

Deploying Microservices and Containers with Azure Container Service and DC/OS

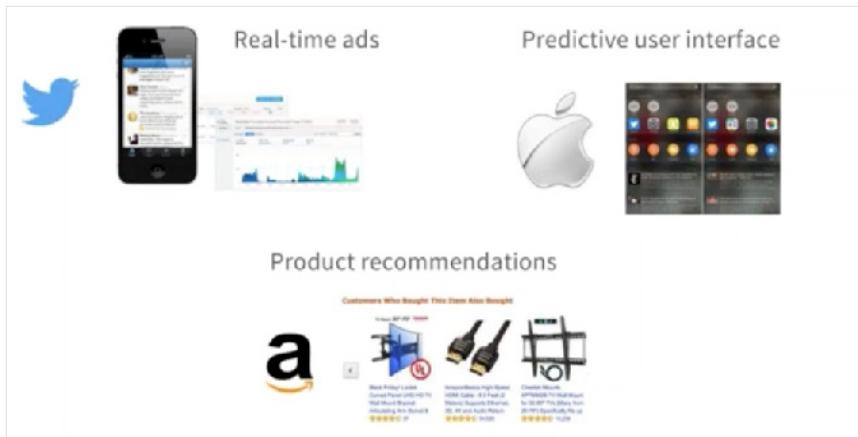
Microsoft Azure



MESOSPHERE

Intro

The explosion of mobile devices, data, and sensors everywhere has enabled the potential for real-time apps for just about everything—paying for products or services, getting directions, banking, healthcare. People expect to be able to tap into the ever-increasing data flowing around us and get precisely what they want and when they want it. Or better yet, before they even know they need it.



While some organizations are moving fast, disrupting their competitors and capitalizing on new opportunities by building personalized, predictive, and real-time applications, others are scrambling to keep up and meet user's expectations. Getting it wrong has terrible consequences for businesses in every industry, but they all face the same challenges in handling the complexity and scale of this new generation of applications:

- They need to ingest, store, and derive insights from vast amounts of complex data in real time.
- They need to be able to quickly respond and adapt to dynamic markets and changing customer demands.
- They need to find more efficient and flexible ways to build and deploy their apps.
- They also have to ensure requirements for security, service quality, and uptime are met.

New infrastructure needed for new applications

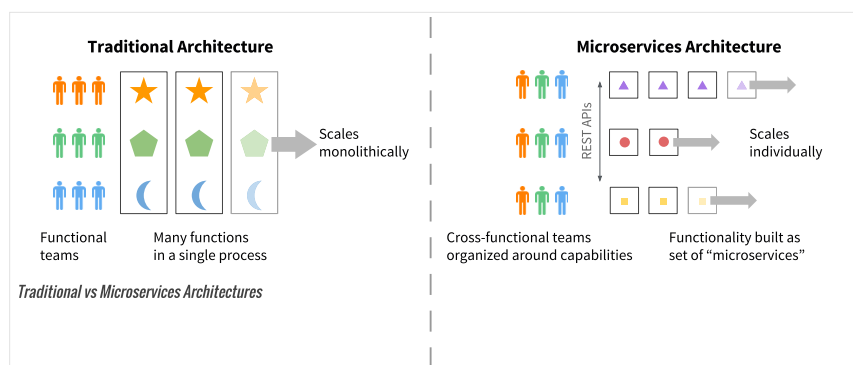
As a result, many companies are recognizing that they need to reinvent how they architect, develop, and operate their applications. Analyst firm Gartner agrees, predicting back in 2015:

“Monolithic, linear application designs (e.g., the three-tier architecture) are giving way to a more loosely coupled integrative approach: the apps and services architecture. Microservice architecture is an emerging pattern for building distributed applications that support agile delivery and scalable deployment, both on-premises and in the cloud. Containers are emerging as a critical technology for enabling agile development and microservice architectures.”¹

Microservices

Microservices provide a flexible approach to developing and deploying modern applications. Instead of one large monolithic application, microservices break down applications into smaller components with key features and functions that all communicate with each other. These services are developed and deployed independently by small, focused teams.

This approach results in less dependency among components, making it easier for developers to build, improve, and scale specific parts of an app without risking the whole application. Teams can focus on developing or refining specific capabilities rather than the entire app, opening the door to daily (or even more frequent) updates and additions to an application. This also makes developing and testing software faster and easier.



