



KodeKloud

# A Guide to becoming a **Kubestronaut** in One Year

Download the  
digital Version of  
this book here





## Disclaimer

The Linux Foundation®, Kubernetes®, Kubestronaut™, Docker®, Kubeflow™, Certified Kubernetes Administrator (CKA)®, Certified Kubernetes Application Developer (CKAD)®, Certified Kubernetes Security Specialist (CKS)®, Kubernetes and Cloud Native Security Associate (KCSA)™, and Kubernetes and Cloud Native Associate (KCNA)™ logos are trademarks or registered trademarks of The Linux Foundation and respective entities in the United States and/or other countries. KodeKloud is neither affiliated with nor endorsed by The Linux Foundation, Docker, Kubeflow, or the Cloud Native Computing Foundation (CNCF). The use of these trademarks and logos is with permission and intended solely for educational purposes and the clarity of readers.

All rights, title, and interest in and to these trademarks remain exclusively with their respective owners. Any references or uses within this publication are purely informational and do not constitute endorsement or approval of the content herein.



# Table of Contents

<b>Why You Should Learn Kubernetes in 2025 and Beyond</b>	3
Build big, Without the Infrastructure Hassle	3
Kubernetes: Fueling AI & ML Innovation	5
<b>What is Kubernetes (K8s)</b>	7
From Monoliths to Microservices	8
A brief overview of the Twelve-Factor App methodology	11
Containers and Docker	15
The Cloud-Native Way	18
Kubernetes: The Orchestrator	21
Kubernetes in the Cloud: Managed Services	22
<b>How to get started with Kubernetes</b>	23
The Simplest Roadmap to start learning Kubernetes	26
<b>Kubernetes Certification Knowledge Graph</b>	29
Kubernetes Certifications	29
How These Certifications Connect – The Knowledge Graph	31
<b>Why Become A Kubestronaut</b>	35
<b>KodeKloud Courses and Learning Paths</b>	38
<b>Boost your team's DevOps expertise with KodeKloud Business</b>	41



# Why You Should Learn Kubernetes in 2025 and Beyond

## Build Big, Scale Without Limits

Think about the biggest breakthroughs in AI—OpenAI, Anthropic, and beyond. Perhaps you want to build the next cutting-edge AI model, a groundbreaking app like Spotify, or a scalable platform like Airbnb. Your main focus should be on crafting brilliant features, perfecting the user experience, and making your product stand out.

However, there's a challenge—these applications all eventually face the same problem: scale. When you're just starting out, infrastructure is usually not your biggest concern. You're focused on building features, acquiring users, and iterating quickly.

But as your app grows, infrastructure issues quickly become the primary challenge. As your user base expands, you'll need more servers, handle more traffic, and manage more moving parts. If your system can't scale reliably, it will break.



KodeKloud

OpenAI

ANTHROPIC

Spotify

airbnb



# kubernetes

That's where Kubernetes comes in. It doesn't eliminate infrastructure challenges—it helps you approach them in a more structured way. It equips you with the tools to manage the complexity of running applications at scale, saving you from reinventing the wheel.

OpenAI January 25, 2021

## Scaling Kubernetes to 7,500 nodes

### Kubernetes for Claude

- Kubernetes helps services communicate smoothly
- Optimizing utilization is easier with pods as an abstraction



Case study: Spotify

Spotify: An Early Adopter of Containers; Spotify Is Migrating from Homegrown Orchestration to Kubernetes

Medium

### Dynamic Kubernetes Cluster Scaling at Airbnb

If you're building something that might one day need to scale to thousands or millions of users, understanding Kubernetes isn't just about simplifying your work—it's about enabling scalability.

## Kubernetes: Fueling AI & ML Innovation

### Cloud Native AI (CNAI)

Cloud Native AI (CNAI) simplifies running, managing, and scaling AI workloads by leveraging modern cloud technologies. It optimizes every part of the AI process—from computing power to storage and networking—for peak performance and cost-efficiency. With CNAI, deploying advanced AI tools becomes more streamlined, scalable, and cost-effective.

At the center of this transformation is Kubernetes, a platform that simplifies container deployment and management. Containers, which package AI models and their dependencies, are lightweight and portable. Kubernetes ensures these containers run efficiently on its clusters, even when different models require conflicting configurations.

