What is Kubernetes?

Kubernetes is open-source orchestration software for deploying, managing, and scaling containers.

Overview

Why Use Kubernetes

Beyond Kubernetes

DevOps and Kubernetes

Example Workflow

❯ More

Kubernetes explained

Modern applications are increasingly built using containers, which are microservices packaged with their dependencies and configurations. Kubernetes (pronounced "koo-ber-net-ees") is open-source software for deploying and managing those containers at scale—and it's also the Greek word for helmsmen of a ship or pilot. Build, deliver, and scale containerized apps faster with Kubernetes, sometimes referred to as "k8s" or "k-eights."

Explore Kubernetes with this simple learning path ❯

How Kubernetes works

As applications grow to span multiple containers deployed across multiple servers, operating them becomes more complex. To manage this complexity, Kubernetes provides an open source API that controls how and where those containers will run.

Kubernetes orchestrates clusters of virtual machines and schedules 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, and those pods scale to your desired state.

Kubernetes also automatically manages service discovery, incorporates load balancing, tracks resource allocation, and scales based on compute utilization. And, it checks the health of individual resources and enables apps to self-heal by automatically restarting or replicating containers.

Watch how Kubernetes works   See Common Kubernetes scenari

Why use Kubernetes?

Keeping containerized apps up and running can be complex because they often involve many containers deployed across different machines. Kubernetes provides a way to schedule and deploy those containers—plus scale them to your desired state and manage their lifecycles. Use Kubernetes to implement your container-based applications in a portable, scalable, and extensible way.

Make workloads portable

Because container apps are separate from their infrastructure, they become portable when you run them on Kubernetes. Move them from local machines to production among on-premises, hybrid, and multiple cloud environments—all while maintaining consistency across environments.

Scale containers easily

Define complex containerized applications and deploy them across a cluster of servers or even multiple clusters with Kubernetes. As Kubernetes scales applications according to your desired state, it automatically monitors and maintains container health.

Build more extensible apps

A large open-source community of developers and companies actively builds extensions and plugins that add capabilities such as security, monitoring, and management to Kubernetes. Plus, the Certified Kubernetes Conformance Program requires every Kubernetes version to support APIs that make it easier to use those community offerings.

Get started with Kubernetes

See how to begin deploying and managing containerized applications.

