TechTarget | Search IT Operations

E-guide

PRO+ Content

What is Container Management and Why is it Important?

In this e-guide

In this e-guide:

In this comprehensive guide that we created just for you, you'll learn:

- **What container management consists of**
- **Benefits and challenges of container management**
- **Container management strategies**
- **Major container management software vendors and tools**
- **Container schedulers, orchestration and deployment tools**
- **Kubernetes implementation considerations**
- **And more**

What is container management and why is it important?

Emily Mell, Site Editor and **Beth Pariseau,** Senior News Writer, SearchITOperations

Container management refers to a set of practices that govern and maintain containerization software. Container management tools automate the creation, deployment, destruction and scaling of application or systems containers.

Containerization is an approach to software development that isolates processes that share an OS kernel -- unlike virtual machines (VMs), which require their own -- and binds application libraries and dependencies into one deployable unit. This makes containers lightweight to run, as they require only the application configuration information and code from the host OS. This design also increases interoperability compared to VM hosting. Each container instance can scale independently with demand.

Modern Linux container technology was popularized by the Docker project, which started in 2013. Interest soon expanded beyond containerization itself, to the intricacies of how to effectively and efficiently deploy and manage containers.

In 2015, Google introduced the container orchestration platform Kubernetes, which was based on its internal data center management software called Borg. At its most basic level, open source Kubernetes automates the process of running, scheduling, scaling and managing a group of Linux containers. With more stable releases throughout 2017 and 2018, Kubernetes rapidly attracted industry adoption, and today it is the de facto container management technology.

IT teams use containers for cloud-native, distributed -- often microservices-based -- applications, and to package legacy applications for increased portability and efficient deployment. Containers have surged in popularity as IT organizations embrace DevOps, which emphasizes rapid application deployment. Organizations can containerize application code from development through test and deployment.

Benefits of container management

The chief benefit of container management is simplified management for clusters of container hosts. IT admins and developers can start, stop and restart containers, as well as release updates or check health status, among other actions. Container management includes orchestration and schedulers, security tools, storage, and virtual network management systems and monitoring.

Organizations can set policies that ensure containers share a host -- or cannot share a host -- based on application design and resource requirements. For example, IT admins should colocate containers that communicate heavily to avoid latency. Or, containers with large resource requirements might require an anti-affinity rule to avoid physical storage overload. Container instances can spin up to meet demand -- then shut down -- frequently. Containers also must communicate for distributed applications to work, without opening an attack surface to hackers.

A container management ecosystem automates orchestration, log management, monitoring, networking, load balancing, testing and secrets management, along with other processes. Automation enables IT organizations to manage large containerized environments that are too vast for a human operator to keep up with.

Challenges of container management

One drawback to container management is its complexity, particularly as it relates to open source container orchestration platforms such as Kubernetes and Apache Mesos. The installation and setup for container orchestration tools can be arduous and error prone.

IT operations staff need container management skills and training. It is crucial, for example, to understand the relationships between clusters of host servers as well as how the container network corresponds to applications and dependencies.

