

Definitive Guide to Enterprise Container Platforms

JUNE 2019

CONTENTS

Digital Transformation and the Software Era.....	3
The Emergence of Container Platforms.....	6
Components of a Complete Container Platform	6
Build.....	7
Share.....	7
Run.....	7
Enterprise Platform Requirements for the Digital Era	8
Speed.....	8
Flexibility	8
Security.....	9
Introducing Docker Enterprise	9
A Complete End-to-End Solution.....	10
How Docker Enterprise Spurs Innovation	12
Taking the Next Steps.....	14

Digital Transformation and the Software Era

Disruption is the only constant. Entire industries are being upended seemingly overnight by simple web and mobile apps that offer better consumer experiences at lower cost. Established companies are being outmaneuvered by more nimble global competitors that have access to world-class talent, technology and processes and can innovate and respond to new markets and opportunities. In order to compete, businesses everywhere must accelerate their timeline for digital transformation; transformation of their legacy systems, delivering new innovative solutions for their internal processes and new engaging experiences for their customers. To keep up with the pace of change, every company must become a software company. This is the reality today where software has become the foundation for achieving core business objectives and achieving competitive advantage; it's what enables organizations to deliver and scale innovation across the business.

“Our biggest area of need has been our transformation from what has historically been an industrial manufacturing company ... into a software, a hardware and firmware design and manufacturing company.”

Brian Magnusson
VP, Innovation and Technology
Lindsay Corporation

Today, IT organizations are looking at ways to reallocate investments from business-as-usual systems and applications, which consume 71% of the budget, to innovation projects. At the same time there's a strong desire to build new applications quickly and deploy them in a variety of scenarios from on-premises, hybrid and multi-cloud strategies to edge computing.

For many years, taking advantage of the cloud and new distributed application architectures for this new digital era meant tearing everything down and building everything new. This approach dismissed the importance and value of an organization's existing applications and intellectual property and created schisms in organizations that divided a company's resources between “the old” and “the new”.

Today's mandate is clear: organizations need to drive high-velocity innovation across their entire application portfolio - including the old and the new - and be prepared for what comes next. This is now possible due to the advent of container technology and container platforms.

BACKGROUND: INTRODUCTION TO CONTAINERS

A container is a standard unit of software that packages up code and all its dependencies so an application can run quickly and reliably from one computing environment to another. Docker popularized container technology when it launched in 2013 with an open source [Docker Engine](#). It leveraged existing computing concepts from the Linux world, specifically primitives known as cgroups and namespaces, and made it easy to use. Now available for both Linux and Windows-based applications, Docker containers will always run the same, regardless of the infrastructure because containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

Containers revolutionized computing because it introduced a new way to build and run applications:

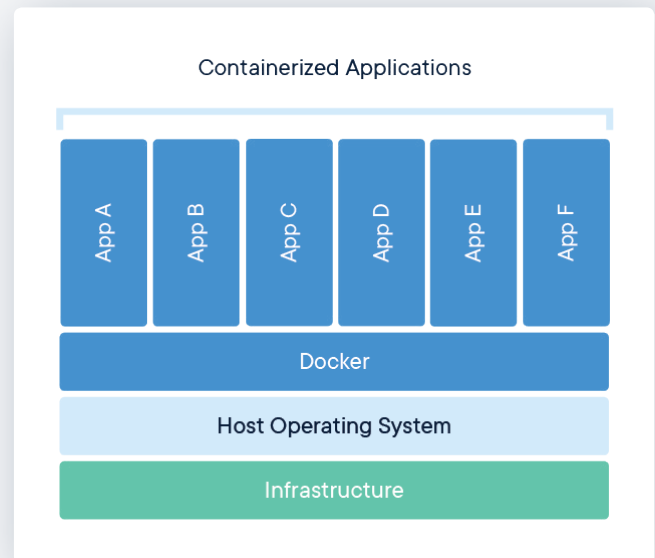
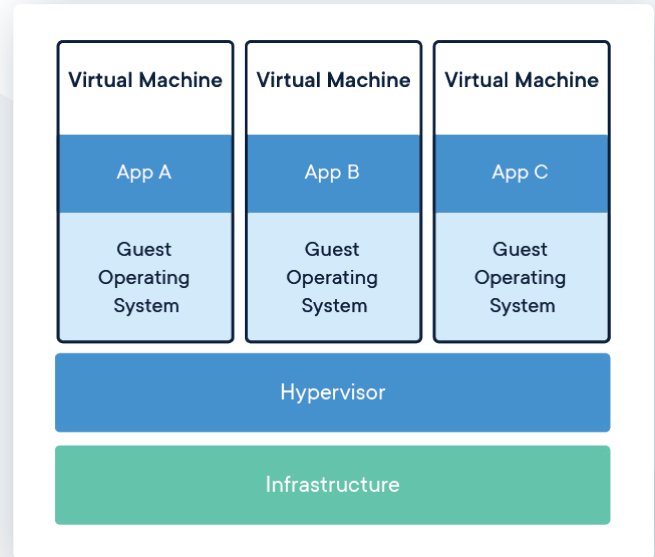
- Standard: Docker created the industry standard for containers, so they could be portable anywhere
- Lightweight: Containers share the machine's OS system kernel and therefore do not require an OS per application, driving higher server efficiencies and reducing server and licensing costs
- Secure: Applications are safer in containers and Docker provides the strongest default isolation capabilities in the industry

