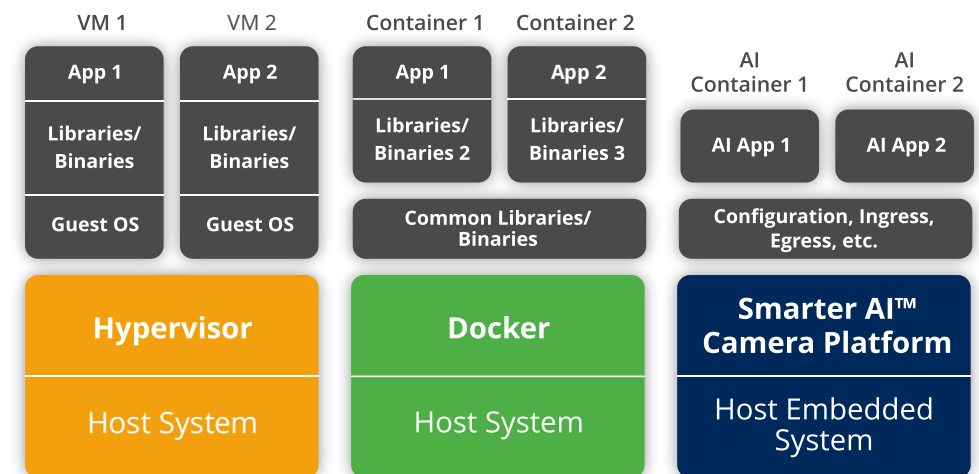


AI Containers™ vs. Docker & Kubernetes

AI Containers™ vs. Docker

- For more information about AI Containers™, please visit the [AI Containers™](#) page.
- [Containers](#) and [Docker](#), in particular, have been a significant (r)evolution in IT and in the way to code, test, deploy, and manage applications. It started with the [Virtual Machine](#) revolution, 20 years ago, enabling IT administrators to optimize IT infrastructure (compute, storage, and network) as well as increase availability.
- Containers, often referred to as OS-level virtualization, remove the OS from virtualization, enabling Applications to run on top of other applications contained in images. This technology allowed for software-centric deployments (code, test, deployment, and maintenance), known today as [DevOps](#).
- Docker and Containers, in general, are very useful to deploy applications on IT and cloud systems; however, deploying AI Applications to [embedded devices](#) with real-time video, audio and sensors has its own set of challenges, mostly related to plumbing and configuration:



- Ingress: which imager/sensor has to be connected to an AI Application / AI Model?
- Egress: how to standardize the format of inferences and inference quality metrics from the AI Application / AI model?
- Data handling: How to resize/transform data streams, easily, efficiently, and automatically on an embedded device?
- Scale: How to deploy Edge AI without requiring expensive and hard to find software developers?
- Configuration: how to standardize the configuration of these AI Applications?
- How to provide easy access to [AI Acceleration](#), Storage, [Crypto Cores](#), and [Vaults](#).

Unfortunately, Docker containers do not provide solutions to those challenges, worst they are adding an additional layer of software, using compute, and complexity lengthening development cycles. Finally, a good way to understand AI Containers™ is to compare them to [Android/iOS](#) applications. Most interactions between the device hardware and the Application is standardized (e.g., camera, microphone, etc.), AI Containers™ do the same for AI Applications.

We designed AI Containers™ to facilitate the deployment of AI Applications to the edge by System Administrators with a graphical web tool.

AI Containers™ vs. Kubernetes



kubernetes

[Kubernetes](#) is an open-source container orchestration system. In other words, it's a convenient tool to orchestrate the deployment of containers on host machines, whether those are in a local data center or are in a cloud system. Kubernetes has a lot of fancy features that are heavily biased towards enterprise-class IT, such as automating deployment, [scaling](#), and operations of application containers across [clusters](#) of hosts. Kubernetes, which is the foundation of many [PaaS](#) & [IaaS](#) systems, works with Docker and other container systems.

Deploying AI Applications to millions of embedded devices in the field with intermittent network connections (e.g., [4G/LTE](#), [5G](#), [Wi-Fi](#)), [Over-the-Air](#) (OTA) is a completely different ball game. Furthermore, as compute capability is limited and varies a lot on embedded devices, you have to understand the compute performance level required, as well as the sensor types needed to run a specific AI Application. On top of that, system administrators want to have one single interface to manage their camera networks, like deploying AI Applications as well as Firmware and OS updates.

[Console](#) is an [API based](#) web portal to let you configure & manage large deployments of AI-Driven and connected camera networks. [Console](#) enables system administrators to push AI Containers™ to edge devices, [Over-the-Air](#). On top of that, Console allows system administrators to update [OS](#), [firmware](#), and camera apps “en masse” and Over-the-Air. AnyConnect offers a wealth of AI Applications in its marketplace: the AI Store™. Deploying an AI Application through the AI Store™ is almost as easy as installing an App on your phone. Console is API based and integrates with existing deployment management systems.

Summary – AI Containers™ vs. Docker & Kubernetes

	AI Containers™	Docker & Kubernetes
Solution for containerized AI Applications mostly at the edge.	✓	⚠
Simple and graphical data ingress management	✓	⚠
Simple and graphical inference data egress management	✓	⚠
Simple and graphical AI Application Configuration management	✓	⚠
Designed for OTA deployment	✓	✗
Designed for OTA deployment on difficult and intermittent networks (Cellular/Wi-Fi, etc.)	✓	✗
Encrypted deployment	✓	3rd party for channel encryption
Integrated Access Control management	✓	Optional and complex to setup

Automated and simple deployment of containers on new edge devices	✓	Requires scripting
Mass deployment on a group of edge devices	Yes, one click.	Yes, through Swarm, with complex configuration and clusters (designed for Datacenters).
Inference data gathering	Automatic	Manual
Monitoring AI Application's health and resources consumption	Automatic	Requires scripting
Automated advisory if the device has the right hardware feature & resources	Yes, Automatically	Resource only
Redundancy/replication	Not needed on embedded systems, automated deployment	Through Swarm & Clusters
Solution for general-purpose containerized application, mostly in datacenters.	✗	✓

AI at the edge has never been that simple.

Contact Us