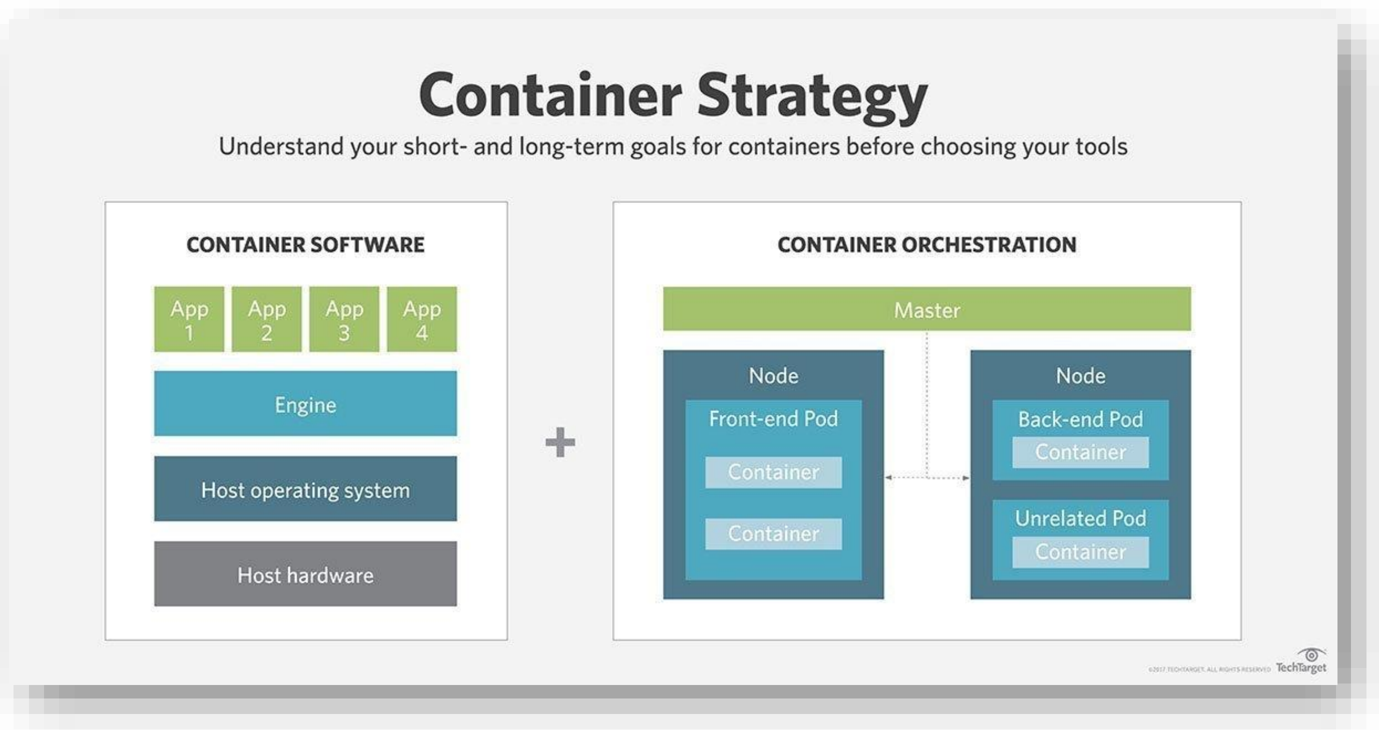
Issues of persistence and storage present significant container management challenges. Containers are ephemeral -- designed to exist only when needed. Stateful application activities are difficult because any data produced within a container ceases to exist when the container spins down.

Container security is another concern. Container orchestrators have several components, including an API server and monitoring and management tools. These pieces make it a major attack vector for hackers. Container management system vulnerabilities mirror standard types of OS vulnerabilities, such as those related to access and authorization, images and intercontainer network traffic. Organizations should minimize risk with security best practices -- for example, identify trusted image sources and close network connections unless they're needed.

Container management strategy

Forward-thinking enterprise IT organizations and startups alike use containers and container management tools to quickly deploy and update applications.

IT organizations must first implement the correct infrastructure setup for containers, with a solid grasp of the scope and scale of the containerization project in terms of business projections for growth and developers' requirements. IT admins must also know how the existing infrastructure's pieces connect and communicate to preserve those relationships in a containerized environment. Containers can run on bare-metal servers, VMs or in the cloud -- or in a hybrid setup -- based on IT requirements.



Container Strategy

Understand your short- and long-term goals for containers before choosing your tools

In addition, the container management tool or platform should meet the project's needs for multi-tenancy; user and application isolation; authentication; resource requirements and constraints; logging, monitoring and alerts; backup management; license management; and other management tasks.

IT organizations should understand their hosting commitment and future container plans, such as if the company will adopt multiple cloud platforms or a microservices architecture.

