A Broadcom Company
technologies

White Paper

A Broadcom® Point of View

How to Effectively Manage and Monitor Cloud Native Applications with AIOps from Broadcom

Executive Summary

With enterprises now adopting cloud native applications in production, IT teams are being faced with a variety of new monitoring challenges. In order to effectively monitor these highly complex, modern applications, organizations need a monitoring solution that can provide full-stack observability into these dynamic environments combined with the actionable insights to help resolve problems quickly before the customer experience suffers.

What is Cloud Native?

Before we dive into what it takes to monitor cloud native applications, it's best to first understand what cloud native actually means. According to the Cloud Native Computing Foundation (CNCF), the definition of cloud native is as follows:

"Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds."

There's one more interesting part of the definition – "*Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.*"

So, another way to recognize a cloud native application is not just that it runs on a cloud platform, but rather that the application components are containerized and most likely orchestrated with some form of Kubernetes (K8S). Another interesting aspect of clouds, platforms-as-a-service, containers, orchestration and even serverless runtimes is that the lines are blurring between hardware infrastructure, software infrastructure (or platforms), and applications.

The Rise of Cloud Native Application Adoption

When Software-as-a-Service and cloud-based solutions began to dominate the technology landscape, not everyone jumped on the bandwagon. For a myriad of reasons, enterprise organizations resisted the urge and migration to cloud-based infrastructure, applications, and business systems. But it was only a matter of time before the forays of enterprise cloud applications began—enabling faster time to market, improved flexibility, increased scalability and resilience, and better operational cost models. Now, enterprises are more than just trying out cloud technologies, they are actually running cloud-native applications in production. In fact, according to a 2019 survey conducted by the CNCF, 78% of enterprises are running Kubernetes in production—a large jump from just a year prior.

Whether trying to achieve the next level of digital transformation or creating standardized deployment and operating processes that bring economies of scale to dozens or hundreds of applications, enterprises are more widely adopting and deploying these modern technologies—but with the benefits also comes challenges.

The Challenges of Managing Cloud Native Application Performance

Ultimately, the pains that usually send Operations teams looking for monitoring solutions are around gaining visibility into some kind of application black box. Containers and orchestration fall right into this ideal—with levels of virtualization and change control that require a new level of visibility.

What does this mean for teams responsible for ensuring the availability and performance of these modern applications? Like enterprise applications built on classic infrastructure and platforms, there are three questions that must be answered to have control over any application environment:

1. What technologies make up the applications and infrastructure?
2. How can we get the most comprehensive performance and health data about each entity?
3. How can we make sense of the data to answer critical questions about the application services?

Whether managing a Java EE or cloud-native orchestrated microservice application, these three questions must be answered in order to deliver properly performing, scalable applications.

AIOps from Broadcom® uniquely answers these questions with the ability to correlate data across users, applications, infrastructure and network services; then apply machine learning, advanced analytics, and automation to deliver a new level of visibility and actionable insights into today's complex enterprise environments. Built on an open, scalable data lake, the solution generates actionable, predictive insights by ingesting and analyzing diverse data sets, including metric, topology, text, and log data.

In the following sections, we will examine the capabilities of the AIOps solution from Broadcom that help answer the key questions of cloud native application monitoring.

Part One – What Makes Up the Application?

There are several key differences between traditional (even SOA based) applications and today's modern versions, and these differences make a big difference when it comes to all three of the key questions.

Where classic applications are written on a single platform, usually tied to a specific programming language (most commonly Java or .Net/C#), cloud-native microservice applications are more typically polyglot, technically meaning that more than one language is used.

But the concept of polyglot actually extends to almost every piece of the application and infrastructure: multiple cloud platforms, web servers, messaging, caching, databases, storage, security, and many more. And as cloud applications evolve, they become even more diverse, with each developer selecting infrastructure, platforms and code based on the needs for their specific service.

The world of database servers offers a microcosm of this issue. At the hands of each developer, they can select one of many classic relational databases, or a cloud version of relational databases (and every cloud has one), a non-relational (or NoSQL) database, an objects database, a database of secrets, and a multitude of other specialty platforms. And every developer selects exactly what they need to deliver their service.

For monitoring, handling this polyglot of platforms requires a methodology that has the ability to discover all the entities that are involved in applications and gather the information about each entity's configuration. And just as important, understand how the entities are related to all the other entities.

With disparate technologies, data sets, dependencies and actions, IT teams are quickly overwhelmed with both the sheer amount of data, as well as critical gaps within the system. Even worse, when examining a single entity or piece of data, there's no context as to how the object, service, or network device fits within the broader scope—nor what the data, metric, or trace actually means.

