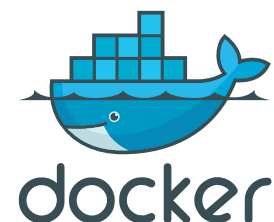


# Modern App Architecture for the Enterprise

Delivering agility, portability and control with Docker  
Containers as a Service (CaaS)



# Executive Summary

Developers don't adopt locked down platforms. In a tale akin to Goldilocks and the Three Bears, app teams have been searching for the right mix of freedom and control for their app environments. Along the way, they discovered that Platform as a Service (PaaS) models were too high-level, abstracted and restrictive - sacrificing flexibility for a completely locked down, black box style environment. Similarly, Infrastructure as a Service (IaaS) offerings with their respective container services are not sufficient as it provides a myopic view of what resides solely in that infrastructure silo. In looking for that right-sized solution, organizations are offering a Docker Containers as a Service (CaaS) environment to provide agility for development teams, control for operations teams and portability of apps across any infrastructure - from on-premises datacenters to public cloud, across a vast array of network and storage providers.

The Docker platform provides an integrated suite of capabilities for an infrastructure agnostic CaaS model). With this solution, IT operations teams are able to secure, provision and manage both infrastructure resources and base app content while developers are able to build and deploy their apps in a self- service manner.

In this whitepaper, we will discuss the drivers toward new software models, the capabilities of the Docker platform, detail the requirements of a CaaS and specify how they are critical to solving core challenges in building, shipping and running apps anywhere.

## Key Takeaways Include:

- How the cloud, data and microservices are changing the software that runs business
- Understanding the Docker journey
- Docker Containers as a Service (CaaS) capabilities and benefits

# Transforming business through software

Gone are the days of private datacenters running off-the-shelf software and giant monolithic code bases that you updated once a year. Everything has changed. Whether it is moving to the cloud, migrating between clouds, modernizing legacy or building new apps and data structure, the desired results are always the same – speed. The faster you can move defines your success as a company.

Software is the critical IP that defines your company even if the actual product you are selling may be a t-shirt, a car, or compounding interest. Software is how you engage your customers, reach new users, understand their data, promote your product or service and process their order.

To do this well, today's software is going bespoke. Small pieces of software that are designed for a very specific job are called microservices. The design goal of microservices is to have each service built with all of the necessary components to “run” a specific job with just the right type of underlying infrastructure resources. Then, these services are loosely coupled together so they can be changed at anytime, without having to worry about the service that comes before or after it.

This methodology, while great for continuous improvement, poses many challenges in reaching the end state. First it creates a new, ever-expanding matrix of services, dependencies and infrastructure making it difficult to manage. Additionally, it does not account for the vast amounts of existing legacy apps in the landscape, the heterogeneity up and down the app stack and the processes required to ensure this works in practice.

## The Docker Journey and the Power of AND

In 2013, Docker entered the landscape with app containers to build, ship and run apps anywhere. Docker was able to take software and its dependencies package them up into a lightweight container. Similar to how shipping containers are today, software containers are simply a standard unit of software that looks the same on the outside regardless of what code and dependencies are included on the inside. This enabled developers and IT ops teams to transport them across infrastructures and various environments without requiring any modifications and regardless of the varying configurations in the different environments. The Docker journey begins here.



### **Agility:**

The speed and simplicity of Docker was an instant hit with developers and is what led to the meteoric rise in the open source project. Developers are now able to very simply package up any software and its dependencies into a container. Developers are able use any language, version and tooling because they are all packaged in a container, which “standardizes” all that heterogeneity without sacrifice.

At the core of the Docker journey is the power of AND. Docker is the only solution to provide agility, portability and control for developers and IT operations team across all stages of the app lifecycle. From these core tenets, Containers as a Service (CaaS) emerges as the construct by which these new apps are built better and faster.