Major container management software vendors and tools

Kubernetes forms the basis of diverse distributions from various IT tool vendors. Some commercial vendors support open source container management components, including Kubernetes, or embed those components into their own products. Kubernetes deprecated Docker container image support as of version 1.20 in December 2020 in favor of the Open Container Initiative (OCI) format; the two are largely identical, although OCI relies on a command-line interface. The Kubernetes platform also continues to develop Windows support.

The container software market evolves constantly. Organizations must consider many factors to choose the right container management software for their particular needs -- and be flexible in those plans. Some options to explore include the following:

- Amazon Elastic Container Service and Elastic Kubernetes Service
- Microsoft Azure Kubernetes Service
- Canonical Charmed Kubernetes
- Cloud Foundry
- D2iQ Kubernetes Platform
- Google Kubernetes Engine
- IBM Red Hat OpenShift
- Mirantis Kubernetes Engine (formerly Docker Enterprise)
- Rancher Labs' Rancher (acquired by SUSE in 2020)
- VMware Tanzu (formerly Enterprise Pivotal Container Service)
- Open source Kubernetes

Container schedulers, orchestration and deployment tools

Many projects, from service mesh to cluster managers to configuration file editors, are designed to improve one aspect of the main container management technologies. Kubernetes support and partnerships crop up and evolve frequently. For example, service mesh technologies, such as Istio, work alongside Kubernetes to simplify networking. And some container management software, such as IBM Red Hat OpenShift, offers an integrated service mesh layer based on Istio or other technology.

Apache Mesos, an open source project designed for large-scale container deployments, manages compute clusters, including container clusters and federation. Mesos differs from Kubernetes in how it handles federation: Mesos treats it as a peer group of cooperating deployments, whereas in Kubernetes the master unites the operators to support the common goal.

Mesosphere DC/OS from D2iQ (formerly known as Mesosphere), is a commercial product based on Mesos that orchestrates containers with hybrid cloud portability. The Apache Mesos project remains available upstream, but D2iQ is now primarily focused on Kubernetes support.

