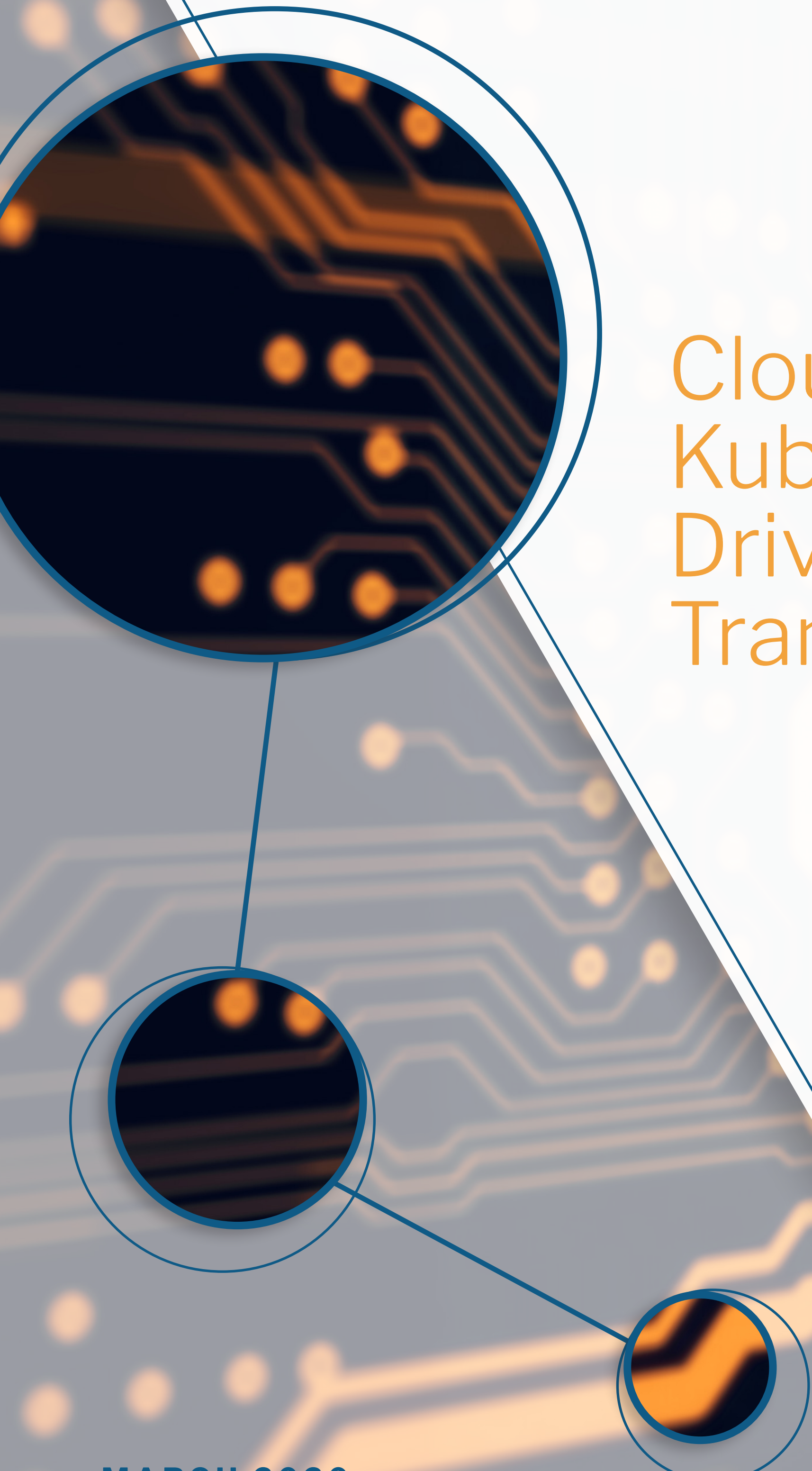
451 Research
PATHFINDER REPORT



Cloud Native, Kubernetes Drive Digital Transformation

COMMISSIONED BY

vmware

MARCH 2020

About this paper

A Pathfinder paper navigates decision-makers through the issues surrounding a specific technology or business case, explores the business value of adoption, and recommends the range of considerations and concrete next steps in the decision-making process.

ABOUT THE AUTHOR

JAY LYMAN

PRINCIPAL ANALYST,
CLOUD NATIVE AND DEVOPS

Jay Lyman is a Principal Analyst with 451 Research's Applied Infrastructure & DevOps Channel. He covers infrastructure software, primarily private cloud platforms, cloud management and enterprise use cases that center on orchestration, the confluence of software development and IT operations known as DevOps, Docker and containers. Jay's analysis encompasses evolving IT operations and software release models, as well as the technology used to create, deploy and support infrastructure and applications in today's enterprise and service-provider markets. Key areas of research also include OpenStack, PaaS and enterprise end users.

Introduction

Today's modern enterprises can drive high-impact digital transformation via technology resources such as cloud computing and emerging methodologies for application development and deployment, including cloud-native approaches like containers, Kubernetes and service mesh. Nevertheless, adoption of these new technologies can bring significant challenges, including operating efficiently at scale; a growing complexity and diversity of technology components to monitor, manage and secure; and cultural challenges that center on efficient organizational collaboration and 'legacy' enterprise IT inertia.

This paper examines how cloud-native technology and practices enable organizations to effectively leverage microservices to achieve greater development velocity and greater overall efficiency, and set the stage for successful digital transformation. It is aimed at both enterprise IT decision-makers and technology practitioners, so they can better understand the drivers, challenges and perceptions that are shaping cloud-native approaches that include Kubernetes and service mesh.

