

## REPORT REPRINT

# Cloud-native approaches are making infrastructure invisible

**WILLIAM FELLOWS**

**14 MAY 2018**

Serverless, containers and microservices are making infrastructure invisible, benefitting technology consumers that expect to access, assemble and pay for digital services in a simple, seamless and automated manner, without requiring any specific knowledge of the underlying physical infrastructure.

---

THIS REPORT, LICENSED TO PIVOTAL, DEVELOPED AND AS PROVIDED BY 451 RESEARCH, LLC, WAS PUBLISHED AS PART OF OUR SYNDICATED MARKET INSIGHT SUBSCRIPTION SERVICE. IT SHALL BE OWNED IN ITS ENTIRETY BY 451 RESEARCH, LLC. THIS REPORT IS SOLELY INTENDED FOR USE BY THE RECIPIENT AND MAY NOT BE REPRODUCED OR RE-POSTED, IN WHOLE OR IN PART, BY THE RECIPIENT WITHOUT EXPRESS PERMISSION FROM 451 RESEARCH.



©2018 451 Research, LLC | [WWW.451RESEARCH.COM](http://WWW.451RESEARCH.COM)

Serverless, containers and microservices are making infrastructure invisible, benefitting technology consumers that expect to access, assemble, and pay for digital services in a simple, seamless and automated manner, without requiring any specific knowledge of the underlying physical infrastructure. The 'invisible infrastructure' must be instantly available, operate and scale regardless of specific requirements, and be billed and metered in a manner the customer prescribes. It just works. However, this does not mean it is any less important. In fact, the paradox is that the more important the role infrastructure plays in our lives, the more important it becomes to shield users from having to directly interact with, or even consider, it.

---

## THE 451 TAKE

The impact of cloud-native approaches (including containers, serverless and microservices) will have a profound effect on the IT landscape. Infrastructure (and associated complexity) becomes invisible (to the consumer), and cloud may eventually disappear as a separate IT category. How 'cloud native' plays out with regard to existing/alternative application-focused approaches including PaaS remains to be seen. It is relatively immature by comparison, but the disruption it causes will be significant. No wonder most IT vendors and service providers are embracing a cloud-native position. The 'building blocks' of cloud native are used mostly to assemble other blocks and address cloud-native challenges, rather than being provided as packaged offerings that solve business problems. However, stacks that use these blocks will emerge, and anyone not paying attention to the opportunities and challenges risks becoming irrelevant.

---

451 Research data shows that the majority of cloud-native or cloud-enabled software is developed internally, rather than purchased commercially (including SaaS). This suggests that enterprises are currently using the building blocks of cloud native (containers, microservices, serverless, etc.) to create their own software. The data (see Figure 1) from our Voice of the Enterprise: Cloud, Hosting & Managed Services, Workloads and Key Projects 2018 survey also shows that most enterprises design cloud-native software to run on multiple clouds, while about one-third of respondents are designing it for one specific public cloud and 17% design for their own private cloud. Taken together, this suggests that the opportunity for commercial cloud-native software is mostly in front of us, and that offerings which support multiple cloud environments are likely to be more successful than those that only target one.

Figure 1

Q: How much of the software developed internally at your organization is cloud-native or cloud-enabled?

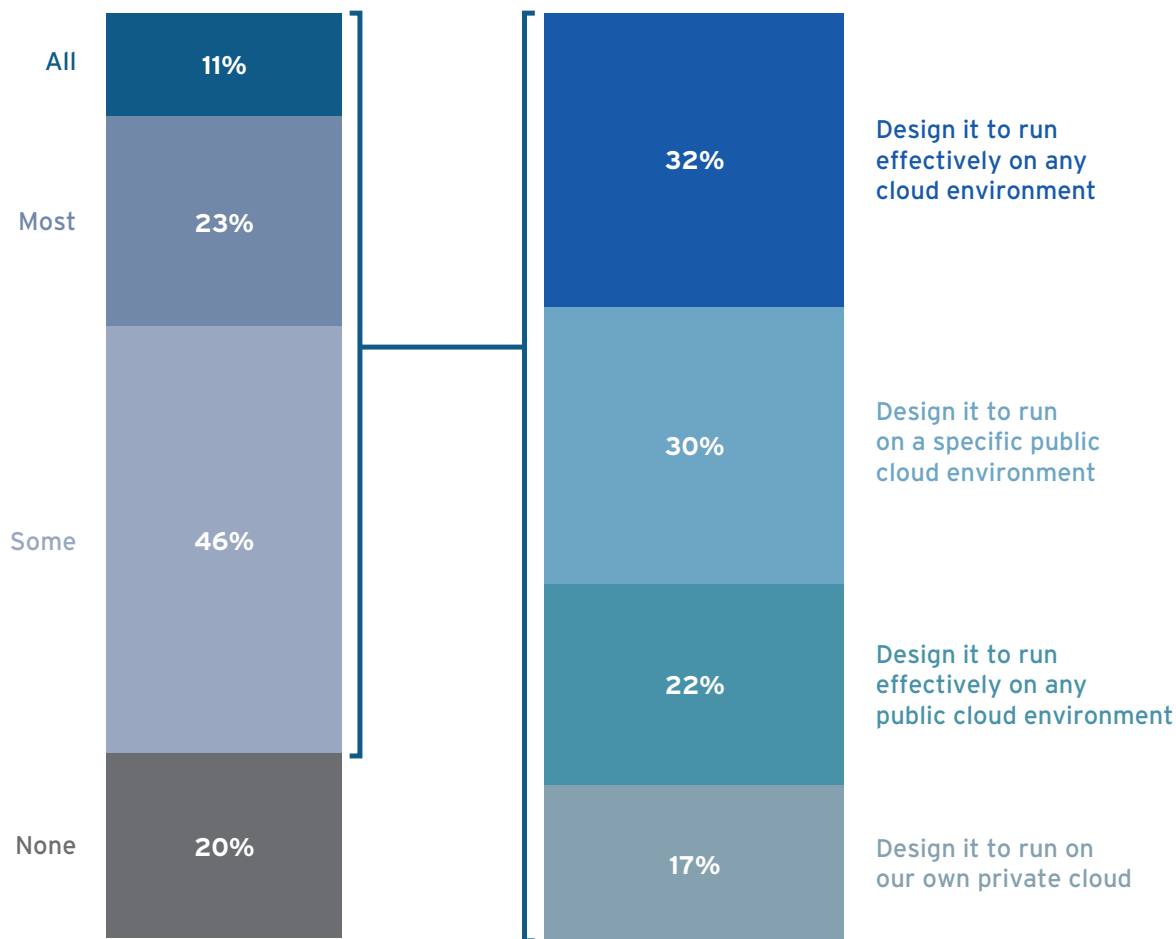
Q: When developing cloud-native software, which, if any, of the following approaches does your organization take to designing that software?

