Perfect 🔥 — here’s your complete Kubernetes learning roadmap, designed especially for DevOps engineers who want to go from beginner ➜ expert ➜ production-ready.

This roadmap will give you concepts + tools + practice projects at every stage.  
Let’s start 👇

**Kubernetes Learning Roadmap (DevOps Focus)**

🧩 **Phase 1: Basics & Foundations**

🎯 **Goal:** Understand containers and basic Kubernetes architecture.

🔹 **Learn Prerequisites:---**

--------------------------------------

Before Kubernetes, you must master:

Linux basics

Commands, file system, networking

Docker

Images, containers, volumes, networks

Dockerfile, docker-compose, docker run, docker ps

YAML syntax – for writing Kubernetes manifests

**📘 Practice:**

docker build -t myapp:v1 .

docker run -d -p 8080:80 myapp:v1

**☸️ Phase 2: Kubernetes Core Concepts**

🎯 Goal: Understand how Kubernetes works internally.

**🔹 Core Topics**

Cluster architecture (Master & Worker nodes)

Pods, ReplicaSets, Deployments

Services (ClusterIP, NodePort, LoadBalancer)

Namespaces

ConfigMaps & Secrets

Volumes & PersistentVolumes

Resource Requests & Limits

**📘 Commands to Practice:**

kubectl get pods

kubectl describe pod podname

kubectl apply -f deployment.yaml

kubectl expose deployment myapp --type=NodePort --port=8080

**📘 Practice Project:**  
Deploy a simple Nginx web app with:

Deployment

Service

ConfigMap (for index.html)

PersistentVolume (for logs)

**⚙️ Phase 3: Controllers & Scheduling**

🎯 Goal: Learn how Kubernetes manages workload automatically.

**🔹 Topics**

ReplicaSets, DaemonSets, StatefulSets

Jobs & CronJobs

Labels, Selectors, Annotations

Node scheduling, affinity, taints & tolerations

Probes (liveness, readiness, startup)

**📘 Practice Project:**  
Deploy a backend + frontend app with readiness & liveness probes.

**🧠 Phase 4: Networking & Security**

🎯 Goal: Secure communication and control access.

**🔹 Topics**

Cluster networking (CNI basics)

Services & DNS (CoreDNS)

Ingress & Ingress Controller (NGINX)

Network Policies

RBAC (Roles, RoleBindings, ServiceAccounts)

Pod Security Admission (PSA)

TLS certificates & Secrets

**📘 Practice Project:**  
Set up a Kubernetes Ingress for multiple services under one domain.

**🧱 Phase 5: Storage & Data Management**

🎯 Goal: Persist and manage app data safely.

**🔹 Topics**

PersistentVolume (PV)

PersistentVolumeClaim (PVC)

StorageClass

CSI drivers (EBS, AzureDisk, NFS)

Dynamic provisioning

**📘 Practice Project:**  
Deploy a MySQL database using StatefulSet + PersistentVolumeClaim.

**⚡ Phase 6: Scaling & Autoscaling**

🎯 Goal: Automatically scale apps and clusters.

**🔹 Topics**

Horizontal Pod Autoscaler (HPA)

Vertical Pod Autoscaler (VPA)

Cluster Autoscaler

Resource Quotas & LimitRanges

**📘 Practice Project:**  
Deploy a Node.js app that auto-scales based on CPU usage.

🧰 **Phase 7: Advanced Operations**

🎯 Goal: Operate Kubernetes efficiently in production.

**🔹 Topics**

Rolling updates & rollbacks

Helm (package manager)

Kustomize

Monitoring (Prometheus + Grafana)

Logging (EFK Stack: Elasticsearch + Fluentd + Kibana)

Metrics Server

Audit Logs

**📘 Practice Project:**  
Create a Helm chart for your application and deploy it with monitoring enabled.

🧩 **Phase 8: Automation & GitOps**

🎯 Goal: Automate deployments with CI/CD.

🔹 **Topics**

GitOps with ArgoCD or Flux

CI/CD pipelines with Jenkins, GitHub Actions, or GitLab CI

Deployment strategies:

Blue-Green

Canary

Rolling

Webhooks & Admission Controllers

**📘 Practice Project:**  
Build a GitOps pipeline using ArgoCD + GitHub repo.

**☁️ Phase 9: Cloud & Managed Kubernetes**

🎯 Goal: Work with cloud-managed clusters.

**🔹 Topics**

AWS EKS, Azure AKS, GCP GKE basics

Node groups, IAM roles

Cluster upgrades, autoscaling

Load balancers, external DNS

**📘 Practice Project:**  
Deploy your app on AWS EKS with auto-scaling and Ingress.

**🧠 Phase 10: Expert Level (Real-World Skills)**

🎯 Goal: Build and maintain enterprise-grade Kubernetes systems.

**🔹 Topics**

Service Mesh (Istio / Linkerd)

Operators & Custom Resource Definitions (CRDs)

Multi-cluster management (KubeFed)

Backup & recovery (Velero)

Security best practices (OPA Gatekeeper, Falco)

Edge Kubernetes (K3s / MicroK8s)

Performance tuning and troubleshooting

**📘 Practice Project:**  
Create a multi-service microservice app with:

Istio service mesh

Canary deployment

Monitoring & tracing with Jaeger

**🧾 Bonus: Tools & Resources**

**🧰 Tools to Install**

kubectl

minikube or kind (local cluster)

helm

k9s (visual cluster manager)

lens (Kubernetes IDE)

Prometheus & Grafana

ArgoCD / Flux

📚 **Learn From:**

Official Docs: https://kubernetes.io/docs/

Playgrounds:

https://killercoda.com

https://katacoda.com

Practice Labs:

<https://play-with-k8s.com>

| **Project** | **Description** |
| --- | --- |
| **E-commerce microservice app** | Frontend, backend, and DB running as Pods |
| **CI/CD with GitOps** | Auto-deploy on Git push |
| **Monitoring stack** | Prometheus + Grafana dashboards |
| **Logging setup** | EFK or Loki stack |
| **Blue-Green deployment** | Zero downtime upgrades |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

🏆 **Final Stage**

Once you complete all phases, you’ll be ready for:

CKA (Certified Kubernetes Administrator)

CKAD (Certified Kubernetes Application Developer)

DevOps Engineer roles in top companies 🚀

Top of Form

Bottom of Form