

# Ensuring Visibility into Microservices and Containers

How to capture the benefits of continuous application development without unnecessary risks.



## Introduction

As more enterprises embrace DevOps practices and move their workloads to the cloud, application architects are increasingly looking to design choices that maximize the speed of development and deployment. Two of the fastest growing are containers and microservices.

Named by Gartner<sup>1</sup> in the top 10 technology trends impacting IT infrastructure and operations, containers and microservices are playing a crucial role in cloud adoption and application-driven innovation. Benefits include ease of implementation and operation; acceleration of time-to-market; and streamlined, lower-resource processes.

This white paper describes how containers and microservices work, the benefits and challenges of using them, and how a unified view of the enterprise stack and effective application performance monitoring (APM) can help to fortify their benefits and address their challenges.



## DevOps: Collaborative Application Development

In the old-school development model, there was little integration of the separate functions within IT operations. DevOps is an evolved model that facilitates collaboration among operations, development, and quality assurance teams. They collaborate and communicate throughout the software development process.

DevOps is not a separate role held by a single person or group of individuals. Rather, it conceptualizes the structure needed to help operations and development work closely together so that application development and deployment is faster and high quality.



## Containers: What Are They?

A container is a lightweight, standalone, executable software package that includes everything needed to run it: code, runtime, system tools, system libraries, and settings. Available for both Linux and Windows, containerized software will always run the same, regardless of the environment. Architecturally, containers are similar to virtual servers; however, unlike VMware or MS Hyper-V, they don't need the overhead of a hypervisor management layer to function.

## Containers: What Are They? *(continued)*

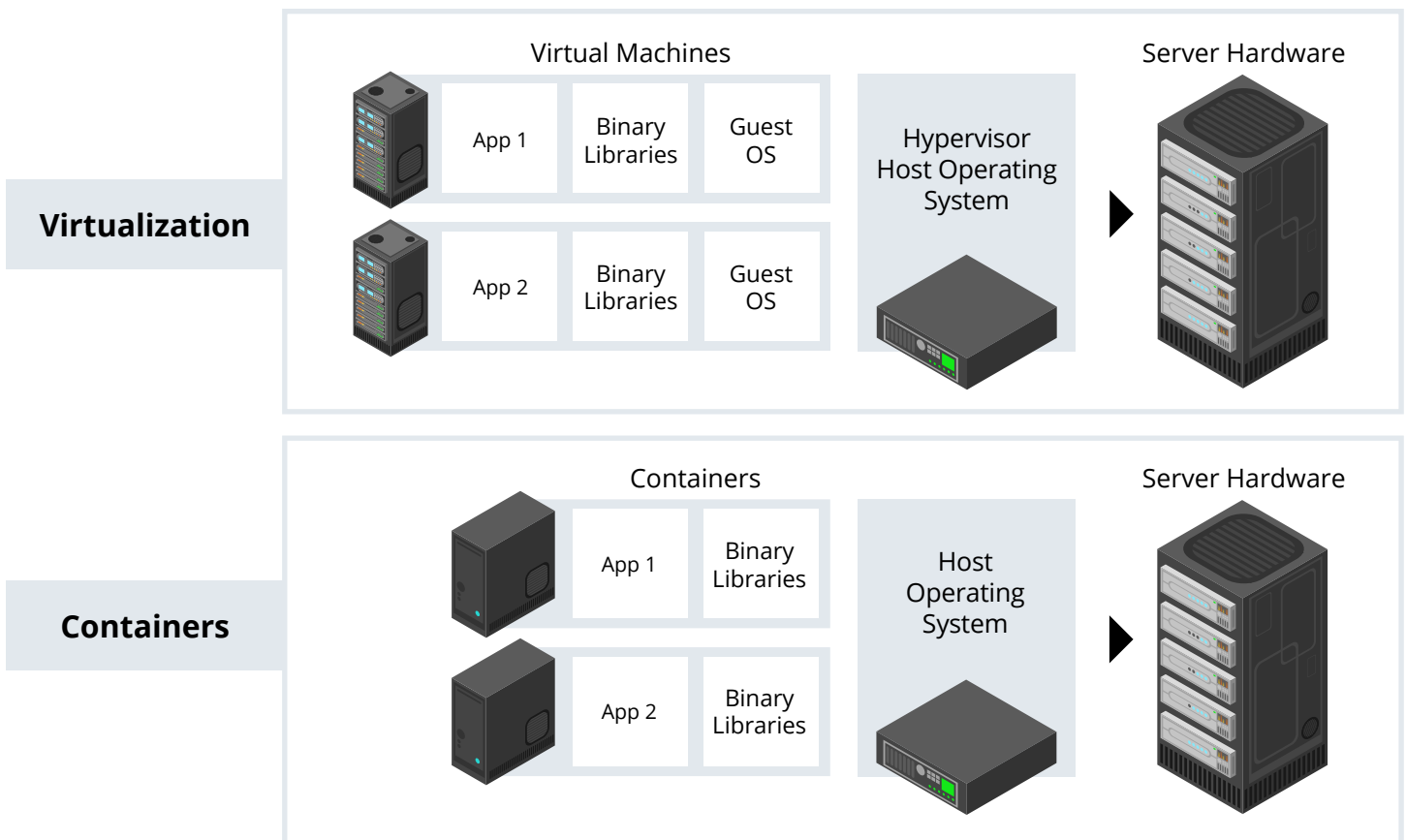
This lack of external dependencies makes containers very easy to deploy when compared to traditional application deployment models and a very good fit for DevOps and Continuous Delivery (CD), which promote rapid development and frequent release cycles.

