REPORT REPRINT

Red Hat's OpenShift Serverless for hybrid, legacy and greenfield

FEBRUARY 21 2020

By William Fellows

Kubernetes is complex and difficult to deploy – it autoscales based on available resources, not on requests themselves. OpenShift Serverless is designed to resolve this complexity to deliver the benefits 'as advertised' of quicker time to market and faster recovery.

Introduction

Kubernetes is complex and difficult to deploy – for example, it autoscales based on available resources, not on requests themselves. OpenShift Serverless is designed to resolve this complexity (and without the need to write YAML) to deliver the benefits 'as advertised' of quicker time to market and faster recovery.

451 TAKE

The promise of serverless applications – in which reusable functions and triggers are assembled into software that works independently of the infrastructure that executes it – has attracted a wave of startups, open source projects and cloud providers because the potential benefits are too compelling to ignore: faster development, hands-off provisioning and dramatically lower costs. Rather than having VMs sitting idly by – burning cash, compute resources and energy – these applications invoke compute resources only when needed, operating on a pay-per-use rather than pay-per-provision basis, as virtualized hardware does. Although the technology for building and operating serverless applications at scale still has rough edges (partly due to the lack of an open standard), born-in-the-cloud companies are adopting a 'serverless first' strategy for many applications, given the favorable economics and speed of development that it makes possible – it's a first-class citizen in enterprise IT (see Figure 1 below). Red Hat's OpenShift Serverless is designed to provide a one-stop shop for enterprises seeking to optimize their use of serverless across hybrid environments, and for both new and existing applications.

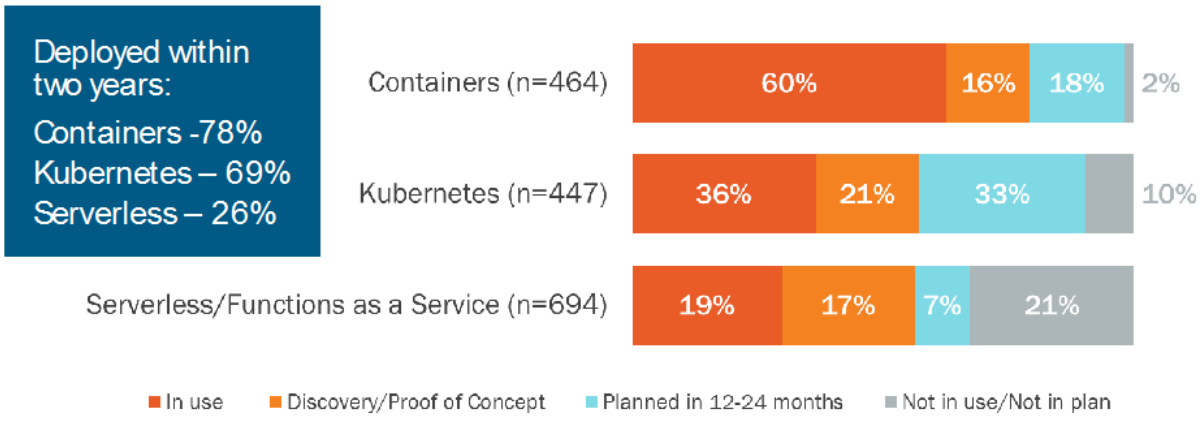
Problem set

Kubernetes provisions containers based on resource availability, which means there are times when the resources are over-provisioned (when there are few requests), and times when they are under-provisioned (when there are many requests). Serverless can address this, and bring the agility of the cloud (and the illusion of public cloud infinite resources) to on-premises, multi-cloud and hybrid.

It enables event-driven applications that can also integrate with traditional applications. Developers can focus on business logic, leave infrastructure to platform and services engineers, and run code without needing to know anything about the underlying platform specifics. Furthermore, serverless provides consistent and scalable operations across multiple applications.

Figure 1: Adoption of cloud-native technologies

Source: 451 Research's Voice of the Enterprise: DevOps Q1 2019 and Digital Pulse: Budgets & Outlook 2019
Q. Please indicate your organization's adoption status for the following technologies...



Red Hat OpenShift Serverless

OpenShift Cloud Functions was originally a FaaS implementation using Apache OpenWhisk serverless. Red Hat has pivoted this away from OpenWhisk to Knative, which enables developers to build and run serverless applications on Kubernetes (and is not opinionated about the FaaS), and has rebranded the assets here as OpenShift Serverless (which comes with OpenShift and therefore runs wherever OpenShift runs: cloud, on-premises, etc.).

Once it has settled with serverless for all of the container services, it will get to the functions – that is, the runtime that runs on top – and OpenShift Cloud Functions will run on top of OpenShift Serverless. It expects some users will want a FaaS for new application development or migrating from a cloud (AWS Lambda) to an on-premises service (OpenShift).

Right now, OpenShift Serverless encompasses microservices, functions and applications, and events; uses containers as the atomic unit; and takes advantage of OpenShift, Knative, Istio (service mesh) and Keda (event-driven autoscaling for containers).

While OpenShift serverless comes free with OpenShift, Red Hat makes money on the use of OpenShift and the additional services customers will need around it, such as API gateways, messaging and storage.