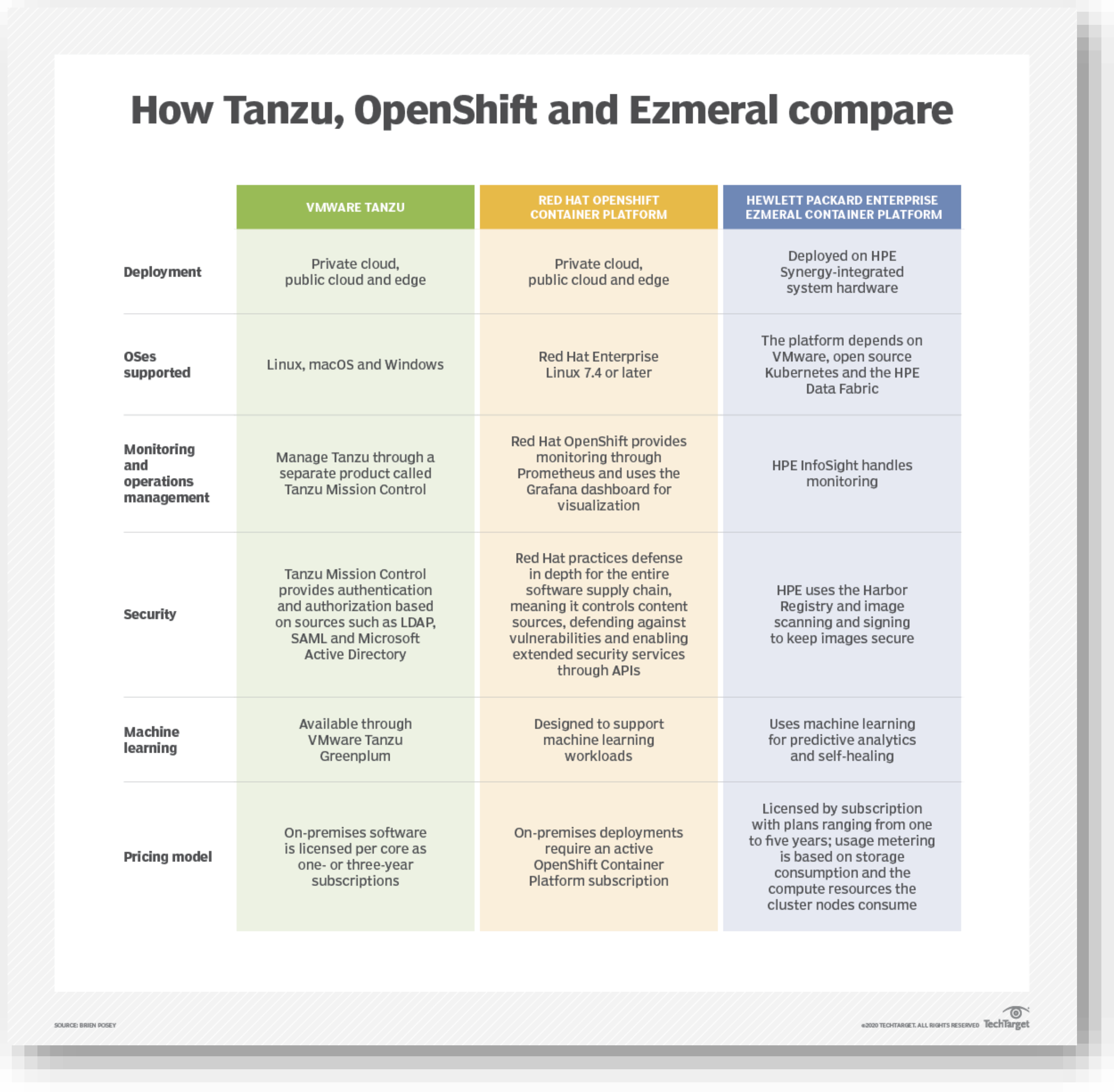
Docker's swarm mode is another open source cluster management utility for containers. Mirantis acquired Docker Inc.'s Docker Enterprise business in 2019, including a commercial version of Docker Swarm, and has continued to support and expand it.

The lines between container management software categories -- orchestration, security, networking and so on -- blur as container orchestration platforms add native support for additional management capabilities. Container management technology has been folded into or connected with larger management suites for server hosts and VMs.

Integrated Kubernetes platforms

Integrated container management packages appeal to many organizations because they simplify deployment and management challenges. Examples include IBM Red Hat OpenShift Container Platform, the VMware Tanzu suite and HPE Ezmeral Container Platform. (Here's a comparison of OpenShift vs. Tanzu vs. Ezmeral). Commercial container management products are available in various configurations and versions with distinct feature sets.

In this e-guide



How Tanzu, OpenShift and Ezmeral compare

	VMWARE TANZU	RED HAT OPENSHIFT CONTAINER PLATFORM	HEWLETT PACKARD ENTERPRISE EZMERAL CONTAINER PLATFORM
Deployment	Private cloud, public cloud and edge	Private cloud, public cloud and edge	Deployed on HPE Synergy-integrated system hardware
OSes supported	Linux, macOS and Windows	Red Hat Enterprise Linux 7.4 or later	The platform depends on VMware, open source Kubernetes and the HPE Data Fabric
Monitoring and operations management	Manage Tanzu through a separate product called Tanzu Mission Control	Red Hat OpenShift provides monitoring through Prometheus and uses the Grafana dashboard for visualization	HPE InfoSight handles monitoring
Security	Tanzu Mission Control provides authentication and authorization based on sources such as LDAP, SAML and Microsoft Active Directory	Red Hat practices defense in depth for the entire software supply chain, meaning it controls content sources, defending against vulnerabilities and enabling extended security services through APIs	HPE uses the Harbor Registry and image scanning and signing to keep images secure
Machine learning	Available through VMware Tanzu Greenplum	Designed to support machine learning workloads	Uses machine learning for predictive analytics and self-healing
Pricing model	On-premises software is licensed per core as one- or three-year subscriptions	On-premises deployments require an active OpenShift Container Platform subscription	Licensed by subscription with plans ranging from one to five years; usage metering is based on storage consumption and the compute resources the cluster nodes consume

SOURCE: BRIEN POSEY

In this e-guide

Another option is Cloud Foundry, an open source platform that uses containers as part of a larger collection of integrated tools. One difference between Cloud Foundry and OpenShift is that Cloud Foundry is more positioned for development, while OpenShift highlights capabilities for the rest of the application lifecycle.