Containers also isolate software from its surroundings — for example, differences between test and production environments — and help reduce conflicts running different software on the same infrastructure.

Containers are currently dominating the application development scene, especially in (but not limited to) cloud computing environments.<sup>2</sup> Popular container platforms include Docker and Kubernetes.

Benefits of a container-based architecture include:

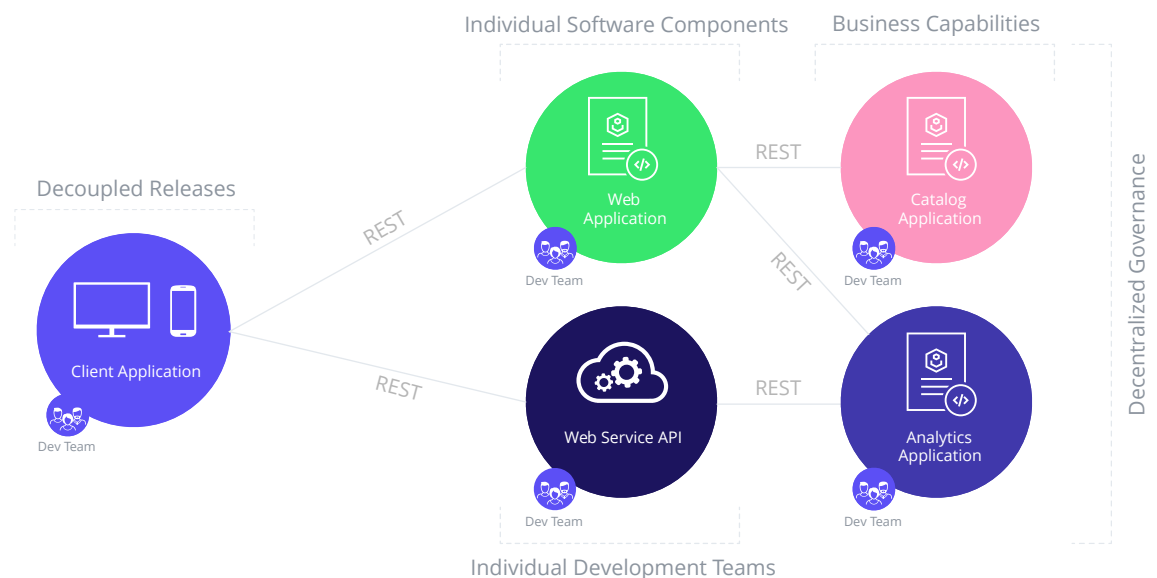
- Faster development
- Flexible deployment
- Improved isolation
- Microservices architecture
- Cost savings



# Microservices: What Are They?

Microservices are a type of software architecture where application functionality is implemented through the use of small, self-contained units working together through APIs. Each service has a limited scope, concentrates on a particular task, and is highly independent. Some key characteristics of a microservice architecture<sup>3</sup> as defined by Martin Fowler are:

- **Componentization:** Microservices are independent units that are easily replaced or upgraded. The units communicate with each other through remote procedure calls or web services.
- **Business capabilities:** Legacy application development often splits teams into areas like the “server-side team” and the “database team”. Microservices development is built around business capability, with responsibility for a complete stack of technical functions, plus UX and project management.
- **Products rather than projects:** Instead of focusing on a software project that is delivered following completion, microservices treat applications as products of which they take ownership. They establish an ongoing dialogue with a goal of continually matching the app to the business function.
- **Dumb pipes, smart endpoints:** Microservice applications require that all logic be maintained within the application and allow the transports for communicating between services to be lightweight.
- **Decentralized governance:** Teams are encouraged to choose the right tool for their own particular use case. However, some libraries may be shared among teams to avoid duplicative work.