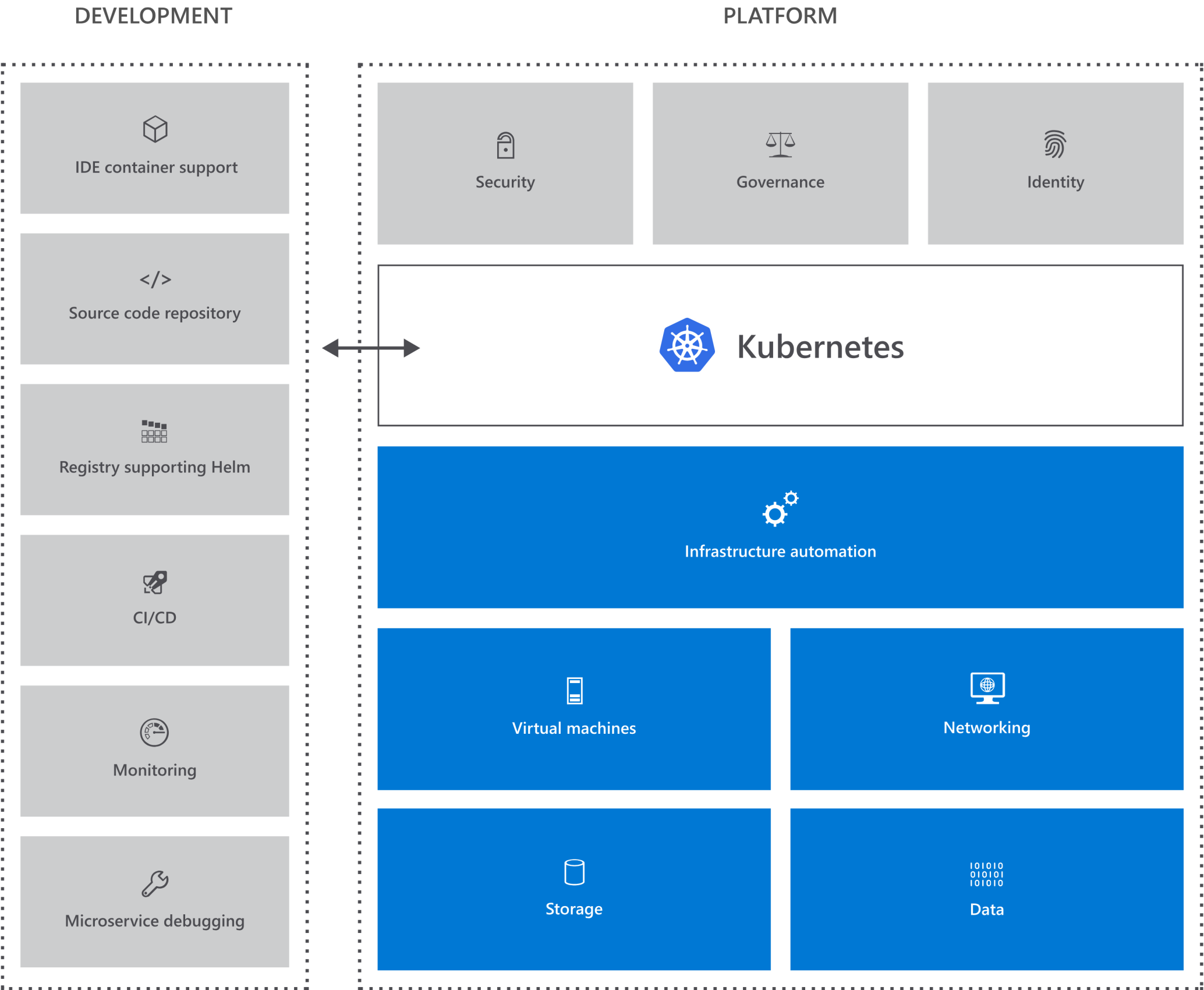Learn more

Follow the learning path

Learn practical skills to get up and running with Kubernetes.

Get the guide

Build on a complete Kubernetes platform

While Kubernetes itself offers portability, scalability, and extensibility, adding end-to-end development, operations, and security control allows you to deploy updates faster—without compromising security or reliability—and save time on infrastructure management. As you adopt Kubernetes, also consider implementing:

✓ Infrastructure automation or serverless Kubernetes to eliminate routine tasks like provisioning, patching, and upgrading.

✓ Tools for containerized app development and continuous integration and continuous deployment (CI/CD) workflows.

✓ Services to manage security, governance, identity and access.

DEVELOPMENT

PLATFORM

IDE container support

Source code repository

Registry supporting Helm

CI/CD

Monitoring

Microservice debugging

Security

Governance

Identity

Kubernetes

Infrastructure automation

Virtual machines

Networking

Storage

Data

Harness Kubernetes with DevOps practices

As a Kubernetes app grows—adding containers, environments, and teams—release frequency tends to increase, along with developmental and operational complexity. Employing DevOps practices in Kubernetes environments allows you to move quickly at scale with enhanced security.

Deliver code faster with CI/CD

While containers provide a consistent application packaging format that eases the collaboration between development and operations teams, CI/CD can accelerate the move from code to container and to Kubernetes cluster in minutes by automating those tasks.

Set up CI/CD for Kubernetes >

Manage resources effectively with infrastructure as code

Infrastructure as code establishes consistency and visibility of compute resources across teams—reducing the likelihood of human error. This practice works with the declarative nature of Kubernetes applications powered by Helm. Combining the two allows you to define apps, resources, and configurations in a reliable, trackable, and repeatable way.

Deploy a Kubernetes cluster with Terraform >

## Accelerate the feedback loop with constant monitoring

Shorten the time between bugs and fixes with a complete view of your resources, cluster, Kubernetes API, containers, and code—from container health monitoring to centralized logging. That view helps you prevent resource bottlenecks, trace malicious requests, and keep your Kubernetes applications healthy.

See how real-time container insights work >

Balance speed and security with DevOps

Bring real-time observability into your DevOps workflow without sacrificing velocity. Apply compliance checks and reconfigurations automatically to secure your build and release pipeline—and your Kubernetes application as a result.
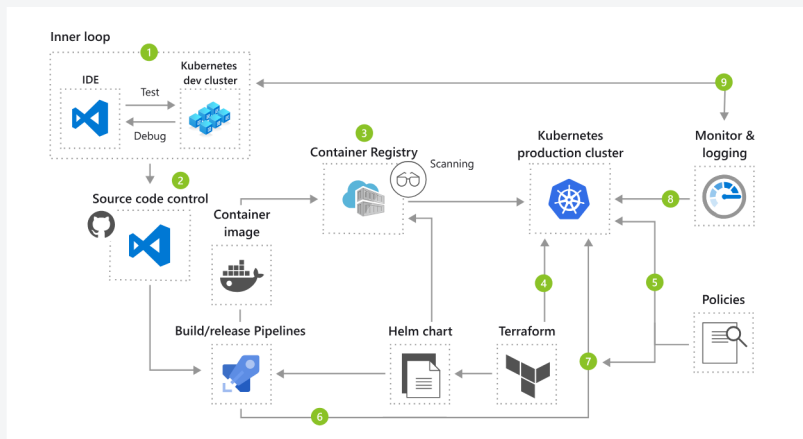
Example DevOps workflow with Kubernetes

