REPORT REPRINT

The 451 Take on cloud-native: truly transformative for enterprise IT

MARCH 15 2019

By Jay Lyman, Fernando Montenegro, Matt Aslett, Owen Rogers, Melanie Posey, Brian Partridge, William Fellows, Simon Robinson, Mike Fratto, Liam Rogers

Helping to shape the modern software development and IT operations paradigms, cloud-native represents a significant shift in enterprise IT. In this report, we define cloud-native and offer some perspective on why it matters and what it means for the industry.

451 Research®

In this report, 451 Research presents our definition of cloud-native and the key technologies and methodologies that are representative of the trend, including containers, Kubernetes, service mesh and serverless. We recognize the importance of cloud-native based on our survey research and conversations with enterprise providers and end users. Containers and serverless are among the top IaaS features in use and planned for use, according to our Voice of the Enterprise: Digital Pulse, Budgets & Outlook, 2019 survey. Cloud-native technologies and methodologies – a departure from monolithic applications and waterfall release processes – are being driven by a desire for speed, efficiency, and support for applications and services that are distributed across hybrid infrastructure such as public clouds, private clouds and on-premises environments. There are, nevertheless, significant challenges with cloud-native approaches, mainly around complexity and lack of available skills and experience. Indeed, access to talent is becoming a key constraint for enterprises transforming around cloud and cloud-native (see Figure 1 below).

We expect the cloud-native trend to continue to grow, fueled in part by intersections with adjacent technologies and trends, including data and analytics, AI and ML, security, and IoT/edge computing – all of which play a role in facilitating digital transformation. We also expect the cloud-native market, populated by a burgeoning number of startups, as well as established giants, to undergo consolidation as vendors seek to gain talent and the market matures.

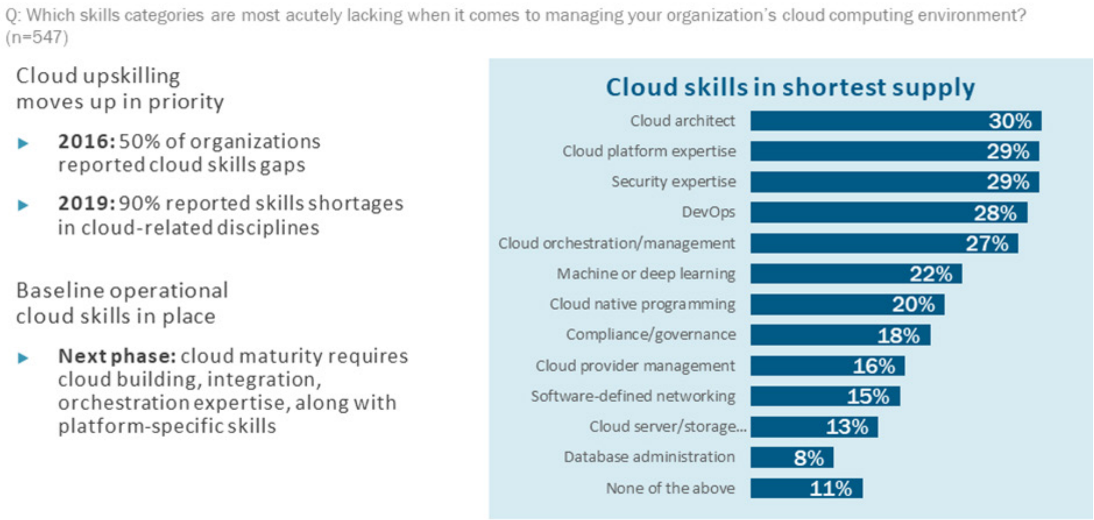
451 TAKE

Just like DevOps, cloud-native technologies and methodologies are now being attached to digital transformation efforts, and are expanding their presence in enterprise IT. Right now, the cloud-native trend consists mainly of containers, microservices, Kubernetes, service mesh and serverless, but we may see intersections of these different approaches with adjacent trends, as well as new ones. In addition to application development and deployment in the public cloud, cloud-native is connected to private and hybrid clouds and the ability to run applications consistently across different IT environments. Kubernetes, for example, is not only a container management and orchestration software, it is also a distributed application framework – one that is timed well with enterprise use of hybrid environments that span multiple clouds, as well as on-premises infrastructure.