From Monolithic AppDev to Microservices

The shift from traditional application development to 'cloud native' centers on the transition from monolithic applications to an approach that relies heavily on microservices – a software development method that breaks large applications into lightweight and modular parts, often in containers, that can scale more simply and horizontally. A microservices approach segments application functionality into composable building blocks held together via RESTful APIs.

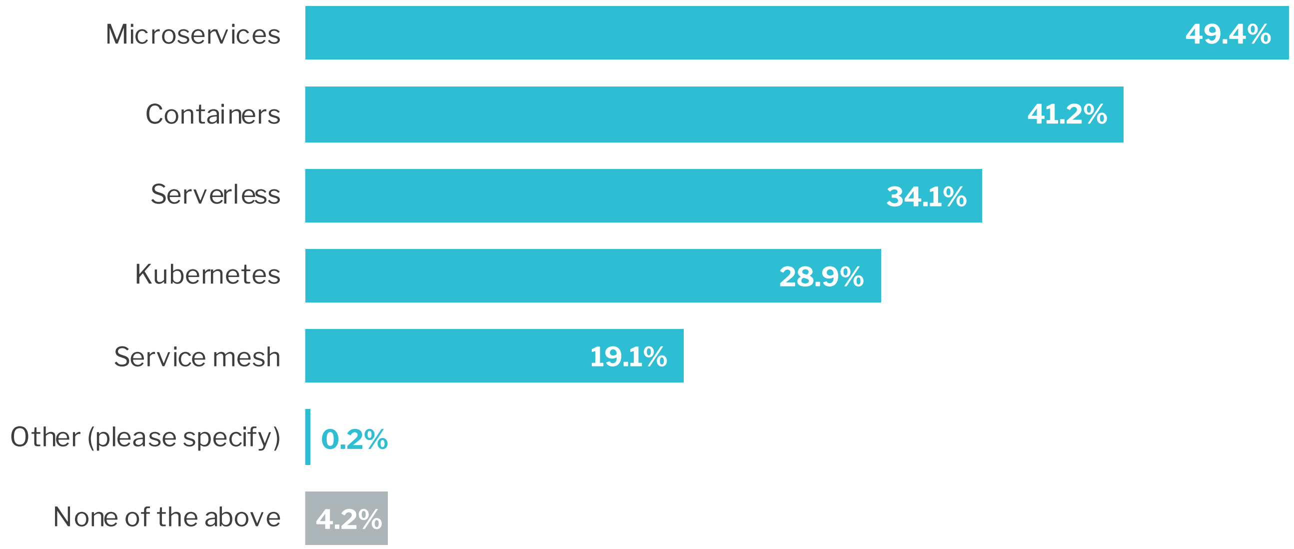
By breaking down monolithic applications from their massive, complex internal architectures into numerous smaller individual services that can be independently scaled, microservices can make development, as well as updating and deploying, of applications less complex and therefore more efficient. Additional advantages to using microservices include efficient workload portability across hybrid IT infrastructures, increased modularity and enhanced management benefits. Nevertheless, microservices present unique challenges, such as increased difficulty of monitoring and event logging, as well as performing testing and debugging processes across a decentralized, loosely coupled application environment. Other challenges include disjointed security policies and event logs, and the diverse, fragmented libraries and code bloat typical of today's polyglot programming.

DevOps Driving Cloud Native, Digital Transformation

Our research shows a strong connection between cloud-native technology and enterprise DevOps implementation focused on speeding software development and releases while making IT operations more efficient. In our Voice of the Enterprise (VotE) DevOps H1 2019 survey of 483 enterprise IT decision-makers and DevOps practitioners, more than 90% of respondents said they believed cloud native was very or somewhat important to their organization's DevOps implementation. We also see enterprise DevOps teams now using a variety of cloud-native technologies, from containers and Kubernetes to serverless and service mesh (see Figure 1).

Figure 1 – Cloud-Native Technologies Critical to Enterprise DevOps

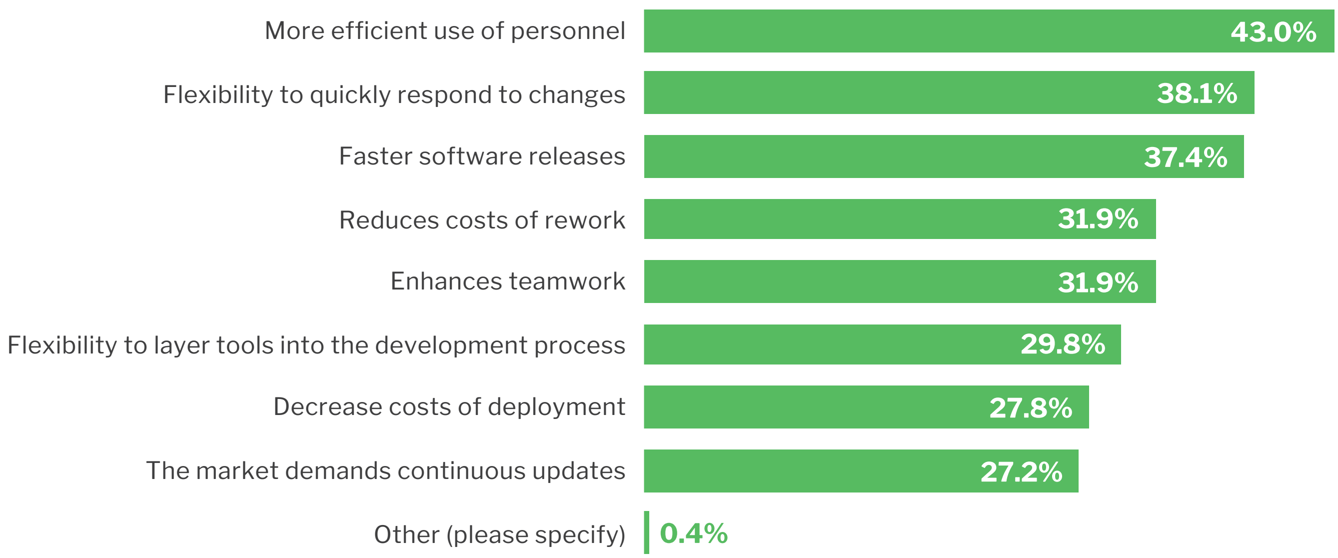
Source: 451 Research's Voice of the Enterprise: DevOps 2H 2019
Q. Which cloud-native technologies and methodologies are most critical to your organization's DevOps implementation? (Select all that apply)
Base: Multiple cloud native trends have some importance to organization's DevOps implementation (n=426)



