Kubernetes vs. Docker

Build, deliver, and scale apps faster with container technologies that work better together.

Start free (/en-us/free/services/kubernetes-service/)

The Kubernetes vs. Docker question

The conversation around Kubernetes vs. Docker is often framed as either-or: should I use Kubernetes or Docker? This is like comparing apples to apple pie, and it's a common misconception that you must choose one or the other.

The difference between Kubernetes and Docker is more easily understood when framed as a "both-and" question. The fact is, you don't have to choose—Kubernetes and Docker are fundamentally different technologies that work well together for building, delivering, and scaling containerized apps.

Docker and the rise of containerization

Docker is open-source technology—and a container file format—for automating the deployment of applications as portable, self-sufficient containers that can run in the cloud or on-premises. Docker, Inc., although it shares a similar name, is one of the companies that cultivates the open-source Docker technology to run on Linux and Windows in collaboration with cloud providers like Microsoft.

While the idea of isolating environments is not new and there are other types of containerization software, Docker has grown to be the default container format in recent years. Docker features the Docker Engine, which is a runtime environment. It allows you to build and run containers on any development machine; then store or share container images through a container registry like Docker Hub or Azure Container Registry.

As applications grow to span multiple containers deployed across multiple servers, operating them becomes more complex. While Docker provides an open standard for packaging and distributing containerized apps, the potential complexities can add up fast. How do you coordinate and schedule many containers? How do all the different containers in your app talk to each other? How do you scale many container instances? This is where Kubernetes can help.

Kubernetes and container orchestration

Kubernetes is open-source orchestration software that provides an API to control how and where those containers will run. It allows you to run your Docker containers and workloads and helps you to tackle some of the operating complexities when moving to scale multiple containers, deployed across multiple servers.

What is Kubernetes? (/en-us/topic/what-is-kubernetes/)

Kubernetes lets you orchestrate a cluster of virtual machines and schedule containers to run on those virtual machines based on their available compute resources and the resource requirements of each container. Containers are grouped into pods, the basic operational unit for Kubernetes. These containers

and pods can be scaled to your desired state and you're able to manage their lifecycle to keep your apps up and running.

What's the difference between Kubernetes and Docker?

While it's common to compare Kubernetes with Docker, a more apt comparison is Kubernetes vs. Docker Swarm. Docker Swarm is Docker's orchestration technology that focuses on clustering for Docker containers—tightly integrated into the Docker ecosystem and using its own API.

A fundamental difference between Kubernetes and Docker is that Kubernetes is meant to run across a cluster while Docker runs on a single node. Kubernetes is more extensive than Docker Swarm and is meant to coordinate clusters of nodes at scale in production in an efficient manner. Kubernetes pods—scheduling units that can contain one or more containers in the Kubernetes ecosystem—are distributed among nodes to provide high availability.

