. Apply different usage patterns and test in various geographies. You can use Kubernetes to orchestrate new test environments. However, diagnosis of new issues becomes more challenging.

The most complex tests and the most challenging methodologies are often most closely aligned with actual production environments. Given this, how can organizations achieve the best results from tests? The answer lies in enterprise observability.

IBM Instana advantages in pre-production

IBM Instana™ adds functionality to make testing work across the application spectrum. To make pre-production testing effective for launching new products and features, the same enterprise observability functionality is extremely valuable:

Automated discovery: IBM Instana automatically discovers every application and infrastructure component the very moment it is installed. You can start benchmarking and comparing all application components, nodes, containers, and architectural components almost instantly.

Architecture monitoring: By monitoring your architecture and your applications, IBM Instana gives a better view of the impact of your applications on your architectural components, and the effects of your architecture on your applications. Seeing upstream and downstream effects enables you to identify the root cause of issues very quickly, limiting time spent on triage and issue resolution.

No sampling ever: The whole point of testing is to verify the performance of applications every time they act. Legacy applications spot-check transactions and sample only elements of traces. IBM Instana never samples, so it delivers an enhanced version of the same metrics.

Request tracing: During end-to-end traces, IBM Instana traces every request through all the systems it moves through. This tracing is automated, freeing up time for your developers. Following traces through systems and architectures unveils changes under more circumstances and helps your teams find and fix more issues. By delivering full context around each trace, IBM Instana gives you the information you need to triage more quickly and create more resilient applications before deploying to production.

One-second granularity: A new infrastructure snapshot every second ensures up-to-date measurements every time and doesn't miss changes that might take place during longer times between measurements.

Why IBM Instana?



With DevOps, SRE, Platform, ITOps, and Developer teams under pressure to keep applications performant while operating faster and smarter than ever, observability must be easy, simple, and transcend technology stacks to achieve the modernization the business demands.

To operate even more quickly and proactively, these teams need observability that is powered by AI and operating on precise high-fidelity data—no sampling, which in the cloud native world can miss critical anomalies that impact end-users.

IBM Instana democratizes observability by providing a solution that anyone across DevOps, SRE, Platform, ITOps, and Development can use to get the data they want with the context they need.

IBM Instana offers simple and predictable pricing, ensures no lock-in due to intense upfront setups, and has no limits on what you and your teams observe. Advanced skill sets are not required and getting started takes minutes.

Why IBM Instana?