The main reasons these DevOps teams are using containers, microservices, Kubernetes, serverless and service mesh center on developer onboarding and productivity, operational efficiency, and portability across hybrid and multicloud infrastructure that includes on-premises, private cloud and multiple public cloud environments. The benefits of cloud native are also consistent with the benefits of implementing DevOps, which include more efficient use of personnel, flexibility to quickly respond to changes, and faster software releases, according to our VotE DevOps Q3 2019 survey (see Figure 2). Other DevOps benefits include reduced costs of rework, enhanced teamwork, and flexibility to layer tools into the development process – all cases where cloud-native software such as containers, Kubernetes and service mesh can play critical roles.

Figure 2: Benefits of Implementing DevOps

Source: 451 Research, Voice of the Enterprise: DevOps 2H 2019
Q. How is a DevOps approach benefitting your organization? (Check all that apply)
Base: All respondents (n=486)



Just as there is a strong connection between cloud native and DevOps, we see both of these trends playing a significant role in enterprise digital transformation. The VotE DevOps Q1 2019 survey indicated that digital transformation 'leaders' were most likely to have all-sanctioned, centrally managed DevOps processes, while digital transformation 'learners' were most likely to have all-sanctioned but distributed management for DevOps. Digital transformation 'laggards,' meanwhile, were most likely to have some sanctioned and centrally managed DevOps, but some unsanctioned or distributed DevOps deployment.

This illustrates how DevOps has grown beyond a bottom-up trend driven primarily by developers. In today›s DevOps market, adoption is driven as much or even more so by company leadership, managers and central IT teams, which are focused on better serving developers, driving efficiency in IT operations and creating enhanced value for their organizations. Our research has found that 79% of organizations are at some level of DevOps adoption, further confirming the trend as an enterprise priority for digital transformation.

Challenges of Cloud Native

While it can contribute to successful DevOps implementation and digital transformation, cloud native also comes with its own set of challenges. Among the biggest issues for enterprises adopting containers, microservices, Kubernetes and other cloud-native technology and methodology is complexity. As an application is broken into microservices, to enable secure service-to-service communications, each service would need to have its own networking and security stack. Moreover, each development team would need to implement its own networking and security stack, and may not have domain expertise.

Keeping up with Kubernetes is hard enough with the myriad of projects that are used to support software like Kubernetes, including Helm package management, Prometheus monitoring, etcd data store and others. Then the developers have the burden of implementing a consistent network and security stack across the thousands of services that make up the application. This is a tall order.

Use of containers and microservices can also mean many points to connect, secure and monitor. Doing so consistently can be an even greater challenge, particularly in the face of application silos from multiple environments spanning on-premises, private and public clouds. Our research has also consistently showed a lack of DevOps and cloud-native skills and talent amid increasingly high demand.

Definitions

Cloud Native

'Cloud native' refers to 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 can also be defined by the different technologies and methodologies that fall under its umbrella: containers, microservices, Kubernetes, serverless and service mesh.