Cloud-native software is also closely intertwined with open source. Nearly all of the key software components are open source projects, and we believe open source to be table stakes for cloud-native software. There is still ample commercial opportunity around cloud-native. We would also highlight that cloud-native is also not limited to public cloud platforms, with on-premises environments increasingly serving as the basis for cloud-native approaches. We also see cloud-native crossing over with adjacent trends. Such intersections and integrations bode well for continued growth and significance of cloud-native approaches. It remains to be seen which approach within the cloud-native arena will be most effective and which combination of different technologies paves the best path forward for enterprise and service-provider organizations, but we will continue to track the technology, use cases and impact of cloud-native going forward, including survey data, market sizing and other research.

Figure 1: Cloud Skills Gaps – Roadblock to Optimized Cloud Leverage

Source: 451 Research, Voice of the Enterprise: Cloud, Hosting & Managed Services, Organizational Dynamics 2018



Q: Which skills categories are most acutely lacking when it comes to managing your organization's cloud computing environment? (n=547)

Cloud upskilling moves up in priority

- **2016:** 50% of organizations reported cloud skills gaps
- **2019:** 90% reported skills shortages in cloud-related disciplines

Baseline operational cloud skills in place

- **Next phase:** cloud maturity requires cloud building, integration, orchestration expertise, along with platform-specific skills

Cloud skills in shortest supply
Cloud architect 30%
Cloud platform expertise 29%
Security expertise 29%
DevOps 28%
Cloud orchestration/management 27%
Machine or deep learning 22%
Cloud native programming 20%
Compliance/governance 18%
Cloud provider management 16%
Software-defined networking 15%
Cloud server/storage... 13%
Database administration 8%
None of the above 11%

Cloud-native defined

451 Research defines cloud-native software as: applications designed from the ground up to take advantage of cloud computing architectures and automated environments, and to leverage API-driven provisioning, auto-scaling and other operational functions. Cloud-native architecture and software include applications that have been redesigned to take advantage of cloud computing architectures, but are not limited to cloud applications – we see cloud-native technologies and practices present in on-premises environments in the enterprise. We can also define cloud-native by the technologies and approaches that characterize the trend, all intended to make software development and deployment more fluid and composable – containers, microservices, Kubernetes, service mesh and serverless.

Our research and conversations indicate that these different types of cloud-native application development and deployment are by no means exclusive in enterprise organizations, which are typically leveraging multiple cloud-native technologies and methodologies across their many different releases and teams. Rather than competing components, tools and methods, the different technologies of cloud-native software are similar to hybrid cloud, which is representative of a best-tool-for-the-job or Best Execution Venue (BEV) approach. We also contend that cloud-native is far broader than application development and deployment. Cloud-native also includes application and infrastructure architecture and organizational approach.

From an economic point of view, cloud-native technologies enable the true value of cloud by allowing applications to scale and evolve in much shorter timelines than previously. This scalability creates new opportunities for the business in terms of revenue growth, efficiency improvements or a better customer experience. However, cloud entropy means that scalability leads to great complexity, which is where the likes of Kubernetes, Istio, Prometheus and others come into play. The raison d'etre for these open source components is to keep track of the fluid and complex deployments of cloud-native services.

In terms of applications, we see cloud-native methodologies and technologies used for a breadth of both internal and consumer-facing applications, led by data services and analytics applications, IT optimization and automation, digital user enhancement, and industry-specific software.

