APPDYNAMICS

Getting Started With Containers and Microservices:

A Mini Guide for Enterprise Leaders

An introduction on how to effectively implement and monitor containers and microservices.

AppDynamics is now part of Cisco.

Introduction

Across a variety of large, global companies, DevOps teams are adopting containers and microservices, two of many new tools and practices that are offering solid business benefits.

- **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; and they provide a solution for application-portability challenges because they are not tied to software on physical machines.

- **Microservices** are a type of software architecture that is light and limited in scope. Single-function applications comprise small, self-contained units working together through APIs that are not dependent on a specific language. A microservices architecture is faster and more agile than traditional application architecture.

And while containers and microservices exist independently and serve different purposes, they're often used together. For example, microservices are often delivered in one or more containers.

Developers use containers and microservices in many tasks, from app delivery to migration of legacy systems to cloud servers. The increased popularity in containers and microservices can be attributed to their impact on agile cloud environments, with benefits that include increased efficiency, greater speed of delivery, and the ability to do more with existing resources.

And while the benefits of containers and microservices are causing an increase in usage, there are also some complexities to be aware of. For example, while usage is exploding, many developers are still learning how to use these relatively new tools and practices. This means organizations must maintain data security, system reliability, and other service levels during the learning curve.

As far as ongoing use, enterprises must work to prevent application sprawl, unauthorized use, problematic code, and other issues that could go undetected if unmonitored.

> Over half of technology professionals surveyed for the Cloud Foundry Global Perception Survey on containers indicated they were evaluating/using containers, with 64% anticipating mainstream usage within a year.
>
> — Cloud Foundry Foundation

Understand the Impact: Resources, Strategy, and Monitoring

Containers and microservices have implications for resource management, strategic goals, and monitoring capabilities. Use these questions to begin to understand and document what those implications could be.

Resource Management

How much of our workload should be moved to microservices?

Where is the best place to start using containers?

How will usage impact our current stack today and in the future? Fault tolerance? Proliferation of languages, caching, and requests?

Strategic Goals

How can we ensure the development group and operations group are working together to use containers and microservices in alignment with enterprise needs?

Do we need to restructure our teams so that we are using containers and microservices according to enterprise goals?

Monitoring Capabilities

Will we build our own system to manage container assignment, clustering, etc.? Or should we use third-party vendors that will need to be monitored?

Will we be able to monitor code inside containers and the components that make up microservices with our current application performance management (APM) footprint?

MICROSERVICES FYIs

- **Containers are an enabling technology for microservices:** Because containers are isolated environments, they can be used to deploy microservices quickly, regardless of the code language used to create each microservice.
- **Microservices are making the enterprise IT stack more complex, and thus more difficult to monitor:** Microservices focus on solving a particular problem, which means the technologies they choose to employ can change for any given service. The introduction of more technologies means that DevOps teams will be overwhelmed with data if they choose to use a monitoring tool that is too narrow.
- **Traditional logging is ineffective because microservices are stateless, distributed, and independent:** If attempting to log them, an enterprise would produce way too many logs to easily identify a problem. Logging must be able to correlate events across several platforms.

Do You Need Better Monitoring Capabilities?

After considering the implications of containers and microservices, you may discover that you need to find a new APM solution that will fill the identified coverage and capabilities gaps. To choose an effective solution, use this eight-point plan, which is based on best APM practices.

1 **Document what you want to achieve so that you can share with APM solution vendors.**

Use consistent terminology as you define:

- Vision and goals
- Priorities and plans
- Roles and responsibility

Be sure that you ask about:

- Education and onboarding of new users
- Integration with your main technology stacks
- Use of APM throughout the SDLC (software development life cycle)

2 **Embed ways to share knowledge about the new solution.**

These actions will ensure that knowledge about the new solution can be shared with all users of the APM solution.

- Make a wiki and playbook for users
- Operationalize input from subject matter experts and power users
- Hold regular review/success meetings

3 **Define a measurement framework.**

- Determine KPIs and how they'll be measured
- Determine how to best leverage flow maps and baselines

4 **Keep flows intact.**

In choosing a solution, be sure that it will:

- Provide end-to-end correlation
- Show contextual views of flows

5 **Configure signals from noise.**

Make sure the solution can be calibrated to prioritize needs from noise.

- Ensure Business Transactions are correct
- Baseline all metrics to minimize alerts to truly abnormal activity

6 **Explore your options.**

- Understand the multitude of options at your disposal
- Slice and dice your data to identify what you can do

7 **Secure your privacy.**

Securing enterprise data should be a priority during implementation. Ask these questions of application providers:

- Is sensitive data going to be protected out-of-the-box?
- Is there a way to limit access to within our organization?

8 **Prioritize automation.**

Three areas that can be automated are:

- Agent deployment
- Configurations (advanced)
- Information flow

AppDynamics for APM

AppDynamics' APM solution can help enterprises prepare for safely adopting and scaling the use of containers and microservices, as well as other emerging tools and technologies used by DevOps teams.

- **See all applications and environments in your stack:** Monitoring and visibility are among the top challenges of users who are adopting containers. AppDynamics gives you metrics on the container, the operating system, server, and applications.
- **Lower the risk of migrating to containers:** Because you get visibility into the dependencies your applications have on one another, you can be proactive about managing them.
- **Have the right level of monitoring no matter how many containers or microservices are in use:** AppDynamics scales APM as you grow.

Microservices and Containers Are Here to Stay

As the race to a more agile cloud environment continues, enterprise users will need to understand not only how to develop technology solutions faster, but how to monitor everything that is happening across their technology stack.

Let your teams take full advantage of containers and microservices by ensuring you have an APM solution that will scale as they are spun up, used, and decommissioned over and over again within your technology stack.

This is where AppDynamics excels. With AppDynamics integrated into your technology stack, your enterprise can move fast, without having to worry about breaking things.

Take a guided tour