**This container-based approach resolves compatibility issues and provides teams with the flexibility to deploy AI models seamlessly, regardless of complexity.**



## Built on Kubernetes for AI

Kubeflow builds on Kubernetes to streamline AI workflows and enhance efficiency. Designed for Machine Learning Operations (MLOps), it leverages Kubernetes' scalability to support tasks such as model training, hyperparameter tuning, and deployment.

Kubeflow offers features like the [Training Operator](#) for distributed training, [Katib](#) for hyperparameter tuning, and [KServe](#) for model deployment, simplifying the entire ML lifecycle. This allows teams to manage even the most complex AI projects with greater ease.

Kubeflow demonstrates that Kubernetes is more than an orchestrator—it's a foundational platform for driving innovation in machine learning.



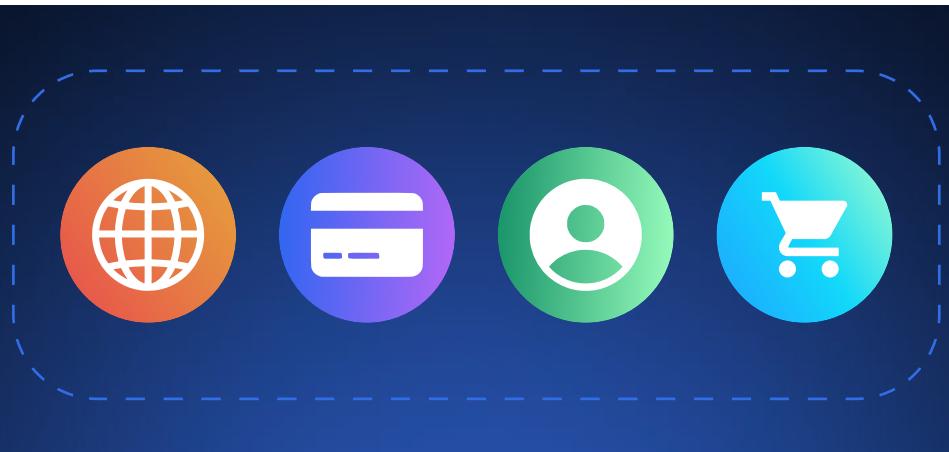
# What is Kubernetes (K8s)

The way we build and deploy applications has drastically evolved over the last few decades, setting the stage for innovations like Cloud Native AI. To understand Kubernetes' role in this journey, let's explore how application development has progressed over time.



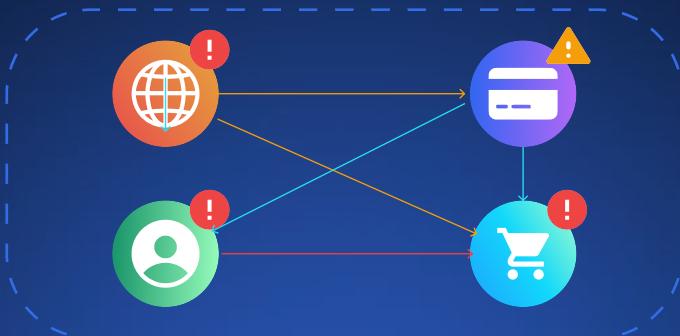
## From Monoliths to Microservices

In the past, applications were developed as single, self-contained units known as monoliths. These monolithic applications tightly coupled all components, including the web interface, payment gateways, and user management.



While this approach was effective initially, it had significant drawbacks.

A failure in any part of the application could bring down the entire system. Additionally, scaling was inefficient and resource-intensive—you had to scale every component together, leading to wasted resources and higher costs.



This led to the rise of microservices, where applications were divided into smaller, independent components, each responsible for a specific function.



Microservices can be scaled or updated independently. If one microservice experiences a failure, it does not cause the entire application to fail.



The microservices approach to building applications necessitated a shift in practices, including adopting methodologies like the [12-factor app](#) methodology, which focuses on:

- Clear separation of codebases and dependencies
- Abstracting configurations from the code
- Agile development processes with scalable and disposable components



## A brief overview of the Twelve-Factor App methodology

I

### CODEBASE

**One codebase,  
multiple deploys**



II

### DEPENDENCIES

**Explicitly declare and  
isolate dependencies**



```
FROM python:3.10-alpine
WORKDIR /kodemkloud-twelve-factor-app
COPY requirements.txt /kodemkloud-twelve-factor-app
RUN pip install -r requirements.txt --no-cache-dir
COPY . /kodemkloud-twelve-factor-app
CMD python app.py
```

