ActualTech Media

PLATFORM9

Tackling Observability in Your Kubernetes Environment

Joep Piscaer

CONTENTS

IN THIS PAPER

Observability in container-based environments requires a re-think of the monitoring tools you use in production. This tech brief presents a natural progression of how teams start with observability and become more fluent in using observability tooling.

There are many tools in the cloud-native and microservices tool chest. Kubernetes is the go-to for container management, giving organizations superpowers for running container applications at scale. However, running an enterprise-grade, production-level Kubernetes deployment is more than running just Kubernetes by itself.

Because containers are ephemeral and transient, monitoring, security, and data protection are fundamentally different from their counterparts in virtualized or bare metal applications. Optimizing the tooling that supports a Kubernetes deployment is not a trivial task. In many cases, this means that tooling aimed at virtualized environments doesn't translate well into containerized platforms. Replacing these tools may be better than retrofitting legacy tooling.

In fact, more modern tooling created to support container environments may help you get the most out of your container platform. In this paper, we'll look at how to optimize observability in your Kubernetes environment. We'll define the types of observability, offer a path for starting and expanding the process, and describe the advantages of cloud-based or Software as a Service (SaaS) monitoring.

Types of Observability and Their Value

Let's go back to basics first. Looking at the *observability* space for container-based microservices landscapes, we can distinguish three separate types of tooling:

1. **Monitoring (or metrics):** collecting operational telemetry about applications, application services, middleware, databases, operating systems, and virtual or physical machines
2. **Logging:** collecting error messages, debug or stack traces, and more detailed messages
3. **Tracing:** collecting user transactions and performance data across a single or distributed system

Note: Be sure to watch the Platform9 webinar on how Kubernetes has transformed monitoring. Another great resource to check out is the Platform9 blog, "Logging & Monitoring of Kubernetes Applications: Requirements & Recommended Toolset."

In a DevOps or Site Reliability Engineering (SRE) world, these three disciplines collectively make up observability.

Each discipline provides valuable insights in all layers of the layer cake that make up the increasingly complex application and infrastructure landscape of containers. DevOps engineers and SREs use the insights from these tools to improve resilience and performance, as well as triage errors, fix bugs, and improve availability and reliability.

> As application landscapes expand due to digital transformation, the number of microservices and individual containers explodes, making it harder to see the inner workings of systems.

Finally, they use these tools to gauge how users are interacting with the system. The tools help figure out which functionality visitors use or don't use, and where performance bottlenecks lie.

As application landscapes expand due to digital transformation, the number of microservices and individual containers explodes, making it harder to see the inner workings of systems. So, it shouldn't be a surprise that executing a good observability strategy is one of the deciding factors of a successful Kubernetes deployment.

While enterprise IT is more important than ever, digital transformation has led many more organizations to create digital and online applications. IT has often become a critical business function, vital for the survival of your business.

Layers of Monitoring

A good place to start with monitoring is by collecting metrics and operational telemetry of the Kubernetes constructs like clusters and pods, as well as collecting metrics on resource usage like CPU, memory, networking, and storage. Starting with the bottom two layers

(see **Figure 1**) for monitoring is relatively easy and a good way of becoming comfortable with observability tooling.

Infrastructure monitoring and logging are key capabilities because it's important to know the activities of your physical infrastructure. A substantial amount of your application's performance and resilience comes from correctly functioning servers and networking.

As the application landscape expands, a well-executed infrastructure monitoring and logging strategy also builds a shared understanding of application performance across teams, preventing miscommunication between application development, cloud platform, and other teams.

> Infrastructure monitoring and logging are key capabilities because it's important to know the activities of your physical infrastructure.

Visibility into infrastructure and the shared understanding it builds is crucial, but of course doesn't give the entire picture. For that, you need to move up the stack, and start with application performance monitoring (APM). For many organizations, the application monitoring journey starts with monitoring (or metrics collection) and logging

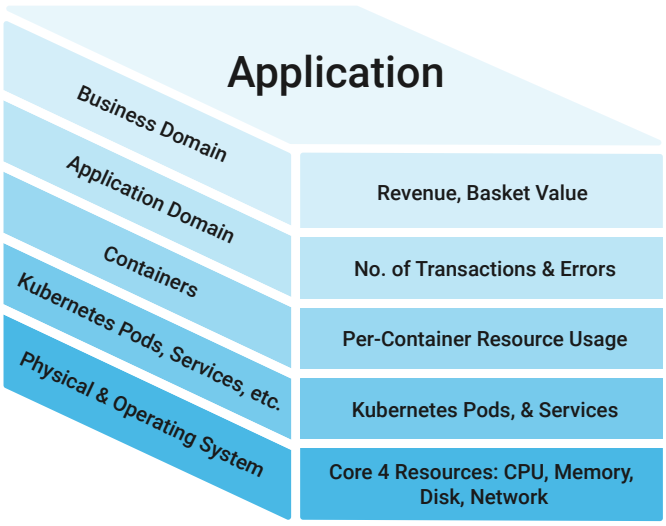

Figure 1: An application layer cake with monitoring examples

containerized workloads. For Kubernetes-based environments, there are natural combinations to start with, like the open source Fluentd and Prometheus, which make it easier to run monitoring and logging.