How a Unified Data Model Solves the Problem

Our AIOps solution is built upon a unified data model which enables teams to gain complete visibility across modern application environments. This open, extensible, ontology-agnostic model allows teams to collect, group, correlate, and visualize more complex performance conditions spanning applications, infrastructure, and networks.

The data model acts as the glue that connects all the pieces of the monitoring puzzle, providing the following information:

- A topology map of all components and their relationships to each other
- Configuration information about each component

Our understanding of topology is based off of our years of domain expertise across application, infrastructure and network monitoring. So our solution is able to provide a visualization of the application infrastructure and service architecture without requiring the Ops team to possess any prior knowledge of the application infrastructure or architecture. The following figure shows one way in which our solution utilizes this data in the context of the performance of a Kubernetes application.

Figure 1: Example of Performance for a Kubernetes Application

These dynamic dashboards automatically update to display metrics related to the specific cluster selected—helping IT teams understand how performance problems impact the larger environment.

Ultimately, the unified data model acts as the bridge to help fill in data and dependency gaps which is traditionally siloed and delivers contextual visibility across modern, distributed architectures where there was blindness before.

Part 2 – Obtaining Application Performance Data

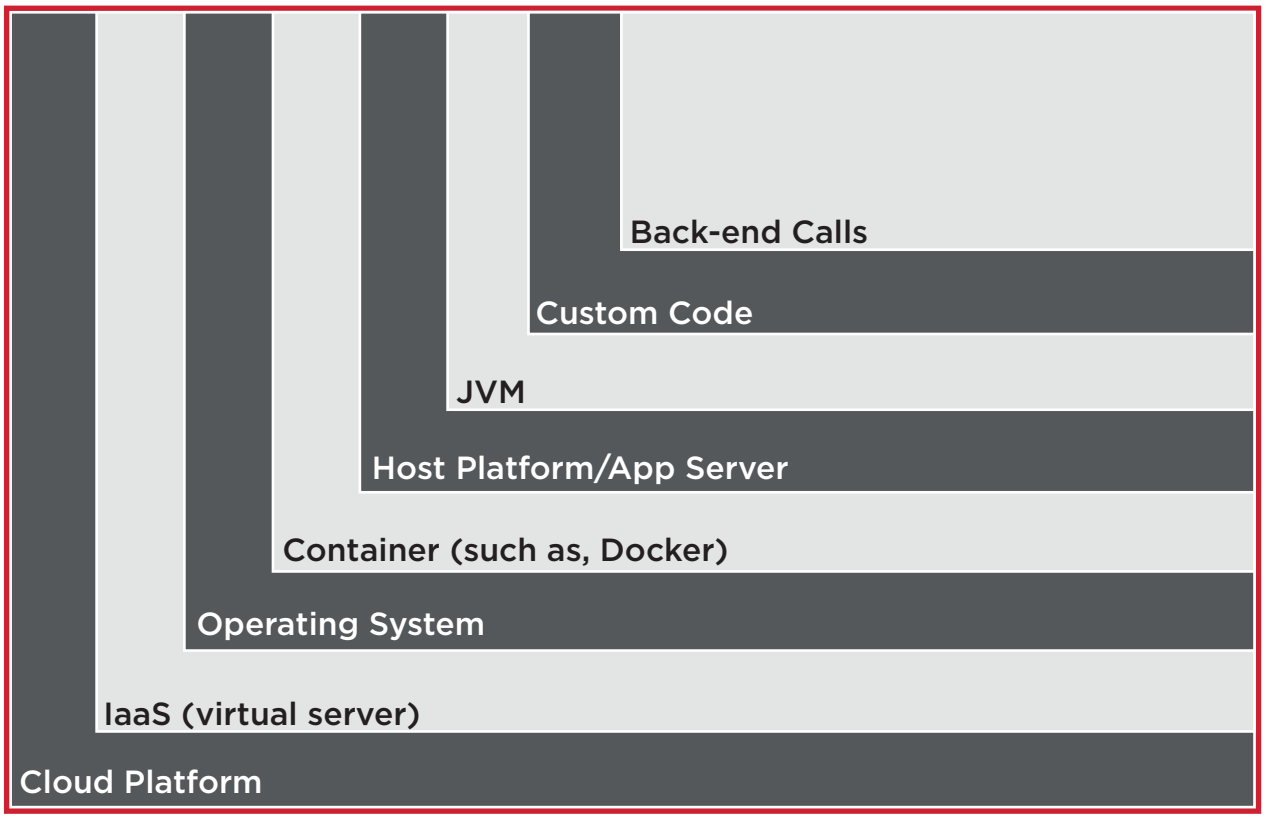
As with discovery and architecture, monitoring a poly-polyglot application is loaded with challenges that begins with the sheer number of platform types, platform vendors, data types and application services. Monitoring each component requires two, and sometimes three levels of detail. This detail includes platform (or infrastructure) data and service performance, as it relates to applications.