III

### CONFIG

**Store configuration in  
environment variables**





IV

## BACKING SERVICES

### Treat services as attached resources

v

## BUILD, RELEASE, RUN

### Separate build and run stages



VI

## PROCESSES

## Execute application as stateless processes

<https://12factor.net/>



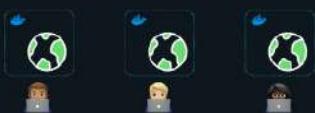
VII

PORT BINDING

**Export services via port binding**



HORIZONTAL SCALING



VIII

CONCURRENCY

**Scale out via process model**

IX

DISPOSABILITY

**Maximize robustness with fast startup/graceful shutdown**





DEV / PROD PARITY

**Keep development  
and production similar**

XI

LOGS

**Treat logs as event  
streams**



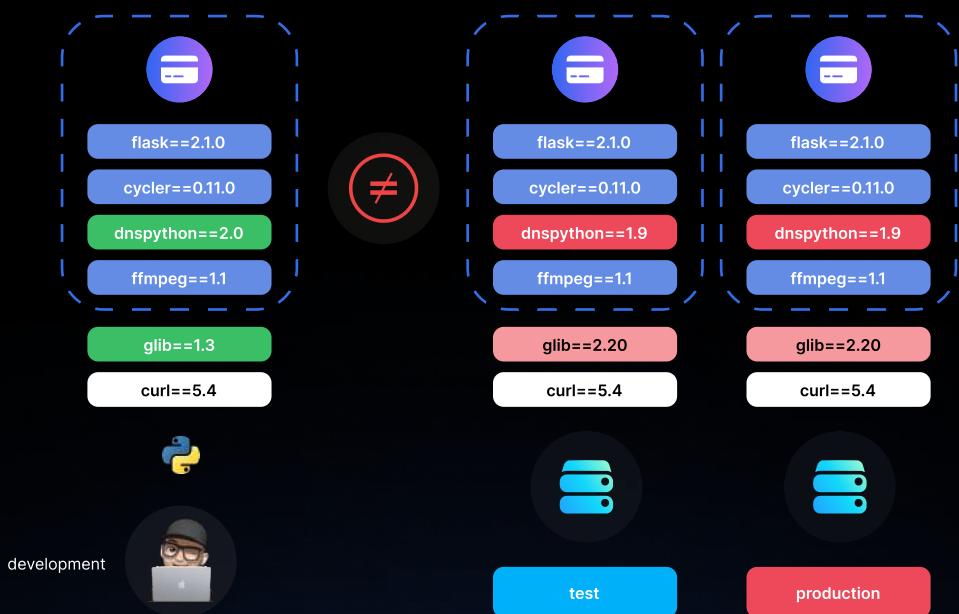
ADMIN PROCESSES

**Run admin/  
management tasks as  
one-off processes**



## Containers and Docker

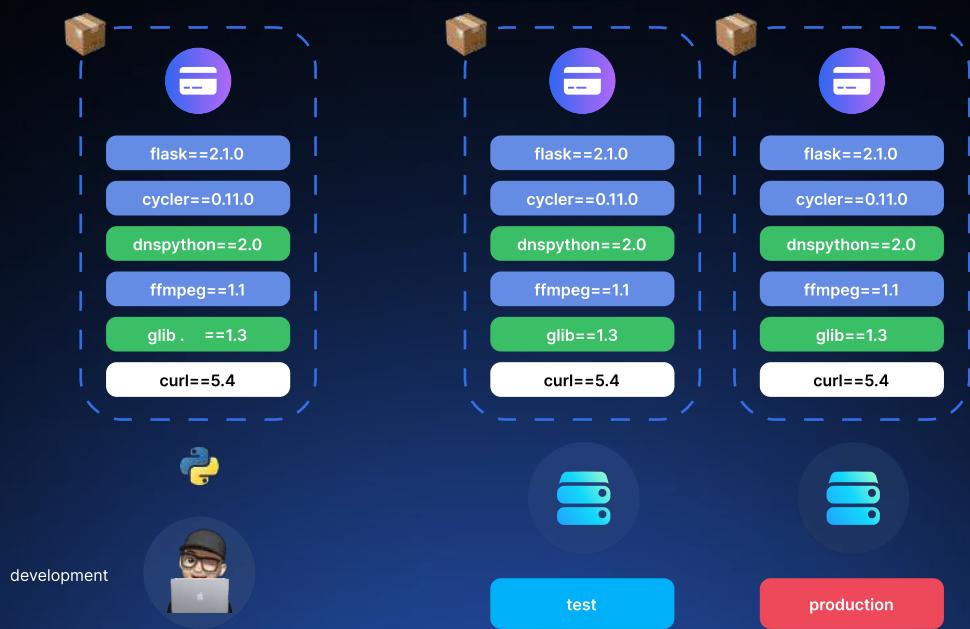
Despite the benefits of microservices, maintaining consistent environments across development, testing, and production remained a significant challenge.