### Internal Software Development: How Cloudy?

Respondents with internal software development for new business functions/apps  
(n=347)

### Cloud-Native Software: Designed To Run Where?

Respondents with cloud-native internal software development for new business functions/apps  
(n=266)



Source: 451 Research, Voice of the Enterprise: Cloud, Hosting and Managed Services: Workloads and Key Projects, 2018

### 'WRITE ONCE, DEPLOY TO ANY SERVERLESS' IS CNCF'S GOAL

Cloud Native Computing Foundation (CNCF) doesn't have a serverless project on its books, and doesn't plan to adopt one – at least at this point. Instead, its serverless working group is focusing on standards – how to write once and deploy to all serverless/function-as-a-service environments, which all have different formats, runtimes and plumbing. It has working demos and reports into the CNCF's technical committee. There are a number of OSS serverless projects, such as IBM-originated Apache Whisk (underpins IBM Cloud Functions), Oracle's (Apache) Fn, Platform9 Fission (uses Kubernetes), Iron.io IronFunctions, OpenFaaS, Bitnami's Kubeless and Riff (basis of Pivotal Function Service).

One of the wider issues that ‘serverless’ impacts is its effect as it reaches into the world of IT operations. Although serverless would appear to abstract away all of the complexity associated with operations (implying that traditional IT ops is not required), this will probably make operations even more important. This will be especially true for regulated industries, in which when, where and how processing takes place is a matter of legal record. In order to be successful, the serverless message will need to be that operations remains an important dimension for enterprises and large organizations. Serverless implementations aren’t going to displace this requirement (or staff).

## SERVICE MESH

A service mesh is an inter-service communication mechanism for microservices. Microservices don’t communicate directly with other microservices; instead, they use software called a service mesh or sidecar proxy. Service meshes variously support some network functions, such as resiliency and service discovery; some also support traffic management, policy enforcement, observability, and security and identity. They act as a control plane, leaving the developer to focus on the business logic. Service meshes are effectively solving the same kinds of problems that enterprise service buses and message-oriented middleware tried to in the world of service-oriented architecture. It means that, for example, content servers, order management and payment processes can all interact. Service mesh implementations include Envoy (via NGINX and HAProxy), which was developed by Lyft; Istio (uses Envoy and is used by Google, IBM, Lyft, Cisco and F5 Networks’ Aspen Mesh, among others); Conduit (via Bouyant); and Linkerd (from a Twitter technology and now a CNCF incubation project).

One obvious consequence will be the need for new economic environments that support micro transactions and micro billing. The likes of Cisco Comptio, ParkMyCloud, Skeddly, RightScale Optima, Spotinst, BidElastic, GorillaStack and FittedCloud operate here, but there is plenty of room for innovation. Moreover, observers note that, for all the benefits of a service mesh, the downsides are a big increase in the number of runtime instances needed and the need for each call to take an extra ‘hop’ via the sidecar. Service meshes don’t address many associated complex issues – and as with immature technologies, it’s not for the technology faint-hearted.

## PAAS AND THE CLOUD-NATIVE WORLD

Viewed through the lens of serverless, is the world of PaaS in danger of being disintermediated? Serverless looks set to deliver, for all intents and purposes, what PaaS should have been doing a decade ago – making infrastructure invisible. Moreover, the astonishing number of value-added services being introduced by the hyperscalers collectively serve as de facto PaaS. Red Hat’s OpenShift PaaS was introduced in 2011, but it’s now more or less pivoted to become a ‘go to’ for enterprise Kubernetes with the introduction of OpenShift Container Platform (Kubernetes is a Google invention).

Meanwhile, Pivotal Software, originally formed by EMC, VMware and General Electric (and now public) to incubate the Cloud Foundry PaaS, also offers the Pivotal Container Service (Kubernetes) and Pivotal Function Service (serverless) alongside the PaaS. Cloud Foundry itself is now the name of the portfolio, which includes all three – the PaaS is renamed Pivotal Application Service (PAS) and ‘cloud native’ is very much in the driver’s seat in terms of positioning. Pivotal – and the associated Cloud Foundry Foundation open source project – is making great progress with the PaaS, which is in use at the very biggest companies. Half the Fortune 500 use the PaaS, and the take-up by Huawei and Alibaba demonstrate its global significance – the US cloud.gov is certified, and the leading IaaS providers, except AWS, support Cloud Foundry workloads.

None of this, however, detracts from the fact that cloud-native approaches, containers, serverless and microservices are beginning to dominate the agenda, and are the shape of things to come. For many attendees we spoke with at Kubecon, the view was that unless you are a large organization seeking to re-platform your application portfolio, why bother with PaaS when these other tools are available?