



# Kubernetes deployment strategy

Find the optimal Kubernetes rollout strategy for your application.

## Know your Kubernetes deployment options

Whether you want to shorten time to market, create resilient deployments without downtime, release apps and features faster or more frequently, or operate with greater flexibility, choosing the right Kubernetes deployment strategy is key to delivering resilient application and infrastructure. Each Kubernetes deployment strategy has its advantages depending on your goals and needs.

[New to Kubernetes? Learn the basics >](#)

## How Kubernetes deployments work

Before you implement a Kubernetes deployment strategy, you'll need to understand the pieces of a Kubernetes deployment and how they all function together.

## What's in a Kubernetes deployment



### YAML file

The desired state for your Kubernetes cluster—the configuration of Pods—that you describe, which serves as the basis for a Kubernetes deployment.



### Pods

The containers, shared resources, and environment your app or workflow needs to run.



## ReplicaSet

Groups of identically configured pods are called ReplicaSets that ensure the type and number of pods described in the YAML file for a Kubernetes deployment are running at all times. If a pod fails, a new one is created.



## Kube-controller-manager

Changes the current state of the cluster to match the desired state described in the YAML, creating new Pods and ReplicaSets as well as updating or removing existing ones.



## Kube-scheduler

Determines how your pods and ReplicaSets are deployed among your worker nodes in addition to distributing traffic to those nodes.

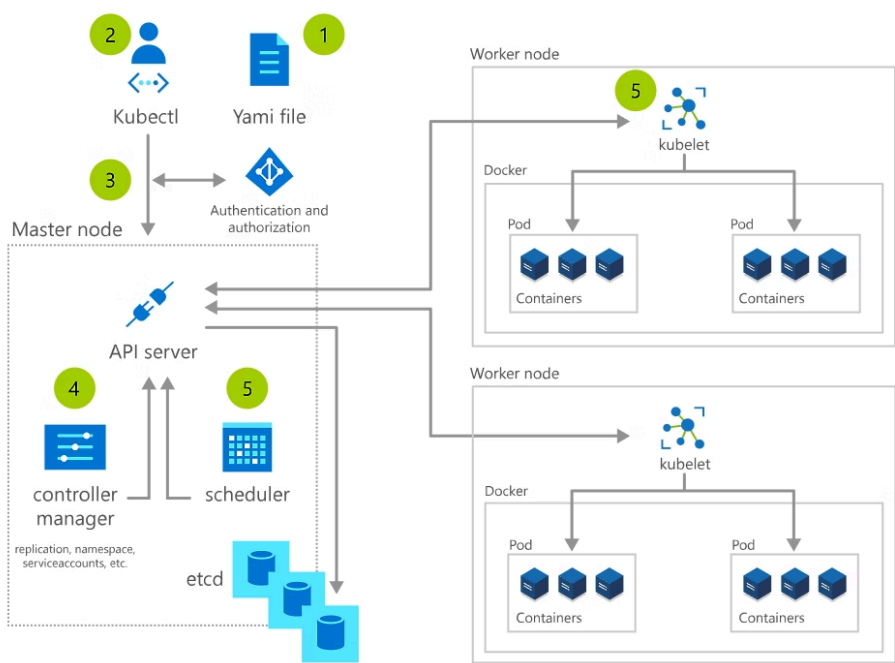


## Rollout

The process of reconfiguring the cluster from its current state to the desired state—achieved in most cases without downtime.

## How Kubernetes rollouts work

- Create a YAML file describing the desired state configuration of the cluster.
- Apply the YAML file to the cluster through kubectl, the Kubernetes command-line interface.
- Kubectl submits the request to the kube-apiserver, which authenticates and authorizes the request before recording the change into a database, etcd.
- The kube-controller-manager continuously monitors the system for new requests and works towards reconciling the system state to the desired state—creating ReplicaSets, deployments and pods in the process.
- After all controllers have run, the kube-scheduler sees that there are pods in the "pending" state because they haven't been scheduled to run on a node, yet. The scheduler finds suitable nodes for the pods then communicates with the kubelet in each node to take control and start the deployment.



