✓ 100 XP

## What is Kubernetes?

7 minutes

The decoupled design of microservices combined with the atomicity of containers makes it possible to scale out apps, respond to increased demand by deploying more container instances, and to scale back if demand is decreasing.

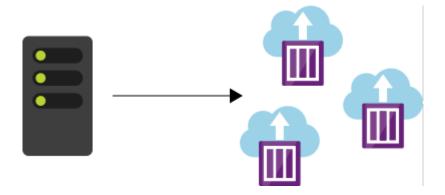
In complex solutions, like the drone-tracking app, the process of deploying, updating, monitoring, and removing containers introduces challenges.

Before looking at what Kubernetes is, here's a summary of a few concepts that are key to containerized workloads.

# What is container management?

Container management is the process of organizing, adding, removing, or updating a significant number of containers.

The drone-tracking app consists of multiple microservices responsible for tasks like caching, queuing, or data processing. Each of these services is hosted in a container and can be deployed, updated, and scaled independently from one another.



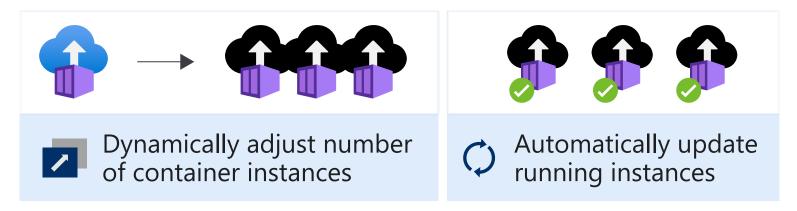
For example, with the drone-tracking app's website, you find that at specific times during the day, you need more instances of the site's caching service to keep up performance, so you add more caching service container instances.

Now, assume that you've increased the number of caching instances and need to roll out a new version of the microservice. You'll have to make sure to update *all* the active containers.

Container management helps you with these otherwise manual tasks.

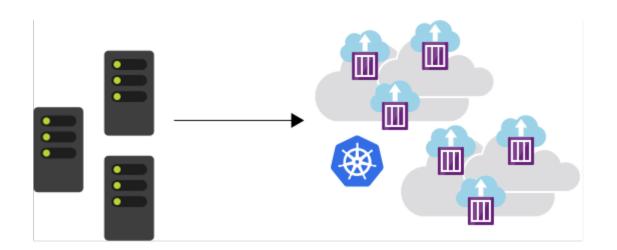
### What is container orchestration?

A container orchestrator is a system that automatically deploys and manages containerized apps. For example, the orchestrator can dynamically respond to changes in the environment to increase or decrease the deployed instances of the managed app, or it can ensure all deployed container instances get updated if a new version of a service is released.



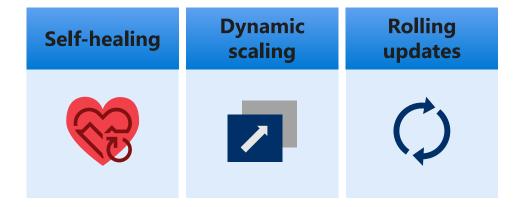
## **Define Kubernetes**

Kubernetes is a portable, extensible open-source platform for managing and orchestrating containerized workloads. Kubernetes abstracts away complex container-management tasks and provides you with declarative configuration to orchestrate containers in different computing environments. This orchestration platform gives you the same ease of use and flexibility you might already know from platform as a service (PaaS) or infrastructure as a service (laaS) offerings.



#### **Kubernetes benefits**

The benefits of using Kubernetes are based on the abstraction of tasks.



#### These tasks include:

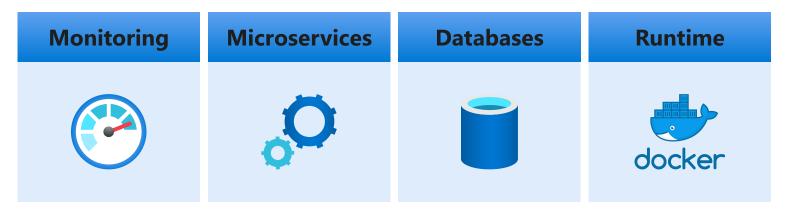
- Self-healing of containers; for example, restarting containers that fail or replacing containers
- Scaling deployed container count up or down dynamically, based on demand
- Automating rolling updates and rollbacks of containers
- Managing storage
- Managing network traffic
- Storing and managing sensitive information such as usernames and passwords

#### (i) Important

Keep in mind that all of the preceding aspects of Kubernetes require configuration and a good understanding of the underlying technologies. For example, you need to understand concepts such as virtual networks, load balancers, and reverse proxies to configure Kubernetes networking.

### **Kubernetes considerations**

With Kubernetes, you can view your datacenter as one large compute resource. You don't need to worry about how and where you deploy your containers, only about deploying and scaling your apps as needed.



However, it's important to understand that Kubernetes isn't a single installed app that comes with all possible components needed to manage and orchestrate a containerized solution:

• Aspects such as deployment, scaling, load balancing, logging, and monitoring are all optional. You're responsible for finding the best solution that fits your needs to address these aspects.

- Kubernetes doesn't limit the types of apps that can run on the platform. If your app can run in a container, it can run on Kubernetes. To make optimal use of containerized solutions, your developers need to understand concepts such as microservices architecture.
- Kubernetes doesn't provide middleware, data-processing frameworks, databases, caches, or cluster-storage systems. All these items are run as containers, or as part of another service offering.
- For Kubernetes to run containers, it needs a container runtime like Docker or containerd. The container runtime is the object that's responsible for managing containers. For example, the container runtime starts, stops, and reports on the container's status.
- You're responsible for maintaining your Kubernetes environment. For example, you need to manage OS upgrades and the Kubernetes installation and upgrades. You also manage the hardware configuration of the host machines, such as networking, memory, and storage.

Cloud services such Azure Kubernetes Service (AKS) reduce these challenges by providing a hosted Kubernetes environment.

These services also simplify deploying and managing containerized apps in Azure. With AKS, you get the benefits of open-source Kubernetes without the complexity or operational overhead of running your own custom Kubernetes cluster.

① Note

Kubernetes is sometimes abbreviated to *K8s*. The 8 represents the eight characters between the K and the s of the word K[*ubernete*]s.

#### **Next unit: How Kubernetes works**

Continue >