## Microservices: What Are They? *(continued)*

Microservices are changing how teams are structured, allowing organizations to create teams centered on specific services, and giving them autonomy and responsibility in a constrained area. This approach lets the company respond to fluctuating business demands without interrupting core activities. These are great benefits, but there are also risks that come with microservices proliferation.

Microservices architecture is much more complex than legacy systems. In turn, the environment becomes more complicated because teams must manage and support many moving parts.

### **Some of the things enterprises must be concerned about include:**

- As you add more microservices, you must ensure they can scale together. More granularity means more moving parts, which increases complexity.
- When more services are interacting, the number of possible failure points increases. Smart developers stay one step ahead and plan for failure, ensuring their service remains operational in a diminished capacity if another service is down.
- Transitioning functions of a monolithic app to microservices creates many small components that constantly communicate. Tracing performance problems across tiers for a single business transaction can be difficult. This can be handled by correlating calls with a variety of methods including custom headers, tokens, or IDs.
- Traditional logging is ineffective. You would produce too many logs to easily locate a problem. Logging must be able to correlate events across several platforms.

Just as with containers, the expanded use of microservices brings challenges, but they are challenges that can be effectively managed with the right APM solution.

## Tech Summary

- **Containers** are isolated workload environments in a virtualized operating system. They speed up workload processes and application delivery because they can be spun up quickly, easily scaled, and provide a solution for application-portability challenges.
- **Microservices** are a type of software architecture that is lightweight and built on small, single-function, self-contained code units working together through APIs. This makes a microservices-based architecture easier to develop, test, and deploy than traditional application development architectures.
- **Containers and microservices** are often used together — for example, microservices are often delivered in one or more containers — but they exist independently and have different purposes.



## How AppDynamics Helps

Enterprises that fail to adequately monitor these new application architecture techniques are risking a lot, from letting bad code releases slip through to making bad decisions based on inaccurately tracked resource utilization. In terms of day-to-day application development, there's increased opportunity to be faster with solution releases and more informed about application performance.

---

**With a comprehensive understanding of how their applications are performing, enterprises can:**

- Find and fix problems faster and more easily
- Make better-informed business decisions about resource allocation
- Have total visibility into all assets and users
- Use the knowledge gained from APM to contribute to enterprise-level strategy planning

---

However, application performance monitoring is not just about managing technology and providing analytics to IT teams and developers. Read more about the AppDynamics APM solution below:

**Comprehensive APM Solution:** AppDynamics provides the ability to monitor the minute details of every component of the enterprise IT stack. This visibility is essential for reducing redundancies and wasted effort, finding the root cause of problems and swiftly adjusting, as well as extracting product/platform data for meaningful strategic analysis.

Without a clear and detailed understanding of how each component of a software stack works alone and in relation to other components, enterprise users cannot adjust their teams, resources, and roadmaps when they need to be modified. AppDynamics' diverse set of reporting options makes data discovery and visibility easy for enterprise users.

## How AppDynamics Helps *(continued)*

**On-Demand Scalability:** Understanding what is happening within your organization's cloud environment and Business Transactions helps you make real-time capacity decisions based upon your company's needs. With the AppDynamics platform, your enterprise can scale up or down as needed based on an up-to-the-minute understanding of current performance. Regardless of how the cloud/container/microservices landscape changes over the coming years, the ability to monitor, understand, and act based on your application performance data will be critical.

**Native Container Monitoring Support:** Most traditional monitoring solutions offer little insight into KPI metrics related to container environments. This lack of visibility has obvious implications for maintaining system performance and the time it takes to identify and resolve application problems surfaced during the SDLC. Enterprises must address this challenge by deploying the latest generation of application monitoring. Increasingly the solution of choice is a platform such as AppDynamics, which provides specific support for monitoring container internals.

By understanding all aspects of their application development infrastructure, enterprises are better prepared to seize opportunities that will fuel their growth while protecting their investments, intellectual property, and user relationships. Containers and microservices are part of a transition to more agile, responsive, and targeted development teams. Grab their benefits by encouraging their use, and do so confidently with the help of AppDynamics.

Check out our free eBook  
*How to Build (and Scale) with Microservices*

APPDYNAMICS

How to build (and scale) with  
microservices

<sup>1</sup> Gartner. "The Top 10 Technology Trends Impacting Infrastructure Operations," December 7, 2016

<sup>2</sup> DZone/Cloud Zone. "Containers in the World of QA," October 6, 2017

<sup>3</sup> MartinFowler.com "Microservices - a Definition of This New Architectural Term," March 25, 2014