1 <http://www.gartner.com/newsroom/id/3143521>

Leading companies including Netflix, Facebook, Google, Twitter, Apple, and Yelp have turned to microservices to develop the applications that run their businesses.

Netflix was one of the earliest adopters of a microservices architecture—enabling them to stream more than 125 million hours of digital entertainment to 81 million customers in over 190 countries every day.

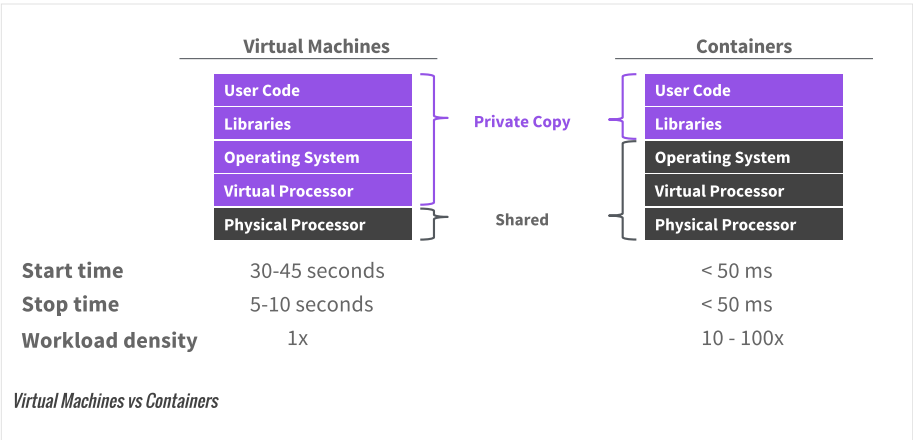
The [Netflix journey to microservices](#) was triggered in August 2008 when the company had a database failure that took four days to fix. They needed a new architecture that could ensure availability, scale, and speed and keep pace with the rapid global expansion of their business. Knowing that they needed the flexibility to scale, they decided to move their platform to the cloud and the only way to take a big complex system from data center to cloud was to redesign and build a microservices architecture.

The Netflix application consists of more than 500 microservices that handle about two billion API requests a day. Every day there are hundreds or even thousands of production changes. There is no central system that gates it all because they want their developers to continue to move fast and innovate. Netflix continues to share best practices about its experiences with microservices at scale and has open sourced a great deal of the technologies powering its microservices. continues to share








Containerization

Applications designed with microservices need a nimble infrastructure for deployment. While microservices can be managed with traditional VMs, Containerization is seen as a promising new technology for simplifying the packaging and deployment of both traditional applications and microservices. Developers bundle an application’s operating system, dependencies, and configuration files together into one package (a container), which can then be pulled from a “repository” and deployed on nearly any server. Developers move more quickly in this model because they can focus much more on application logic and much less on where it will run.

In addition, containers are better than virtual machines because they are faster to spin up (and kill) and allow greater workload density.



While Docker may be the most well known technology when people think of containers, there are many different technology options. Here are a few of the technologies available today:

	<p>Docker is an open-source tool that automates the deployment of applications inside software containers by providing an additional layer of abstraction and automation of operating-system-level virtualization on Linux.</p>
	<p>Apache Mesos is an open-source distributed systems kernel that abstracts the entire datacenter into a single pool of computing resources, using Linux container groups to isolate workload resources..</p>
	<p>Rocket (or rkt), built by CoreOS, is a container manager for Linux clusters that is designed for security, simplicity, and composability within modern cluster architectures.</p>
	<p>App Container (appc) is an open specification that defines several aspects of how to run applications in containers: an image format, runtime environment, and discovery protocol.</p>
	<p>Linuxcontainers.org is the umbrella project behind LXC, LXD, LXCFS and CGManager. The goal is to offer a distro and vendor-neutral environment for the development of Linux container technologies.</p>
	<p>The Open Container Initiative is an open governance structure for the purpose of creating open industry standards around container formats and runtime.</p>
	<p>OpenVZ is a container-based virtualization for Linux that creates multiple secure, isolated Linux containers (otherwise known as VEs or VPSs) on a single physical server.</p>

Containers are especially important for cloud computing and hybrid environments because of their portability—if a machine is running Linux (or, in some cases, Windows), chances are it can launch a container. The workload densities and lower computation overhead of containers mean companies can spend less on capital expenses for their private clouds and operational expenses in the public cloud. “You can’t have a discussion on cloud computing lately without talking about containers,” says Mark Russinovich, CTO of Microsoft Azure. “Organizations across all business segments, from banks and major financial service firms to e-commerce sites, want to understand what containers are, what they mean for applications in the cloud, and how to best use them for their specific development and IT operations scenarios.”