Use cases

Red Hat survey data suggests key serverless use cases among its user base are back-end APIs, web applications, process automation, severless websites, and integration across multiple systems. This is where Red Hat is putting significant effort, with a portfolio of web applications and APIs, data transformation services and scheduling, for use with OpenShift Serverless.

Red Hat OpenShift Serverless uses the open source Camel-K to access the Camel connector catalog (200+) and for stateless service orchestration, and CNCF Strimzi for event streaming in its Red Hat AMQ Streams product. It integrates with Istio offered by its OpenShift Service Mesh, and can also offer API gateway capabilities with the mesh through the 3scale Istio Mixer Adapter. Various partners support a FaaS layer on Knative for OpenShift Serverless including Quarkus.

Given that IBM's stated direction is toward Red Hat services and OpenShift specifically, we expect IBM's own serverless offering, IBM Cloud Functions based on OpenWhisk, to be consolidated on OpenShift and use OpenShift Serverless in future.

The evolution of serverless

Red Hat believes serverless has evolved to have reached a '2.0' era of maturity today. In its view, serverless 1.0 (e.g., Lambda, Functions) was built around the FaaS component and by other services such as API gateways. And while this enables a variety of use cases, Red Hat believes it is far from ideal for general computing, and has room for improvements. Serverless 1.0 is characterized by the use of:

- HTTP and few other sources
- Functions only
- Limited execution time (5-10 minutes)
- No orchestration
- Limited local development experience

Serverless containers heralded the 'serverless 1.5' era where, with the advent of Kubernetes, many frameworks started to auto-scale containers. Cloud providers created offerings using managed services completely abstracting Kubernetes APIs. Serverless 1.5 is characterized by:

- Knative
- Kubernetes-based auto-scaling
- Microservices and functions
- Easy to debug and test locally
- Polyglot and portable

The serverless 2.0 era is emerging with the addition of integration and state, Red Hat believes. The maturity and benefits of serverless are being recognized across the industry, and providers have started adding the missing parts to make serverless suitable for general-purpose workloads and used by the enterprise. Serverless 2.0 is characterized by:

- Basic state handling
- Use of enterprise integration patterns
- Advanced messaging capabilities
- Blended with the enterprise PaaS
- Enterprise-ready event sources
- State and integration

Providing state to serverless approaches (so far, it's mostly been a stateless activity) includes AWS Step Functions, the CNCF Serverless working group that is building a standard for it, and the Cloudstate specification for providing distributed state management patterns, which is included in the Lightbend Platform. State has previously been dealt with by putting it into serverless state management services. Cloudstate keeps it closer to the actual code.

Serverless adoption is growing rapidly, and is a core component of many digital transformation projects in the enterprise – some believe all compute will go serverless. Regardless of whether some, most, or all compute goes serverless, it's going to be a big opportunity, and there's a huge need for integration of serverless with both Kubernetes-based and legacy applications.

TriggerMesh EveryBridge plays here – targeting an opportunity similar to what MuleSoft targeted in providing integration for enterprise service buses. Verticals such as financial services are now embracing the cloud, serverless and data streaming as they seek to manage information flows.

The Project Keda collaboration between Red Hat and Microsoft is an open source project designed to provide event-driven scale capabilities for container workloads. It enables, for example, Azure Functions on any Kubernetes implementation (OpenShift was originally the target). Now it's being extended to any containerized workloads with the integration or KEDA (event pull) and Knative (event push).

Competition

The concentration of serverless around the major vendors (AWS, Azure, Google, IBM, Oracle and VMware) has made independent approaches less viable. Platform9, for example, is not further developing its Fission FaaS. However, some, such as startup Nuweba, perceive that having only big ships that are difficult to steer is an opportunity.

Amazon EventBridge is a serverless event bus that ingests data from a customer's applications, SaaS apps and AWS services, and routes that data to targets. But EventBridge only consumes AWS events. Apart from Google's Knative and Cloud Run efforts, Pivotal Function Service also uses Knative. Microsoft offers the Kubernetes-based KEDA designed with Red Hat. Triggermesh and Pipegears are also aimed at integrating different cloud and serverless services. Other vendors here include NowFloats and Kitsune.

SWOT Analysis

STRENGTHS

Event-driven serverless ensures enterprises don't waste money owning resources when they are not needed, and they do not need to be worried about scale. Red Hat anticipates that everything in the PaaS layer (where applications and development are abstracted from the underlying infrastructure) will ultimately run as some kind of serverless workload.

WEAKNESSES

The industry is making this up as it goes along, and out of the primordial soup that is cloud-native, a few strange animals are likely to emerge – however, serverless is already a strong gene pool.

OPPORTUNITIES

The revenue in serverless is being made in the application space. We believe the major vendors are providing serverless hosting as loss leaders for the add-on services, and that is the reason some independents have abandoned future investment here. As enterprises' cloud-native applications start consuming a greater number of different serverless offerings in the cloud and on-premises, value-added services enabling this will become attractive to buyers, partners and investors alike.

THREATS

Red Hat believes the key barriers to cloud-native, and specifically Kubernetes, adoption include the lack of DevOps maturity, the inability to integrate existing development and receive events from existing systems (and its APIs and processes), and the lack of tools and management. The question is whether OpenShift Serverless can address all of these, and how it (IBM) ultimately stacks up against both competitors and erstwhile partners such as Google and Microsoft.