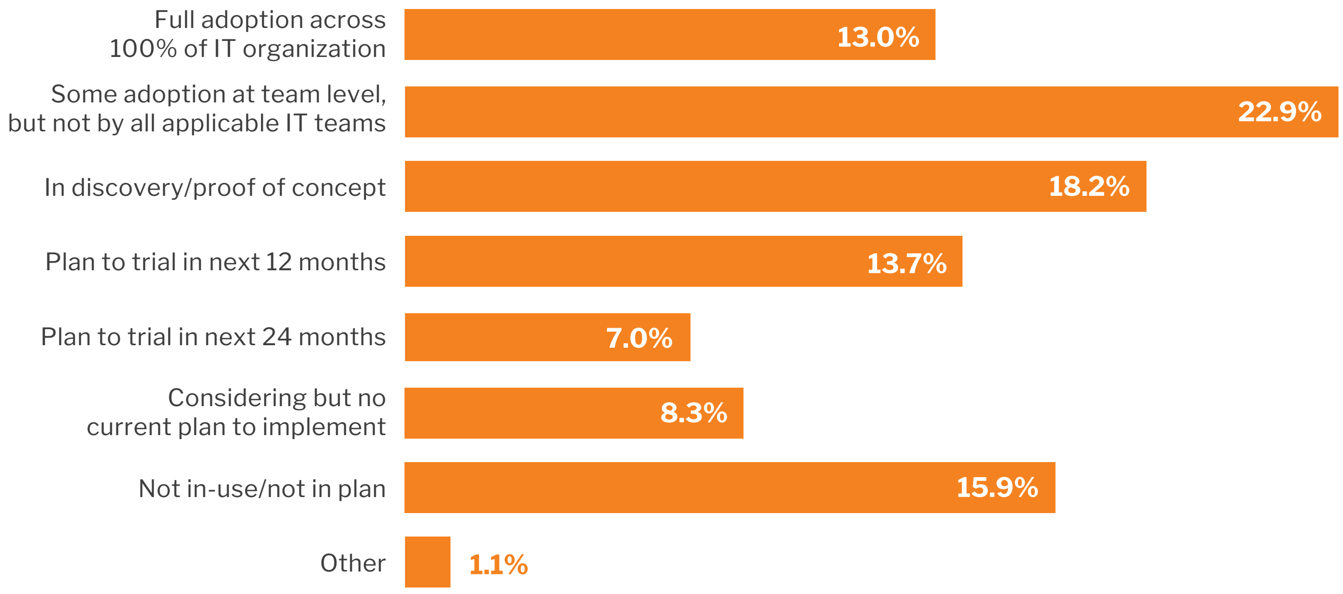
Kubernetes

Kubernetes is an open source container management and orchestration software suite that is also a distributed application framework. This makes it ideal for deploying and managing container clusters and for operating applications across hybrid and multicloud environments.

We see growing adoption of Kubernetes for effective management of containers and microservices, and to drive operational consistency across hybrid and multicloud infrastructures (see Figure 3). Although our research has found that 13% of organizations have Kubernetes already deployed fully across their organization and 23% have adoption among some teams, we still believe it is the early days for Kubernetes. This is especially true considering that a further 40% of organizations surveyed are in discovery/PoC or planning to deploy Kubernetes over the next two years.

Figure 3: Adoption of Kubernetes

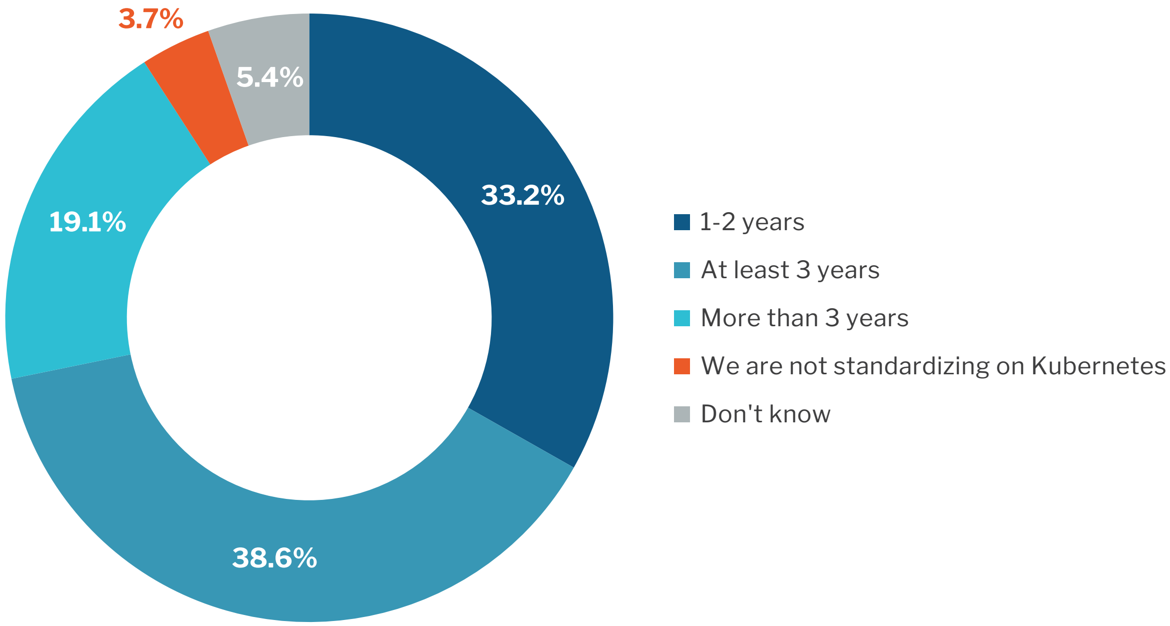
Source: 451 Research, Voice of the Enterprise: DevOps 2H 2019
Q. Please indicate your organization's adoption status for the following technologies: Kubernetes
Base: All respondents (n=446)



Our VotE DevOps Q3 2019 survey provided further evidence that Kubernetes is emerging as a standard operational paradigm in enterprise IT, with a vast majority of organizations indicating Kubernetes standardization plans (see Figure 4).