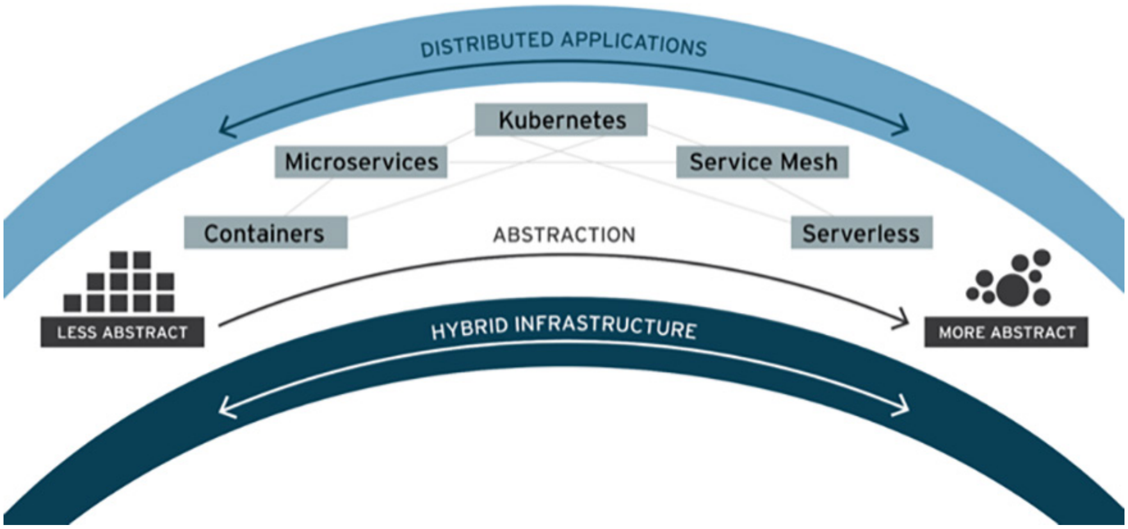
The Spectrum of Abstraction

Contrary to the narrative that 'serverless is killing containers,' we don't see the different approaches and technologies within cloud-native technology competing with or eliminating one another. The same way that containers are living alongside, and sometimes inside of, VMs is indicative of how all of the different aspects of cloud-native will coexist in a mixed-use market. No, serverless is not killing containers; serverless is built on containers. The main distinction between the two is the level of abstraction provided to the end user. Thus, we can also describe cloud-native as a set of technologies that fall somewhere on what we call the Spectrum of Abstraction.

Figure 2

Source: 451 Research, LLC

The Spectrum of Abstraction



On one side of this spectrum is the DIY containers approach, whereby organizations leverage custom code and services and make their own choices on languages, frameworks and APIs. This approach is attractive for certain applications that require low latency, that run longer compute jobs, and for which high traffic can be predicted. On the other end of the spectrum, as functionality becomes more abstracted and invisible, are serverless functions and events, for which there are standardized and opinionated choices that are abstracted away from the end user. In between these two ends of the spectrum are still other levels of abstraction, such as supported Kubernetes distributions and container-as-a-service offerings from the large public cloud vendors and others.

We typically see these different cloud-native technologies adopted in a specific order, starting with containers used for microservices, which break applications into smaller, loosely coupled services; then Kubernetes container orchestration and distributed application management for container clusters; followed by service mesh to abstract for developers and serverless to abstract for IT operators. However, we do see mixed use of the different approaches, and a leap whereby interested customers can skip ahead is feasible. For example, overheard at Kubecon/CloudNativeCon 2018 was the idea that organizations might be able to skip containers, microservices and the complexity of Kubernetes by simply adopting serverless. The reality is not that simple for most enterprise and service-provider organizations, which are more likely to be using the different technologies concurrently.

There is some interesting tension between different approaches that are still playing out in the marketplace – for example, advocates of 'single platform' approaches to cloud-native, such as OpenStack, Pivotal/Cloud Foundry or Red Hat, versus loosely coupled models that will be composed of different coordinated parts. Both require a specific organizational model, and the success or otherwise of each has yet to be determined – enterprises are still undergoing transformation.

Cloud-native isn't only in the cloud

Cloud-native does not necessarily mean applications run only on private or public cloud infrastructure. The hybrid cloud trend, which entails the use of a mix of public and private clouds with on-premises environments, dictates that enterprises will seek to run cloud-native applications atop on-premises infrastructure, as well. Vendors have responded aggressively with offerings such as Azure Software Stack, GKE on Prem and AWS Outposts. PaaS vendors, such as Red Hat with OpenShift or Pivotal with PCF, have also focused on the ability to run applications consistently across public clouds and on-premises infrastructure. In fact, our recent Voice of the Enterprise: Servers and Converged Infrastructure, Vendor Evaluations 2018 survey indicates continued growth of x86 servers and on-premises environments, with nearly one-third of organizations anticipating an increase in their x86 server deployments in the coming year.