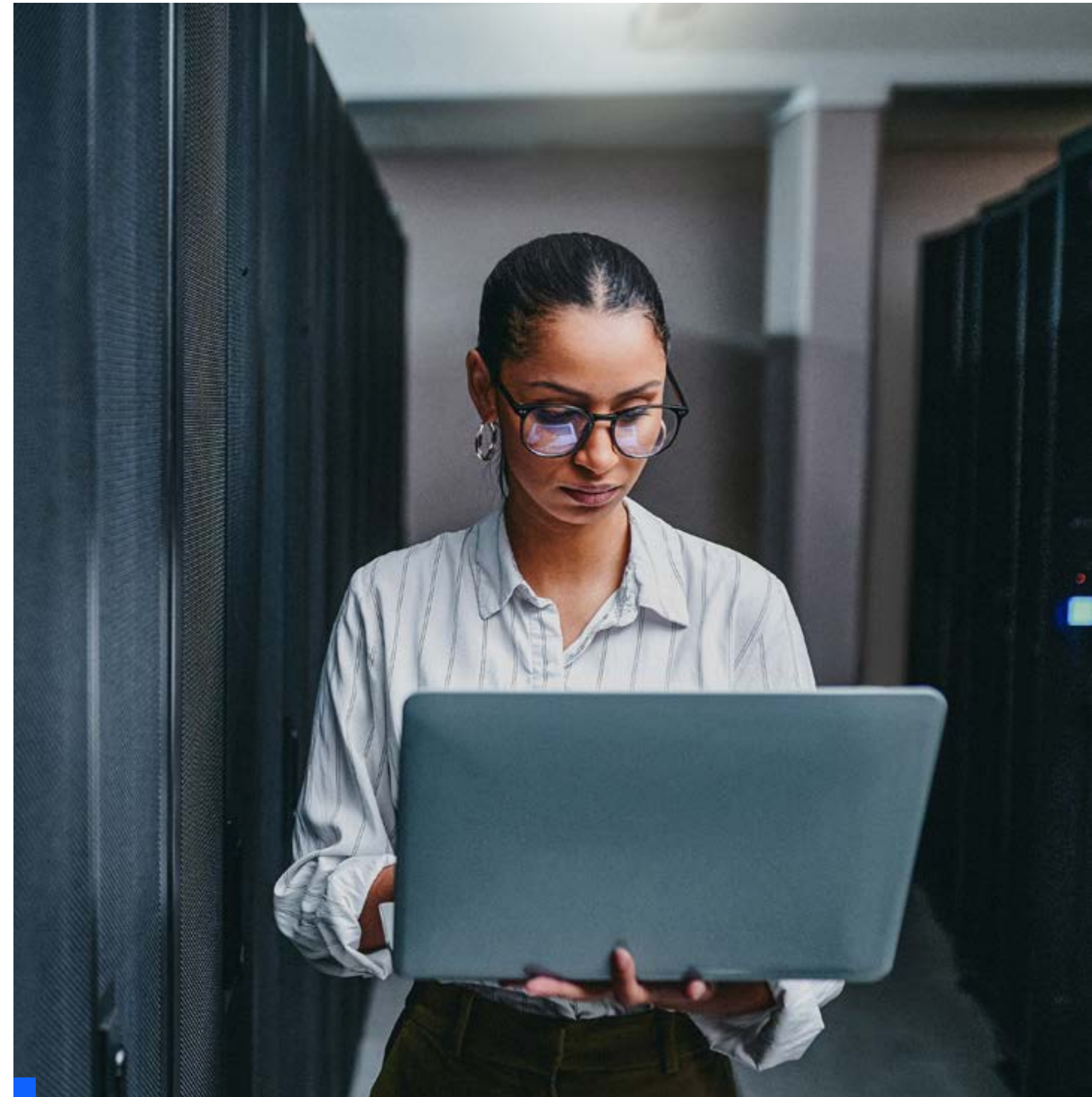
Purpose-built for cloud native yet technology-agnostic, this solution automatically and continuously provides high-fidelity data—1-second granularity and end-to-end traces—with the context of logical and physical dependencies across mobile, web, applications, and infrastructure.

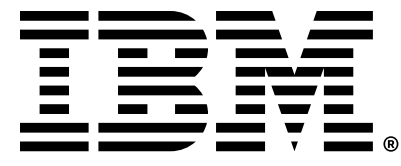
This approach provides immediate feedback on new, more frequent deployments, ensuring issues are identified before they become incidents, while configurable smart alerts further enable proactive performance management.

IBM Instana™ provides real-time observability for your organization's data, to all your teams, with the context you need. It delivers quick time to value while ensuring your observability strategy can keep up with the dynamic complexity of today's environments and tomorrow's. From mobile to mainframe, IBM Instana supports over 250 technologies and growing.

[Explore IBM Instana](#) →

[Try IBM Instana](#) →





© Copyright IBM Corporation 2023

IBM Corporation
New Orchard Road
Armonk, NY 10504

Produced in the United States of America
May 2023

IBM, the IBM logo, IBM Instana and IBM Z are trademarks or registered trademarks of International Business Machines Corporation, in the United States and/or other countries. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on ibm.com/trademark.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

It is the user's responsibility to evaluate and verify the operation of any other products or programs with IBM products and programs. THE INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.