Developers often encountered the common challenge of “It works on my machine.”



Docker solved this problem by introducing the concept of containers. Containers package the application, its dependencies, and the underlying system libraries into a portable unit. This ensures that the application runs consistently across different environments—whether it's a developer's laptop, an on-premises server, or a cloud platform.



With containers, scaling applications became more streamlined and efficient. You can run multiple instances of the same application by issuing a single command.



flask==2.1.0

cycler==0.11.0

dnspython==2.0

ffmpeg==1.1

glib==1.3

curl==5.4

```
# Use the official Python image as the base image
FROM python:3.9-slim

# Set the working directory inside the container
WORKDIR /app

# Install system dependencies for curl and ffmpeg
RUN apt-get update && apt-get install -y \
    \
    curl \
    ffmpeg \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/*

# Install Python dependencies directly
RUN pip install --no-cache-dir \
    flask==2.1.0 \
    cycler==0.11.0 \
    dnspython==2.0

# Copy the application code into the container
COPY . .

# Expose the application port
Expose 5000

# Define the command to run the application
CMD ["python", "app.py"]
```

docker build -t my-flask-app

docker push my-flask-app

docker run my-flask-app

docker run my-flask-app

docker run my-flask-app

docker run my-flask-app





**However, managing these containers at scale—especially across multiple servers—introduced new complexities, such as resource allocation, load balancing, and fault tolerance.**

## The Cloud-Native Way

Applications have evolved to become cloud-native—designed for seamless operation across different environments, including on-premises, public cloud, private cloud, or hybrid setups. By design, cloud-native applications are portable and scalable, using tools like Docker to package and run applications consistently across diverse platforms.

### Portability





## Vertical Scaling

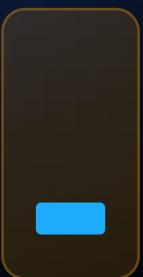
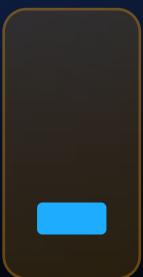
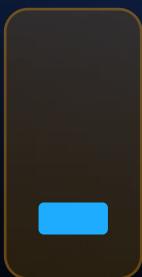


## Horizontal Scaling





# kubernetes



As organizations deployed containers across multiple servers, they needed a way to efficiently manage their lifecycle (including starting, running, stopping, restarting, and scaling). [Kubernetes](#) was introduced to address these challenges and provide a robust container orchestration solution.

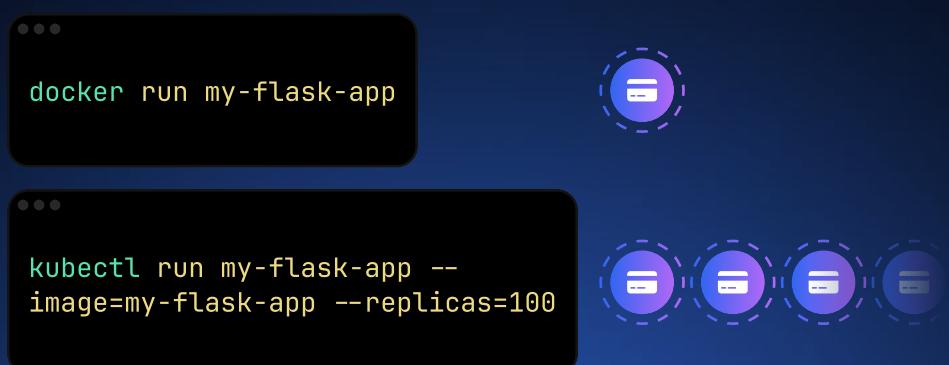


## Kubernetes: The Orchestrator

Kubernetes acts as a container orchestrator, automating container deployment, scaling, and lifecycle management. It serves as a bridge between infrastructure (on-premises or cloud) and containerized applications.

Here's what Kubernetes does:

- **Autoscaling:** Adjusts the number of running containers based on user demand.
- **Self-Healing:** Automatically replaces containers if they crash.
- **Load Balancing:** Distributes traffic evenly across containers.
- **Storage and Networking:** Manages persistent storage and secure networking between containers.