Further evidence of the ties between cloud-native and hybrid cloud can be found in our Voice of the Enterprise: Cloud, Hosting and Managed Services, Workloads and Key Projects 2018 survey, which indicates that most cloud-native software (32%) is designed to run effectively on any cloud environment, with another 22% designed to run effectively on any public cloud environment, rather than for a specific public cloud (30%) or private cloud (17%).

Cloud-native with adjacent trends/sectors

Data, AI and ML

The dynamism that cloud-native architecture and containers provide is ideal for stateless web applications, but it can be problematic for stateful database workloads, given the need for a persistent connection between the application and its associated data volume. Kubernetes, in particular, has been at the forefront of containerization of stateful services, providing elements for persistence and cluster lifecycle management that enable custom deployments for individual databases that could be the beginning of a viable long-term approach. Database vendors are beginning to update their products to take advantage of these features. However, inherent challenges remain in getting databases and containers to work together, and vendors, enterprises and industry consortia must work together to continue to evolve Kubernetes, in order to provide a general-purpose environment for the containerization of multiple stateful services.

Cloud-native methodology and software are also crossing over with artificial intelligence and machine learning, including integrations of TensorFlow, an open source machine learning library, and projects such as Kubeflow for machine learning on Kubernetes. The combination enables data scientists to create and train models in self-contained environments with the necessary data and dependencies; these can then be deployed into production via Kubernetes, which provides autoscaling, failover, and infrastructure monitoring and management, as well as exaction venue abstraction.

Security

Increased adoption of cloud-native technology and delivery patterns will deeply influence how organizations think about security, even as key security principles, such as the need to maintain confidentiality, integrity and availability, remain. The scope of changes will affect both security technology and practices. On the technology front, the key cloud-native technologies (containers, Kubernetes, service mesh and others) have incorporated some security functionality themselves – service mesh supports workload identity and encryption, while Kubernetes includes several policy constructs. This will affect organizations deploying these technologies, as well as vendor offerings, since that functionality becomes the reference point for additional functionality and design decisions. Particularly as organizations adopt high-level services and abstractions (containers as a service and serverless), the focus of security shifts much more to application-level security and data security. This is a shift away from traditional infrastructure security considerations. Lastly, the quickened pace associated with cloud-native deployments will deeply affect security teams – not only will they need to skill up in cloud-native technologies and patterns, but the very pace of deployment will require teams to rethink how they interact with the rest of IT, and what role security can actually play.

IoT and edge computing

While the timing of their arrival on the IT scene was coincidental, it's as though containerization and IoT were born to be together as the match between capability (containers) and need (IoT app developers). The trends are well-aligned as the IoT industry matures, scales, and requires a complicated tapestry of computing venues depending on context and use case.

We believe the successful future of IoT is linked with timely adoption of cloud-native techniques to support the speed and diversity of IoT apps. The reality that a nontrivial portion of IoT apps will actually fail means reducing the cost of doing so is a high priority, and there is a need for iterative updates to software based on feedback from 'the field.' There is also a requirement for a small operating system footprint for low-power edge devices; support for microservices to enable the data- and messaging-intensive characteristics of IoT across and within multiple actors; and platform-independent runtime support using container technologies and orchestration to ensure that workloads are run on the optimal computing platform at the edge, near edge or centralized core.

Networking

There are still significant challenges to cloud-native networking, whether in a cloud service, an on-premises or colocation cloud environment, a virtual machine-based cloud, a container-based cloud, or a mix of services and on-premises. Enterprise IT prefers consistency in capabilities, but cloud-native environments have basic networking capabilities that established networking vendors have been attempting to address by integrating their switch and management software with the container environment and the container management framework. These products unify networking workflows and are familiar to IT, but can also inhibit IT from moving past its traditionally managed infrastructure, which is rigid and slow to adapt to changes. Layer on top service mesh, which offers a more robust technology for cloud-native infrastructure and provides a useful abstraction between application connectivity and the physical or virtual paths interconnecting software and hardware, and much of the intelligent networking capabilities in the physical underlay become irrelevant in the application layer.