Figure 4 – Standardization on Kubernetes

Source: 451 Research, Voice of the Enterprise: DevOps 2H 2019
Q. If your organization is standardizing on Kubernetes, how long will it take?
Base: Organization has kubernetes in use (n=241)

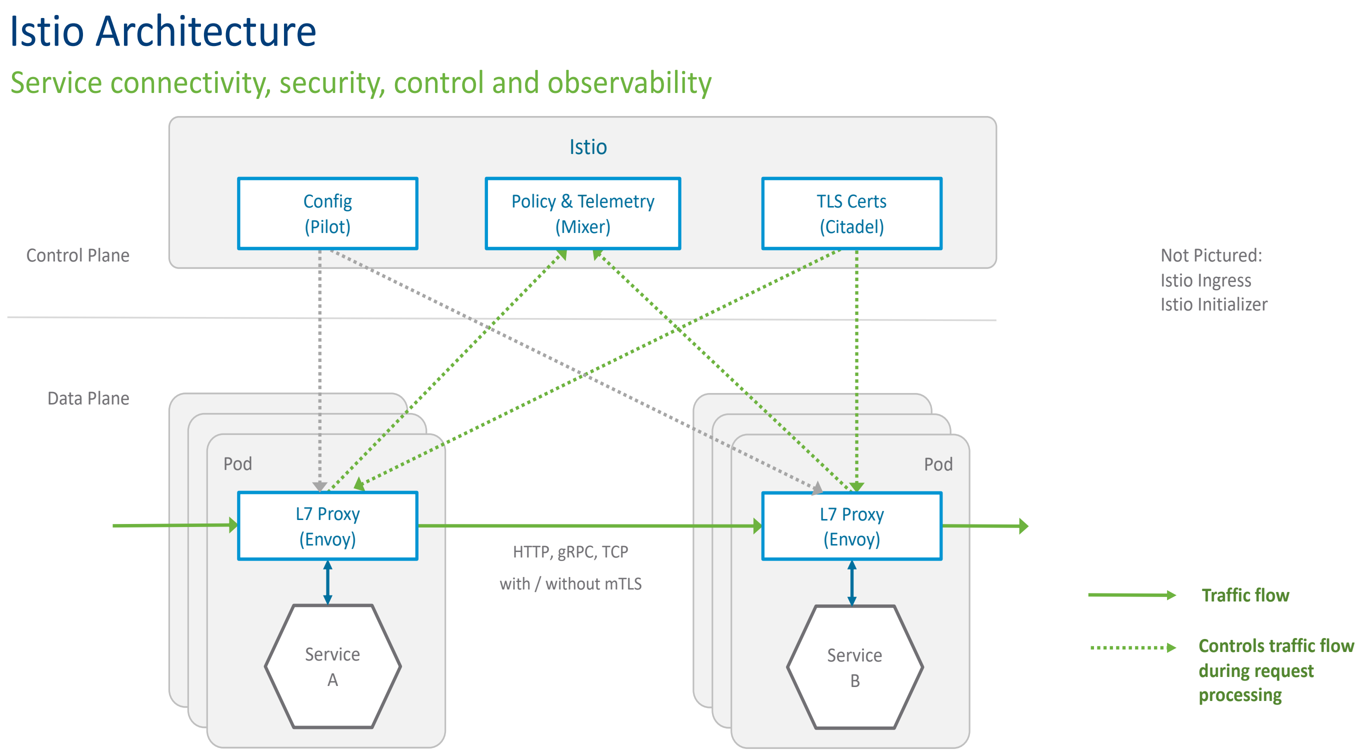


Service Mesh

Rather than communicating directly with each other, microservices rely on software called a service mesh or sidecar proxy for communication. Service meshes usually support some networking functionality, such as resiliency and service discovery. Some service mesh technologies also support traffic management, policy enforcement, observability, and security and identity management.

Figure 5: Illustration of Service Mesh Concept

Source: istio-releases.github.io



In effect, service meshes provide the abstraction layer with consistent networking and security controls, allowing developers to focus on building business logic rather than managing the infrastructure or environments in which it is running. In other words, using an e-commerce application as an example, the service mesh can become the consistent substrate enabling functions such as content services, order management services and payment processing to communicate.

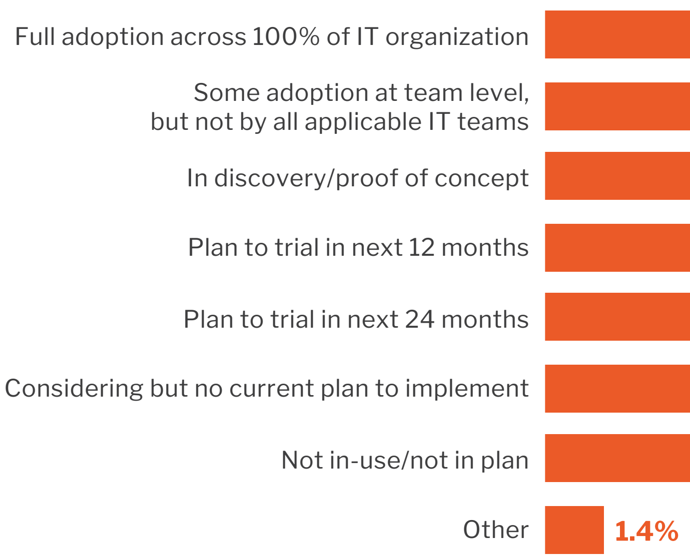
As with other cloud-native software such as containers and Kubernetes, open source software is typically an important part of a service mesh environment. Modern software development tenets such as modularity and componentization enable integration and flexibility with other technologies to achieve a service mesh. Beyond popular open source service mesh technologies such as Istio, Conduit, Envoy and Linkerd, there are additional open source software projects and components that can play a significant role in enterprise deployment of a service mesh. This ecosystem around service mesh and related cloud-native software can add to the overall complexity and confusion, but it also highlights the broad engagement and innovation among enterprise vendors and end users.

Service meshes such as Istio and the commercialized products built around them offer robust technology and useful abstraction to support the requirements of containerized environments and microservices. Service meshes are growing in importance as application infrastructure decisions are increasingly being made by software architects in lines of business who are building both applications and application environments.

