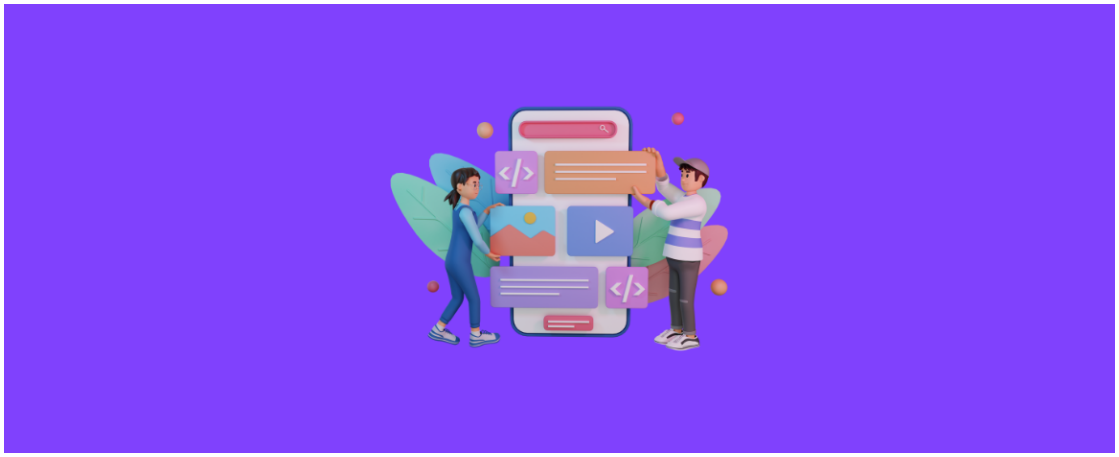
How to Learn Kubernetes (Complete Roadmap & Resources)



Learning Kubernetes can seem overwhelming. It's a complex container orchestration system, that has a steep learning curve. But with the right roadmap and understanding of the foundational concepts, it's something that any developer or ops person can learn.

In this Kubernetes learning roadmap, I have added prerequisites and complete **Kubernetes learning path** covering basic to advanced Kubernetes concepts.

Prerequisites To Learn Kubernetes

Before jumping into learning kubernetes, you need to have a fair amount of knowledge of some of the underlying technologies and concepts.

1. **Distributed system:** Learn about distributed system basics & their use cases in modern IT infrastructure. CAP theorem is good to have knowledge.
2. **Authentication & Authorization**: A very basic concept in IT. However, engineers starting their careers tend to get confused. So please get a good understanding of learning from analogies. You will quite often see these terms in Kubernetes.
3. **Key Value Store**: It is a type of NoSQL Database. Understand just enough basics and their use cases.
4. **API:** Kubernetes is an API-driven system. So you need to have an understanding of RESTFUL APIs. Also, try to understand gRPC API. It's good to have knowledge.
5. **YAML**: YAML stands for *YAML Ain't Markup Language*. It is a data serialization language that can be used for data storage and configuration files. It's very easy to learn and from a Kubernetes standpoint, we will use it for configuration files. So understanding YAML syntax is very important.
6. **Container**: Container is the basic building block of kubernetes.The primary work of Kubernetes is to orchestrate containers. You need to learn all the container basics and have hands-on experience working on container tools like Docker or Podman. I would also suggest reading about Open container initiative and Container Runtime Interface (CRI)
7. **Service Discovery**: It is one of the key areas of Kubernetes. You need to have basic knowledge of client-side and server-side service discovery. To put it simply, in client-side service discovery, the **request goes to a service registry** to get the endpoints available for

backend services. In server-side service discovery, the r**equest goes to a load balancer** and the load balancer uses the service registry to get the ending of backend services.

8. **Networking Basis**

    1. CIDR Notation & Type of IP Addresses
    2. L3, L4 & L7 Layers (OSI Layers)
    3. SSL/TLS: One way & Mutual TLS
    4. Proxy
    5. DNS
    6. IPTables
    7. IPVS
    8. Software Defined Networking (SDN)
    9. Virtual Interfaces
    10. overlay networking

Learn Kubernetes Architecture

Understanding Kubernetes architecture is not an easy task. The system has many moving parts that need to be understood in order for you to get a grip on what's happening beneath the surface. While learning architecture, you will come across the concepts we discuss in the prerequisites.

As Kubernetes is a complex system, trying to understand the core architecture could be a little overwhelming for DevOps Engineers. As you get more hands-on experience, you would be able to understand the core architecture better.