Though easy to launch, containers are difficult to manage at scale

Containerization technology has made it easy to spin up thousands of application instances or services in minutes or even seconds. At the MesosCon 2015 conference in Seattle, for example, Verizon demonstrated how it could launch 50,000 Docker containers in just 72 seconds using the Mesosphere DC/OS platform. Verizon believes this kind of speed, coupled with automated control of containers and the underlying cluster of machines, accelerates application deployment on its network by an order of magnitude.

However, this accelerated development capability also creates entirely new challenges. While individual containers might be relatively small in size and complexity compared with virtual machines, managing a huge volume of containers also creates tremendous operational complexity.

“IT operations teams need to understand the simplicity of containers brings with it significant new challenges that need to be understood before containers can be widely deployed for production applications,” cautions Brian Gracely, an analyst with Wikibon.org.

Moving from development to production can be difficult for several reasons.

For example, running containers in or across clouds requires working in a distributed environment spanning multiple physical and virtual machines and host clusters. Without robust monitoring, management, and orchestration, IT administrators can face a container sprawl of containers left running, dislocated or forgotten. Running containers in production also requires new thinking about security and new demands on network resources, as many large container files might be moving across the network at any given time.

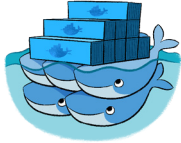



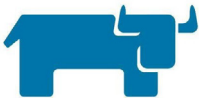

Microservice architecture compounds the challenge. An app might consist of dozens or even hundreds of containers spread across dozens of physical nodes and interdependent containerized services. Combine multiple components with a continuous integration and delivery system, and a cluster may need to spin up and then kill millions of containers per day just to test code. And microservices architectures also notoriously ignore—at least inherently—stateful services such as databases and block storage that many apps require.

Tackling complexity with container orchestration

Before deploying microservices in production, teams need to figure out the best way to do scheduling, cluster management, authentication and service discovery. Container orchestration has emerged as a key enabling technology to help organizations embrace containers and microservices while avoiding pitfalls, or the legwork of building out tooling necessary to do microservices in production.

Orchestration automates the deployment and management of containers and microservices at scale—often including the above considerations—creating the foundation for new service frameworks needed for big data, the Internet of Things, and other modern applications and workloads. However, one shortcoming of many container orchestration platforms is the issue of stateful services, or how reliably you run your databases and storage systems on shared infrastructure.

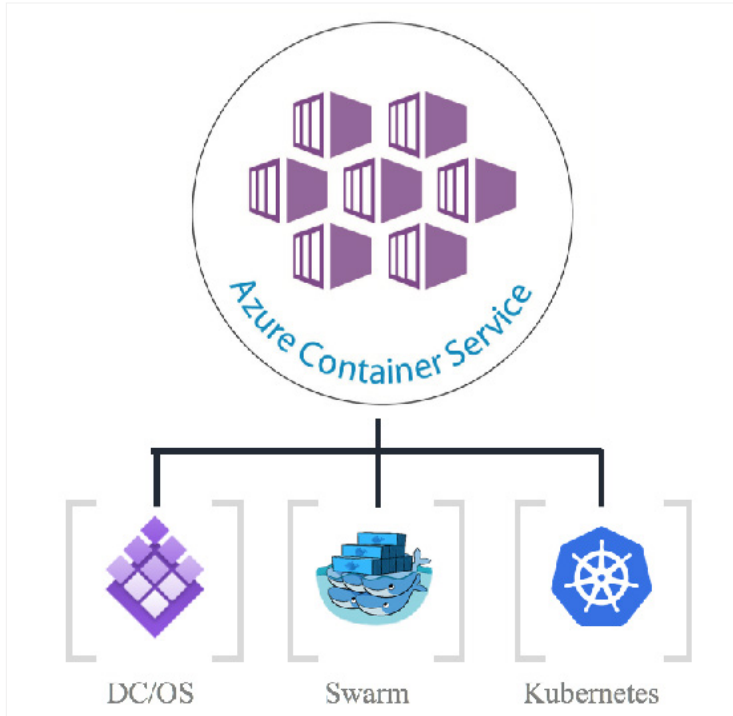
There are [several technology options](#) to choose from for container orchestration. This area is evolving rapidly with many competing products, platforms and frameworks, all with different capabilities. Some of the options are outlined below, all with differing capabilities for scheduling, resource management and service management.