For example, instead of manually using the `docker run` command to start multiple containers, you can use Kubernetes' `kubectl` command to deploy and manage 100 instances of an application using a single command. If one instance fails, Kubernetes automatically starts a replacement to maintain availability.

## Kubernetes in the Cloud: Managed Services

Every major cloud platform today offers its own managed Kubernetes service to simplify the process. With these



AWS offers EKS



Google provides GKE



Azure provides AKS

managed services, you can create a fully provisioned Kubernetes cluster with minimal effort. Once the cluster is ready, you can focus on deploying and managing your applications on Kubernetes, eliminating the need to set up or maintain the underlying cluster infrastructure.



# How To Get Started With Kubernetes

Kubernetes can seem overwhelming at first, but following a structured approach can help streamline the learning process and make it more manageable.

1



Engage

2



Learn

3



Contribute



- **Engage in the Community**

Joining Kubernetes communities is an excellent way to stay informed about the latest trends and best practices. Participate in discussions, ask questions, and contribute where possible.



KubeCon



CloudNativeCon

- **Independent Learning: Blogs, Videos, and Courses**

Independent learning is crucial. Reading blogs, watching tutorials, and enrolling in structured courses can help you build a strong foundation in Kubernetes.

- **Contribute to Open-Source Projects**

Kubernetes is an open-source project, and contributing to it or similar cloud-native initiatives can accelerate your learning. Even small contributions, such as improving documentation, can help you gain a deeper understanding of Kubernetes and its ecosystem.



## TIP

For self-learning, the official Kubernetes documentation is an essential resource. It provides accurate, up-to-date information for understanding Kubernetes and using its features effectively.

**The best way to learn Kubernetes is through [hands-on practice](#). At KodeKloud, we offer structured courses and labs that provide practical experience. Since most Kubernetes certifications require hands-on knowledge, gaining real-world experience is essential for mastering Kubernetes.**

The screenshot shows a dark-themed user interface for a hands-on lab. On the left, a 'Task' window displays a question: 'What is the version of Docker Server Engine running on the Host?' with four options: '23.0.6', '17.12.1-ce', '1.12.1-ce', and '1.13.1-ce'. A 'Hint' button is at the top left, and a 'Next' button is at the bottom right. On the right, a 'Terminal 1' window shows a welcome message: 'Welcome to the KodeKloud Hands-On lab' followed by a stylized purple 'KODEKLOUD' logo. Below the logo, it says 'All rights reserved' and '\$'. The overall theme is professional and modern.



# The Simplest Roadmap to start learning Kubernetes

Before diving into Kubernetes, it's important to build a strong foundation in essential concepts like containers, Linux, and networking. KodeKloud offers a structured learning path for becoming a Kubernetes Administrator, with hands-on labs and practical scenarios. Whether you're new to IT or already familiar with DevOps, following a systematic approach will streamline your Kubernetes learning experience.

The screenshot shows the KodeKloud website with the navigation bar at the top. Below the navigation, the title "Kubernetes Administrator Learning Path" is displayed in a large, bold, blue font. A subtext explains the purpose of the learning path: "The Kubernetes Administrator Learning Path is designed to equip individuals with the skills necessary to become proficient Kubernetes administrators. This learning path offers a comprehensive journey that provides a solid foundation in DevOps practices, containerization, and Kubernetes administration skills." At the bottom of the screenshot, there is a navigation bar with buttons for "I Know", "Nothing", "Container", "Kubernetes", and a link to "Find out here".

**Understand the Fundamentals of DevOps**

**DevOps Prerequisite course**  
6h 30m

**Fundamentals of DevOps**  
2h 30m

KodeKloud Kubernetes Administrator Learning Path:  
<https://kodekloud.com/learning-path/kubernetes-administrator>



1

## Understand IT Prerequisites

If you're new to IT or don't have a coding background, it's helpful to understand the basics of software development and the Software Development Life Cycle (SDLC). You should know how to build and run an application's code.

2

## Learn DevOps Fundamentals

To succeed with Kubernetes, understanding foundational DevOps concepts is crucial. These include:

- **Networking Basics** – Learn about IP addresses, DNS, and firewalls.
- **Security Fundamentals** – Understand authentication, authorization, and encryption.