Here is my suggestion. Learn the high-level architecture and key components involved in Kubernetes. If you are not able to grasp the concept, either you can spend time and do more research on a specific topic or you can learn the concept while doing hands-on. It's your choice.

Check out the Kubernetes Architecture guide to learn about all the Kubernetes components in detail.

Overall you need to learn the following.

1. **Control plane components:** Understand the role of each component like API server, etcd, Scheduler, and Controller manager.
2. **Worker node components**: Learn about Kube Proxy, Kubelet, Container Runtime
3. **Addon Components:** CoreDNS, Network plugins (Calico, weave, etc), Metric Server
4. **Cluster high availability:** Most organizations use managed Kubernetes services (GKE, EKS, AKS, etc). So the cloud provider takes care of the cluster's control plane's high availability. However, it is very important to learn the high availability concepts in scaling the cluster in multi zones and regions. It will help you in real-time projects and devops interviews.
5. **Network Design**: While it is easy to set up a cluster in an open network without restrictions, it is not that easy in a corporate network. As a DevOps engineer, you should understand the Kubernetes network design and requirements so that you can **collaborate with the network team** better. For example, When I was working with kubernetes setup on Google cloud, we used a CIDR pod range that was not routable in the corporate network. As a workaround, we had to deploy IP masquerading for the pod network.

$1000+ Free Cloud Credits to Deploy Clusters

Deploying big clusters on the cloud could be expensive. So make use of the following cloud credits and learn to launch clusters as if you would on a real-time project. All platforms offer managed k8s services.

1. GKE (Google Cloud – $300 free credits)
2. EKS (AWS – $300 free POC credits)
3. DO Kubernetes (Digital Ocean – $200 free credits)
4. Linode Kubernetes Engine (Linode Cloud – $100 Free credits)
5. Vultr Kubernetes Engine (Vultr Cloud – $250 Free Credits)

Use one account at a time. Once the credits are expired. move to the next account. You need to keep a watch on your credits as well as expiry. Or else you could get charged. Also, check the terms and instance usage limits if any.

Also, setting up servers on this platform is very easy and every cloud provider had extensive documentation to get started.

Kubernetes Cluster Setup

As DevOps engineers, it is very important to learn every component and cluster configuration. While there are many options to deploy a Kubernetes cluster, It is always better to learn to deploy multi-node clusters from scratch.

With multi-node clusters, you can learn about all the concepts like Cluster security, High Availability, Scaling, Networking, etc.

It gives you the feeling of working on a real-world project. It will also help you in interviews and you can be confident about production-level cluster configurations.

Following are my cluster setup suggestions.

1. **Kubernetes the Hard Way:** I would suggest you start with Kubernetes the hard way set up. It helps you understand all the configurations involved in bootstrapping a kubernetes cluster. If you want to work on production clusters, this lab will help you a lot. The setup is based on google cloud. You can use the $300 free credits to complete the lab.
2. **Kubeadm Cluster Setup**: Learning kubeadm cluster setup helps you in Kubernetes certification preparation. Also, it helps you automate Kubernetes cluster setup with best practices.
3. **Minikube:** If you want to have a minimal development cluster setup, minikube is the best option.
4. **Kind:** Kind is another local development Kubernetes cluster setup.
5. **Vagrant Automated Kubernetes:** If you prefer to have a multi-VM-based local Kubernetes cluster setup, you can try the automated vagrant setup that uses Kubeadm to bootstrap the cluster.

Learn About Cluster Configurations

Once you have a working cluster, you should learn about the key cluster configurations. This knowledge will be particularly helpful when working in a self-hosted Kubernetes setup.

Even if you use a managed Kubernetes cluster for your project, there may be certain cluster configurations that you need to modify.

For example, if you set up a cluster in a hybrid network, you may need to configure it with an on-premises private DNS server for private DNS resolution. This can be done via CoreDNS configuration.

Refer to the Kubernetes Cluster configurations guide for more details.

Also, having a solid understanding of cluster configurations will help you with Kubernetes certifications (CKA & CKS) where you need to troubleshoot cluster misconfiguration and issues.

Understand Kubeconfig File

`Kubeconfig` file is a YAML file that contains all the cluster information and credentials to connect to the cluster.

As a Devops Engineer, You should learn to connect to kubernetes clusters in different ways using the Kubeconfig file. Because you will be responsible for setting up cluster authentication for CI/CD systems, providing cluster access to developers, etc.

