Ten Priorities for Container Management and DevOps in Production and at Scale in 2018

EMA Top 3 Report and Decision Guide for Enterprise

ENTERPRISE MANAGEMENT ASSOCIATES® (EMA™) REPORT
PREPARED FOR MORPHEUS

Written by Torsten Volk, Q1 2018

IT & DATA MANAGEMENT RESEARCH • INDUSTRY ANALYSIS • CONSULTING

CONTENTS

ABSTRACT

DevOps and containers enable enterprises to deliver better customer value in a more cost-effective manner by institutionalizing an automated, continuous, and rapid development, build, test, release, and deployment process for software.

Based on primary research data obtained from 300 U.S. enterprises, this report presents ten key priorities for successfully leveraging containers in production and at scale, within a DevOps context. These priorities will be the foundation for the EMA Top 3 awards presented to the outstanding vendors in each container management-related software category.

This report focuses on providing guidance that will help enable enterprises to transform into digital attackers by leveraging containers and DevOps to serve their customers faster, better, and cheaper on a continuous basis.

Visit EMA Top 3 website.

CONTAINER MANAGEMENT AND DEVOPS AT SCALE:

Ten key priorities for 2018, based on quantitative data from 300 enterprise developers, IT operators, and business staff.

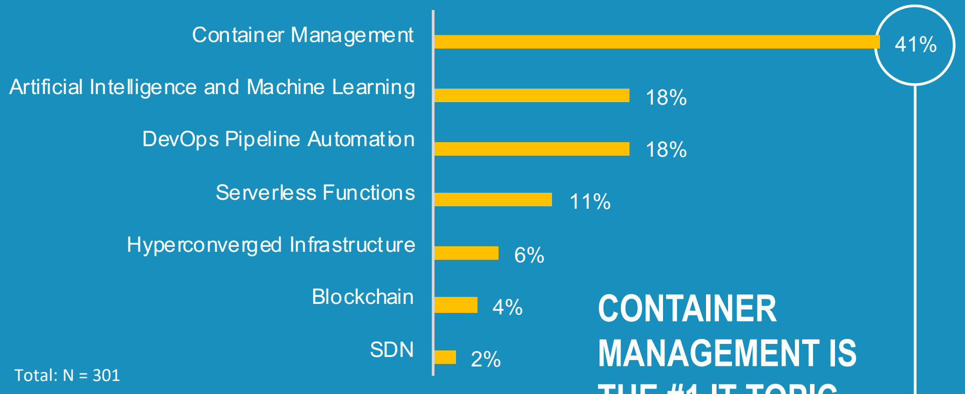
EMA TOP 3:

EMA presents its Top 3 award to vendors that are best aligned with today's customer priorities and pain points.

LEVERAGE CONTAINERS FOR ENHANCED DEVOPS:

Examination of how container technologies can help address today's DevOps challenges.

Which IT topic is most important for you in 2018?



CONTAINER MANAGEMENT IS THE #1 IT TOPIC IN 2018

TEN CONTAINER AND DEVOPS PRIORITIES FOR 2018

- Integration with existing enterprise IT
- Consistent security and compliance framework
- Getting IT operators back into the driver's seat
- Hybrid infrastructure management for containers
- Transitioning to hybrid apps and microservices
- Application-centric container management
- Transition to a declarative management paradigm
- Combining containers, PaaS, and FaaS
- Prioritization of applications for containerization
- Optimal Kubernetes management

WHAT ARE THE EMA TOP 3 PRODUCTS?

EMA Top 3 products address the ten key container management priorities in 2018 in a convincing and innovative manner. This report will inform and inspire influencers and decision makers in their project planning and vendor selection process.

EMA TOP 3 VENDORS ENABLE ENTERPRISES TO LEVERAGE CONTAINER TECHNOLOGIES TO BETTER SERVE THEIR CUSTOMERS.

This report is not a feature-by-feature comparison. In certain cases, EMA included products that are still in late beta or preview stages simply to recognize a vendor's excellent alignment with customer challenges. On the other hand, readers may miss more traditional products that did not make it into the report, because newcomers took their spot. EMA created this guide as a resource for enterprises to learn from their peers and benefit from hundreds of product briefings, cases studies, and demonstrations.

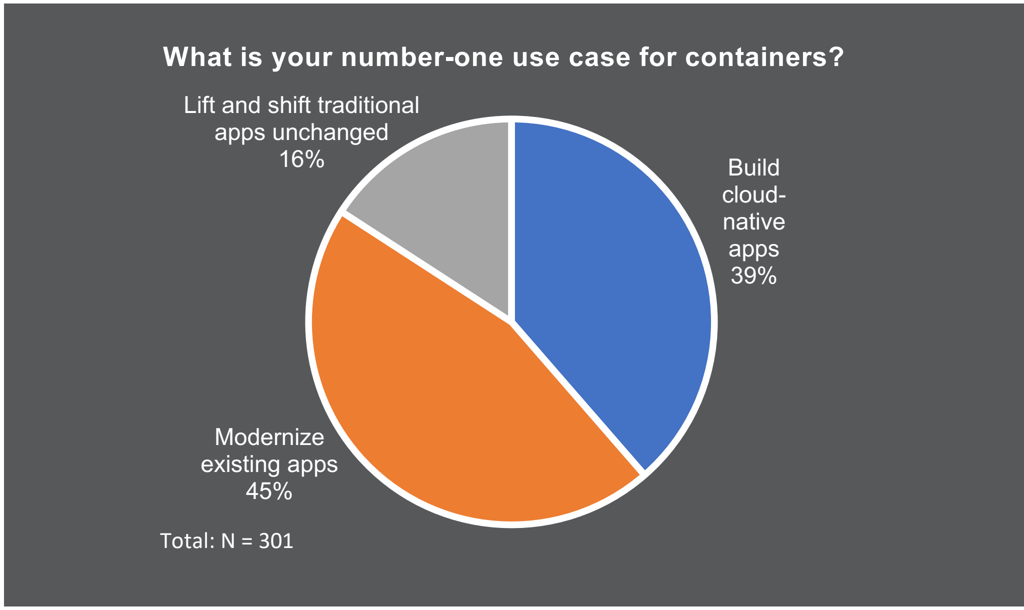
Container Momentum in 2018

In 2018, many developers and IT operators regard containers as a magic potion that enhances software delivery and lowers operations cost. This is expressed in the staggering 45 percent of survey respondents who indicated that in their opinion, containers will replace VMs within two years. However, when drilling down into customer experiences, priorities, and pain points, it becomes clear that in reality, containers are one more type of infrastructure and will coexist with VMs and other infrastructure types, such as bare metal, VMs, FaaS, and PaaS. Due to today's enormous market traction of containers, every software and hardware vendor today needs to have a clear container story and strong integration capabilities with container technologies. DevOps groups and enterprise IT are expected to leverage containers for faster, better, and cheaper software delivery and lifecycle management.

> **"The EMA Top 3 report gets its credibility from its empirical foundation. It provides me with insights on which vendors I might want to look at, without claiming to know what I should buy."**
>
> – Director, Application Platforms, Large University

Digital Transformation Must Aim to Increase Customer Value

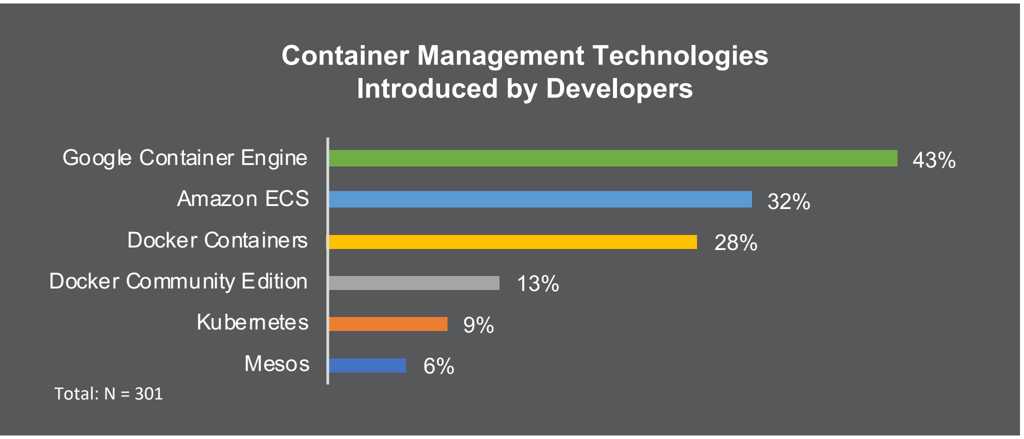
Digital transformation is often perceived as stale, with a connotation of long-term projects aimed to re-architect enterprise applications for cloud deployment. Viewing digital transformation as a series of technology projects misses the point, since the true goal is to increase the organization's capacity to facilitate continuous change and to rapidly and proactively address new customer requirements. Along with organizational, procedural, and cultural changes, containers are a critical technology component to accelerate this transformation.