Monitoring Kubernetes and Orchestrated Containers

Remember that the definition of cloud native applications includes the statement that many of these applications make use of microservices, containers and orchestration like Kubernetes. Kubernetes is an open-source container orchestration system, essentially acting as an Operating System for containers. Containers, in turn, are like little specialized virtual machines with just enough resources to get by.

Thus, a microservice service is actually a complex stack on its own:

Figure 2: Kubernetes Container Orchestration



Monitoring this single microservice component requires visibility at nine different levels of technology, understanding how each level, and its respective data, is correlated with each of the other levels of the stack —and then how each service works with other services to serve up applications and execute user requests.

What About Observability?

The big question as organizations shift from custom-code centric applications to specialized microservices is just how you can get the application performance we've all come to expect.

That has led to the rise of a new concept, especially for containers and orchestration—observability. The easiest way to think about observability versus monitoring is that observability is the art of making an application component expose the pieces of performance it needs to for monitoring tools, while monitoring tools use what's observable to visualize and model the performance of the applications.

Observability can be a big part of cloud native applications as many developers and Dev teams use open source protocols like OpenTracing or Jaeger to build visibility into their individual code pieces.

Automatically Gain Observability with Universal Monitoring Agent

In order to simplify the process to gaining insights into these modern, containerized applications, our solution now utilizes a Universal Monitoring Agent which acts as single deployment that automatically discovers and monitors cloud and container infrastructures and containerized application processes. With Universal Monitoring Agent, Operations teams can easily capture data from multiple sources including OpenTracing applications, Zipkin, Prometheus, Istio, and more —and pull it into our unified data lake for further analysis.

In fact, think of the stack above—a single ontological database can store and describe any complete stack, no matter what underlying infrastructure it is on, from physical hosts to orchestrated containers, delivering a full picture of each and every component.

Another benefit our solution uniquely offers is that it allows any data, from any source to be brought in—including data from third party monitoring tools. Our AIOps solution ingests this data and treats it as a first-class citizen to allow for the analysis that makes part three possible—understanding what the data means for the overall performance of the application.

Part Three – Understanding What That Data Means

So far, we've discussed complexity in the system, complexity in the attack and complexity in observability. We've also hinted at the ephemeral nature of containerized environments, especially orchestrated containerized applications. The result of this complexity and constant change is that humans (either developers or ops) can't possibly understand how all the pieces are working together at any given time, especially as it changes.

In the fast-paced world of agile development, microservices, CI/CD and other operational advancements, monitoring gaps present new challenges. These gaps can impact overall service quality and user experience—not just in seeing and understanding, but also in how actions are taken to mitigate any problems.

An interesting example of this would be software updates. As CI/CD operations take hold, applications are measured not in releases per year, but rather in updates per day (or maybe even shorter time periods). In such an environment, continuous monitoring for problems manually becomes problematic:

- Knowing which software components have been updated
- Understanding whether any new dependencies (upstream or downstream exist)
- Alerting to any new errors within requests

- Capturing the service response time and application response time—after the update
- Deciding whether those metrics indicate success or failure
- Rolling back previous versions when problems are indicated

Any one of those tasks can take a person or legacy tool minutes (or longer) to determine for each update. An individual and even a team would be overwhelmed quickly. Ultimately, the inability to handle the task of verifying a software update could preclude an application delivery team from even moving forward with a continuous delivery strategy.

The best way to ensure that service monitoring and quality assurance keep pace with delivery is to automate the entire list of steps. Of course, there are other scenarios where automation across the monitoring lifecycle optimizes processes, from QA to prototyping. This is just one example.

Examining another example, root cause analysis is at another level entirely. After all, application stakeholders expect their application monitoring tools to help them solve problems when they occur. In cloud native applications, the number of technology platforms, polyglot of languages, complexity of relationships, and the dynamic nature of the applications make troubleshooting difficult.