Making Observability Work for Your Business

To gain benefits from observability, you need to think about your business requirements over the next few years and choose a platform that meets your needs. This section describes how to think about requirements, tools, and the IT organization.

ALIGN MONITORING TO BUSINESS OBJECTIVES

This paper has presented a natural progression of how teams use observability, starting with the infrastructure basics, working their way up the stack into the realm of applications, and even tracing users across the application landscape, monitoring their behavior and transactions.

This journey up the stack is an opportunity to align monitoring, logging, and tracing to business objectives, mining more insights from the increased visibility. It allows teams to gain visibility into more than just technical metrics, generating business-oriented metrics, too.

> To gain benefits from observability, you need to think about your business requirements over the next few years and choose a platform that meets your needs.

By measuring business-oriented metrics (such as the dollar value of the shopping basket, the number of abandoned baskets, and metrics on popular or even disused features), product owners can align development priorities to what their users really want, optimize performance in areas where it actually matters, and fix technical debt to accommodate further growth. Naturally, these insights fuel business growth and revenue.

When tooling is aligned to the business and customer experience, the tools can be used by more than just IT teams, allowing business teams to gain insights into their applications and its users.

THINK MID-TERM TO LONG-TERM

The tools you choose for observability should serve your needs for several years. This requires you to think about how your business is changing and how that will change your observability requirements in the long-term.

The cost of migrating to a new, more capable APM platform can be significant, but won't immediately give you additional functionality. This additional functionality requires additional engineering and implementation before these capabilities are fully unlocked.

And let's not forget that moving to another APM platform requires you to retrain staff and needs time to regain confidence in the metrics and insights, all of which reduce the value the APM platform brings in the short term. So, it makes sense to choose your tooling wisely from the start, keeping the long-term goals in mind.

> The tools you choose for observability should serve your needs for several years.

In other words, while you won't need the most complex or feature-rich solution now, look at what features you'll need to support evolving requirements in the future. Invest in your team and people and start with the APM capabilities you need now. You don't need to enable, implement, and incorporate every feature the tooling provides from the get-go. It's OK to start simple, build up confidence along the way, continuously evolve your knowledge of the tool, and expand its use in-sync with changing requirements.

THE OBSERVABILITY PLATFORM TEAM

As your organization grows and your use of metrics becomes commonplace, it makes sense to create a dedicated team focused on the continual improvement of the observability platform, and help application development teams gather the metrics that matter to them.

The dedicated team owns the platform and executes an observability strategy across many applications and teams. The observability platform team's expertise speeds up troubleshooting, helps with application architecture optimization and can help teams to quickly pinpoint and solve bottlenecks and fragile areas prone to failure.

> As your organization grows and your use of metrics becomes commonplace, it makes sense to create a dedicated team focused on the continual improvement of the observability platform, and help application development teams gather the metrics that matter to them.

With their expertise and knowledge of the environment, they can "travel" from application team to application team to grow each team's knowledge and expertise more efficiently, preventing re-inventing the wheel and other local optimizations within each team, instead letting all teams learn from the collective knowledge, driving up maturity more quickly.

With this team structure, the application teams can focus on gathering metrics, refining the metrics they collect, and improving the signal-to-noise ratio inherent in gathering metrics, so the telemetry gathered is optimally serving business objectives and the team isn't spending any time on managing or operating the observability platform.

CLOUD-BASED OBSERVABILITY (SAAS)

If your teams are busy setting up and managing servers, they'll be less effective at their real job: improving the usage of APM across the organization. By adopting a SaaS offering, the observability platform team can focus on the functional side of observability instead of the day-to-day toil of managing and operating the platform.

That means they're not bogged down with installing yet another security patch or forced to think about scaling the observability platform. Instead, they have more time

to help application development teams with their observability challenges or to improve the platform itself.

SaaS also helps an organization get started with an observability platform. Instead of having to make the hard design choices at the start of a project (with little to no experience), you can use the choices offered by the SaaS vendor, confident that its experts have vetted the choices.

Using a SaaS service lets the vendor take on the operational burden of upgrades, scalability, and performance for the APM software, freeing up your team to work on broader and deeper implementation of APM across application teams.

This will speed up the pace at which your observability platform matures, allowing teams to gain a deeper and more user-oriented understanding of the application landscape more quickly. This increases the value of the investments made in the observability space in two ways: You spend less time installing, configuring, and getting started with the tooling, and the insights gained from the tooling can be applied more quickly to optimizing revenue and fueling growth.

In effect, you can leapfrog your APM journey, skipping the traditionally hard first steps in getting started with monitoring.

> Using a SaaS service lets the vendor take on the operational burden of upgrades, scalability, and performance for the APM software, freeing up your team to work on broader and deeper implementation of APM across application teams.

Platform9's free managed Kubernetes service deploys, upgrades, scales, patches, and manages your clusters. It has observability tooling built in: Prometheus, Grafana, and FluentD (in early access), and has a unique SaaS view of alarms across clusters globally. This built-in observability platform lets you leapfrog into mature application performance management with ease.

To experience observability as a service for free, sign up for Platform9's freedom plan.