DECISION MAKERS AND SELECTION CRITERIA FOR CONTAINER TECHNOLOGIES

Developers Introduced Containers to Make Life Easier

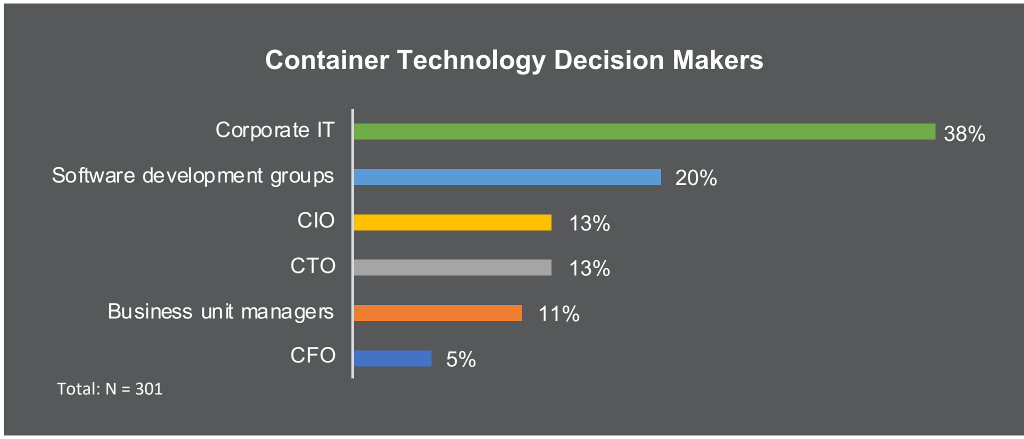
Initially, developers adopted containers to simplify application deployments and upgrades and to decrease the time spent on creating and troubleshooting test, staging, and production environments. Today, containers are an excellent fit for the transition to hybrid applications consisting of numerous microservices, each with its own development team and release lifecycle. Simply speaking, containers make it easy for each developer to deliver these microservices, including all of their dependencies, without deploying or debugging infrastructure.


Container Management Technologies Introduced by Developers

In 2018, Corporate IT is Taking Back Control

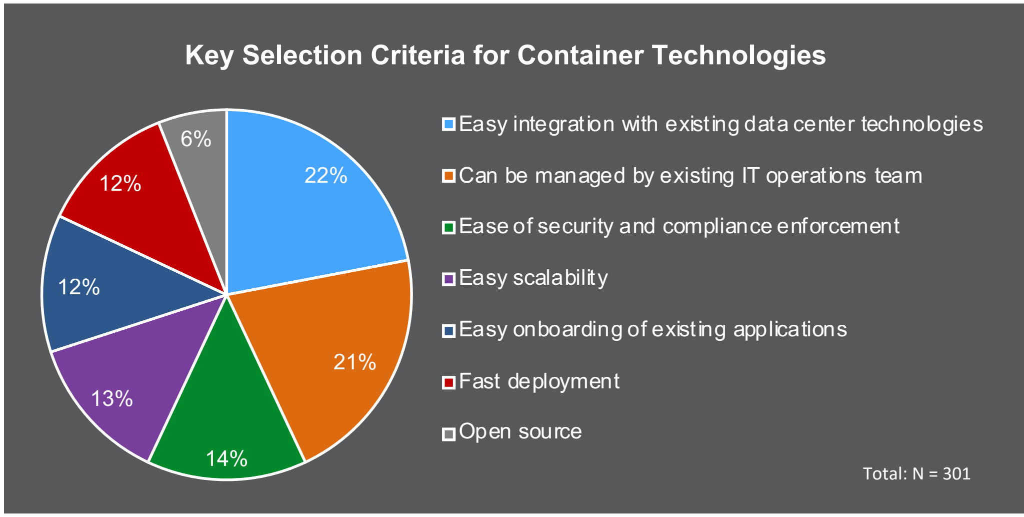
Seventy-two percent of organizations are still faced with the challenge of bringing unsanctioned container environments back under central control, in order to be able to ensure service-level agreements (SLAs), security, policy compliance, performance, and cost efficiency.

72% **OF ENTERPRISES ARE STRUGGLING WITH UNSANCTIONED CONTAINER ENVIRONMENTS.**


Container Technology Decision Makers

Integration, Simplicity, Security, and Compliance are Essential

Today's key container technology selection criteria are predominately focused on DevOps topics. Easy integration with existing data center technologies, container management by the existing IT operations team, and ease of security and compliance enforcement lead the list of selection criteria.


Key Selection Criteria for Container Technologies

CONTAINERS IN PRODUCTION AND AT SCALE

To use containers in production across the organization, even for business-critical applications, there is one central piece of technology debt that must be resolved: transforming “pets” into “cattle.”

Containers are Cattle

Containers are stateless entities that run on their host kernel without the requirement for custom changes to the operating system (OS). All runtime requirements and dependencies are built into the container images used to launch an application. Updating the application simply means replacing the containers already running with new ones derived from a new image. This eliminates the risk of updating applications, since multiple versions can live on the same container host. In case of problems, the updated container versions can be switched out for the previously working version without rebooting the OS.

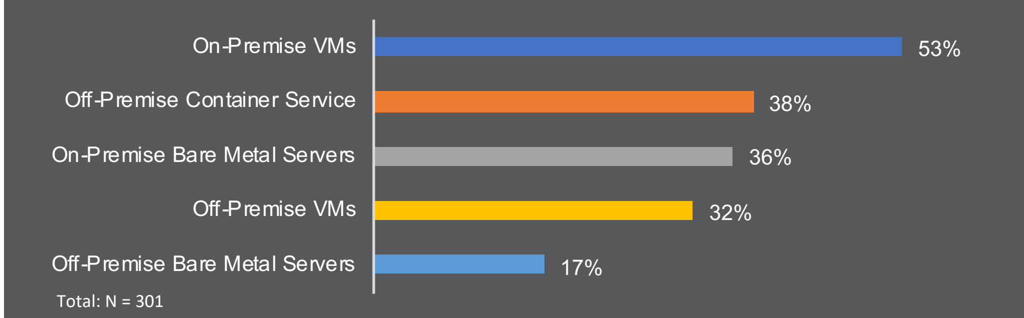
The Underlying Infrastructure is Still Based on Pets

While containers are managed like cattle, the underlying server, storage, network, and hypervisor infrastructure are typically still managed as pets with individualized deployment, configuration, and upgrade procedures. In order for IT infrastructure to constitute a scalable foundation for containers, the underlying hardware and virtualization components need to also be managed as cattle, allowing automatic mass deployments, upgrades, and terminations as required by the container scheduling platform.

Managed Containers are Only Part of the Solution

Many container vendors have recently launched container as a service (CaaS) offerings that provide IT operations and developers with an interface to manage containers without having to worry about the underlying virtual and hardware infrastructure. While this works well as a hosted service, data locality rules and compliance regulations often require applications and databases to remain inside the corporate data center. This makes on-premise CaaS deployments mandatory and still requires the enterprise to transform the underlying hypervisor, server, storage, and network infrastructure into cattle, so it can effortlessly scale up and down depending on container requirements. All ten priorities identified in this research report link back to solving this specific challenge.