Container Orchestration <ul style="list-style-type: none"> • Scheduling • Resource Management • Service Discovery 		Docker Swarm provides native clustering capabilities to turn a group of Docker engines into a single, virtual Docker Engine.
		Kontena is a container and microservices platform, built on Docker.
		Nomad by HashiCorp is a distributed, highly available datacenter-aware scheduler, also does VMs.
Container Platform <ul style="list-style-type: none"> • Container Lifecycle Management 		Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.
		Rancher natively supports and manages Kubernetes, Swarm and Mesos clusters, provides more than just orchestration.
Distributed Operating System <ul style="list-style-type: none"> • Container Operations • Distributed System Services • Services Catalog + App Ecosystem 		DC/OS combines container orchestration and platform system into a single platform. Based on Mesos, includes everything you need to elastically run microservices in production, including both containers and stateful data services.

Depending on your needs and capabilities, you can choose a single orchestration tool, or a more integrated solution.

Azure Container Service (ACS) supports many of the orchestrators mentioned above, in this e-book we will focus on just one, DC/OS.

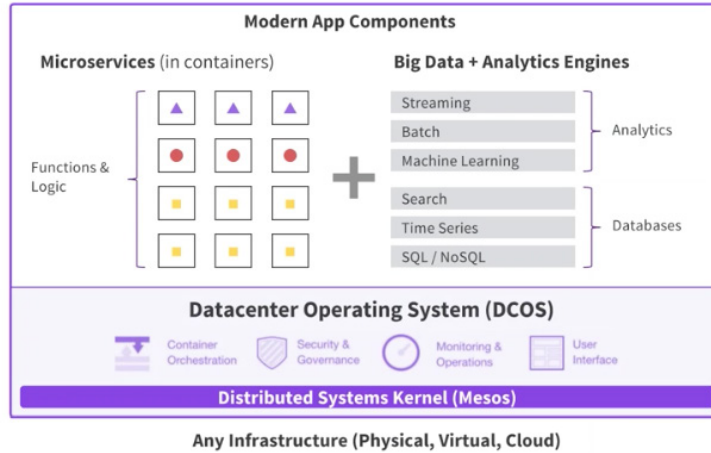
Orchestrating modern apps with Azure Container Service



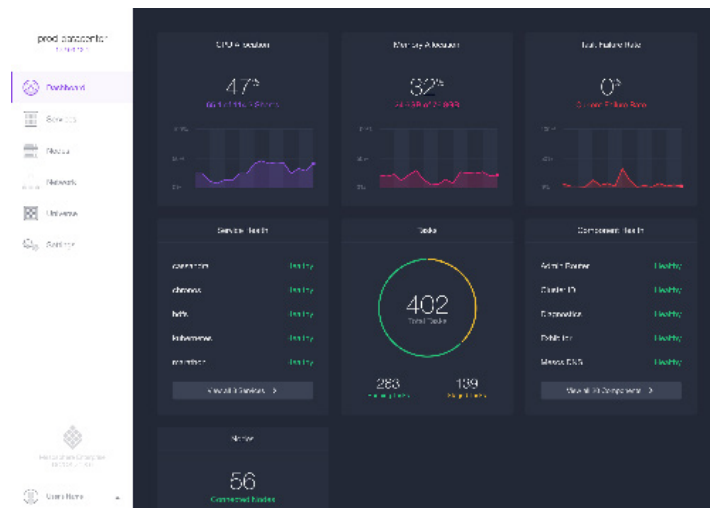
In order to address many of the challenges of adopting and managing containers, Microsoft and Mesosphere have collaborated on a solution that enables the orchestration and overall management of modern applications. The Azure Container Service (ACS), powered by DC/OS, is a fully-managed, open source cloud service that makes it simple to create, configure, and manage modern applications.

For the purposes of this ebook, the two primary pieces of ACS we are focusin on are the Microsoft Azure cloud and Mesosphere DC/OS. Azure, of course, is one of the world's largest public clouds and is used by many of the largest enterprises. Azure features a wide variety of products—ranging from storage to IoT to artificial intelligence— and is located in more than 30 regions across the globe.