Cloud providers' managed Kubernetes services

Major public cloud providers offer hosted Kubernetes services that handle cluster management. These services include Amazon Elastic Kubernetes Service, Google Kubernetes Engine and Microsoft's Azure Kubernetes Service.

While these as-a-service choices reduce the administrative overhead of deploying and maintaining Kubernetes, they can hamper workload portability in multi-cloud environments. Enterprises should carefully consider these factors before they commit to a cloud-based managed Kubernetes service. Organizations must also assess whether the cloud services are compatible with on-premises deployments and management tools.

Container security tools

Secrets management tools keep track of passwords and tokens in secure environments. Docker secrets management tech exists in Kubernetes as well as Mesosphere, CISOfy's Lynis and HashiCorp's Vault.

For fraud protection, Docker Notary and similar tools certify container images as they move between test, development and production environments.

Static image and runtime container security scanning tools inspect container images before they deploy and track behavior on the network after installation. This software is available from several vendors, including Aqua Security, Deepfence, NeuVector and Twistlock.

Some general network security platforms, such as Trend Micro Deep Security, also support containers.

Container networking tools

Container-specific virtual networking tools are available from Contiv, Weaveworks and open source projects, such as Project Calico, which focuses on Kubernetes container network management.

Virtual network management platforms that address infrastructure also support container technology. Examples include Ansible Container, VMware NSX, Cisco Application Centric Infrastructure and OpenShift Virtualization.

Service mesh technology aids communication between application services within container clusters. It is a unified abstraction layer for container networking. However, without proper management and skills, a service mesh can increase complexity within a containerized environment. Service mesh technologies include open source projects such as Linkerd, Envoy, Istio and Kong Mesh, as well as offerings from cloud and container management tool vendors.

Container monitoring tools

The more containers in use, the more difficult they become to monitor efficiently. Organizations should make automation capabilities a primary requirement as they evaluate container monitoring tools. Vital container monitoring capabilities include the following:

- health monitoring
- remediation
- root cause analysis
- broad system integration
- inactive container detection

Specialized monitoring tools track performance, bugs and security in containerized workloads. Options for container-specific monitoring tools include Sysdig, Google's cAdvisor and the Prometheus tool for Kubernetes.
Some DevOps monitoring platforms track containers in addition to other hosting architectures. These products come from companies such as New Relic, Datadog, AppDynamics, Dynatrace, Sumo Logic and SignalFx.