So spend some time, understanding the Kubeconfig file structure and associated parameters.

Check out the complete Kubeconfig file guide to learn everything about the Kubeconfig file.

Understand Kubernetes Objects And Resources

You will quite often come across the names "**Kubernetes Object**" and "**Kubernetes Resource**"

First, you need to Understand the difference between an object and a resource in kubernetes.

To put it simply, anything a **user creates and persists in Kubernetes is an object**. For example, a namespace, pod, Deployment configmap, Secret, etc.

Before creating an object, you represent it in a YAML or JSON format. It is called an **Object Specification (Spec)**. You declare the desired state of the object on the Object Spec. Once the object is created, you can retrieve its details from the Kubernetes API using Kubectl or client libraries.

As we discussed earlier in the prerequisite section, everything in Kubernetes is an API. To create different object types, there are **API endpoints provided by the Kubernetes API server**. Those **object-specific api-endpoints are called resources**. For example, an endpoint to create a pod is called a **pod resource**.

So when you try to create a Kubernetes Object using Kubectl, it converts the YAML spec to JSON format and sends it to the Pod resource (Pod API endpoint).

You can refer to the Kubernetes objects vs resource guide for more details.

Learn About Pod & Associated Resources

Once you have an understanding of Kubernetes Objects and resources, you can start with a native Kubernetes object called Pod. A pod is a basic building block of Kubernetes.

You should learn all the **Pod concepts** and their associated objects like **Service, Ingress, Persistent Volume, Configmap, and Secret**. Once you know everything about a pod, it is very easy to learn other pod-dependent objects like deployments, Daemonset, etc.

First, learn about the Pod Resource Definition (YAML). A typical Pod YAML contains the following high-level constructs.

1. Kind
2. Metadata
3. Annotations
4. Labels
5. Selectors

Refer to Kubernetes Pod Explained blog to learn all the basics about Pod.

Once you have a basic understanding of the above, move on to hands-on learning. These concepts will make more sense when you do hands-on.

Following are the **hands-on tasks** to learn about Pod and its associated objects.

1. Deploy a pod
2. Deploy pod on the specific worker node
3. Add service to pod
4. Expose the pod Service using Nodeport
5. Expose the Pod Service using Ingress
6. Setup Pod resources & limits
7. Setup Pod with startup, liveness, and readiness probes.
8. Add Persistent Volume to the pod.
9. Attach configmap to pod
10. Add Secret to pod
11. multi-container pods (sidecar container pattern)
12. Init containers
13. Ephemeral containers
14. Static Pods
15. Learn to troubleshoot Pods

Few advanced pod scheduling concepts.

1. Pod Preemption & Priority
2. Pod Disruption Budget
3. Pod Placement Using a Node Selector
4. Pod Affinity and Anti-affinity
5. Container Life Cycle Hooks

Learn Pod Dependent Objects

Now that you have a better understanding of Pod and independent kubernetes resources, you can start learning about objects that are dependent on the Pod object. While learning this, you will come across concepts like HPA (Horizontal Pod Autoscaling) and VPA (Verification Pod Autoscaling)

1. Replicaset
2. Deployment
3. Daemonsets
4. Statefulset
5. Jobs & Cronjobs

Learn Ingress & Ingress Controllers

To expose applications to the outside world or end users, kubernetes has a native object called ingress.

Many engineers get confused with Ingress due to less knowledge of Ingress controllers. Ensure you go through the concept of Ingress and Ingress controllers and understand it correctly. Because it is the base of exposing applications to the outside world.

You can start with the following comprehensive guides.

Also, learn about the Kubernetes Gateway API. it provides advanced features over Ingress.

Learn End to End Microservices Application Deployment on Kubernetes

Once you understand the basics of these objects, you can try deploying an end-to-end microservices application on Kubernetes. Start with simple use cases and gradually increase complexity.

I would suggest you get a domain name and try setting up a microservice application from scratch and host it on your domain.

You don't need to develop an application for this. Choose any open-source microservice-based application and deploy it. My suggestion is to choose the open-source pet clinic microservice application based on spring boot.

Following are the high-level tasks.

1. Build Docker images for all the services. Ensure you optimize the Dockerfile to reduce the Docker Image size.
2. Create manifests for all the services. (Deployment, Statefulset, Services, Configmaps, Secrets, etc)
3. Expose the front end with service type ClusterIp
4. Deploy Nginx Ingress controller and expose it with service type Loadbalancer
5. Map the load balancer IP to the domain name.
6. Create an ingress object with a DNS name with the backend as a front-end service name.
7. Validate the application.