Containers and virtual machines have similar resource isolation and allocation benefits, but function differently because containers virtualize the operating system instead of hardware.

Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers. The hypervisor allows multiple VMs to run on a single machine. Each VM includes a full copy of an operating system, the application, necessary binaries and libraries - taking up tens of GBs. VMs can also be slow to boot.

Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), can handle more applications and require fewer VMs and Operating Systems. By encapsulating and isolating everything in a container, the container will always run the same, regardless of the environment it is running in.

Containers can uniquely turn very diverse set of application services into standardized software units.



BACKGROUND: INTRODUCTION TO KUBERNETES

[Kubernetes](#) is an open-source container orchestration engine for automating deployment, scaling, and management of containerized applications. Originally created by engineers at Google and now hosted by the Cloud Native Computing Foundation ([CNCF](#)), Kubernetes has seen rapid growth and adoption in a few short years.

Kubernetes is primarily focused on the running of containerized applications, including the following features:

- Service discovery and load balancing
- Service placement and bin packing
- Self-healing
- Storage orchestration
- Automated rollouts and rollbacks
- Secret and configuration management
- Horizontal scaling
- Batch execution

Docker packages a certified distribution of Kubernetes in Docker Desktop and Docker Enterprise, along with additional tools that make a more complete, enterprise-ready solution.



75% of global organizations will be running containers in production by 2022

“Best Practices for Running Containers and Kubernetes in Production”, Gartner, February 2019

The Emergence of Container Platforms

Over the past few years, the industry has standardized around the container as the format to solve some of the key challenges around application portability and developer productivity. Docker containers famously solved the “it works on my machine” problem by abstracting away application dependencies and spurred the growth of microservices-based application architectures.

But containers on their own are just a technology. Just like standard shipping containers introduced a new way to package physical goods, it is the entire transportation system including trains, cargo ships, and sophisticated loading and unloading equipment that has been built around the shipping container that has truly revolutionized goods distribution. In a similar way, today’s container platforms are built on industry-leading standards like the Docker container runtime and Kubernetes orchestration. However, it is the system of tools that operationalizes containers that will transform a company. A container platform goes beyond Kubernetes and orchestration in building a complete solution around the lifecycle of the containerized application, including how those applications are created, where they are stored, and how they get integrated into other IT tools and processes.

Components of a Complete Container Platform

A container platform should provide a complete solution for the building, sharing and running of containerized applications. Oftentimes, focus is placed strictly on the running of containerized applications, however the impact of containers on an organization is the transformational change it brings to the entire software development process. A container platform should address the needs of both developers and operators who often work in different paradigms. Container platforms should also improve the handoff between these groups.



BUILD

- Developers work on code
- Compile, assemble and build apps



SHARE

- Teams collaborate
- Source content from others



RUN

- DevOps teams deploy apps
- Manage, monitor and patch

From concept to fruition, a containerized application passes through multiple lifecycle stages. Each of these stages introduce new requirements for the container platform:

 **Build**

Containerized applications usually start with the developers who are building applications in a local environment - typically a desktop or laptop. Container platforms need to provide developers a simple way to build containerized applications, either from existing code or from scratch. It should easily integrate with existing software development tools like their preferred IDE (e.g. Visual Studio, Eclipse, IntelliJ), and it should work across different languages and frameworks (e.g. .NET, Java, Node.JS). Developers also need a quick and simple way to test their applications locally before checking their work into the broader software delivery pipeline.