1. Rapidly iterate, test, and debug different parts of an application together in the same Kubernetes cluster.

2. Merge and check code into a GitHub repository for continuous integration. Then, run automated builds and tests as a part of continuous delivery.

3. Verify the source and integrity of container images. Images are held in quarantine until they pass scanning.

4. Provision Kubernetes clusters with tools like Terraform. Helm charts installed by Terraform define the desired state of app resources and configurations.

5. Enforce policies to govern deployments to the Kubernetes cluster.

6. The release pipeline automatically executes pre-defined deployment strategy with each code.

7. Add policy audit and automatic remediation to the CI/CD pipeline. For example, only the release pipeline has permission to create new pods in your Kubernetes environment.

8. Enable app telemetry, container health monitoring, and real-time log analytics.

9. Address issues with insights and inform plans for the next sprint.

Learn more about Kubernetes deployment strategy >

Build on the strengths of Kubernetes with Azure

Automate provisioning, upgrading, monitoring, and scaling with the fully managed Microsoft Azure Kubernetes Service (AKS). Get serverless Kubernetes, a simpler development-to-production experience, and enterprise-grade security and governance.

Learn more about AKS >

Draw inspiration and innovation from the Kubernetes community

Kubernetes was created by—and thrives because of—the thousands of individuals and hundreds of organizations who have given their wisdom, code, and continuing support to the people who use it. Build the success of your software on top of their impassioned contributions.

35,000

Contributors

180,000

Commits

Top project

in GitHub

Microsoft contributions to Kubernetes

Bringing open-source ingenuity to enterprises

To make Kubernetes easier for organizations to adopt—and easier for developers to use—Microsoft has tripled the number of employees who participate in the open source project in just three years. Now the third-leading corporate contributor, Microsoft works to make Kubernetes more enterprise-friendly and accessible by bringing the latest learnings and best practices from working with diverse customers to the Kubernetes community.

FAQs – Kubernetes

Expand all | Collapse all

> Where should I start?

> What are some common use cases for Kubernetes?

> What are Kubernetes best practices?

> What is a Kubernetes deployment?

> How do I deploy to Kubernetes using DevOps practices?

> What is Kubernetes vs. Docker?

Resources

More about Kubernetes

Learn Kubernetes basics

See Kubernetes best practices

Learn more about containers

Learn about AKS

Explore Azure Kubernetes Service (AKS)

Watch AKS videos and on demand Azure webinars for demos, top features, and technical sessions.

See Azure quickstart templates for Kubernetes

See AKS regional availability

Join other AKS users on GitHub, at KubeCon, or at a Kubernetes Meetup near you.

Follow step-by-step AKS tutorials:

Create container images from an application

Upload container images to the Azure Container Registry

Deploy an AKS cluster

Run container images in Kubernetes

Scale an application and Kubernetes infrastructure

Update an application running in Kubernetes

Upgrade AKS cluster

Open source and Azure

Find out more about open source on Azure

Ready when you are—try Kubernetes free on Azure   Start free

Get the Azure mobile app

Explore Azure
Explore Azure
What is Azure?
Get started
Global infrastructure
Datacenter regions
Trust your cloud
Customer enablement
Customer stories
Products and pricing
Products and pricing
Products
Pricing
Free Azure services
Flexible purchase options
Cloud economics
Optimize your costs
Solutions and support
Solutions and support
Solutions
Resources for accelerating growth
Solution architectures
Support

Azure demo and live Q&A

Partners

Azure Marketplace

Find a partner

Join ISV Success

Resources

Training and certifications

Documentation

Blog

Developer resources

Students

Events and webinars

Analyst reports, white papers, and e-books

Videos

Cloud computing

What is cloud computing?

What is cloud migration?

What is a hybrid cloud?

What is AI?

What is IaaS?

What is SaaS?

What is PaaS?

What is DevOps?

Change language

English (US)

Your Privacy Choices