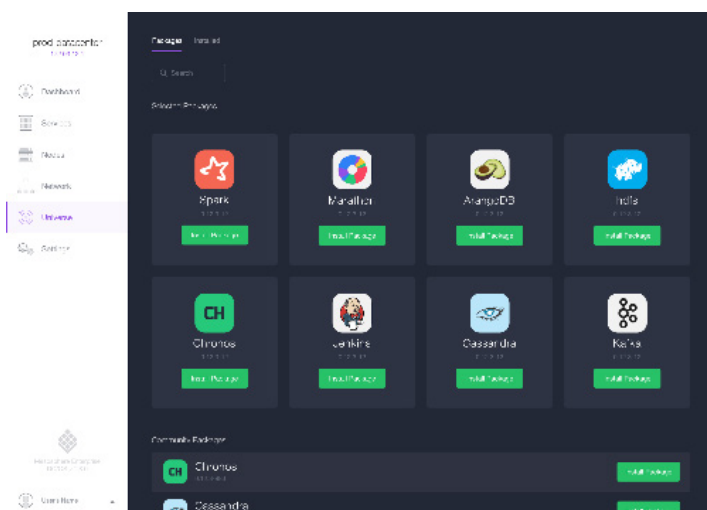
Mesosphere DC/OS provides a single, scalable platform for running modern applications, providing a single platform to easily deploy and manage containers and stateful data services. It is built around Apache Mesos (see table above), which underpins the infrastructure and operations at leading companies including Apple, Twitter, Yelp and Netflix.



Mesosphere DC/OS offers a simple UI for installing, deploying, and managing all these components, as well as a powerful set of command line-based tools for advanced configurations. The DC/OS top-level dashboard provides a single view of total infrastructure health and capacity, spanning all of your services and compute.

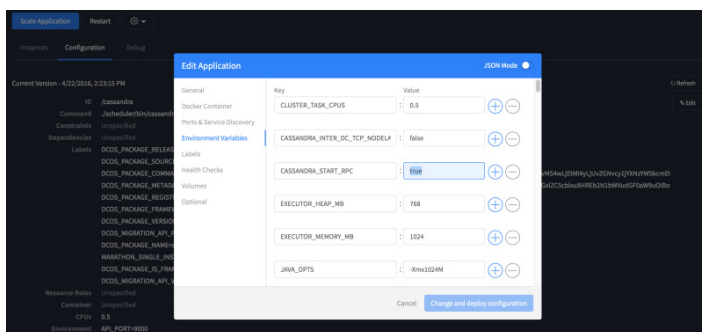


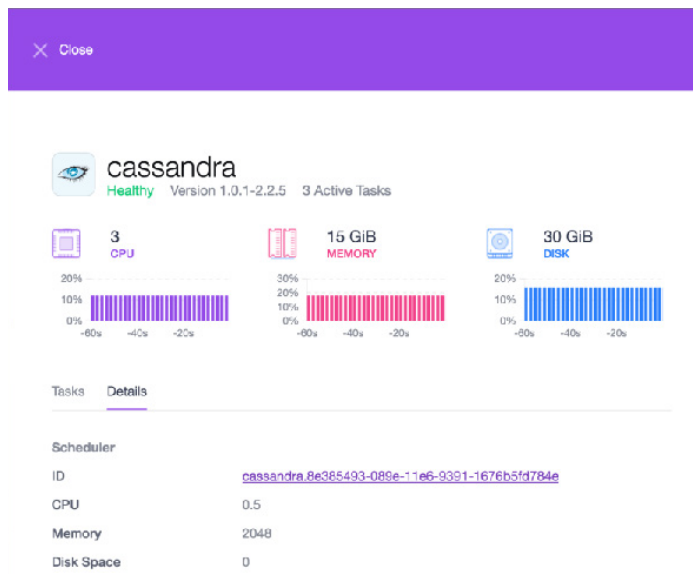
Mesosphere DC/OS turns all of your server resources into a single, logical computer and then gives you an app store-like interface for deploying and updating a vast library of software and services.



The three images below illustrate how simple it can be to install (in this case, using the command line), manage, and even troubleshoot an instance of the Cassandra distributed database on DC/OS.

```
$ dcos package install cassandra
```





When taken as a whole, ACS offers an unparalleled foundation for container orchestrations. It's open; it's portable; and it's proven in production in the world's largest companies. These are some of the core capabilities:

Optimized for open source

Every component of ACS is open source. ACS optimizes the configuration of popular open source tools and technologies for Azure. You get an open solution that offers portability for both your containers and your application configuration. You select the size, the number of hosts, and orchestrator tools, and Azure Container Service handles everything.

