

Docker, Kubernetes, and CKA

8-week detailed curriculum | Target: Certified Kubernetes Administrator (CKA)

A message to you, Peter

Peter, I built this guide to give you a clear path into technology without expecting you to already know the language. You are not supposed to understand everything on day one. Your job is to show up consistently, practice, ask questions, and keep evidence of what you build. I will help you connect the pieces, but the hands-on work is where the confidence will come from.

Study rule: Do not only watch videos. Type the commands, build the labs, break things, fix them, and explain what happened in your own words.

Your outcome for this phase

Develop command-line speed and practical skill deploying, configuring, maintaining, and troubleshooting Kubernetes clusters.

Expected pace: 2-3 hours Monday through Friday, 3-4 hours Saturday, and Sunday reserved for rest or catch-up. Each week ends with something you can demonstrate.

Phase learning resources

Resource	How to use it
Play with Docker	Browser-based Docker environment.
Killercodea	Free browser-based Linux, Docker, and Kubernetes scenarios.
Kubernetes Basics	Official interactive Kubernetes tutorial.
KodeKloud Engineer	Task-based Linux and DevOps practice.
CKA official page	Current domains, exam format, and official details.

How to prove you learned it

- Explain the concept without reading notes.
- Complete the lab twice: once with instructions and once from memory.
- Push scripts, diagrams, screenshots, or written reflections to GitHub.
- Write down one failure, the cause, and the fix each week.
- Answer practice questions only after performing hands-on work.

Week 1 - Containers and Docker basics

Weekly deliverable: Complete the assigned lab, update your GitHub notes, and record a 3-5 minute explanation of what you learned.

Day	Exact learning assignment	Proof of work
Mon	Images, containers, layers, registries	Notes + commands/screenshots
Tue	Dockerfile instructions and build context	Notes + commands/screenshots
Wed	Ports, volumes, environment variables, networks	Notes + commands/screenshots
Thu	Docker Compose and multi-container apps	Notes + commands/screenshots
Fri	Lab: containerize a Python web app	Notes + commands/screenshots
Sat	Repeat the lab from memory, create a command cheat sheet, and complete a timed Killercoda scenario.	Completed lab + weekly recap

End-of-week questions

- 1 What problem does this technology or concept solve?
- 2 What are the most important commands, services, or components?
- 3 What broke during the lab, and how did you identify the cause?
- 4 How does this week connect to cloud or DevOps work?

Week 2 - Kubernetes fundamentals

Weekly deliverable: Complete the assigned lab, update your GitHub notes, and record a 3-5 minute explanation of what you learned.

Day	Exact learning assignment	Proof of work
Mon	Cluster architecture and API objects	Notes + commands/screenshots
Tue	kubectl workflow, YAML, namespaces, labels, selectors	Notes + commands/screenshots
Wed	Pods, ReplicaSets, Deployments	Notes + commands/screenshots
Thu	Services and basic networking	Notes + commands/screenshots
Fri	Lab: deploy, expose, scale, update, rollback	Notes + commands/screenshots
Sat	Repeat the lab from memory, create a command cheat sheet, and complete a timed Killercoda scenario.	Completed lab + weekly recap

End-of-week questions

- 1 What problem does this technology or concept solve?
- 2 What are the most important commands, services, or components?
- 3 What broke during the lab, and how did you identify the cause?
- 4 How does this week connect to cloud or DevOps work?

Week 3 - Workloads and scheduling

Weekly deliverable: Complete the assigned lab, update your GitHub notes, and record a 3-5 minute explanation of what you learned.

Day	Exact learning assignment	Proof of work
Mon	ConfigMaps and Secrets	Notes + commands/screenshots
Tue	Requests, limits, probes	Notes + commands/screenshots
Wed	Jobs, CronJobs, DaemonSets, StatefulSets	Notes + commands/screenshots
Thu	Taints, tolerations, affinity, topology spread	Notes + commands/screenshots
Fri	Lab: resilient application deployment	Notes + commands/screenshots
Sat	Repeat the lab from memory, create a command cheat sheet, and complete a timed Killercoda scenario.	Completed lab + weekly recap

End-of-week questions

- 1 What problem does this technology or concept solve?
- 2 What are the most important commands, services, or components?
- 3 What broke during the lab, and how did you identify the cause?
- 4 How does this week connect to cloud or DevOps work?

Week 4 - Services and networking

Weekly deliverable: Complete the assigned lab, update your GitHub notes, and record a 3-5 minute explanation of what you learned.

