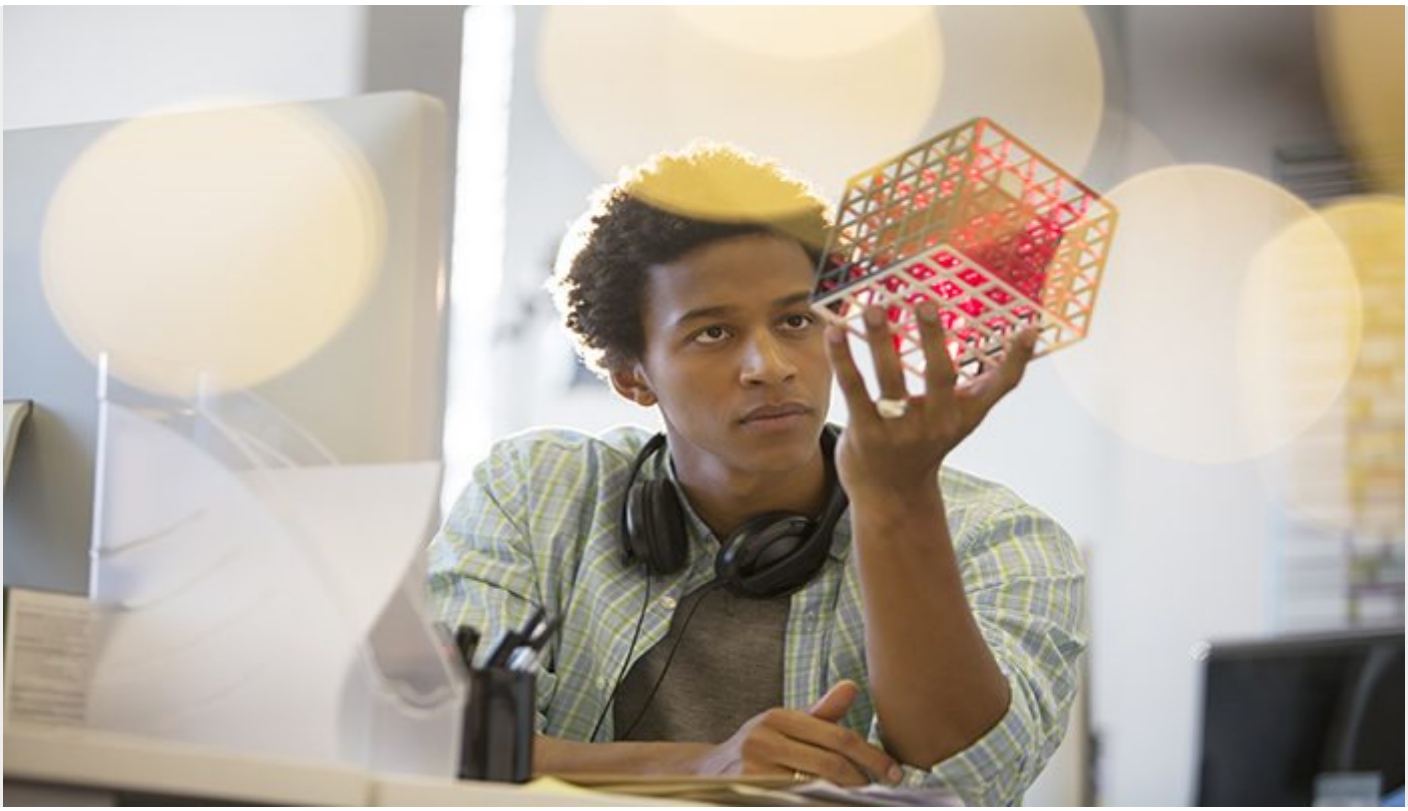
OBSERVABILITY VS MONITORING: WHAT'S THE DIFFERENCE?



Enterprise IT and software-driven consumer product development are increasingly complex. The internet delivers IT infrastructure services from vast data centers at distant geographic locations. Companies consume these services as distributed functions like microservices and containers, across layers of infrastructure and platform services. Consumers expect rapid feature improvements through new releases via the internet.

To meet these end-user requirements, IT service providers and business organizations must streamline performance and improve stability and predictability of backend IT infrastructure operations—amid the inherent complexity of the IT systems. To do so, we closely observe and monitor metrics and datasets related to infrastructure performance in order to optimize system dependability.