The Majority of Containers Run on VMs Today



Total: N = 301

CONTAINERS ARE NOT INFRASTRUCTURE AWARE

The container host treats any performance or availability problem as a resource shortage, where the answer is to simply provide additional resources. Enterprise IT is left with the challenge to monitor whether the issue is due to legitimate application use, or if the problem is rooted in failures of the virtual or physical infrastructure.

CONTAINERS FOR DIGITAL ATTACKERS

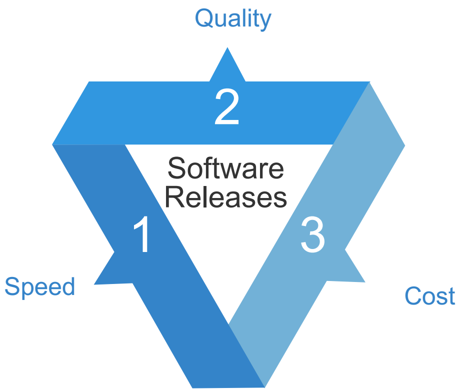
Digital attackers are companies that disrupt industries by continuously delivering better customer value compared to established incumbents. To win back customers, or to not lose them to digital attackers in the first place, incumbents need to complete their own digital transition by following the three core principles discussed below. Containers are the catalyst that can dramatically accelerate the transformation into a digital attacker.

62% **OF ENTERPRISES ARE DISSATISFIED WITH THE QUALITY, SPEED, AND COST OF THEIR APPLICATION RELEASES. THIS IS WHERE CONTAINERS COME IN.**

Principle #1: Deliver Better Customer Value Faster and at Lower Cost

Digital attackers must break through the traditional "vicious triangle" that describes the tradeoff between development speed, quality, and cost. The upfront investment into containers, pervasive release, and infrastructure automation is critical to facilitate a much faster release schedule that simultaneously absorbs less operator time and increases release quality by minimizing the potential for human error.

> **"Containers provide the flexible and scalable infrastructure layer to move loosely coupled components through the DevOps pipeline."**
>
> – Development Lead,
> U.S. Investment Bank



Principle #2: Minimize Infrastructure Operations Cost

Digital attackers have transitioned their IT processes into a state of centralized infrastructure, application, policy, and security automation where IT operators focus on creating and monitoring policies instead of completing manual provisioning and management tasks. Containers are no shortcut for this exercise, but they are an excellent platform to provide developers with the resources they require, while enabling operators to manage policies and SLAs.

> **"Lack of automation and automation silos were killing us, if not from a speed perspective, then through the added manual steps to ensure compliance."**
>
> – VP, Software Engineering, U.S. Grocery Chain

Principle #3: Experiment Often and Fail Early

Teams need to be able to rapidly test new capabilities and receive customer feedback. Containers are an excellent platform for experimentation and prototyping because new resources can be spun up and down nearly instantly and at minimal cost.

#1 **OPERATIONAL SIMPLICITY IS THE #1 REQUIREMENT FOR CONTAINER TECHNOLOGIES.**

OVERVIEW: TEN PRIORITIES FOR CONTAINER MANAGEMENT AND DEVOPS IN 2018

Based on responses from 300 enterprises, there are ten priorities around container adoption in 2018.

1

INTEGRATION WITH EXISTING ENTERPRISE IT

Easy integration of containers with existing data center technologies is the main selection criteria for container technologies today. Primary integration points are: hypervisor, storage, network, hyperscale clouds, server automation, and configuration management tools.

2

CONSISTENT SECURITY AND COMPLIANCE FRAMEWORK

Including containers into the corporate security and compliance framework is the primary container-related pain point today. Difficulties in this area originate from differences in container lifecycle management compared to managing traditional VM environments.

3

PLACE IT OPERATORS BACK IN CONTROL

The ability for current infrastructure, platform, and application operators to manage container environments is critical for most enterprises. Container adoption is rising fast and as more business-critical applications are containerized, enterprises seek scalability, security, compliance, and cost efficiency in container operations, which can only be achieved by leveraging existing IT operations resources.

4

SUPPORT HYBRID INFRASTRUCTURE

Enterprises mostly leverage multiple container services and 53 percent are running containers on VMs in the corporate data center. Twenty-six percent of enterprises are asking for their container management solutions to also manage FaaS.

5

OPTIMAL KUBERNETES MANAGEMENT

Kubernetes is today's de-facto standard container scheduling and orchestration framework, consisting of a set of services to deploy and manage containers in production and at scale. The key value of Kubernetes is that it provides developers with standard APIs that enable them to deploy their code, based on policies defined by corporate IT, without having to get involved in time-consuming infrastructure configuration.

TRANSITIONING TO MICROSERVICES

Microservices are functional components with standardized API interfaces that can be combined into applications or added to traditional enterprise applications. Each microservice can have its own development and operations team and release cycle. Microservices enable enterprises to rapidly provide new capabilities without requiring a major release of the entire application.

6

APPLICATION-CENTRIC CONTAINER MANAGEMENT

Application-centric container management provides policy-driven deployment, monitoring, alerting, scaling, updates, and upgrades of containerized applications, without the underlying infrastructure imposing restrictions. This means application-centric CaaS management offers an abstraction layer on top of different container services, enabling customers to freely choose between application deployments in the data center or public cloud.

7

TRANSITION TO DECLARATIVE MANAGEMENT

Declarative management relies on developers and operators defining the desired state of an application or infrastructure, instead of "scripting their way there." This approach relies on involving security and compliance experts, as well as most other development and operations roles throughout the application management lifecycle. Declarative container management constitutes as the foundation for rapid cross-platform scalability and portability, as the scheduler, e.g., Kubernetes, simply executes predefined policy rules.

8

LEVERAGING CONTAINERS, PAAS, AND FAAS

Sixty-two percent named developers needing more flexibility as their main reason for adopting containers on top of PaaS. The key reason for operators to add containers to their existing PaaS deployment was the easier onboarding of legacy enterprise applications. With many organizations aiming to eliminate all baggage that slows down code releases, FaaS will see a rapid rise in 2018. Ultimately, enterprises will require their IT operations tools to consistently manage VMs, containers, PaaS, and FaaS.

9

PRIORITIZE CONTAINERIZATION

Containers are offering a compelling new option to run applications in a scalable, portable, and cost-efficient manner. However, in order to prevent unnecessary increases in operational complexity, enterprises require a clear business case before containerizing an application.

10

EMA TOP 3: APPLICATION-CENTRIC MULTI-CLOUD ORCHESTRATION

QUICK TAKE

As the line of business piles on new IT requirements at a faster pace, and as new technologies become available in rapid sequence, corporate IT has to complete its automation homework to get back into the driver's seat. There is a constant need to deploy, move, scale, and terminate application platform services, especially in a world of dynamic microservices, hybrid apps, and small but frequent software releases. The entity to be managed has changed from the data center infrastructure, toward an application-centric view when hybrid cloud emerged, and toward a release-centric view in today's world of DevOps.

> **“Instead of releasing once per quarter, we seem to be in a constant state of more or less frenzied release preparation. There is constantly a backlog of requests for development, test, and staging resources. This situation becomes even more heated when collaboration across teams or even company boundaries is required.”**
>
> – IT Director, Automotive Industry

EMA DATA POINTS: APPLICATION-CENTRIC AUTOMATION AND MANAGEMENT

Public cloud cost is the #1 IT operations pain point	Enterprises are struggling to consolidate and govern their use of public cloud services.
Integration is the #1 requirement for DevOps and Containers	Due to their mobility and dynamic character, containers and microservices increase complexity and sprawl.
96% are unable to manage Kubernetes	Only 4% of enterprises can deploy Kubernetes without the help of external consultants.
Over 70% use multiple container platforms	Most enterprises leverage containers across the data center and multiple clouds.
PaaS, CaaS, VMs, and bare metal are used in combination	Enterprises use multiple deployment models and require a centralized and application-centric automation backbone.
72% are using DevOps	72% of enterprises have adopted DevOps groups.

PRODUCT	MORPHEUS DATA
	Hybrid and multi-cloud orchestration platform with app deployment templating and DevOps integrations (Jenkins, GitHub, Config. Mgmt).
	Infrastructure-agnostic focus on hybrid app deployments, second-day operations, and cloud cost optimization.
	Automate deployment of traditional or microservices apps across bare metal, HCI, hypervisor, and containers.