Our VotE DevOps H2 2019 survey indicated remarkable adoption of service mesh given its fairly nascent status in the industry (see Figure 6). Nearly 15% of enterprise DevOps teams indicated that service mesh is deployed across the entire organization, while another 19% indicated team-level adoption. We also see a large number of organizations poised to move beyond discovery/PoC and planning deployment of service mesh over the next two years. Granted, our survey was focused on enterprise DevOps implementers, but our research indicates that DevOps teams and site reliability engineers (SREs) are among the primary users of cloud-native technology such as Kubernetes and service mesh. Meanwhile, only 11% of organizations surveyed said they are not using service mesh and have no plans to do so.

Figure 6: Adoption of Service Mesh

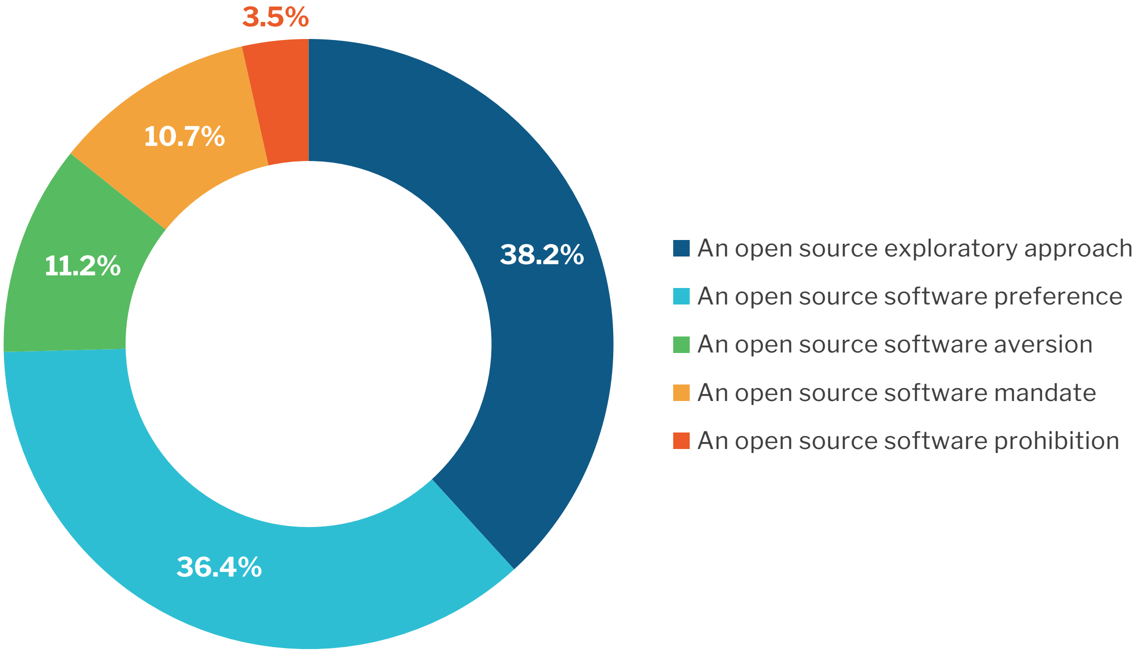
Source: 451 Research, Voice of the Enterprise: DevOps 2H 2019
Q. Please indicate your organization's adoption status for the following technologies: Service mesh
Base: All respondents (n=440)



We would also highlight the importance of open source software in cloud-native technology and methodology. Nearly all of the components of cloud-native projects are open source, including Kubernetes, Istio, Docker and many others. Open source software communities have expanded the scope of development of cloud-native technologies and are also offering enterprise customers a greater opportunity to influence features, priorities and roadmaps. Our VotE DevOps 2H 2019 survey indicated a prevailing open source software preference or mandate (47% of organizations) compared with an open source software aversion or prohibition (15% of organizations – see Figure 7).

Figure 7: Open Source Mandate or Preference Favored

Source: 451 Research, Voice of the Enterprise: DevOps 2H 2019
Q. What is your organization's attitude towards open-source software?
Base: All respondents (n=484)



Critical Elements, Features and Capabilities of Kubernetes

The growth of Kubernetes, a container orchestration and distributed application framework, is timed well with rising enterprise adoption of microservices, as well as multicloud and hybrid cloud infrastructure and strategy. With broad support from an array of providers in many areas – including agile and DevOps software development, data services, storage, networking and security – Kubernetes also represents a global market poised for more growth. As it increasingly becomes a standard for containers, microservices, distributed applications and hybrid cloud, Kubernetes is also gaining support and enhancements from a community of other open source projects, developers and backers. Integration is additionally being driven by other technologies and trends, such as service mesh, machine learning and data analytics.

With a growing ecosystem of providers and open source software developers and projects, Kubernetes represents more than open source container management and orchestration software – it is helping to define modern enterprise IT operations and orchestration. Kubernetes is also offered in a variety of forms, whether on-premises or in the cloud, through a CNCF-certified distribution, or simply via plugins and APIs for data services, networking, storage and other capabilities.

Kubernetes is actually a combination of open source projects. Born out of Google's intelligent container management, it was designed, in typical open source fashion, to be componentized and modular. Thus, there is a burgeoning ecosystem of additional open source projects within and around Kubernetes under the umbrella of the nonprofit Cloud Native Computing Foundation. This includes Helm package management, Prometheus monitoring, Harbor container registry, Grafana analytics and Jaeger distributed tracing. Further highlighting the more prominent role of large enterprise users, many of these Kubernetes-connected projects are open source and born out of end-user experiences and innovations from companies such as Lyft, Netflix, Ticketmaster and Uber.