Invites collaboration from open source community

The [ACS Engine](#) is a new open source project on GitHub that shares how to deploy Mesosphere DC/OS, Swarm, and Kubernetes and collaborate on best practices for orchestrating containers on Azure, both public and on Azure Stack. The community can collaborate and build the best container infrastructure for Azure. With the ACS Engine, you can modify and customize deployments of the service beyond what is possible today.

Provides flexibility and choice

Choose the tools and solution you want to scale and orchestrate: Mesosphere DC/OS, Kubernetes, or Docker Swarm. ACS is compatible with popular open source tools including visualizers, monitoring, continuous integration, command line tools, and even future tools as they become available.

Enables agile development

With a single command, you can easily deploy, operate and upgrade an ecosystem of complex distributed services such as Spark, Kafka, Cassandra, Artifactory, GitLab, Jenkins, Chronos, and more. Use your existing management practices and tools to move container workloads to the cloud. Get into production quickly and confidently operate at scale using advanced monitoring and troubleshooting tools.

Ensures seamless portability

You can migrate container workloads seamlessly to and from Azure. Azure Container Service only uses open source components in the orchestration layers to give you portability of full applications—not only individual containers—to migrate seamlessly to and from Azure at will.

Ready for the enterprise

Running Mesosphere DC/OS on Azure Container Service is a proven platform that offers operational features such as monitoring, security, compliance as well as support from Microsoft and Mesosphere. The self-healing infrastructure allows enterprises to easily maintain uptime with high availability, automatic failure detection and recovery, and non-disruptive upgrades. Global enterprises can share the same infrastructure across multiple teams and geographies while maintaining complete workload and access isolation.

Case Study: esri

Esri's enterprise mapping and geospatial analytics software integrates maps with rich data to provide location-based insights to more than 350,000 businesses, government agencies and NGOs that collectively create 150 million new maps every day.

Esri's clients want to deploy new IoT and data-driven applications that were pushing the boundaries of Esri's current on-premises software solution. Performance and scalability demands of these new applications required Esri to adopt a new technology platform that leverages cloud computing, open source software, containers and container management software.

This service—built on Microsoft Azure Cloud, Azure Container Service and Mesosphere Enterprise DC/OS, working with the Azure IoT Suite—enables Esri customers to do real-time, predictive mapping at whatever scale they need. Click here to see a [demo](#) of Esri's solution.

"With Azure Container Service running on DC/OS, we can serve a new set of customers with entirely new capabilities in terms of the performance and intelligence of their map and analytic applications. And with our cloud-based platform, we can get these solutions up and running in minutes. This gives Esri and our clients a level of innovation and business agility we've never had before," says Adam Mollenkopf, Esri Real-Time & Big Data GIS Capability Lead

Summary

To be competitive in today's world, companies need to be able to create powerful, data rich applications that give their customers immediate access to the experience they want, when they want it. Companies need to be able to embrace new development and deployment paradigms without getting locked into a single provider. The old model of monolithic software applications and enterprise server sales cannot deliver on this expectation.

Microservices, delivered through containers and cloud, are emerging as the best path forward to develop today's modern applications. Containers can run on any hardware, on any cloud, or in a hybrid environment. They offer an open and portable solution for agile development.

Mesosphere DC/OS on Azure Container Service makes deploying and operating containers easy. Combined, they are a proven platform for managing the next generation of modern applications. Mesosphere and Microsoft provide a simple way for enterprises to scale containers in production.

Resources

Get started or deepen your knowledge with these selected resources:

Azure Container Service 101

<https://channel9.msdn.com/Shows/Azure-Friday/Azure-Container-Service-101?ocid=player>

Building applications using Azure Containers Service

<https://channel9.msdn.com/Events/Build/2016/B822?ocid=player>

Deep dive into Azure Container Service

<https://channel9.msdn.com/Events/Ignite/2016/BRK4008-TS>

Container Orchestration Wars (video webcast)

<https://mesosphere.com/resources/container-orchestration-wars/>

Meet Marathon: Production-ready container orchestration at scale

<https://mesosphere.com/blog/2016/02/17/marathon-production-ready-containers/>

Microsoft Azure



MESOSPHERE