Critical components:

- Desktop solution for local container development
- Developer tools for integration into the software development pipeline

 **Share**

Today's digital economy requires the ability to innovate rapidly on top of previous innovation and close collaboration among teams. In mid-sized to large organizations, this increasingly involves DevOps teams who embrace Continuous Integration and Continuous Delivery (CI/CD) principles for agile delivery. A container platform should include tools that make it both simple and secure to collaborate on applications while assisting in the rapid delivery of applications through the software pipeline.

Critical components:

- Secure image repository with access controls
- Public and private library of images, including images provided directly from ISVs

 **Run**

Once ready to be deployed, containerized applications will run in clusters on-premises or in hybrid and multi-cloud environments. These environments need to be secure while being easy to operate, which starts at the foundation of secure container runtime and a standard set of APIs and tools. Finally, the platform needs to provide a highly scalable operator experience that integrates well with storage, networking, logging and monitoring tools and provides a management plane for ongoing operations.

Critical components:

- Secure container runtime
- Highly scalable management plane

There are different container platform options available in the market including the ability for organizations to build their own container platform piece by piece. Organizations will need to be able to assess what their requirements are and determine which platform fits their needs.

Enterprise Platform Requirements for the Digital Era

As traditional businesses transform into application companies, speed, flexibility and security become absolutely essential and yet increasingly complicated given technical debt, vendor stacks and the risks and requirements around protecting data and systems in distributed hybrid and multi-cloud scenarios. Container platforms enable a new wave of software-based innovation, but they need to support the broader company objectives. Below are 3 common objectives that are important to companies from all different industries and sizes and where some container platforms fall short:

Speed

Today's market dynamics and competitive environment are changing so rapidly that companies who can respond and react quickly will be the most successful. However, most CIOs surveyed think they are falling behind in their digital transformation efforts. Things that slow an organization down include unnecessary barriers to developer productivity, friction between developers and operators and steep learning curves for new technologies. The container platform must be designed for high-velocity innovation across the entire software development lifecycle and should be intuitive for both developers and operators. Platforms should extend to developers' systems to enable "shift left" testing and security.

Be wary of:

- Platforms that force adoption of specific software development patterns or require learning specific coding techniques
- Platforms that require additional tooling for each new type of technology or application
- Platforms that only allow "late" testing, risking design defects that are not detected until applications are deployed to staging or production
- Platforms that focus more on deploying Kubernetes than the software development process

Flexibility

New technology stacks and frameworks are being introduced every day. The container platforms that organizations use must work with both existing technology investments and next generation technologies alike. Container platforms should be independent of both the underlying infrastructure as well as the applications and frameworks, providing a consistent and uniform operating model for different application types intended for different operating environments to prevent lock-in.

Be wary of:

- Platforms specializing in limited types of application frameworks or technology stacks
- Platforms that support only one infrastructure or operating system
- Platforms that cannot address the full spectrum of existing and new applications



Security

The new digital economy has also introduced a new wave of cybercrime. Security has become more complex and high-stakes and organizations are faced with the challenge of empowering developers to be productive while protecting themselves against risk. The new application architectures are highly dynamic and scalable, outgrowing traditional security models so the container platform needs to be built from the ground up with security in mind.

Be wary of:

- Platforms that require 3rd party security tools due to lack of built-in capabilities
- Platforms that ignore the developers' systems as a key to creating safe applications
- Platforms that cannot support required application security models, such as Active Directory authentication for Windows applications

Introducing Docker Enterprise

In the [Forrester New Wave™: Enterprise Container Platform Software Suites, Q4 2018 report](#), Docker was cited as a leader in the enterprise container platform category with Docker Enterprise receiving a “differentiated” rating in eight of ten criteria. Forrester added that Docker Enterprise “leads the pack with a robust container platform well-suited for the enterprise”. Docker Enterprise is the only independent container platform that enables developers to seamlessly build and share any application — from legacy to modern — and operators to securely run them anywhere - from hybrid cloud to the edge. Docker Enterprise is the only container platform in the market that can deliver the following:

High-Velocity Innovation - Docker Enterprise enables organizations to rapidly deliver engaging new customer experiences and transform existing processes. It does this by enabling developers to accelerate time-to-productivity and production, with faster developer on-boarding and workflows. Operators are then able to bring applications to market faster by modernizing the way they're built, managed and secured.