MORPHEUS

LEARN MORE

www.morpheusdata.com

PRIORITY #1 – INTEGRATION WITH EXISTING ENTERPRISE IT

QUICK TAKE

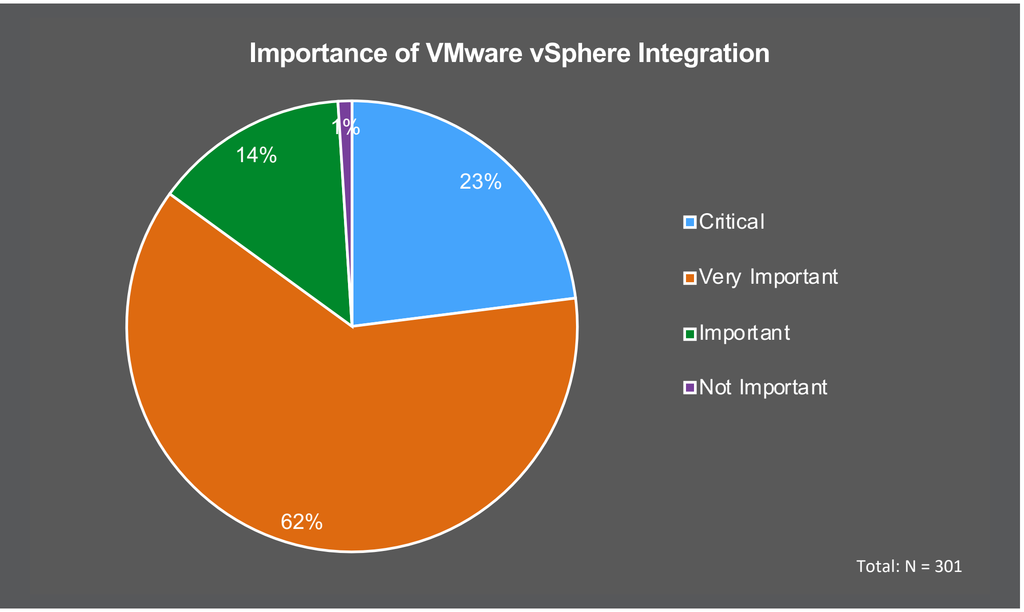
Easy integration of containers with existing data center technologies is the number-one selection criteria for container management products. Key integration points are: hypervisor, storage, network, hyperscale clouds, server automation, and configuration management tools.

Containers and Traditional Enterprise IT Will Coexist

Containers are a critical catalyst for digital transformation, but enterprises have found that it is neither feasible nor economical to immediately move all of their traditional applications and web apps to containers. The integration of containers with corporate IT and existing enterprise applications has taken center stage on their list of requirements.

Hypervisor Integration is Key

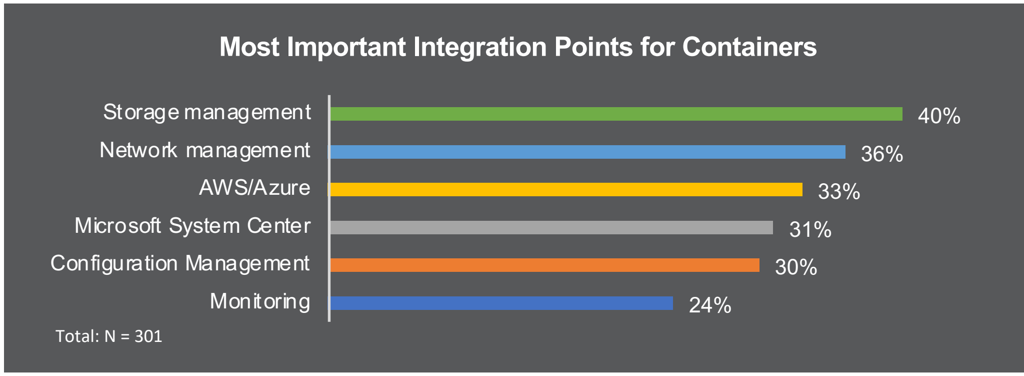
Approximately 75 percent of containers run on VMs today. These VMs are either based on-premises or hosted by a third party. VMware's vSphere hypervisor was named the crucial integration point, with 85 percent describing vSphere integration as critical or very important. The remaining container deployments are either hosted on bare metal or on a managed container service (CaaS).


Importance of VMware vSphere Integration

Total: N = 301

More Critical Integration Points

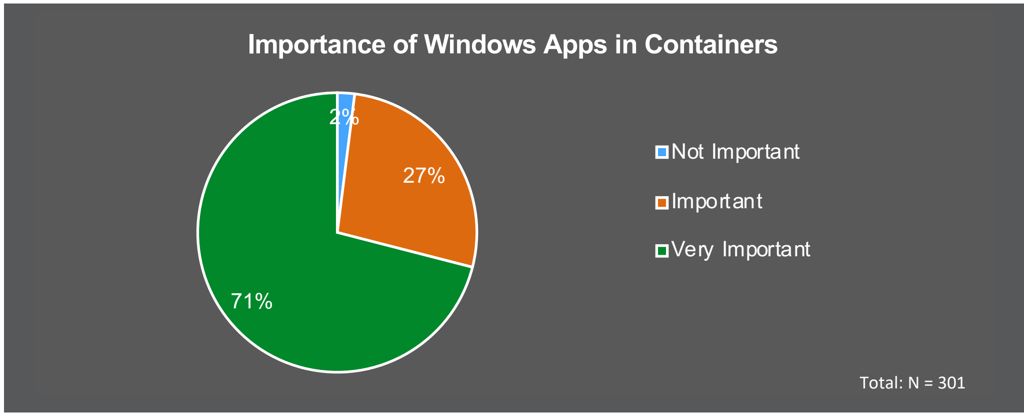
In addition to hypervisor integration, respondents indicated storage (40%) and network management (36%) as the second and third most important integration points, followed by integration with hyperscale clouds (AWS, Azure, Google Compute Engine, IBM Cloud), and configuration management tools (Chef Automate, Puppet Enterprise, Ansible Tower, and Microsoft System Center Configuration Manager).


Most Important Integration Points for Containers

Total: N = 301

Windows Containers are the Next Frontier

Central management of Linux and Windows applications is a key pain point that enterprises are seeking to address in 2018. Kubernetes 1.9 supports Windows Server Containers as a beta feature, enabling a central Linux-based Kubernetes management plane to run Linux and Windows container clusters.


Importance of Windows Apps in Containers

Total: N = 301

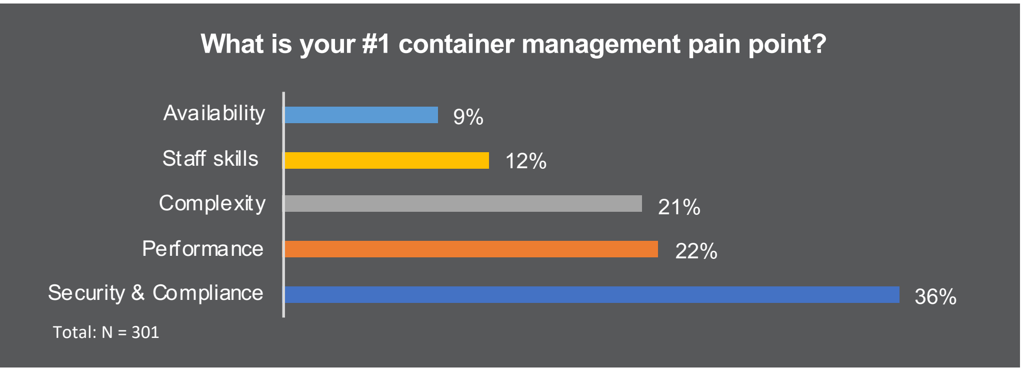
PRIORITY #2 – CONSISTENT SECURITY AND COMPLIANCE FRAMEWORK

QUICK TAKE

Enterprises are seeking to implement security and compliance for containers in a manner consistent with their existing security and compliance frameworks that current staff can operate. The key challenge lies in the fundamental architectural difference between container environments and traditional VM-based IT.