To effectively troubleshoot, the system must be able to ascertain the difference between messages, warnings, and actual problems. Furthermore, Kubernetes and containers make traditional errors in monolithic applications disappear with the ability to spin up new server instances in microseconds. In this world, the ability to monitor for real service problems, then analyze the data to identify the root cause, is critical—especially as teams are built of more generalists and fewer specialists. These two cases, at opposite ends of the deployment lifecycle, showcase why intelligent automation is required to deliver effective service levels.

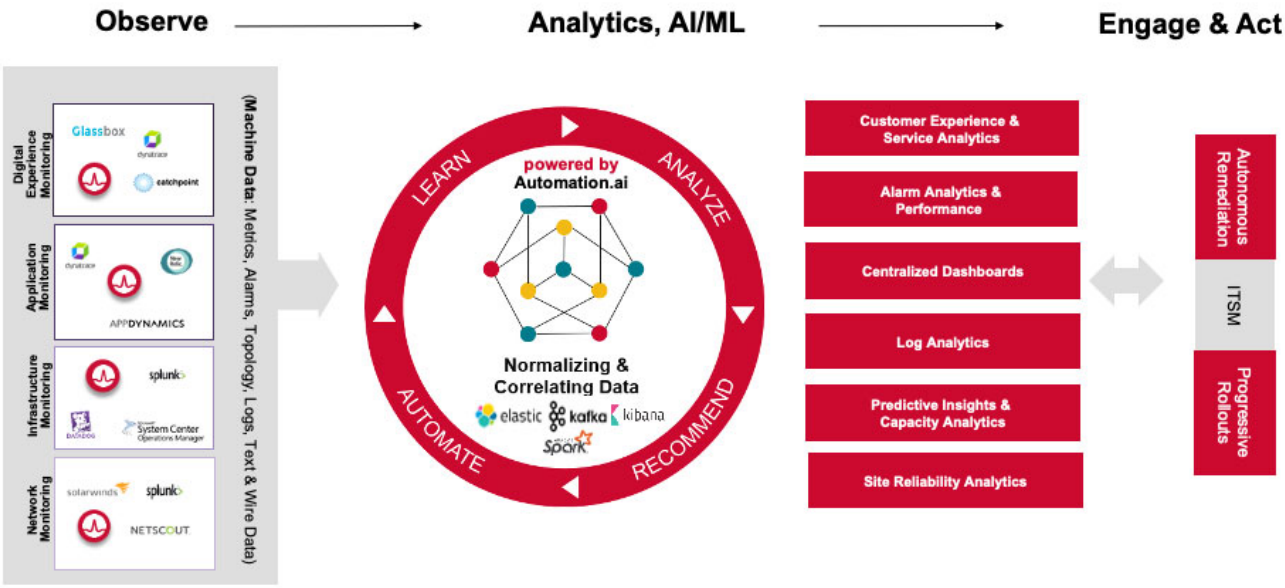
And automation *is* the singular requirement, It is not a question of additional human resources. The complexity of these application environments makes it practically impossible for a team of any size to handle manually. After all, it doesn't take many components before there are thousands of interactions handling millions of requests.

Actionable Insights Delivered by Automation.ai

That brings us to the last critical piece of the puzzle—Automation.ai. Automation.ai is the software intelligence platform that powers our AIOps solution. Our solution harnesses the power of advanced AI, machine learning, and Internet-scale, opensource frameworks to transform massive volumes of enterprise data into actionable insights.

Automation.ai can ingest data from any source including Broadcom monitoring tools, customer observability data or other third-party tools. The tool quickly analyzes the data and makes recommendations to Ops and Dev users to help optimize performance, discover and fix hotspots, and ensure that service levels are maintained.

Figure 3: AIOps from Broadcom—Powered by Automation.ai



Through our AIOps solution, these insights are delivered in a centralized place where everybody involved with applications, from developers and Ops to business managers and IT execs, can get the same data, same analysis and same answer to critical questions—which becomes even more important when managing the complexity of cloud native applications.

Conclusion

As companies move to the latest innovation in application platforms, cloud native, they don't have to reinvent the monitoring wheel. They just need to re-think how to use the monitoring solutions they already know and love in a way that makes sense for orchestrated containerized microservice applications.

AIOps from Broadcom provides the visibility, observability, and analysis needed by Ops teams to practically and effectively handle the complex ever-changing world of microservices. From extended topology to take on application components, a unified data lake to combine any and all data about the infrastructure, applications and transactions, to the advanced AI and machine learning to analyze the data and provide programmatic understanding of how those applications are operating.

To learn more about how our solution can enable effective monitoring for cloud native and containerized applications, visit www.broadcom.com/kubernetes-monitoring.