Freedom of Choice - Organizations have optimal agility and flexibility to meet their business needs. They can easily adapt to the next technologies on their own timeline, while leveraging existing knowledge and processes. Developers have the freedom to select the best tools, languages, application stacks, and deployment environments for each project. Operators have the freedom to pursue the right operational strategy for the business - any app, any OS, any infrastructure.

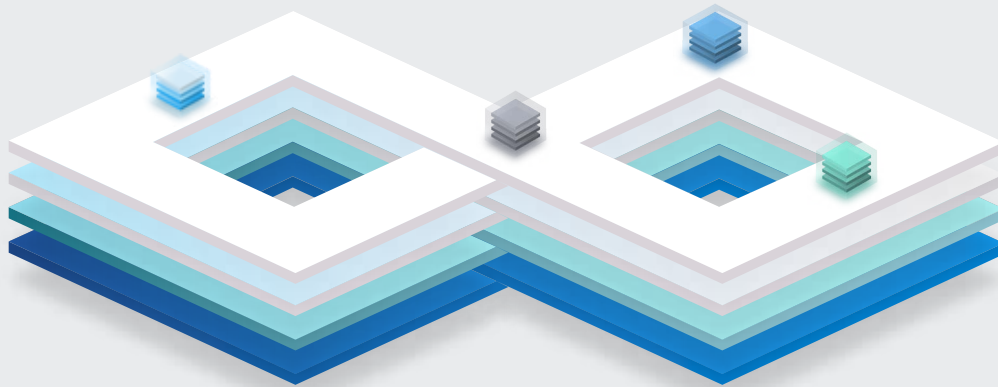
Intrinsic Security - Organizations can continuously ensure compliance and mitigate risk without slowing down innovation. Docker Enterprise is the only platform that can provide trusted and certified end-to-edge security with automated governance and compliance throughout the application lifecycle.

A Complete End-to-End Solution

Docker Enterprise is the only container platform that spans from desktop to servers, providing an integrated approach to help organizations innovate across the entire lifecycle of a containerized application. It is built on the foundation of the world's most widely used Docker Engine and world's largest library of container images with Docker Hub.

Docker Enterprise 3.0

Securely build, share and run any application, anywhere



BUILD

Developer Productivity:
Docker Desktop Enterprise



SHARE

Secure Registry and
Collaboration:
Docker Trusted Registry
Docker Hub



RUN

Application Runtime and
Orchestration:
Docker Engine Enterprise
Docker Universal Control Plane
Kubernetes and Swarm

Docker Enterprise features the following key end-to-end services:

Docker Orchestration Services (featuring Docker Kubernetes Service)

- The first and only desktop-to-server certified Kubernetes distribution that supports Helm charts, Kubernetes YAML and Docker Compose files
 - Automated lifecycle management capabilities that simplify initial installation and configuration and ongoing Day 2 operations
 - Interchangeable orchestration between Swarm and Kubernetes for Linux and Windows*
-

Docker Security Services

- Secure from end-to-edge, with a multi-layered security approach that is delivered by design and by default
 - Scalable security solutions built on a policy-based governance model that supports secure multi-tenancy
 - Security that integrates with the software development process to operate at the speed of DevOps
-

Docker Content Services

- Collaborate and share content amongst a team, multiple teams, or the broader Docker Community with Docker Hub
 - Build and innovate on top of official and certified images that have been tested and validated in Docker Hub
 - Manage and operate a globally consistent private content repository for distributed development teams with the most advanced private registry for container images with Docker Trusted Registry.
-

Docker Application Services

- Docker Application is a new first-class object across the full Docker platform that allows organizations to easily package and collaborate on applications that consist of group of related containers that work together to form an application, all based on the CNAB standard
 - Define and build Docker Applications with Docker Desktop Enterprise, leveraging Application Templates and Application Designer tools to deliver production-ready applications.
 - Collaborate, manage and scan Docker Apps in Docker Hub and Docker Trusted Registry
 - Easily deploy Docker Apps to different environments with parameterized fields for simple runtime customization
-

Docker Operations Management

- Automated lifecycle management of the container environment for an easy and repeatable way to install, configure, backup, manage and scale the platform across hybrid/multi-cloud
 - Manage all system components from an integrated web console including enhanced health status dashboards that provide greater insight into node and container metrics and allow for faster troubleshooting of issues
 - Distribute, manage and secure Docker Desktop Enterprise environments across development teams
-

In addition, Docker Enterprise delivers:

Extensibility that connects the container platform easily into other enterprise tools including preferred storage and networking solutions, logging and monitoring tools.