### **Portability:**

Just by the nature of the Docker technology, these very same developers have realized their app containers are now portable - in ways not previously possible. They can ship their apps from development, to test and production and the code will work as designed every time. Any differences in the environment did not affect what was inside the container. Nor did they need to change their app to work in production. This was also a boon for IT operations teams as they can now move apps across datacenters for clouds to avoid vendor lock in.

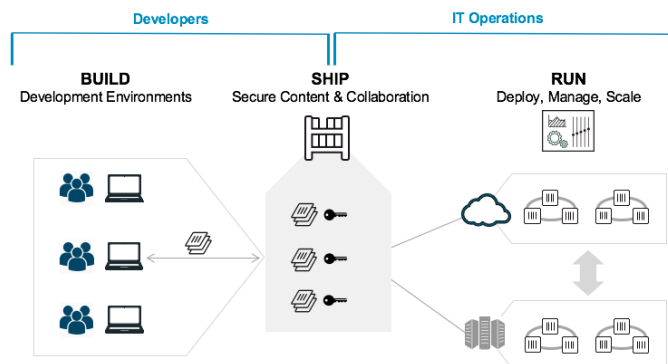


### **Control:**

As these apps move along the lifecycle to production, new questions around security, manageability and scale need to be answered. Docker “standardizes” your environment while maintaining the heterogeneity your business requires. Docker provides the ability to set the appropriate level of control and flexibility to maintain your service levels, performance and regulatory compliance. IT operations teams can provision, secure, monitor and scale the infrastructure and apps to maintain peak service levels. No two apps or businesses are alike and the Docker allows you to decide how to control your app environment.

## Docker Containers as a Service (CaaS)

What is Containers as a Service (CaaS)? It is an IT managed and secured app environment of infrastructure and content where developers can in a self service manner, build and deploy apps.



In the CaaS diagram above, development and IT operations team collaborate through the registry. This is a service in which a library of secure and signed images can be maintained. From the registry, developers on the left are able to pull and build software at their pace and then push the content back to the registry once it passes integration testing to save the latest version. Depending on the internal processes, the deployment step is either automated with tools or can be manually deployed.

The IT operations team on the right in above diagram manage the different vendor contracts for the production infrastructure such as compute, networking and storage. These teams are responsible provision the compute resources needed for the app and use the Docker Universal Control Plane to monitor the clusters and apps over time. Then, they can move the apps from one cloud to another or scale up or down a service to maintain peak performance.

## Key Characteristics and Considerations

The Docker CaaS provides a framework for organizations to unify the variety of systems, languages and tools in their environment and apply the level of control, security or freedom required for their business. As a Docker native solution with full support of the Docker API, Docker CaaS can

seamlessly take the app from local development to production without changing the code and streamlining the deployment cycle.

The following characteristics form the minimum requirements for any organization's app environment. In this paradigm, development and IT operations teams are empowered to use the best tools for their respective jobs without worrying of breaking systems, each other's workflows or lock-in.

1. **Dev and Ops.** Many tools specifically address the functional needs of only one team; however, CaaS breaks the cycle for continuous improvement. To truly gain acceleration in the development to production timeline, you need to address both users along a continuum. Docker provides unique capabilities for each team as well as a consistent API across the entire platform for a seamless transition from one team to the next.
2. **Any app architecture.** Organizations have a variety of apps in their portfolio from monoliths to microservices. A modern app platform needs to be able to address all the different apps within the infrastructure in a common framework.
3. **Any app stage.** From continuous integration to delivery and DevOps, these practices are about eliminating the waterfall development methodology and the lagging innovation cycles with it. By providing tools for both the developer and IT operations, Docker is able to seamlessly support an app from build, test, stage to production.
4. **Any language.** Developer agility means the freedom to build with whatever language, version and tooling required for the features they are building at that time. Also, the ability to run multiple versions of a language at the same time provides a greater level of flexibility. Docker allows your team to focus on building the app

instead of thinking of how to build an app that works in Docker.