# Kubernetes deployment use cases

Roll out containerized apps and workflows through Kubernetes deployments in four ways. The Kubernetes deployment strategy you use may employ one or more of these uses cases.



## Create

Roll out entirely new Kubernetes pods and ReplicaSets.



## Update

Declare a new desired state and roll out new pods and ReplicaSets in a controlled fashion.



## Rollback

Revert the Kubernetes deployment to a previous state—helpful if the current state is not stable.

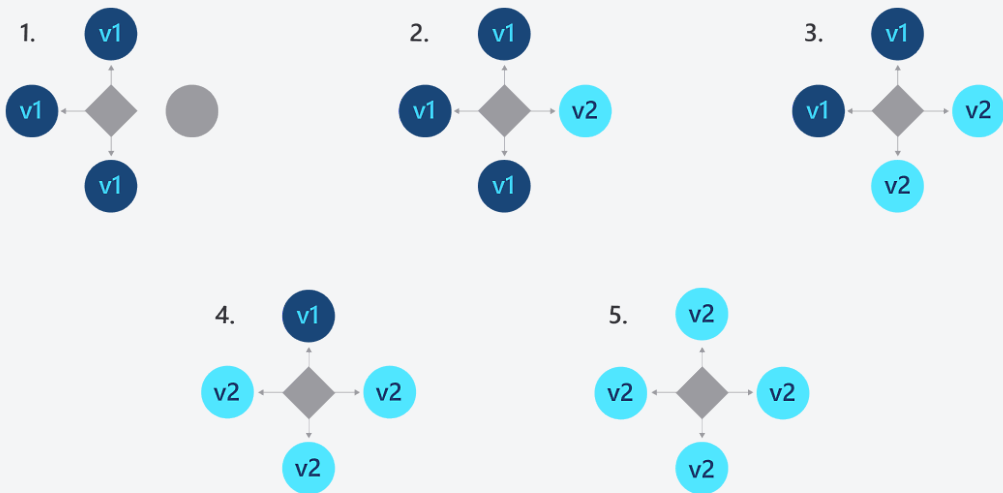


## Scale

Increase the number of pods and ReplicaSets in the Kubernetes deployment without changing them.

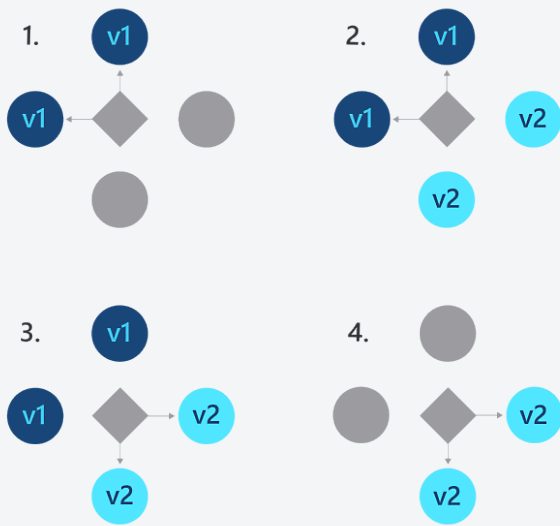
# Kubernetes deployment strategies

The best Kubernetes deployment strategy for your scenario depends on many factors: how much downtime you can spare (if any), your deployment environment, how confident you are in the stability of a new version or platform, whether or not you need to test as well as what you hope to learn, resource cost and availability, and business goals. Here are four common deployment strategies that organizations use in production.