Day	Exact learning assignment	Proof of work
Mon	ClusterIP, NodePort, LoadBalancer	Notes + commands/screenshots
Tue	Ingress and Gateway concepts	Notes + commands/screenshots
Wed	CoreDNS and service discovery	Notes + commands/screenshots
Thu	CNI and NetworkPolicy	Notes + commands/screenshots
Fri	Lab: diagnose service connectivity	Notes + commands/screenshots
Sat	Repeat the lab from memory, create a command cheat sheet, and complete a timed Killercoda scenario.	Completed lab + weekly recap

End-of-week questions

- 1 What problem does this technology or concept solve?
- 2 What are the most important commands, services, or components?
- 3 What broke during the lab, and how did you identify the cause?
- 4 How does this week connect to cloud or DevOps work?

Week 5 - Storage

Weekly deliverable: Complete the assigned lab, update your GitHub notes, and record a 3-5 minute explanation of what you learned.

Day	Exact learning assignment	Proof of work
Mon	Volumes, PV, PVC	Notes + commands/screenshots
Tue	StorageClasses and dynamic provisioning	Notes + commands/screenshots
Wed	Access modes and reclaim policies	Notes + commands/screenshots
Thu	Stateful workload patterns	Notes + commands/screenshots
Fri	Lab: deploy database with persistent storage	Notes + commands/screenshots
Sat	Repeat the lab from memory, create a command cheat sheet, and complete a timed Killercoda scenario.	Completed lab + weekly recap

End-of-week questions

- 1 What problem does this technology or concept solve?
- 2 What are the most important commands, services, or components?
- 3 What broke during the lab, and how did you identify the cause?
- 4 How does this week connect to cloud or DevOps work?

Week 6 - Cluster administration

Weekly deliverable: Complete the assigned lab, update your GitHub notes, and record a 3-5 minute explanation of what you learned.

Day	Exact learning assignment	Proof of work
Mon	kubeadm workflow and cluster components	Notes + commands/screenshots
Tue	RBAC and service accounts	Notes + commands/screenshots
Wed	Helm, Kustomize, CRDs, operators	Notes + commands/screenshots
Thu	Cluster upgrades, backup, restore, etcd concepts	Notes + commands/screenshots
Fri	Lab: RBAC plus Helm installation	Notes + commands/screenshots
Sat	Repeat the lab from memory, create a command cheat sheet, and complete a timed Killercoda scenario.	Completed lab + weekly recap

End-of-week questions

- 1 What problem does this technology or concept solve?
- 2 What are the most important commands, services, or components?
- 3 What broke during the lab, and how did you identify the cause?
- 4 How does this week connect to cloud or DevOps work?

Week 7 - Troubleshooting

Weekly deliverable: Complete the assigned lab, update your GitHub notes, and record a 3-5 minute explanation of what you learned.

Day	Exact learning assignment	Proof of work
Mon	Node and control-plane troubleshooting	Notes + commands/screenshots
Tue	Pod states, events, logs, exec, debug	Notes + commands/screenshots
Wed	DNS, services, endpoints, network troubleshooting	Notes + commands/screenshots
Thu	Resource pressure, scheduling, image failures	Notes + commands/screenshots
Fri	Lab: complete timed broken-cluster scenarios	Notes + commands/screenshots
Sat	Repeat the lab from memory, create a command cheat sheet, and complete a timed Killercoda scenario.	Completed lab + weekly recap

End-of-week questions

- 1 What problem does this technology or concept solve?
- 2 What are the most important commands, services, or components?
- 3 What broke during the lab, and how did you identify the cause?
- 4 How does this week connect to cloud or DevOps work?

Week 8 - CKA performance preparation

Weekly deliverable: Complete the assigned lab, update your GitHub notes, and record a 3-5 minute explanation of what you learned.

Day	Exact learning assignment	Proof of work
Mon	Imperative command speed and YAML generation	Notes + commands/screenshots
Tue	Documentation navigation and bookmarks	Notes + commands/screenshots
Wed	Timed domain drills with emphasis on troubleshooting	Notes + commands/screenshots
Thu	Killer.sh simulations and error review	Notes + commands/screenshots
Fri	Final mock exam and readiness decision	Notes + commands/screenshots
Sat	Repeat the lab from memory, create a command cheat sheet, and complete a timed Killercoda scenario.	Completed lab + weekly recap

End-of-week questions

- 1 What problem does this technology or concept solve?
- 2 What are the most important commands, services, or components?
- 3 What broke during the lab, and how did you identify the cause?
- 4 How does this week connect to cloud or DevOps work?

Final phase assessment

Do not move forward based only on time spent. Move forward when you can perform and explain the following:

- Complete the main project without copying every step.
- Troubleshoot at least three intentionally introduced failures.
- Explain the architecture or workflow using a diagram.
- Score at least 80% on two separate practice assessments when a certification is targeted.
- Teach the core concepts back to another person in plain language.

Certification note

Target: **Certified Kubernetes Administrator (CKA)**. Certification dates should be adjusted based on readiness. Passing quickly is less important than retaining the skill and being able to use it in a real environment.