There are opportunities for application delivery controller (ADC) vendors that can deeply embed themselves into enterprise IT by offering to offload a number of critical capabilities from application owners, such as intelligent load-balancing, high availability and security functions, to purpose-built platforms that can augment applications and keep developers focused on building features versus infrastructure. ADC vendors are also finding ways to embed their products into application infrastructure by enabling scale-out architecture via robust APIs and replacing container environment components like the ingress controller to a container pod.

Storage

There is a shift in how storage is being run as both startups and established vendors offer more storage capabilities (ranging from the storage controller to the backup application) in containers. The alternative is to have them run in VMs, as one would find in HCI-style deployments, or on a dedicated operating system like in proprietary appliances. This brings new flexibility to storage management since the various capabilities of storage platforms can be orchestrated and automated using the same tooling as the applications they are supporting.

Another consideration in the storage industry is providing containerized applications with storage as vendors evolve their offerings to take into account Docker volume drivers and Kubernetes Container Storage Interface drivers to support flexible storage consumption for containerized, stateful applications. This will be increasingly important as containers are used for stateful applications, whether they are net new or traditional and legacy apps that are being containerized for use in the cloud.

Heavily open source

Considering the most successful software components of cloud-native, open source software is a critical part of the trend. Nearly all cloud-native software components are open source, including Docker containers, Kubernetes management and orchestration, Helm package management, Prometheus monitoring, Istio service mesh, and Knative serverless. It is also noteworthy in the context of cloud-native that modern open source software projects and communities include not only vendors, but also end users, which are among project supporters and sponsors in the cloud-native market. The open source nature of cloud-native also means that traditional rivals, such as Microsoft and Google or Pivotal and Red Hat, work together on many of these open source projects in the cloud-native ecosystem. Cloud-native is also all about collaboration, meaning it must accommodate DevOps by offering something for developers and IT operators, as well as other stakeholders, including security teams, data analytics and data science teams, and line-of-business leaders.

Cloud-native competition and outlook

The industry is moving toward containers, microservices, Kubernetes, serverless and other cloud-native constructs. While there are other flavors available, Kubernetes has the wind in its sails and has all but won the battle for container orchestration. Many cloud-native entrants have a 'Kubernetes first' posture in terms of platform architecture and service delivery. Incumbent vendors, service providers and integrators are rewriting and retooling for cloud-native. Cloud-native is a part of every conversation with customers. Most enterprises are already working at some level with cloud-native constructs and exploring what new outcomes can be achieved. Every company is becoming a service provider – seeking to better engage with customers, partners, and suppliers with new digital services and experiences, and to compete in the digital economy. Companies will need to raise their software IQ, and cloud-native will be the basis of this, supported on the cloud operating and delivery model. Cloud-native practices such as CI/CD enable companies to access speed and agility not previously available, and will require new organizational approaches to development.

With many vendors across the different subsegments (containers, Kubernetes, service mesh and serverless), we expect further consolidation of the market. The need for cloud-native talent and expertise – our VotE survey data indicates cloud functions/tools such as containers and microservices are among the most acute skills shortages – will also likely drive mergers and acquisitions in the space. However, it may take some time since different enterprise and service-provider customers have very different needs, and thus support a broad array of providers in the market. The cloud-native market is highly competitive, with no dominant player yet established, although the hyperscale public cloud providers and large vendors that embraced containers early on are the clear leaders.

We also expect that, driven largely by digital transformation and the need to embrace and leverage new technology, cloud-native approaches will more deeply permeate large enterprise organizations. Similar to the DevOps trend, this means increasingly pulling in additional stakeholders, including administrators and line-of-business leaders. This means cloud-native technology and methodology will probably follow the pattern of agile and DevOps to reach half or more of organizations within the next few years. It is also important to note that the concept of cloud-native was meant to mean more than containers, Kubernetes or serverless, leaving room for the next technology, which may be a combination of existing ones; integration with adjacent trends, such as DevSecOps, data analytics, AI and ML; or something currently unknown.