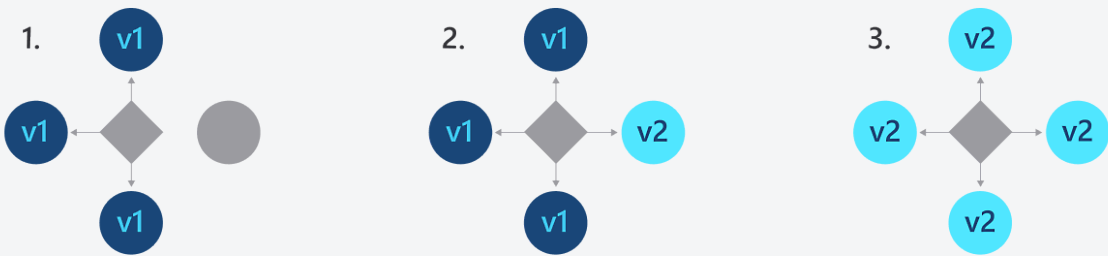
## Ramped

Kubernetes’s default rollout method is a ramped or rolling deployment. This deployment slowly replaces pods one at a time to avoid downtime. Old pods are scaled down only after new pods are ready. If your deployment encounters problems, you can pause or cancel the Kubernetes deployment without taking the entire cluster offline.

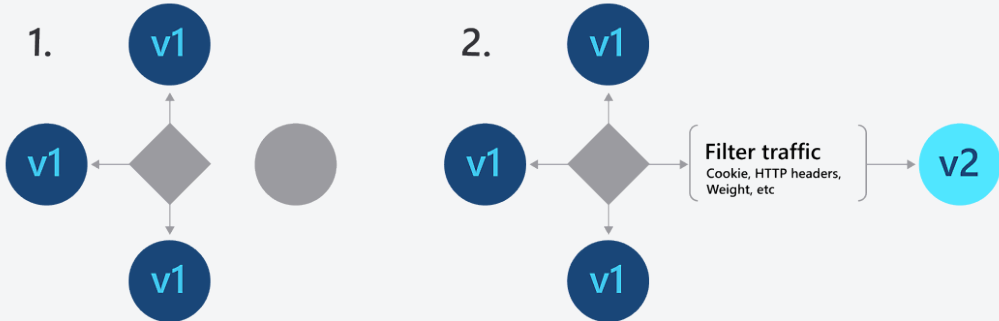


### Blue/green

In a blue/green deployment, you release a new version (blue) of your application or workflow while your current version (green) is still running. This allows you to test the blue version in production while only exposing users to the green, stable version. Once tested, the blue version gradually replaces the green version.



**Canary**  
 Allow your customers to test your Kubernetes deployment by releasing the new version to a small group of them. You'll run one ReplicaSet of the new version along with the current version and then, after a specified period of time without errors, scale up the new version as you remove the old version.



## A/B testing

Much like the canary Kubernetes deployment strategy, an A/B testing strategy targets a specific group of customers. However, an A/B testing deployment seeks to establish more than the stability of a version—it's used to test how effective the version is at achieving business goals. The new version is distributed to users based on factors like cookies, geolocation, operating system, and device type, and it's frequently run alongside the current version—scaling up as the new version proves its worth.

## Kubernetes deployment tools

While you can use the command line or YAML files to execute any of those Kubernetes deployment strategies, gain greater leverage over your clusters and deployments by implementing Kubernetes deployment tools and services like these.



## Azure DevOps

A complete application supply chain to automate Kubernetes deployments. Balance speed and security as you deliver code faster at scale.

[Learn more](#) >



## Helm

An open-source packaging tool. Install, upgrade, and manage Kubernetes applications using charts you create, version, share, and publish.

[Learn more](#) >



## Azure Kubernetes Service (AKS)

A highly available, secure, and fully managed Kubernetes service. Deploy and manage containerized apps in the cloud.

[Learn more](#) >

## Kubernetes quickstarts and technical guidance

[Learn about Kubernetes vs. Docker](#)

## Open-source community

[Join Kubernetes users on GitHub](#)


[Attend KubeCon](#)





[Attend a Kubernetes meetup group](#)



Ready when you are—try Kubernetes for 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

Partners

Azure Marketplace

Find a partner

Join ISV Success

Resources

Resources

Training and certifications

Documentation

Blog

Developer resources



Students

Events and webinars

Analyst reports, white papers, and e-books

Videos

Cloud computing

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