Kubernetes and Docker—better together

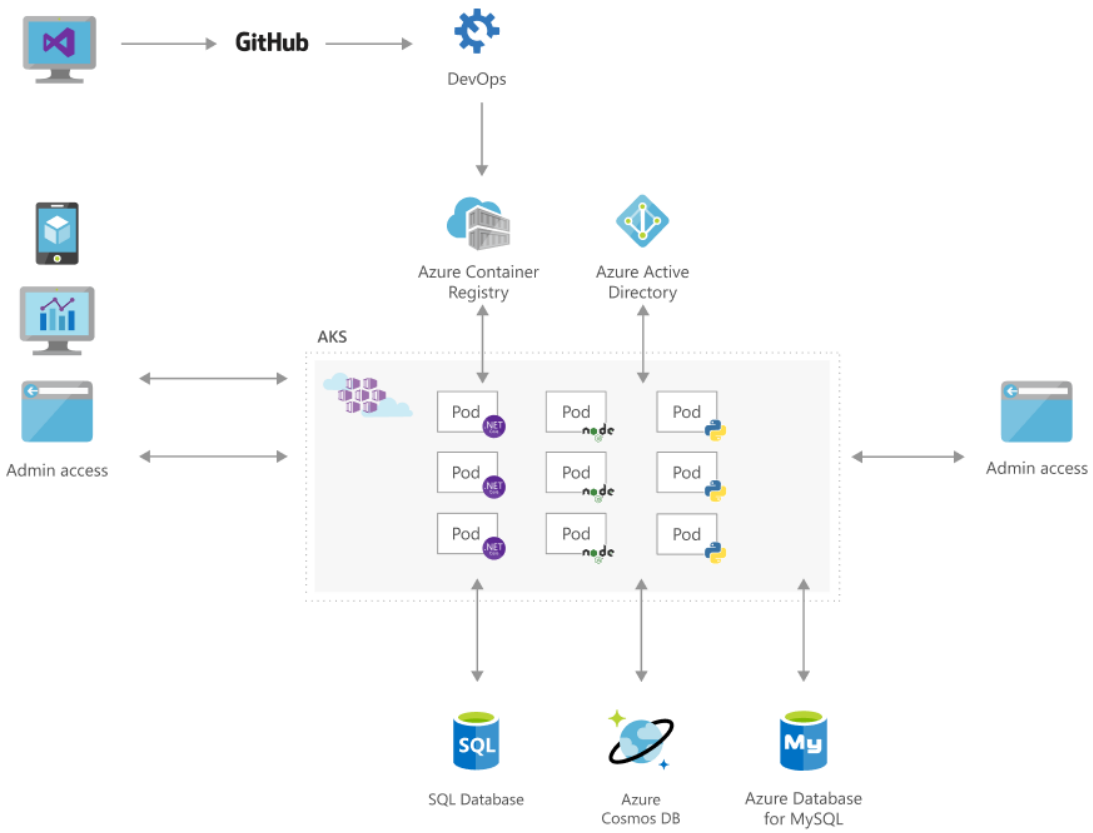
While the promise of containers is to code once and run anywhere, Kubernetes provides the potential to orchestrate and manage all your container resources from a single control plane. It helps with networking, load-balancing, security, and scaling across all Kubernetes nodes which runs your containers. Kubernetes also has built-in isolation mechanism like namespaces which allows you to group container resources by access permission, staging environments and more. These constructs make it easier for IT to provide developers with self-service resource access and developers to collaborate on even the most complex microservices architecture without mocking up the entire application in their development environment. Combining DevOps practices with containers and Kubernetes further enables a baseline of microservices architecture that promotes fast delivery and scalable orchestration of cloud-native applications.

In short, use Kubernetes with Docker to:

- ✔ Make your infrastructure more robust and your app more highly available. Your app will remain online, even if some of the nodes go offline.
- ✔ Make your application more scalable. If your app starts to get a lot more load and you need to scale out to be able to provide a better user experience, it's simple to spin up more containers or add more nodes to your Kubernetes cluster.

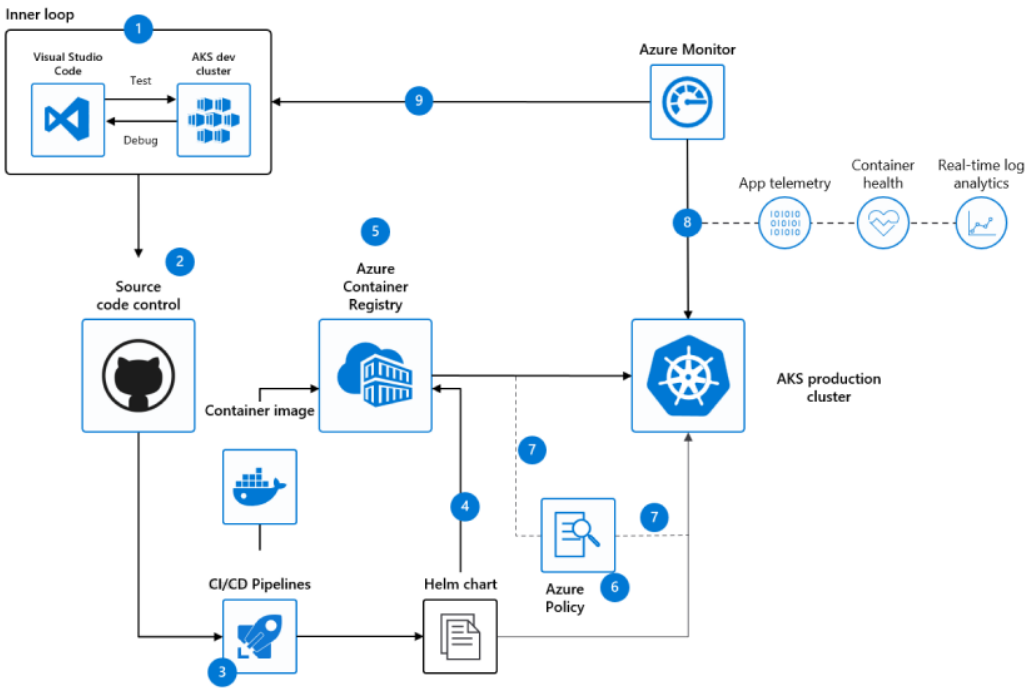
Kubernetes and Docker work together. Docker provides an open standard for packaging and distributing containerized applications. Using Docker, you can build and run containers, and store and share container images. One can easily run a Docker build on a Kubernetes cluster, but Kubernetes itself is not a complete solution. To optimize Kubernetes in production, implement additional tools and services to manage security, governance, identity, and access along with continuous integration/continuous deployment (CI/CD) workflows and other DevOps practices.