Security and Compliance are the Main Pain Points

Security and compliance are the most important container-related pain points in 2018 (36%). This finding is underlined by IT operators naming the ease of security and compliance enforcement as the universal number-two selection criterion for new container-related technologies.



HIPAA leads the list of the most important compliance regulations that apply to the corporate container strategy in 2018 (43%), ahead of PCI (26%), FISMA (25%), and SOX (24%).

Policy Staff not yet Part of DevOps

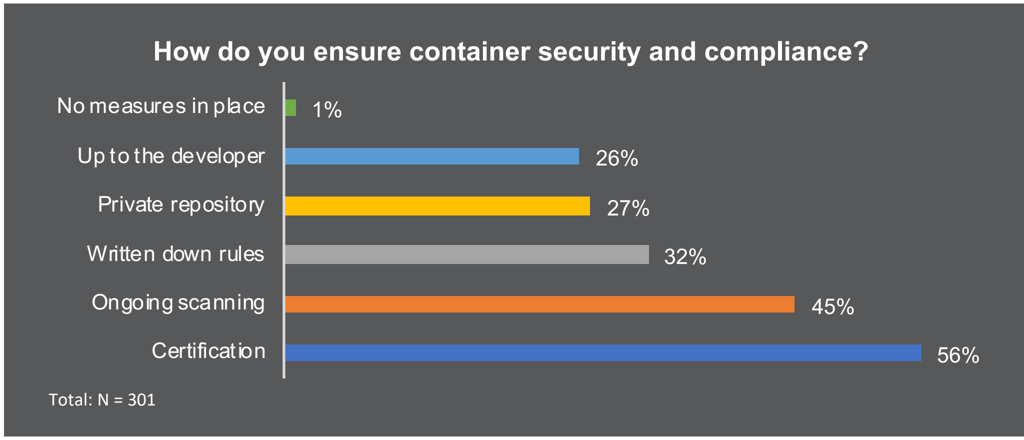
While enterprises are openly acknowledging security and policy challenges today, they still have not made the critical transition toward including security architects and policy roles into the DevOps pipeline. However, this inclusiveness is key for accelerating the release schedule and enabling rapid releases that are secure, compliant, and cost-efficient, even during the development, test, and staging phases.

58% OF POLICY STAFF ARE NOT PART OF THE DEVOPS PROCESS.

52% HAVE NO SOLID COMPLIANCE PLAN IN PLACE.

Most are Investing in Container Security and Compliance

Only 27 percent leverage private container registries. The use of public registries opens the door to development teams introducing non-compliant container images across the enterprise. Fifty-six percent have implemented container image certification to ensure compliance of their production container environments, and 45 percent are continuously scanning deployed container applications for policy and security problems. While these metrics must increase dramatically in 2018, the good news is that only one percent have no measures in place to secure their container images.



PRIORITY #3 – BRING BACK CONTROL FOR CORPORATE IT OPERATIONS

QUICK TAKE

While software developers initially adopt and sometimes even operate containers, corporate IT still has to guarantee SLAs, security, performance, and compliance. Therefore, container management solutions cannot only be optimized for developers, but also must deliver on the requirements of corporate IT.

Who Will Operate Containers?

As containers become the tool of choice for DevOps, enterprises struggle to translate operational requirements from the VM world into the world of containers. Traditional infrastructure and application platform operators still own the SLA, while developers often control the container framework. If the operations team is not in control of container management, these operators cannot be held responsible for SLAs of containerized applications.

66% OF IT OPERATIONS TEAMS ARE RESPONSIBLE FOR SLAS OF CONTAINERIZED APPS.

Unsanctioned Container Environments are a Problem

Seventy-two percent of enterprises today are confronted with unsanctioned container management platforms. In response, they are seeking to implement solutions that developers and operators can both access.

72% OF ENTERPRISES HAVE UNSANCTIONED CONTAINER PLATFORMS

45% IN 45% OF ENTERPRISES, CONTAINERS ARE NOT MANAGED BY CORPORATE

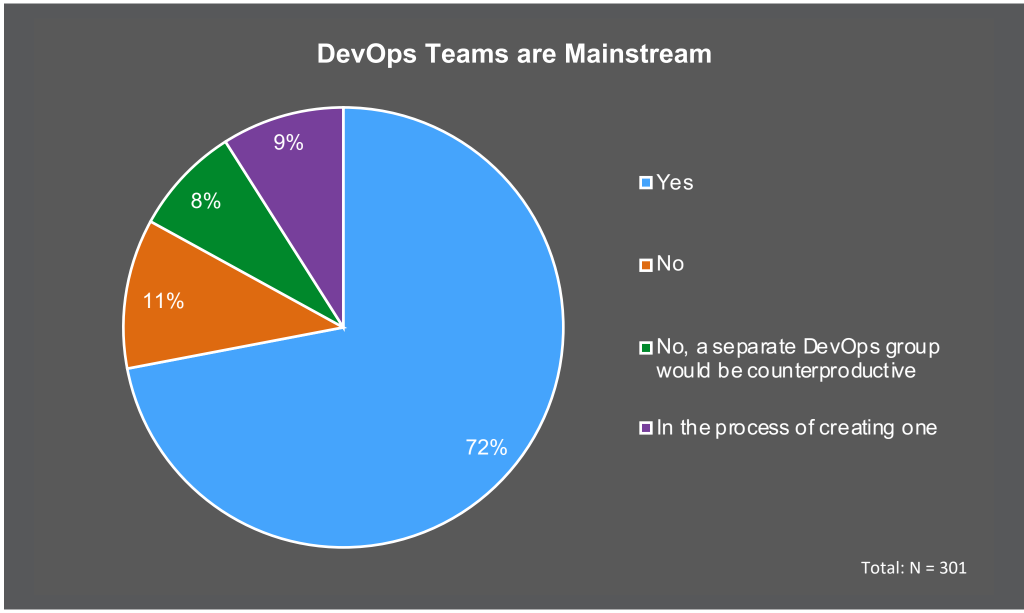
Enterprises Aim to Leverage Their Existing Operations Teams

In 2018, enterprises will select container technology vendors that enable their existing application and infrastructure operations teams to step up and take on container management. Line of business software developers will also be included in the technology selection and management process. While central IT will define provisioning and management policies and operational security, developers will continue to bring their own runtime environments. The key factor for developer adoption will be the availability of undiluted native and up-to-date APIs.

Developers and Operators Require a Joint Container Management Platform

Developers need to become more aware of the operational impact of their code, while operators need to provide an environment that is sufficiently flexible to absorb each application in a secure, cost-effective, highly available, and well-performing manner.

In short, the existing infrastructure and application operations teams will manage and monitor containerized applications in collaboration with developers.


DevOps Teams are Mainstream

Total: N = 301

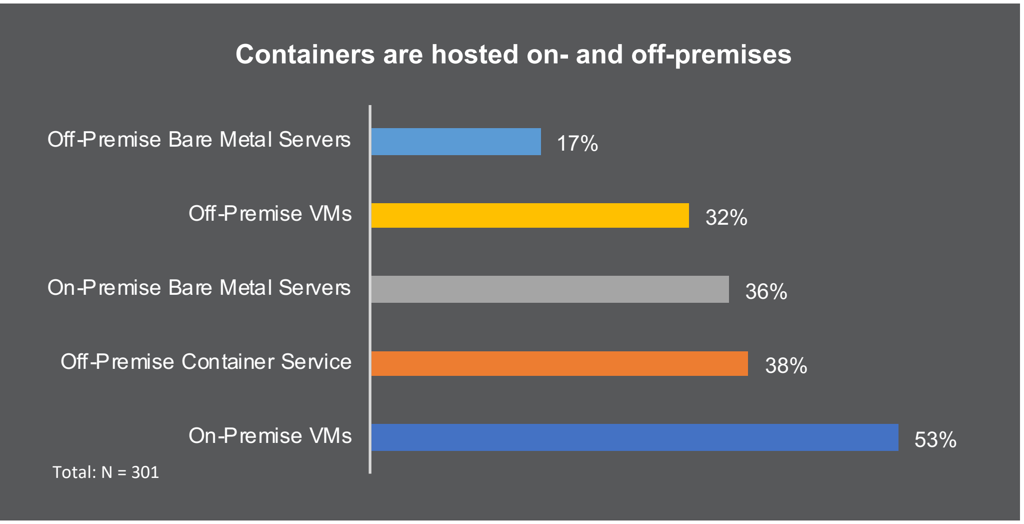
PRIORITY #4 – SUPPORT HYBRID INFRASTRUCTURE

QUICK TAKE

Enterprises mostly leverage multiple container services, and 53 percent are running containers on VMs on-premises. Twenty-six percent are asking for their container management solutions to also manage serverless platforms. Hosted CaaS, Amazon EC2 Container Service, Google Container Engine, and Azure Container Service are in a head-to-head race, with many enterprises using two or even all three of these services.