Container storage tools

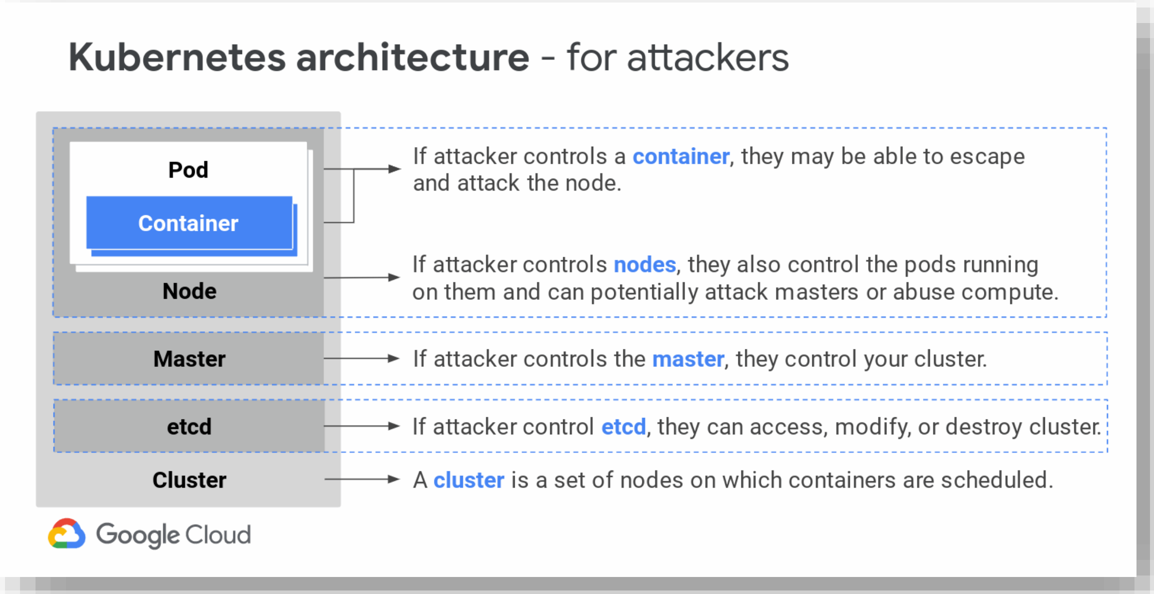
Many container management tools address the challenge of storage and persistence -- albeit not to perfection -- with approaches from attached volumes to plugins and APIs. Container persistent storage tools that offer true container portability for stateful applications come from Portworx (now owned by Pure Storage), Blockbridge Networks, IBM Red Hat Container Storage based on Gluster and IBM Red Hat OpenShift Container Storage based on Ceph.

Kubernetes implementation considerations

As described above, containers are arranged into *pods* in Kubernetes, which run on clusters of *nodes*; pods, nodes and clusters are controlled by a *master.* One pod can include one or multiple containers. IT admins should carefully consider the relationships between pods, nodes and clusters when they set up Kubernetes.

Organizations should plan their container deployment based on how many pieces of the application can scale under load -- this depends on the application, not the deployment method. Additionally, capacity planning is vital for balanced pod-to-node mapping, and IT admins should ensure high availability with redundancy with master node components.

In this e-guide



If attacker controls a **container**, they may be able to escape and attack the node.

If attacker controls **nodes**, they also control the pods running on them and can potentially attack masters or abuse compute.

If attacker controls the **master**, they control your cluster.

If attacker control **etcd**, they can access, modify, or destroy cluster.

A **cluster** is a set of nodes on which containers are scheduled.

IT organizations can address container security concerns by applying some general IT security best practices to containerization. For example, create multiple security layers throughout the environment, scan all container images for vulnerabilities, enforce signed certificates and run the most up-to-date version of any container or application image. Containers introduce the benefits of an immutable infrastructure methodology as well; the regular disposal and redeployment of containers, with their associated components and dependencies, improves overall system availability and security. Additionally, Kubernetes multi-tenancy promises greater resource isolation, but recently revealed security vulnerabilities make multicluster management preferred for now.

Networking is another significant factor. Kubernetes networking occurs within pods, between pods and in user-to-containerized resource connections. Kubernetes enables pods and nodes to communicate without address translation, allocating subnets as necessary.

Lastly, IT admins working with Kubernetes should prepare to troubleshoot common container performance problems, including those caused by unavailable nodes and noisy neighbors, in an implementation.

Further reading

In this e-guide

About SearchITOperations

SearchITOperations.com covers the latest industry news, technologies and trends that impact today's IT professionals tasked with designing and running applications created using DevOps, microservices and cloud-native paradigms.

As trends like hybrid cloud, containers, DevOps and more continue to infiltrate data center infrastructures, IT operations pros are facing a new set of responsibilities, such as integrating next-gen apps into their environments, running and supporting these apps, and collaborating with Agile/DevOps teams – that's where the expert content on SearchITOperations.com come into play.

For further reading visit us at https://searchitoperations.techtarget.com

Images; Fotalia