Kubernetes and Docker solution architectures in production



Microservices with AKS (https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/microservices-with-aks/)

Use AKS to simplify the deployment and management of microservices based architecture. AKS streamlines horizontal scaling, self-healing, load balancing, secret management.



Secure DevOps for AKS (https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/secure-devops-for-kubernetes/)

DevOps and Kubernetes are better together. Implementing secure DevOps together with Kubernetes on Azure, you can achieve the balance between speed and security and deliver code faster at scale.

Build on the strength of Kubernetes with Azure

Deploying and managing your containerized applications is easy with Azure Kubernetes Service (AKS). AKS offers serverless Kubernetes, an integrated CI/CD experience, and enterprise-grade security and governance. Unite your development and operations teams on a single platform to rapidly build, deliver, and scale applications with confidence.

Learn more about AKS (/en-us/services/kubernetes-service/)

Resources

Videos, webinars, demos, technical sessions

Learn Kubernetes basics (https://aka.ms/k8s/lightboard)

See Kubernetes best practices (https://docs.microsoft.com/en-us/azure/aks/best-practices)

Learn more about containers on Azure (/en-us/product-categories/containers/)

Explore Azure Kubernetes Service (AKS) (/en-us/services/kubernetes-service/)

Tutorials and docs

See Kubernetes deployment strategies (https://docs.microsoft.com/en-us/azure/devops/pipelines/ecosystems/kubernetes/deployment-strategies?view=azure-devops)

Install a Docker extension with Visual Studio (https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-docker)

Run container images in Kubernetes (https://docs.microsoft.com/en-us/azure/aks/tutorial-kubernetes-deploy-application)

Scale an application and Kubernetes infrastructure (https://docs.microsoft.com/en-us/azure/aks/tutorial-kubernetes-scale)

Update an application running in Kubernetes (https://docs.microsoft.com/en-us/azure/aks/tutorial-kubernetes-app-update)

Upgrade AKS cluster (https://docs.microsoft.com/en-us/azure/aks/tutorial-kubernetes-upgrade-cluster)

Take the self-paced Azure Kubernetes workshop (https://aka.ms/k8s/learn/aksworkshop)

See Azure quickstart templates for Kubernetes (/en-us/resources/templates/?term=Kubernetes)

Open source community

Join other AKS users on Github (https://github.com/Azure/AKS), at KubeCon (https://events.linuxfoundation.org/kubecon-cloudnativecon-north-america/), or at a Kubernetes meetup (https://www.meetup.com/topics/kubernetes/) near you.

Find out more about open source on Azure (/en-us/overview/open-source/)

See APIs, SDKs, and open source projects from Azure (https://github.com/Azure)

Get started with Kubernetes and Docker

Start free on Azure (/en-us/free/services/kubernetes-service/)