Learn About Securing Kubernetes Cluster

Security is a key aspect of Kubernetes. There are many ways to implement security best practices in Kubernetes starting from building a secure container image.

Following the native ways of implementing security in kubernetes.

1. Service account
2. Pod Security Context
3. Seccomp & AppArmor
4. Role Based Access Control (RBAC)
5. Attribute-based access control (ABAC)
6. Network Policies

The following are the open-source tools you need to look at.

1. Open Policy Agent
2. Kyverno
3. Kube-bench
4. Kube-hunter
5. Falco

Learn About Kubernetes Configuration Management Tools

Now that you have a good understanding of all Kubernetes objects and deploying applications on Kubernetes, you can start learning about Kubernetes configuration management tools.

When you start working on a real-time project in an organization, you will see the usage of configuration management tools to deploy applications on Kubernetes.

Because in organizations, there are different environments like dev, stage, pre-prod, and production. You cannot create individual YAML files for each environment and manage them manually. So you need a system to manage Kubernetes YAML configurations effectively.

Following are the popular and widely adopted Kubernetes tools to manage YAML.

1. Helm (Templating Engine)
2. Kuztomize (Overlay Engine)

Learn About Kubernetes Operator Pattern

Kubernetes Operators is an advanced concept.

To understand operators, first, you need to learn the following Kubernetes concepts.

1. Custom resource definitions
2. Admission controllers
3. Validating & Mutating Webhooks

To get started with operators, you can try setting the following operators on Kubernetes.

1. Prometheus Operator
2. MySQL Operator

If you are a Go developer or you want to learn to extend/customize kubernetes, I would suggest you **create your own operator** using Golang.

Learn Important Kubernetes Configurations

While learning kubernetes, you might use a cluster in open network connectivity. So most of the tasks get executed without any issues. However, it is not the case with clusters set up on corporate networks.

So following are the some of the custom cluster configurations you should be aware of.

1. Custom DNS server
2. Custom image registry
3. Shipping logs to external logging systems
4. Kubernetes OpenID Connect
5. Segregating & securing Nodes for PCI & PII Workloads

Learn Kubernetes Best Practices

Following are the resources that might help and add value to the Kubernetes learning process in terms of best practices.

1. **12 Factor Apps**: It is a methodology that talks about how to code, deploy and maintain modern microservices-based applications. Since Kubernetes is a cloud-native microservices platform, it is a must-know concept for DevOps engineers. So when you work on a real-time kubernetes project, you can implement these 12-factor principles.
2. **Kubernetes Failure Stories**: Kubernetes failure stories is a website that has a list of articles that talk about failures in Kubernetes implementation. If you read those stories, you can avoid those mistakes in your kubernetes implementation.
3. **Case Studies From Organizations:** Spend time on use cases published by organizations on Kubernetes usage and scaling. You can learn a lot from them. Following are some of the case studies that are worth reading.

    1. Scheduling 300,000 Kubernetes Pods in Production Daily
    2. Scaling Kubernetes to 7,500 Nodes

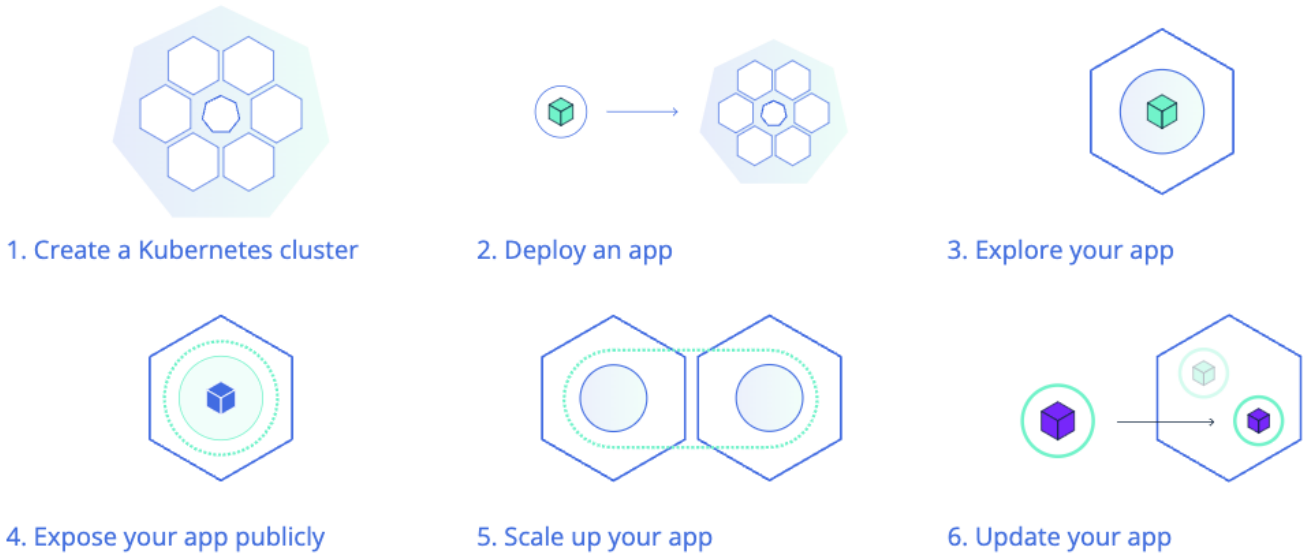
The Best Resources to Learn Kubernetes Online