Enterprise support and services with advisory capabilities to help resolve potentially complex process, orchestration, integration and workflow challenges.

Certification to assure interoperability with the ecosystem and other data center platforms.

How Docker Enterprise Spurs Innovation

Docker has a unique approach that provides a broader view of digital transformation. It is one that focuses not just on the new microservices and cloud-native applications; it also looks at an organization's existing application portfolio and how to bring them forward into the digital era:

- **Replatform legacy applications** - For a subset of applications, the cost or effort to rewrite these applications is not worthwhile, but there is still a need to "keep the lights on". Containerizing these applications can help organizations reduce their costs through further server consolidation or make them easy to move to the cloud. Once containerized, these applications are easier to patch, maintain and secure. Docker has a set of tools and methodologies to help customers containerize these legacy applications at scale, without requiring code changes.
- **Modernize brownfield applications** - For another set of applications that are being actively maintained, containerizing is the first step to further application modernization. Once containerized, organizations often begin the work of segmenting off specific capabilities and building new microservices to replace old monolithic architectures or replacing key building blocks and moving them to cloud-based services (e.g. moving to DBaaS).
- **Accelerate greenfield applications** - Finally, organizations need to build new and compelling experiences for their customers and flexible, responsive systems for their business. New greenfield applications can come in many different flavors and architectures. Docker's methodology puts the emphasis in the innovation process so that developers can make technology and architecture choices that best fit the needs of the application. With Docker's focus on choice and flexibility, organizations have the freedom to use existing stacks or explore new ones.

Finally, Docker's approach is to prepare organizations for whatever is next, whether that is new business models, new opportunities, or responding to new competitive threats.

“What we sought out to do (with Docker Enterprise) is create a multi-lane highway that could accelerate application delivery into the cloud in a way that gave us better portability, better speed, and better agility for our development teams...a model where we could containerize our traditional legacy applications and get them to the cloud, modernize some of them into microservices, and fuel innovation around net-new microservices.”

Eric Drobisewski
Senior Architect
Liberty Mutual

In addition, Docker Enterprise has tangible benefits that immediately deliver a high return-on-investment:

- **Unified operations.** When everything is standardized and follows the same operational patterns, it's easier for IT teams to explore new technology areas – and for the company to adapt and embrace new services. [Franklin American runs a single cluster](#) that supports the development, test and production environments.
- **Leverage existing teams and processes.** This goes back to standardization. With a common platform, processes become repeatable. It's easier and faster to experiment or just make iterative changes.
- **Respond to risks and threats.** The agility and standardization offered by a container platform makes it easier to apply consistent security to protect the organization from threats.
- **Increase data center utilization by 3x.** Even with virtualization, most data centers operate at 20 percent utilization – at best. Containerization increases utilization 50 or 60 percent by eliminating redundant operating systems and further consolidating systems.
- **Decrease IT operating costs.** Lifecycle management and infrastructure standardization make system patching, application updates and even rollbacks much faster. Cornell University accelerated application deployment times by 14x; [Kadaster](#), the Dutch land registry, went from one new deployment a month to as many as 500.
- **Fund innovation.** As [a Fortune 100 insurance company put it in their DockerCon presentation](#), companies can “self-fund innovation” since the savings from a container platform can get reinvested in innovation.

Taking the Next Steps

What organizations should do next depends on where they are now, and what they want to achieve.

- **For organizations that have not started on the containerization journey:** Identify a first project. Show early success bringing a set of containerized applications to production - either legacy, brownfield, or greenfield - and begin planning an innovation team.
- **For organizations that are learning and experimenting with containers or have deployed a container engine for a single application:** Found an innovation team to explore what else is possible. Invite other teams to provide input. Find out what they need. It's possible to go from a very limited pilot to a platform by identifying opportunities where containerization can have a big impact.
- **For organizations that are up and running with container orchestration:** If there isn't an innovation initiative, now is the time to start one. Take the early success to other parts of the organization. Invest in [training and certification](#) to make sure teams know how to get the most out of a container platform.



[Click here](#) to learn more and get started with Docker Enterprise or [contact sales](#) for more information.





www.docker.com