3

## Learn Linux Basics

Since Kubernetes and Docker containers typically run on Linux (although Windows support is available), knowing how to work with the Linux command line is essential. Understanding file structures, user management, and basic shell scripting will give you an advantage.



4

## **Understand Containers Before Kubernetes**

Since Kubernetes is a container orchestration tool, it's important to understand containers first. Docker is the most popular container tool, and learning how to build, run, and manage containers will make Kubernetes much easier to grasp.

5

## **Learn Kubernetes Fundamentals and Follow Certification Paths**

Once you understand containers, start learning Kubernetes basics. You can then follow Kubernetes certification paths, such as:

- **Certified Kubernetes Administrator (CKA)**
- **Certified Kubernetes Application Developer (CKAD)**
- **Certified Kubernetes Security Specialist (CKS)**

Certifications provide structured learning and clear goals, helping you progress systematically.

# Kubernetes Certification Knowledge Graph

## Kubernetes Certifications

Linux Foundation(LF) offers five official Kubernetes Certifications, each tailored to specific roles and experience levels. Whether you're an administrator, developer, or security specialist, choosing the right certification helps you develop expertise in Kubernetes.

### Certified Kubernetes Administrator (CKA) – Hands-On

For Kubernetes administrators who want to learn how to configure, manage, and troubleshoot Kubernetes clusters.





## Certified Kubernetes Application Developer (CKAD) – Hands-On

For application developers who want to understand Kubernetes fundamentals and deploy applications efficiently.



## Kubernetes and Cloud Native Associate (KCNA) – Multiple-Choice

For beginners who want to learn foundational Kubernetes and cloud-native concepts before advancing further.



## Certified Kubernetes Security Specialist (CKS) – Hands-On

For security professionals focusing on Kubernetes security, with CKA as a prerequisite.



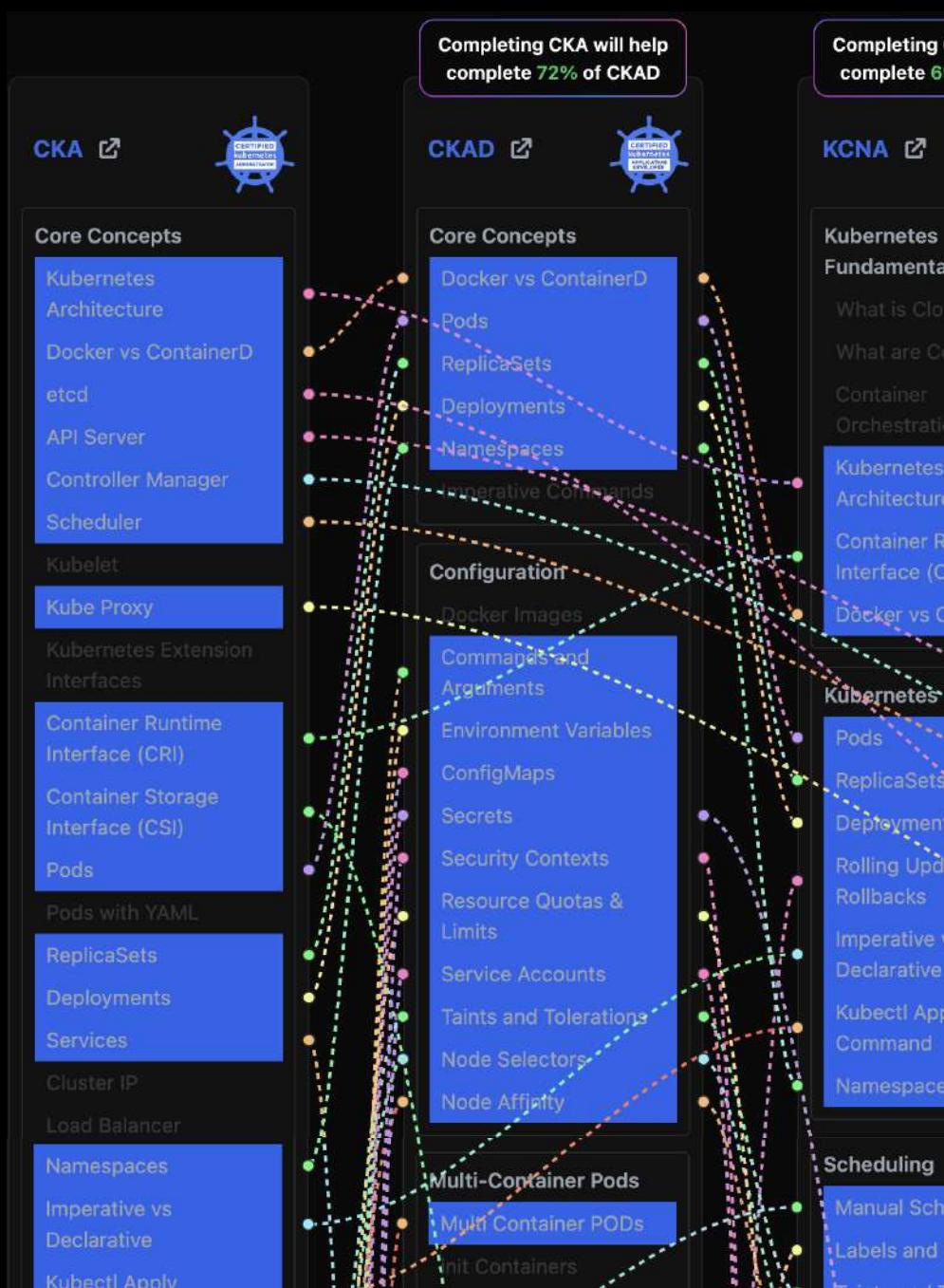
## Kubernetes and Cloud Native Security Associate (KCSA) – Multiple-Choice