Following are the list of the best online resource to learn Kubernetes practically.

1. The official Kubernetes Basics Tutorial

The official Kubernetes website has browser-based hands-on kubernetes basic tutorials powered by Katacoda scenarios. It covers the following.

1. Kubernetes basics
2. Kubernetes configurations
3. Stateless Application deployment
4. Stateful application deployment
5. Kubernetes services
6. Kubernetes Security



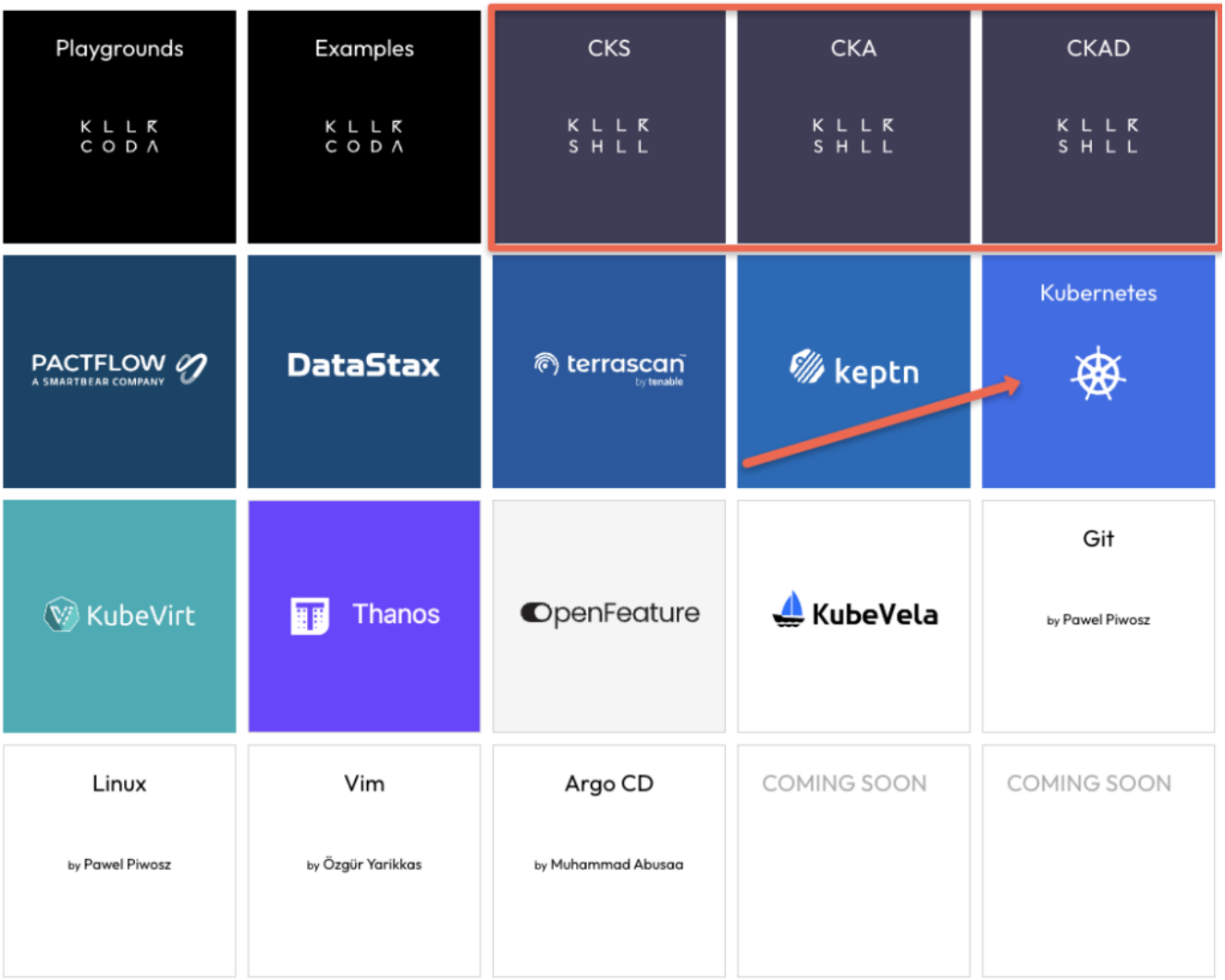
You can also look at official Kubernetes tasks to learn to implement Kubernetes concepts practically. It will also help you prepare for Kubernetes certifications.

