

Kubernetes Hands-On Field Guide

Every command from the series, with what it does. It grows as new episodes air.

Episodes 1-2 · updated 2026-10-03 · the latest version is always at aislabs.ai/kubernetes

EPISODE 1 Your first cluster heals itself

<code>kind create cluster --name demo --wait 60s</code>	Create a local Kubernetes cluster inside Docker, and wait until it is ready.
<code>kubectl get nodes</code>	List the machines in the cluster and whether each one is Ready.
<code>kubectl get pods -n kube-system</code>	See the cluster's own parts (API server, etcd, scheduler, DNS) running as pods.
<code>kubectl create deployment web --image=nginx:alpine --replicas=3</code>	Ask for three copies of a web server.
<code>kubectl get pods -w</code>	Watch pods change state live. Ctrl+C stops watching.
<code>kubectl get deployment,replicaset,pods</code>	See the chain: a deployment owns a replica set, which owns the pods.
<code>kubectl delete pod POD_NAME</code>	Delete one pod. The replica set starts a replacement within seconds.
<code>kubectl delete pods -l app=web</code>	Delete every pod with that label. All of them are replaced.
<code>kubectl scale deployment web --replicas=5</code>	Change how many copies you want.
<code>kubectl delete deployment web --cascade=foreground</code>	Remove the app for real: nothing asks for the pods any more, so none come back.
<code>kind delete cluster --name demo</code>	Delete the whole cluster.

EPISODE 2 Docker containers explained

<code>docker pull nginx:alpine</code>	Download an image: a packaged filesystem plus a default command.
<code>docker images</code>	List the images stored on this machine.
<code>docker run -d --name web -p 127.0.0.1:8080:80 nginx:alpine</code>	Start a container in the background, with port 8080 on the loopback address mapped to port 80 inside.
<code>docker ps</code>	List running containers.
<code>curl -s localhost:8080 head -n 4</code>	Fetch the page the container serves.
<code>docker logs --tail 2 web</code>	Show the last lines the container printed.
<code>docker exec web nginx -v</code>	Run a command inside the running container.
<code>docker inspect -f '{{.State.Status}}' web</code>	Read one field of everything Docker knows about the container.
<code>docker stop web</code>	Stop the container. It stays on disk, exited.
<code>docker ps -a</code>	List every container, stopped ones included.
<code>docker rm web</code>	Remove a stopped container. The image stays.
<code>docker rm -f blue green</code>	Stop and remove running containers in one step.

Kubernetes Hands-On is a free, screen-recorded series from AIS: every command here was typed live on a real cluster. POD_NAME stands for a name from your own `kubectl get pods` output. Kubernetes is a registered trademark of The Linux Foundation; this series is independent and not affiliated with or endorsed by it.