VMs are the Preferred Platform for Containers

On-premise VMs are today's most popular deployment platform for containers (53%), ahead of hosted container services (38%), and hosted VMs (32%). This illustrates that VMs do have a future within a containerized world, mainly due to the maturity of their management platforms, and organizations require their container management framework to support different deployment targets.

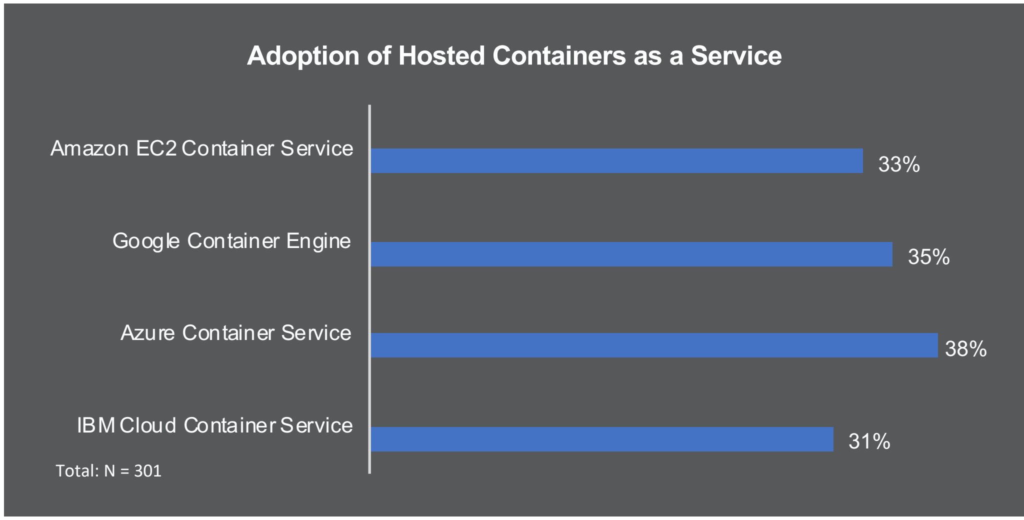


CaaS and Serverless Support is a Must

All hyperscale clouds and numerous on-premises IaaS infrastructure vendors offer managed Kubernetes container services today. Many vendors are now also adding on FaaS and serverless containers. The latter concept enables developers to simply upload containers without interacting with the container management plane, by specifying a few parameters to define scalability and cost. Due to the low threshold of these offerings, developers are quick to swipe their credit cards and give them a try, so there are often numerous different types of CaaS and serverless container platforms deployed in an enterprise.

Consistent Management of Hybrid System is Crucial

Modern development teams take advantage of whatever technology platform enables them to create the required features and capabilities with the least effort, at the lowest cost, and highest quality. In some cases, this can mean to simply tap a database that is still running on bare metal infrastructure, while in others this may mean to use IBM Watson or AWS SageMaker APIs to quickly train a machine learning model for a very specific purpose. Enterprises need to adopt technology platforms that can manage any type of application environment, on- or-off premises.



PRIORITY #5 – OPTIMAL KUBERNETES MANAGEMENT

QUICK TAKE

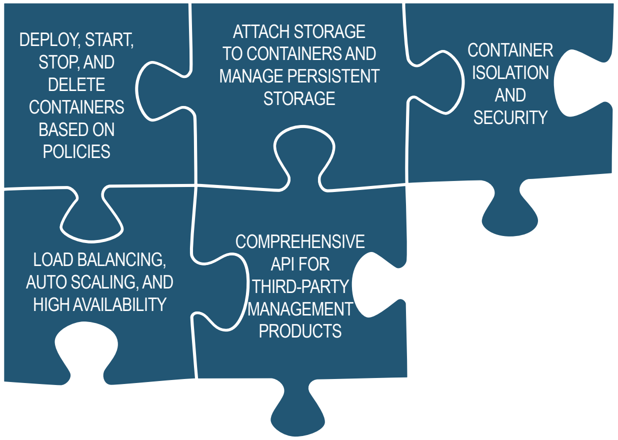
Kubernetes is a scheduling and orchestration framework consisting of a set of services to deploy and manage containers in production and at scale. The key value of Kubernetes is that it provides developers with a set of standard APIs that enables them to deploy their code builds based on policies defined by corporate IT and without having to get involved in time-consuming infrastructure configuration.

Kubernetes: The de facto Standard for Container Scheduling and Orchestration

The open-source Kubernetes platform originated from Google's Borg project. Google open-sourced Borg in 2014 to convince enterprises to adopt its public cloud offering for their container applications. Kubernetes is the de facto standard for container scheduling and orchestration; however, less than ten percent of organizations want to take on Kubernetes management directly, without third-party tools. Ninety-six percent leverage the help of a third party for Kubernetes implementation and maintenance. Amazon, VMware, Microsoft Azure, Google, IBM, Cisco, Red Hat, and most other vendors are offering fully managed Kubernetes (CaaS) platforms where developers can deploy container apps without standing up and managing the platform.

What Kubernetes Does Well

The key advantage of containers is their ability to encapsulate the entire application runtime, including libraries and databases, independently of the underlying operating system. As a result, containerized apps will run on any container host, on-premises or in the cloud. Kubernetes offers the following five key capabilities that enable enterprises to run containers in production and at scale.

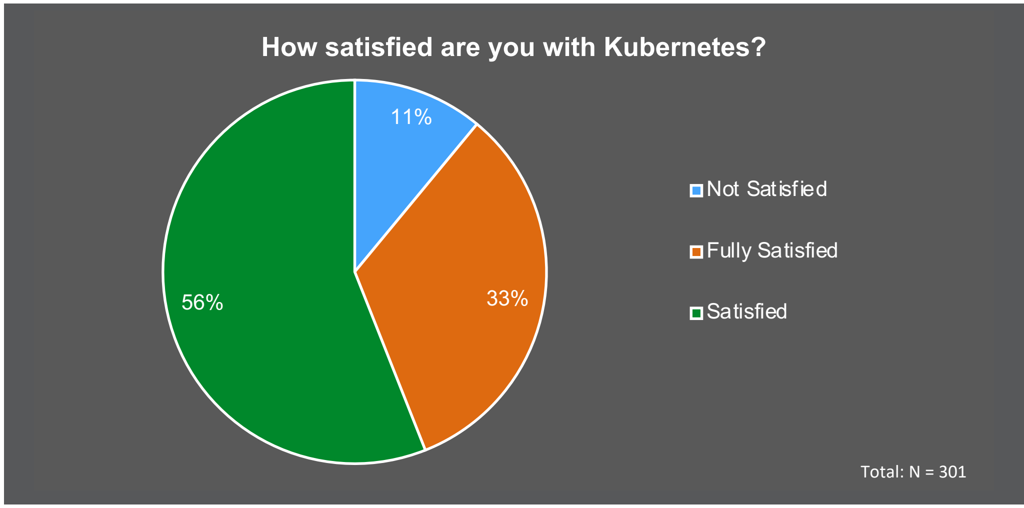

DEPLOY, START, STOP, AND DELETE CONTAINERS BASED ON POLICIES
ATTACH STORAGE TO CONTAINERS AND MANAGE PERSISTENT STORAGE
CONTAINER ISOLATION AND SECURITY
LOAD BALANCING, AUTO SCALING, AND HIGH AVAILABILITY
COMPREHENSIVE API FOR THIRD-PARTY MANAGEMENT PRODUCTS

However, Kubernetes management involves significant CLI and API programming skills, which has led to corporate IT operations struggling to play its role as the guardian of SLAs, performance, high availability, security, and compliance. Therefore, 90 percent of enterprises are looking to commercial container management software vendors to simplify Kubernetes operations.