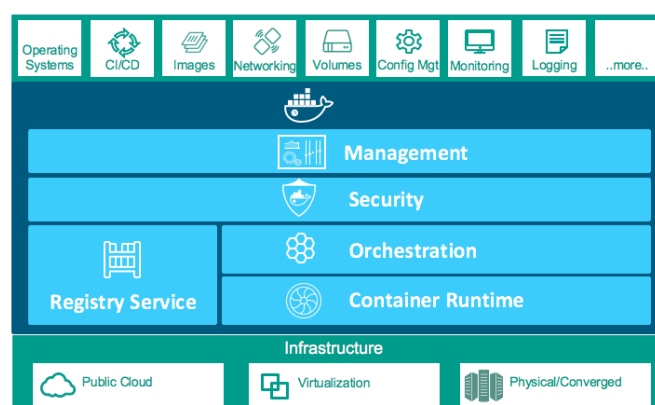
5. **Any operating system.** The vast majority of organizations have more than one operating system. Some tools just work better in Linux while others in Windows. App platforms need to account and support this diversity, otherwise they are solving only part of the problem. Originally created for the Linux community, Docker and Microsoft are bringing forward Windows Server support to address the millions of enterprise apps in existence today and future apps.
6. **Any infrastructure.** When it comes to infrastructure, organizations want choice, backup and leverage. Whether that means you have multiple private data centers, a hybrid cloud or multiple cloud providers, the critical component is the ability to move workloads from one environment to another, without causing app issues. The Docker technology architecture abstracts the infrastructure away from the app allowing the app containers to be run anywhere and portable across any other infrastructure.
7. **Open APIs, pluggable architecture and ecosystem.** A platform isn't a platform as an island to itself. Implementing new technologies is not possible if you need to re-tool your existing environment first. A fundamental principle of Docker is a platform that is open. Being open means APIs and plugins to easily integrate into your existing environment and processes. This openness invites a rich ecosystem to flourish and provide you with more flexibility and choice in adding specialized capabilities.

Although many, these characteristics are critical as the new bespoke application paradigms only invite in greater heterogeneity into your technical

architecture. The Docker CaaS platform is fundamentally designed to support that diversity, while providing the appropriate controls to manage at any scale.

### Docker CaaS Platform Technology

The Docker CaaS platform provides an integration solution of container runtime, orchestration, registry services, security and management. This platform is open and extensible to existing systems in your environment and can run on any infrastructure.



### Docker Datacenter

For organizations who need to keep their IP within their network, Docker Trusted Registry and Docker Universal Control Plane can be deployed on-premises or in a VPC and connected to your existing infrastructure and systems like storage, Active Directory/LDAP, monitoring and logging solutions. Trusted Registry provides the ability to store and manage images on your storage infrastructure while also managing role based access control to the images. Universal Control Plane provides visibility across your Docker environment including Swarm clusters, Trusted Registry repositories, containers and multi container applications.

Your Docker CaaS can be designed to provide a centralized point of control and management or allow for decentralized management to empower individual application teams. The flexibility allows you to create a model that is right for your business, just like how you choose your infrastructure and implement processes. CaaS is an extension of that to build, ship and run applications.

Many IT initiatives are enabled and in fact, accelerated by CaaS due to its unifying nature across the environment. Each organization has their take on the terms for the initiatives but they range from things like containerization, which may involve the “lift and shift” of existing apps, or the adoption of microservices to continuous integration, delivery and DevOps and various flavors of the cloud including adoption, migration, hybrid and multiple. In each scenario, Docker CaaS brings the agility, portability and control to enable the adoption of those use cases across the organization.

## The Power of AND

In conclusion, the shifts in cloud, applications and data have changed the technology and business conversation from “how are you reducing my costs” to “how are you accelerating my business.” As you embark on your journey, Docker provides the added flexibility to choose where you store your application content and host your control plane. Whatever your business, Docker Containers as a Service (CaaS) platform delivers agility, portability and control with the ability to build the best applications as fast and often as possible and providing peak performance of these services at optimal costs with no lock in.