For those interested in understanding Kubernetes security concepts without requiring deep hands-on expertise.



**To help you understand how these certifications relate, we've created a clear knowledge graph that illustrates their connections.**

LINK: <https://kubestronaut.kodekloud.com>



CKA will help  
9% of KCNA



Is  
Cloud Native?  
Containers?

on  
e  
untime  
(RI)  
ContainerD

Resources

sts  
s &  
s  
s  
s

eduing  
Selectors

Completing CKA will help  
complete 27% of CKS

**CKS** ↗

Requires CKA



### Understanding the Kubernetes Attack Surface

The 4Cs of Cloud  
Native Security

### Cluster Setup and Hardening

CIS benchmarks

Kube-bench

#### Kubernetes Security

Primitives

Authentication

Service Accounts

TLS in Kubernetes

PKI Certificates & API

KubeConfig

API Groups

Authorization

Role Based Access

Controls (RBAC)

Cluster Roles

Attribute Based Access

Control (ABAC)

Kubelet Security

Kubectl Proxy & Port

Forward

Kubernetes Dashboard

Completing CKA will help  
complete 40% of KCSA

**KCSA** ↗



### Overview of Cloud Native Security

The 4Cs of Cloud  
Native Security

Cloud Provider Security

Infrastructure Security

Kubernetes Isolation  
Techniques

Artifact Repository  
Security

### Image Security

Workload and  
Application Code  
Security

### Kubernetes Cluster Component Security

API Server

Controller Manager

Scheduler

Kubelet Security

Container Runtime  
Interface (CRI)

Kube Proxy

etcd

Container Networking

Kubectl Proxy & Port  
Forward



There are overlapping topics across the different certifications: KCNA covers foundational concepts, with CKA and CKAD focusing on hands-on skills and CKS and KCSA focusing on security.

While you can take the certifications in any order based on your experience, here are two suggested learning paths based on different starting points:

## PATH 1

### **Administrator-Focused** **CKA → CKAD → KCNA → CKS** → **KCSA**

- Ideal for those who prefer to start with cluster administration before exploring development and security.
- This path focuses on building strong Kubernetes operations skills before specializing in security.





## PATH 2

## Beginner-Friendly Path

KCNA → CKAD → CKA → CKS → KCSA

This path is ideal for beginners who start with KCNA before advancing to application deployment and cluster administration, ensuring a smooth learning curve by progressing from foundational theory to practical experience.



Both paths lead to a well-rounded Kubernetes skill set, and you can choose the order that best suits your comfort and experience level. The key is to build practical knowledge alongside certifications to maximize your learning experience.

# Why Become A Kubestronaut



A Kubestronaut is someone who has earned all five Kubernetes certifications. Achieving this milestone is not just about earning certificates—it's about gaining comprehensive mastery of Kubernetes. It requires a deep understanding of Kubernetes concepts, extensive hands-on experience, and the ability to configure, deploy, and secure Kubernetes clusters at an expert level.



Becoming a Kubestronaut means you've gained expertise in:

- ✓ **Kubernetes Fundamentals** – Understanding Kubernetes architecture and key concepts.
- ✓ **Cluster Administration** – Setting up and managing Kubernetes clusters.
- ✓ **Application Deployment** – Deploying and managing applications in Kubernetes.
- ✓ **Security & Compliance** – Building secure and compliant Kubernetes environments.