Kubernetes is not yet Mature

Kubernetes cannot be compared to today's popular hypervisor platforms because many capabilities are only available via CLI and API, while others require plugins and custom scripts. In addition to lacking a comprehensive graphical user interface, Kubernetes lacks many out-of-the-box enterprise capabilities, leaving the field to commercial container management products. Here are some of the key capabilities available through these third-party vendors:

- Kubernetes cluster deployment
- Kubernetes platform management services for upgrades, disaster protection, high availability
- Policy-driven app deployment and security management
- Advanced analytics for infrastructure and app performance management and root cause analytics



PRIORITY #6 – TRANSITIONING TO MICROSERVICES

QUICK TAKE

Microservices consist of a small number of mostly stateless functions that receive input and provide their functionality through standard APIs. Modern applications can consist of numerous microservices that run as separate processes on individual containers. Each microservice has its own release lifecycle and can be consumed by any number of apps.

Definition of Microservices

Based on EMA's definition, microservices have the following key characteristics:

Functional components aligned with the business: Microservices are functional components that serve a specific business purpose. Multiple microservices can be connected into a hybrid application that offers a unified user experience.

Shared between apps: Microservices are standalone functional modules that can be consumed by any authorized application or end user across the enterprise. This enables the reuse of microservices for multiple applications at the same time.

Development efficiency: Typically, teams of 10-15 developers work on one microservice. As each microservice can stand on its own, there is minimal coordination needed with other teams beyond the standardized input and output definitions for the REST API interface.

Gradual pipeline automation: Instead of turning DevOps pipeline automation into one large and risky project, enterprises can separately automate the release cycle of each individual microservice.

Granular scalability: Instead of adding or terminating additional server resources for the entire application, corporate IT can now only scale the specific microservice that is causing problems.

Why Microservices Will Succeed Where SOA Failed

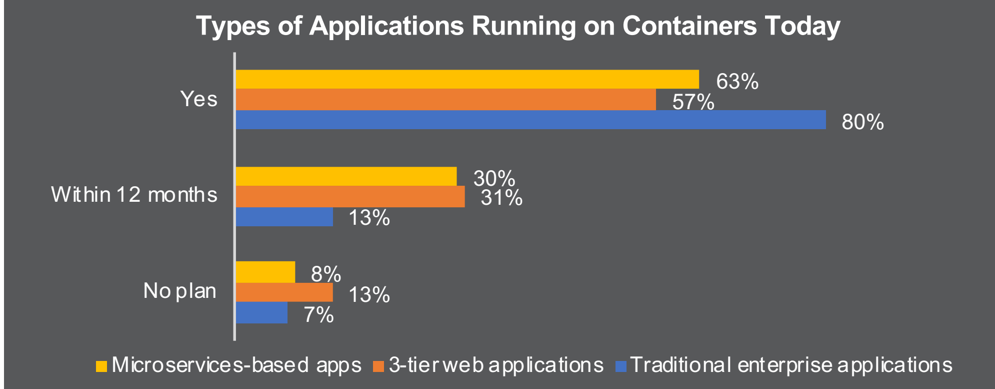
The explanation for optimism regarding the success of microservices is in the technological advancements since the failure of SOA. Today, enterprises are striving to complete their DevOps pipeline automation capabilities and they focus on the simplification of how applications communicate and integrate. Once the DevOps pipeline is automated, testing compatibility between microservices is only one more box to check during the release process.

The Role of Containers

The immutable character of containers makes them the ideal vehicle for deploying, running, and scaling microservices. Whenever a new version of a microservice is released, the container scheduler starts using the updated container image to instantiate new containers. The typically small size of microservices compared to monolithic applications makes the switch over and, if needed, the rollback between releases fast and simple.

How to Transition From Monolithic Apps to Microservices

The microservices concept takes the idea of agile software development further by breaking down a software product into its individual functional components. Existing enterprise apps can simply access the capabilities of microservices via API and vice versa. This facilitates a gradual transition toward more and more functional modules released as microservices. Microservices will run alongside web apps and traditional enterprise applications on containers for years to come.


Types of Applications Running on Containers Today

Total: N = 301

PRIORITY #7 – APPLICATION-CENTRIC CONTAINER MANAGEMENT

QUICK TAKE

Application-centric container management enables DevOps teams to continuously monitor, integrate, and deliver software. The container management platform provides the policy-driven deployment, monitoring, alerting, scaling, updates, and upgrades of containerized applications independent of the underlying infrastructure. This means application-centric CaaS management offers an abstraction layer on top of any container service, enabling enterprise customers to freely choose between application deployments to the data center or public cloud.

The Application-Centric Paradigm

An application-centric container management approach facilitates optimal collaboration between developers and operators to optimize cost, quality, compliance, and speed. This entails a DevOps-centric workspace that delivers a full panoramic view of all development and operations-related processes, metrics, notifications, logs, configuration items, and release information around an application. Instead of managing abstract infrastructure, IT operators manage along the lines of applications or, ideally, around application releases. This enables them to view the immediate impact of infrastructure events, not only on the present application's health, but also on the health of future releases.

Completing the Journey From Pets to Cattle

Containers are a tool to easily and quickly deploy and manage applications throughout their lifecycle without spending time and resources on operating the underlying infrastructure. Kubernetes enables this abstraction by consuming storage, memory, and CPU from its host servers and dynamically assigning it to containers, without spinning up an entire operating system. In short, applications take what they need from the container host, depending on their inherent resource needs and on their usage profiles.

Mind the Gap:
Kubernetes Needs Help With Infrastructure Management

Similar to OpenStack, Kubernetes cannot be used to gloss over automation deficiencies in the data center. In fact, Kubernetes is entirely unaware of what goes on at the server, network, and storage levels. This means Kubernetes will try to remediate a problem by attempting to, for example, spin up more containers or container hosts in response to high CPU utilization or storage latency. This makes sense in response to increasing user demand, but it would not make sense if host-level problems that could be caused by problems of the host operating system or hardware create the CPU utilization issue.

EXAMPLE: COMBINED MONITORING OF CONTAINERS AND INFRASTRUCTURE IS CRITICAL

A faulty RAID array could lead to higher storage latency. This might only be detrimental for certain containers, while others may be less sensitive to storage performance. Kubernetes will spin up more container instances for the affected application across different hosts, but the Kubernetes scheduler does not have the operations data available that would be required to diagnose the RAID issue. Unless corporate IT diagnoses and fixes the RAID array, the issue could be a time bomb that will potentially take down the entire container host and subsequently force Kubernetes to evacuate all containers. In the best-case scenario this will lead to resource waste, but it could also cause a degradation in the end-user experience, depending on the availability of another sufficiently powerful container host.

PRIORITY #8 – TRANSITION TO DECLARATIVE AUTOMATION AND MANAGEMENT

QUICK TAKE

Containerized applications and their microservices should be immutable, meaning they are replaced instead of reconfigured, fixed, scaled, or upgraded. A declarative approach to IT automation simply describes the targeted state of the application environment, in order for the orchestration engine to determine and resolve the delta between this target state and the status quo. Container scheduling and orchestration policies are defined in a declarative manner to ensure absolute consistency across environments at any given time.

The Changing Role of Configuration Management

Based on EMA data, 71 percent of enterprises are leveraging IT automation software such as Chef, Puppet, Ansible, Saltstack, AWS CloudFormation, or HashiCorp TerraForm. However, containers are replaced instead of upgraded, reconfigured, or fixed. By definition, this rules out configuration drift in its traditional form, but it does not decrease the importance of IT automation, especially to enforce security and compliance.

The declarative approach abstracts the application from its underlying infrastructure, enabling easy deployment to any type of target environment from bare metal to CaaS and FaaS.

