ıto the mainstream

After years of experimentation, enterprises are going all-in with containers and microservices, building and updating better apps at a faster clip than ever before

By Eric Knorr

Editor in Chief, InfoWorld

MAR 9, 2020 3:00 AM PDT

When Edison invented the lightbulb, it had a problem: It needed to be hardwired to the lamp. Hence the Edison screw, which became the standard that, to this day, allows almost any bulb to be twisted into almost any light fixture, be it desk lamp or chandelier.

A decade ago, Solomon Hykes' invention of Docker containers had an analogous effect: With a dab of packaging, any Linux app could plug into any Docker container on any Linux OS, no fussy installation required. Better yet, multiple containerized apps could plug into a single instance of the OS, with each app safely isolated from the other, talking only to the OS through the Docker API.

That shared model yielded a much lighter weight stack than the VM (virtual machine), the conventional vehicle for deploying and scaling applications in cloudlike fashion across physical computers. So lightweight and portable, in fact, that developers could work on multiple containerized apps on a laptop and upload them to the platform of their choice for testing and deployment. Plus, containerized apps start in the blink of an eye, as opposed to VMs, which typically take the better part of a minute to boot.

To grasp the real impact of containers, though, you need to understand the microservices model of application architecture. Many applications benefit from being broken down into small, single-purpose services that communicate with each other through APIs, so that each microservice can be updated or scaled independently (versus traditional monolithic applications, where changes require you to bring down and tinker with the whole deal). As it turns out, microservices and containers are a perfect fit.

Featured in this series:

- Containers and Kubernetes: 3 transformational success stories (CIO)
- Kubernetes meets the real world (InfoWorld)
- Essential things to know about container networking (Network World)
- How Visa built its own container security solution (CSO)
- PaaS, CaaS, or FaaS? How to choose (InfoWorld)
- Containers on the desktop? You bet—on Windows 10X (Computerworld)

But how do you get containerized microservices to work in concert as an application? That’s where, at least for larger microservices applications, Kubernetes comes in. This open source orchestration engine enables you to deploy, manage, scale, and ensure the availability of a microservices-based application – and move it all of a piece across platforms if you need to.

If all this sounds like a whole bunch of moving parts, it is (some question whether Kubernetes is necessary except in a small slice of cases). But make no mistake: The microservices era is upon us and the ability to scale or swap in new services on the fly is essential for a big swath of modern applications. No matter how those services are managed, containers have established themselves as their standardized, streamlined receptacles.

Rolling containers into production

In “Containers and Kubernetes: 3 transformational success stories,” Contributing Writer Bob Violino explores how Expedia, Clemson University, and the finserv firm Primerica have tackled Kubernetes. Bob’s article follows on “Kubernetes meets the real world” by UK Group Editor Scott Carey, which delves into similar efforts by Bloomberg, News UK, and the travel data provider Amadeus. The consensus? As Primerica CTO Barry Pellas says, “enabling teams with the right skill sets to properly develop within the [Kubernetes] environment can be challenging.” But challenging or not, Kubernetes today is the broadly accepted solution for orchestrating containerized services at scale.

The usefulness of Kubernetes extends to the knotty problem of networking containers. As Network World contributor John Edwards explains in "Essential things to know about container networking," networking containers bears little resemblance to data center networking. Not only is container networking completely software-defined, Kubernetes itself handles all routing requests and network connections without human intervention. All those connected services together are referred to as a service mesh, which yet another open source project, Istio, is designed to handle – enabling admins to manage traffic, control policies, discover services, and so on.

Istio also provides some measure of security, such as TLS-secured communications among services. But the world of containers in production is all pretty new – and some large enterprises have decided to take security into their own hands. CSO Senior Writer Lucian Constantin explains "How Visa built its own container security solution" for container monitoring, security policy enforcement, and incident detection and remediation. According to Lucian, it was a classic build-versus-buy decision: What happens when existing solutions look a little shaky or lack the right mix of features? Do it yourself.

At the other end of that spectrum are the CaaS (containers-as-a-service) offerings from cloud providers, perhaps more accurately described as Kubernetes-as-a-service solutions. Amazon Web Services, Google Cloud Platform, and Microsoft Azure all offer their own CaaS flavors. But as Contributing Editor Isaac Sacolick observes in "PaaS, CaaS, or FaaS? How to choose," CaaS isn't your only container management option. Instead, you might choose a PaaS (platform as a service), which typically trades configurability for faster, easier development and deployment. FaaS (functions as a service) offerings, also known as serverless computing platforms, offer an even higher level of abstraction, enabling developers to assemble services quickly from small, discrete functions. Yes, FaaS solutions run containers under the hood, but developers don't even see them, let alone need to manage them.

And the end-user benefit of such container solutions? Basically, better software that can be updated and improved at a faster pace. As explored in "Containers on the desktop? You bet – on Windows 10X," Microsoft has introduced a novel type of container that ensures legacy applications run properly on the innovative Windows 10X operating system for dual-screen devices. This particular container advance may help free Microsoft from backward-compatibility issues that have constrained Windows' progress for many years.

In the end, containers are all about that much-ballyhooed IT benefit, agility. They can be moved around easily and plugged into a panoply of platforms. They eliminate unnecessary dependencies. They can be reused and recombined into different applications. And as an agile enabler of microservices infrastructure, containers help sustain small, distributed teams, each responsible for their own microservice – a healthy division of labor that yields better software faster.

On a purely technical level, like the Edison screw, containers are a modest advance, but one with momentous implications for the applications you haven't yet developed, and for the applications you'll use for many years to come.

This story, "Containers march into the mainstream" was originally published by InfoWorld.

Eric Knorr is the Editor in Chief of IDG Enterprise.

Follow