The journey extends beyond certification. Kubestronauts are recognized by the Cloud Native Computing Foundation (CNCF) and receive exclusive benefits to further their professional development:

**Get 30% off Sitewide on  
certifications and bundles.**

Use Code: **30KODE**

**30 %  
OFF**





**Kubestronaut Jacket** – A special jacket to recognize your achievement.



**Credly Badge** – A digital badge for your professional profile.



**Private Community Access** – Join an exclusive **Kubestronaut Slack channel and mailing list** to connect with other experts.



**Certification Discounts** – Get **50% off** five Kubernetes certifications per year (to use personally or share).



**Event Discounts** – Enjoy **20% off** three CNCF events annually, including **KubeCon + CloudNativeCon** and **KubeDays**.

Kubernetes is a critical component of modern cloud applications, and learning it can unlock new career opportunities. Whether you're a DevOps engineer, cloud architect, or AI/ML enthusiast, mastering Kubernetes in 2025 will provide a significant advantage in the tech industry.



# KodeKloud Courses and Learning Paths

## **Kubernetes Learning Path**

[https://kodekloud.com/  
learning-path/kubernetes/](https://kodekloud.com/learning-path/kubernetes/)



## **DevOps Learning Path**

[https://kodekloud.com/  
learning-path/devops/](https://kodekloud.com/learning-path/devops/)



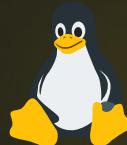
## **Cloud Learning Path**

[https://kodekloud.com/  
learning-path/cloud/](https://kodekloud.com/learning-path/cloud/)



## **Linux Learning Path**

[https://kodekloud.com/  
learning-path/linux/](https://kodekloud.com/learning-path/linux/)





## **CI/CD Learning Path**

[https://kodekloud.com/  
learning-path/ci-cd/](https://kodekloud.com/learning-path/ci-cd/)



## **AI Learning Path**

[https://kodekloud.com/  
learning-path/ai](https://kodekloud.com/learning-path/ai)



## **IaC Learning Path**

[https://kodekloud.com/  
learning-path/iac/](https://kodekloud.com/learning-path/iac/)



## **HashiCorp Learning Path**

[https://kodekloud.com/  
learning-path/hashicorp/](https://kodekloud.com/learning-path/hashicorp/)



## **Platform Engineer Learning Path**

[https://kodekloud.com/  
learning-path/platform-engineer/](https://kodekloud.com/learning-path/platform-engineer/)





## DevOps Engineer Learning Path

<https://kodekloud.com/learning-path/devops-engineer>



## Cloud Engineer Learning Path

<https://kodekloud.com/learning-path/cloud-engineer/>



## System Administrator Learning Path

<https://kodekloud.com/learning-path/system-administrator/>



## Kubernetes Administrator Learning Path

<https://kodekloud.com/learning-path/kubernetes-administrator/>



## Site Reliability Engineer Learning Path

<https://kodekloud.com/learning-path/site-reliability-engineer/>



## Kubernetes Developer Learning Path

<https://kodekloud.com/learning-path/kubernetes-developer/>





# Boost your team's DevOps expertise with **KodeKloud Business**

— providing hands-on labs, structured courses, and practical Kubernetes training to empower your team with real-world skills.

<https://kodekloud.com/business>

## 3 Weeks

Time needed for the team at VMWare to achieve **Kubernetes Certifications** with KodeKloud.

Source: Success Stories

## 50%

**Reduction in downtime** for softwares & Apps, when team are sufficiently trained.

Source: State of DevOps

## 2x

Team capabilities in deploying containers when teams are sufficiently trained.

Source: CNCF

For any inquiries, contact  
[sales@kodekloud.com](mailto:sales@kodekloud.com)



KodeKloud



KodeKloud helped myself and my solutions architecture staff upskill across key digital technologies such as Kubernetes, Git, Terraform and Ansible. We've seen exponential uptake in these areas with our staff thanks to the excellent quality of the materials.



**Anthony Andrades**

Director of Solutions Architecture



Kubernetes Certified in 3 Weeks – Our K8s talent pool is growing exponentially. My advice: take KodeKloud courses.



**Jacky Lee**

Director of CloudNative



(4.8) based on 900 reviews



4.8 out of 5 stars