IT Automation as the Backbone for Compliance, Security, and Scalability

Modern IT automation solutions focus on the implementation of a declarative approach, where applications are defined based on their runtime requirements. This definition also includes declarations for performance, security, availability, scalability, and upgrade requirements. The approach abstracts the application from its underlying infrastructure, enabling easy deployment to any type of target environment from bare metal to CaaS and FaaS. It also enables all stakeholders, such as application administrators, developers, security staff, and compliance experts, to define and monitor the desired state of how the application should function and what its infrastructure requirements are.

Kubernetes Enables Declarative Management

The entire Kubernetes object model can and should be managed following the declarative paradigm. For example, when upgrading an application, Kubernetes checks the affected components of the current environment, and then applies only the changes necessary to establish the desired state.

KEY ADVANTAGES OF DECLARATIVE MANAGEMENT

Consistency across hybrid environments at any point in time	Instant implementation of security or compliance changes
Easy adoption of future technologies	Continuous audit compliance

Declarative management is a core requirement for successful DevOps implementation and container management, since it assures the instant and infrastructure-independent definition of application environments.

PRIORITY #9 – COMBINING VMS, CONTAINERS, PAAS, AND FAAS

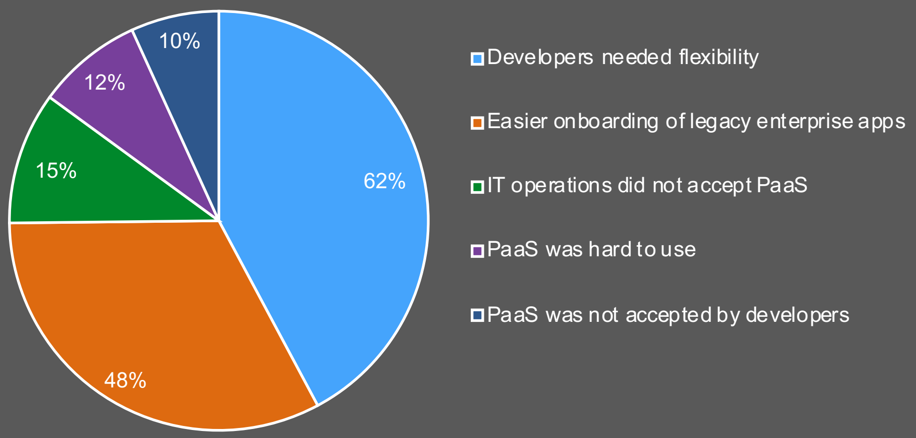
QUICK TAKE

Ninety-five percent of PaaS users also use containers. Sixty-two percent named developers needing more flexibility as their number-one reason for adopting containers in addition to PaaS. The key reason for operators to add containers to their existing PaaS deployment was the easier onboarding of legacy enterprise applications. With many organizations aiming to eliminate all baggage that slows down code releases, functions as a service (FaaS) will see a rapid rise in 2018. Ultimately, enterprises will require their IT operations tools to consistently manage VMs, containers, PaaS, and FaaS.

VMs, Containers, PaaS, and FaaS are Here to Stay

The rapid rise of containers has proven positive for PaaS, since developers and operators often regard containers as the solution that fills the gap between traditional VM-based applications and PaaS. Ninety-five percent of PaaS adopters have now also adopted containers as a solution that gives them the desired flexibility to create a cloud-native app without having to always adhere to an opinionated PaaS framework.

Why did you adopt containers in addition to PaaS?



Total: N = 301

95% **of PaaS users are using containers in addition**

26% **of enterprises aim to make FaaS part of their DevOps process**

FaaS and Serverless Containers

Twenty-six percent of container-adopting enterprises aim to make FaaS part of their DevOps process. FaaS enables developers to directly upload and run code without having to worry about infrastructure provisioning time and cost. FaaS offers inherent scalability, while at the same time enabling small pilot projects without delay or sunk cost. In 2018, a wave of container and cloud vendors started to offer FaaS in a hybrid manner, enabling enterprises to choose between a public cloud or data center as the deployment target for their functions.

Continuous Compliance

Continuous compliance is a critical goal for any application modernization strategy and can only be achieved through consistent management and the automation of applications across platforms and environments. Deploying a Kubernetes-driven application to Amazon Kubernetes Service (EKS) must adhere to exactly the same compliance and security standards as if this application were deployed to Google Kubernetes Engine (GKE), or the corporate data center. The modernized application environment will then intrinsically and consistently enforce application-centric compliance independently of the infrastructure.

PRIORITY #10 – PRIORITIZE CONTAINERIZATION

QUICK TAKE

Whether or not it makes sense to containerize an application depends on a set of economical considerations specific to each enterprise and situation. Only a small minority of enterprises (16%) believes that legacy applications should be containerized wholesale and without first modernizing architecture, codebase, and APIs.

Considerations Before Containerizing an Application

While there are tools to simplify the application containerization process, it is critical to consider many factors in order to ensure successful container transformation. These factors are not only in the application itself, but also in the integration of container technologies with existing corporate IT systems, and the skill and experience of the IT operations teams.

AN INCOMPLETE LIST OF ELEMENTS TO CHECK BEFORE CONTAINERIZING AN APPLICATION

Host dependencies, hardware dependencies, hardcoded environment variables, non-standard access permissions, custom kernel requirements, access requirements to the local file system, source code not or only partially available, communication through the local file system, has artifacts of sloppy coding that cumulatively claim resources without releasing them again, run batch jobs as separate processes, requires root access or different access levels, dependencies on database schemas, does the application require multiple cores to run, extreme performance requirements, data locality requirements, needs to run on Linux or Windows host, requires a graphical user interface.

Scalability: While the container scheduler can easily create additional container instances of an application, without modernizing the application architecture first, this may result in entirely separate instances that write to separate databases, use separate storage, and run under a separate hostname. In some cases, users can load-balance these instances; however, scaling may still be inefficient and risky.

Portability: Applications that rely on specific precompiled infrastructure components will not run on any container host without replicating their native environment. Using VMs in combination with live migration capabilities of the hypervisor could be the more economical solution.

Simple upgrade and rollback: More vendors today deliver their software in container format, making upgrade and rollback seamless and mostly free of risk. For applications where this is not the case and for custom applications, development and operations teams have to define new processes and verify the ability of their configuration management and continuous integration tools to support containers.

ONLY 16% FOCUS ON CONTAINERIZING LEGACY APPLICATIONS WITHOUT MODERNIZING THEM FIRST.

Can Corporate IT Handle it?

Once businesses have decided that an application can be containerized as is, or after modernization, it is critical for them to determine whether the current IT operations team has the skills, experience, and systems at its disposal to manage the application in containerized form. Trivializing this challenge made the ability of container technologies to integrate with traditional IT systems and the ability of current corporate IT staff to manage containers the most important requirements for the selection of new container technologies. In addition, the corporate IT team has to transition to a declarative automation approach and understand container policies, networking, storage, monitoring, and numerous other management topics.

Prime Candidates for Containerization Today

Cloud-native applications, often consisting of loosely coupled microservices, can be easily containerized because they typically do not have environmental dependencies. These are usually stateless apps running sometimes entirely on a PaaS platform, such as CloudFoundry. EMA research data shows that a significant share of enterprises have recently moved from PaaS to containers, mostly for flexibility reasons.

About Enterprise Management Associates, Inc.

Founded in 1996, Enterprise Management Associates (EMA) is a leading industry analyst firm that provides deep insight across the full spectrum of IT and data management technologies. EMA analysts leverage a unique combination of practical experience, insight into industry best practices, and in-depth knowledge of current and planned vendor solutions to help EMA's clients achieve their goals. Learn more about EMA research, analysis, and consulting services for enterprise line of business users, IT professionals, and IT vendors at www.enterprisemanagement.com or blogs.enterprisemanagement.com. You can also follow EMA on Twitter, Facebook, or LinkedIn.

Corporate Headquarters:
1995 North 57th Court, Suite 120
Boulder, CO 80301
Phone: +1 303.543.9500
Fax: +1 303.543.7687
www.enterprisemanagement.com
3680-Morpheus.022818