Among the critical technical aspects of Kubernetes is the concept of Operators, which are software extensions for packaging, managing and deploying Kubernetes applications and clusters. Operators, which are managed with Kubernetes tools such as kubectl, exemplify the extensibility of cloud-native software like Kubernetes, allowing integrations via application programming interfaces (APIs). The Operator Framework is an open source project that serves as the runtime for Kubernetes Operators and comprises a variety of tools that include a software developer kit (SDK), metering and lifecycle management. There is also a growing variety of popular Kubernetes Operators for alerting, auto-scaling, cloud services, continuous deployment, continuous integration, distributed databases, data store, messaging, networking, secrets management, security and other tasks supported by both open source projects and commercial vendors.

Given the number of vendors involved and its importance in digital transformation and hybrid cloud, Kubernetes is somewhat deserving of the intense level of hype and attention around the software project, open source community and commercial market. We expect drivers such as developer speed, operations efficiency, portability and consistency to fuel further Kubernetes growth and visibility. Nevertheless, vendors and end users alike will have to contend with key challenges, including complexity, lack of skills and integration with existing infrastructure such as data management, storage and networking.

Today's Kubernetes market looks very similar to what we previously saw with OpenStack: Many enterprise and service provider end users have attempted to roll out Kubernetes on their own, but have run into difficulty with upgrading, routine monitoring, storage, networking connections and overall complexity. Thus, we expect customers will increasingly seek out vendor-backed Kubernetes distributions and support. We may also see customers switching out different vendors as they go through the process of adopting and integrating Kubernetes with their existing infrastructure and operations. We would also highlight integrations of Kubernetes with other leading-edge technology such as machine learning, artificial intelligence, data analytics and edge computing/IoT, which are likely to grow over time.

Critical Elements, Features and Capabilities of Service Mesh

Service mesh can not only be an enabler of microservices and net-new cloud-native application development and deployment; it can also support enterprise migration of legacy applications, such as those written in Java, to the cloud and the modernization of applications for the cloud – often through container technologies. While enterprises have embraced containers and microservices for primarily cloud-native net-new applications thus far, the overall approach is being extended for use with legacy applications in the hope that they, too, can be supported more efficiently and effectively in the cloud. Service mesh can also accompany and complement the use of Kubernetes container orchestration and distributed application frameworks, which is well timed with the growing enterprise use of multicloud and hybrid cloud architectures.

There is a distinction between service mesh and networking, as well as potential synergies with network and infrastructure-layer innovations such as network virtualization, software-defined networking (SDN) and software-defined storage (SDS). While a service mesh typically provides some network benefits (e.g., resiliency and service discovery), it is not designed to address the wide spectrum of networking challenges at all layers (L2 to L7) of the software stack, such as automating security. We believe that best practices dictate a layered approach to supporting dynamic application environments. This approach should leverage service mesh along with other technologies, such as SDN and SDS, to address the different problems at different layers – which also vary based on the needs of stakeholders including developers, IT operators, combined DevOps teams, network administrators, storage administrators and security administrators.

Service mesh software and methodology encompass key elements such as control plane components for the management of microservices, data plane components that transport data along with applications, APIs and integration points. As for service mesh features and capabilities, customers generally view simplified workflows, granular troubleshooting and distributed tracing tools as among the most critical. Additional service mesh capabilities center on: discovery and 'just knowing what's there'; federation for establishing consistency amid the complexity; observability/visibility through logging, monitoring and alerting; connectivity and traffic management to avoid bottlenecks and outages; control and unified policy across services for easier management; and improved security capabilities to reduce overall risk exposure.

The foundational elements of service mesh are important for addressing microservices challenges around security and compliance, complexity, consistency and visibility. The benefits and challenges of microservices – and, thus, the utility of service mesh – can also vary greatly depending on the perception of the stakeholder. While IT leadership, directors and C-level individuals place a high priority on security and compliance, developers and IT administrators are more focused on resolving issues around complexity and inconsistency. Service mesh is typically top of mind across all stakeholders in matters of visibility and observability, highlighting how it serves as a central control and viewing plane for microservices.

Workload portability is also driving a shift to microservices, particularly among larger enterprises adopting and leveraging hybrid cloud and multicloud strategies that involve various public clouds, private clouds and on-premise IT environments. Additional microservices drivers include improving scalability, simplicity, faster deployment time, enabling modularity, and overall management benefits – all of which can be enabled by and supported via service mesh.

Istio and Other Service Meshes

The most widely known and adopted service mesh is Istio, an open source technology started by Google, IBM and Lyft, with growing involvement from IT vendors and end users. Istio is designed to allow its users to connect, control, secure and observe microservices. Core features of Istio focus on traffic management, platform support across hybrid infrastructures, integration and customization, observability and security.

Now nearing its 1.5 release, the Istio service mesh technology represents the latest and most stable version of the software. In addition to its focus on scaling high-volume and more complex microservices, Istio 1.5 is designed to let developers and operators also control microservices management, data and security. Additional enhancements in the works for Istio include unified support and experience across public clouds and on-premises environments.