These days, observability might seem like a buzzword; in fact, this traditional concept drives monitoring processes. Both the system observability and monitoring play critical roles in achieving system dependability—but they're not the same thing. Let's understand the differences between observability and monitoring, and how they are both critical to visibility and control in cloud-based enterprise IT operations.

What is observability?

Observability is the ability to infer internal states of a system based on the system's external outputs. In control theory, observability is a mathematical dual (follows a direct conceptual mapping) to

controllability, which is the ability to control internal states of a system by manipulating external inputs. In practice, however, controllability is difficult to evaluate mathematically; therefore, system observability is the method for evaluating outputs to reach meaningful conclusions about internal states of the system.

In enterprise IT, distributed infrastructure components operate through multiple abstraction layers of software and virtualization. This environment makes it impractical and challenging to analyze and compute system controllability.

Instead, common practice is to observe and monitor infrastructure performance logs and metrics to understand the performance of individual hardware components and systems. Advanced log analytics and AI (AIOps) evaluate incidents and events related to hardware performance in order to predict potential impact on system dependability. Then, your IT team can proactively adopt corrective measures to reduce the impact on end-users.

What is monitoring?

Observability is the ability to infer a system's internal states. Monitoring, then, is defined as the actions involved in observability: observing the quality of system performance over a time duration. The monitoring action, which tools and processes support, can describe the performance, health, and relevant characteristics of a system's internal states. In enterprise IT, monitoring refers specifically to the process of translating infrastructure log metrics data into meaningful and actionable insights.

A system's observability property includes how well the infrastructure log metrics can infer the performance characteristics associated with infernal components. Monitoring tools analyze the infrastructure log metrics to deliver actions and insights.

Comparing observability and monitoring

Let's use an example of a large, complex data center's infrastructure system that's monitored using log analysis and monitoring and ITSM tools. Analyzing too many data points continuously will generate volumes of unnecessary alerts, data, and false flags. The infrastructure may present low observability characteristics, unless the correct metrics are evaluated and the unnecessary noise is carefully filtered using AI-based infrastructure monitoring solutions.

On the other hand, a single server machine can be easily monitored using metrics and parameters such as hardware energy consumption, temperature, data transfer rates, and processing speed. These parameters are highly correlated with the health of internal system components. Therefore, the system has demonstrated high observability. Using basic monitoring tools, such as energy and temperature measurement instruments, or software-based monitoring tools, the performance, life expectancy, and risk of potential performance incidents can be evaluated proactively.

The observability of a system depends on the system's simplicity, the insightful representation of the performance metrics, and the capability of the monitoring tools to identify the correct metrics. This combination yields the necessary insights to illustrate an accurate representation of the internal states, despite a system's inherent complexity.

Observability in DevOps

The concept of observability is prominent in DevOps software development lifecycle (SDLC) methodologies. In earlier waterfall and agile frameworks, developers built new features and product lines while separate testing and operations teams tested for software dependability. This siloed approach meant that infrastructure operations and monitoring activities were beyond development's scope. Projects were developed for success and not for failure: debuggability of the code was rarely a primary consideration. Infrastructure dependencies and application semantics were not adequately understood by the developers. Therefore, apps and services were built with low inherent dependability. Monitoring failed to yield sufficient information about the known-unknowns, let alone the unknown-unknowns, of distributed infrastructure systems.

The prevalence of DevOps has transformed SDLC. Monitoring goals are no longer limited to collecting and processing log data, metrics, and distributed event traces; monitoring is now used to make the system more observable. The scope of observability therefore encompasses the development segment and is facilitated by people, processes, and technologies operating across the SDLC pipeline.

Collaboration among cross-functional Devs, ITOps, and QA personnel is critical when designing a dependable system. Communication and feedback between developers and operations teams is necessary to achieve observability targets of the system that will help QA yield correct and insightful monitoring during the testing phase. As a result, DevOps teams can test systems and solutions for true real-world performance. Continuous iteration based on performance feedback can further enhance IT's ability to identify potential issues in the systems before the impact reaches end-users.

In many ways, observability has a strong human element, similar to DevOps: it's not limited to technologies but also covers the approach, organizational culture, and priorities in reaching appropriate observability targets, and hence, value of monitoring initiatives.

Additional Resources

Observability for modern applications from **Boaz Ziniman** and **MoovingON**