2. DevOpsCube Kubernetes Tutorials

DevOpsCube has 35+ comprehensive Kubernetes hands-on tutorials for beginners to advanced users. You will learn everything from Kubernetes architecture, Cluster setup, Deployments, best practices, package management, secret management, monitoring, logging, etc.

3. KillerCoda Interactive Tutorials

If you want to learn Kubernetes from the comfort of your browser, Killercoda is a great option. It offers scenario-based learning playgrounds on the browser.



Kubernetes Learning GitHub Repository

The full Kubernetes learning guide is added to the Kubernetes learning path GitHub repo with links to all the free useful resources.

This repo is maintained and contributed by community members and it has the following.

1. Structured Kubernetes learning path
2. Resources to deploy Kubernetes cluster for free.
3. Kubernetes free learning resources
4. Kubernetes hands-on tutorials.
5. Kubernetes production deployment case studies

What is the Best Way to Learn Kubernetes?

Let’s have a look at some of the best ways to learn Kubernetes.

You could fall under any of the following categories.

1. **Self Learning:** If you are a self-learner, you could start with the roadmap and for each topic do further research and learn from blogs, official documentation, free youtube tutorials, and Kubecon Videos.
2. **Guided Learning (Text):** If you like reading guided materials, the best place to learn kubernetes is to start with a beginner's kubernetes book. My suggestion would be "Kubernetes Up and Running" followed by Kubernetes in Action. You could also try educative Kubernetes Course.
3. **Guided Learning (Videos):** If you want to learn Kubernetes by watching guided Video courses, you could start with Udemy, Pluralsight, or KodeKloud Kubernetes courses.
4. **Signup For Certification:** Another way to learn Kubernetes is to prepare for Kubernetes certification. This way, you not only learn Kubernetes but also have a certification to boost your career. If you choose this path, it's important not to use exam dumps. Instead, learn all the concepts properly and take the exam. You could save money on certification using the Kubernetes Certification coupon.

Real-World Kubernetes Case Studies

When I spoke to the DevOps community, I found that a common issue was the lack of real-world experience with Kubernetes. If you don't have an active Kubernetes project in your organization, you can refer to case studies and learning materials published by organizations that use Kubernetes. This will also help you in Kubernetes interviews.

Here are some good real-world Kubernetes case studies that can enhance your Kubernetes knowledge:

What's New in the Latest Kubernetes Release

While learning Kubernetes, it is better to keep track of the latest Kubernetes releases and new features. I will keep updating the latest release information and feature list.

Latest Kubernetes Release	**Details**
Kubernetes v1.27	Named as Chill Vibes
Kubernetes v1.26	Named as Electrifying
Key Features	1. Provision volumes from cross-namespace snapshots 2. You can configure Service Level Indicator (SLI) metrics for kubernetes binaries 3. Support of Mixed Protocols in Services with Type LoadBalancer

Conclusion

In this **learning path to learn Kubernetes**, I have covered all the important concepts you need to learn to master Kubernetes. I will keep adding new features when the new Kubernetes versions get released.

Also, learning a new technical skill takes hours of practice. You'll certainly gain a good understanding of Kubernetes in the learning process, but the truth is that you never stop learning.

I started my Kubernetes journey in 2014 and I learn new concepts and functionalities all the time.