While Istio is the most well-known service mesh in the industry, there are several other service mesh technologies of note, including some that are used within Istio. Envoy, for example, is an open source high-performance proxy used to manage inbound and outbound traffic for the services in a service mesh. Envoy serves as a primary component of Istio and other service mesh software. Another technology to understand is Calico, an open source network policy control plane that can be used with a service mesh to secure applications. Lastly, we'd note Conduit, which is a lightweight open source service mesh for Kubernetes that has its roots in the Linkerd service mesh technology.

While 451 Research's surveys and industry conversations indicate that service mesh is starting to be implemented for production applications and microservices, there are unique challenges to full-scale production use. These include lack of visibility, difficulties in troubleshooting outages, fragile service communication and the potential for cascading failures.

Cloud-Native Customers and Use Cases

Cloud-native software such as Kubernetes and service mesh still represent a fairly nascent and evolving class of technologies, but there are some consistent use cases that have emerged among enterprise organizations we have studied. One of the primary Kubernetes use cases involves managing containers and clusters at scale. Kubernetes is also used for consistent and consolidated management of applications across hybrid and multicloud infrastructures.

For service mesh, one key use case is service discovery – organizations need to understand things such as what services are running, whether and how services are communicating, and which data stores are being accessed. Service mesh can also aid in service discovery by simplifying service registry and enabling the generation of topology maps. Another common service mesh use case is to address management federation, typically in environments that use multiple platforms (IaaS, PaaS, SaaS, etc.) and multiple clouds and infrastructures (public, private, on-premises).

Other cloud-native use cases center on high availability as customers seek to distribute their applications across multiple clusters on a single cloud or across multiple clouds. Cloud-native technology and methodology are also being used to enhance security, drawing on capabilities such as access control and service encryption. Other cloud-native drivers in enterprise IT include observability and visibility for both developers and IT operations, as well as connectivity and traffic management.

As stated earlier, cloud-native technology is also a good match for enterprise DevOps deployments, with support for tasks and requirements of both developers and IT operations teams, as well as other stakeholders including architects, security professionals and SREs.

Conclusion and Outlook

Containers, container orchestration such as Kubernetes, service mesh, and other cloud-native technologies and approaches are integral to successful enterprise IT digital transformation. The promise of greater application agility, portability and scale for cloud-native applications, as well as for migration of existing and legacy applications, is driving rapid adoption of cloud-native technology and methodology, but there are unique challenges these approaches bring in terms of complexity, consistency and security.

Open source projects and commercial vendors are helping customers address these challenges and reap the benefits of emergent cloud-native technology and methodology. While it is still relatively early on, we see Kubernetes and service mesh evolving and maturing rapidly, helped along by the fact that they are being driven and guided by not only vendors, but also end users and the broader open source software communities and ecosystem that are the foundation of cloud native. As these architectures are deployed more broadly by a growing variety of enterprises, we expect cloud native to solidify its place as an effective approach to mitigating the challenges, and being a key enabler, of digital transformation.

CONTENT PROVIDED BY:

vmware

VMware Tanzu Service Mesh, built on VMware NSX, is an enterprise-class service mesh with a Data Plane and a Global Control Plane. The Global Control Plane is offered as a SaaS, with planned support for on-premises deployment. Tanzu Service Mesh extends control and service mesh capabilities to users, services and data.

Tanzu Service Mesh supports multi-cluster, multicloud and multiple workloads. Tanzu Service Mesh can be used with any Kubernetes clusters; it supports VMware PKS portfolio, EKS, upstream Kubernetes, VMware Tanzu Mission Control, vSphere with Kubernetes, GKE and other Kubernetes today. Tanzu Service Mesh provides an open, extensible plug-in framework to support third-party data integrations with security and infrastructure applications on which Tanzu Service Mesh can implement policy decisions.

Tanzu Service Mesh features a unique capability called Global Namespaces, whereby the namespace concept in Kubernetes has been elevated, enabling it to span across clusters and not be limited to a single deployment. Global Namespaces makes it possible to run distributed applications across Kubernetes clusters and on multiple clouds for OPEX, and across locations for customer proximity reasons. The Global Namespaces capability addresses the limitations of namespaces, providing strong isolation and the ability to implement policy independent of the cluster where it resides. While namespaces provide a way to divide cluster resources between multiple Kubernetes users, they do not provide strong isolation between two namespaces running inside.

DevOps/SREs can define a Global Namespace for a given application and then assign policies to it.

Global Namespaces can also be used for service upgrades. In addition, Global Namespaces can support use cases for multicloud application deployments, business continuity/high availability, progressive upgrades, advanced security policies and encryption between services to protect data in flight.

To learn more about the open source service mesh Project Istio, visit *istio.io* and to find out more about VMware's contributions into open multi-mesh federation, read our blog about *Project Hamlet*.

If you want to learn more about Tanzu Service Mesh or request for a technical demonstration, visit *https://www.vmware.com/products/tanzu-service-mesh.html*

About 451 Research

451 Research is a leading information technology research and advisory company focusing on technology innovation and market disruption. More than 100 analysts and consultants provide essential insight to more than 1,000 client organizations globally through a combination of syndicated research and data, advisory and go-to-market services, and live events. Founded in 2000, 451 Research is a part of S&P Global Market Intelligence.

NEW YORK
55 Water Street
New York, NY 10041
+1 212 505 3030

SAN FRANCISCO
One California Street,
31st Floor
San Francisco, CA 94111
+1 212 505 3030

LONDON
20 Canada Square
Canary Wharf
London E14 5LH, UK
+44 (0) 203 929 5700

BOSTON
75-101 Federal Street
Boston, MA 02110
+1 617 598 7200

451 Research®