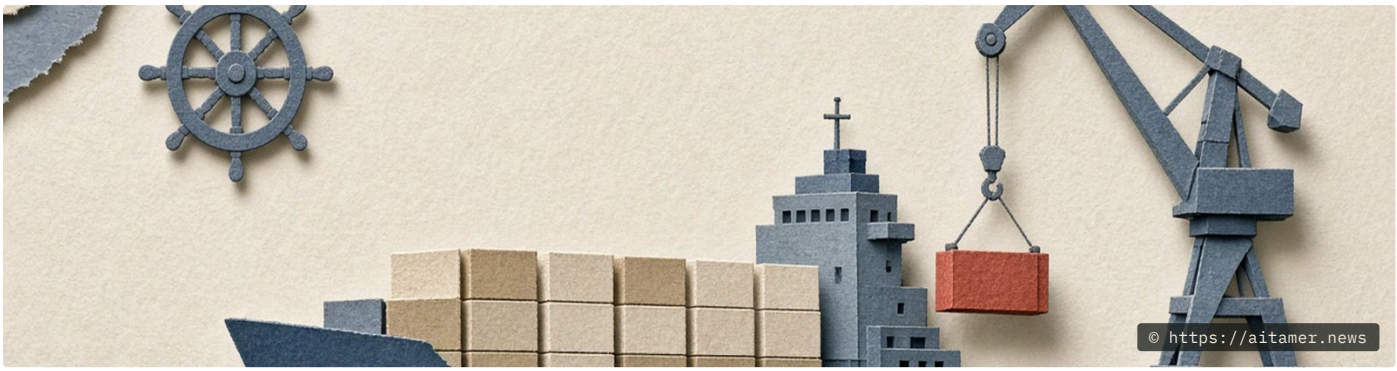


KUBERNETES FROM ZERO · FOUNDATIONS 1 OF 12 · LESSON 1 · LESSON KIT · CHEAT SHEET

Kubernetes from Zero, part 1: what Kubernetes is and why it exists

Words, the desired-state loop, and the commands from Lesson 1.

Ran every command on kind v0.33.0, Kubernetes v1.37.0 node, kubectl v1.37.2, 10 Nov 2026. Cluster: k8s-from-zero.

THE ONE RULE

Kubernetes keeps a running system matching the state you declare. You describe what you want; its controllers keep closing the gap.

Words from the lesson

defined once here, and properly in later parts

WORD	WHAT THE LESSON SAYS	WHERE YOU MEET IT
Container	A process with its own filesystem, its own share of CPU and memory, and its own view of the system, while sharing the host's kernel.	The packaging step, before the cluster.
Image	The packaged filesystem a container starts from.	The `image` on a container.
Cluster	A group of machines. The control plane stores what you asked for and decides. The worker nodes run your containers.	kind runs one node as a Docker container.
Desired state	You say what you want, such as three copies of a web server. Kubernetes makes that true and keeps it true.	An object's `spec`.
spec and status	An object is a record of intent. You write `spec`. Kubernetes writes `status`, what is true right now.	`kubectl get deployment hello -o yaml`
Controller	A loop that watches one kind of object and keeps nudging the real world toward its spec. The lesson compares it to a thermostat.	The ReplicaSet controller in the lab.
Pod	The smallest thing Kubernetes runs: one or more containers that share a network address.	Part 4.
Deployment	Keep N identical Pods of this template running, and update them gradually.	Part 7. The lab creates `hello`.
ReplicaSet	The helper a Deployment creates to hold the exact Pod count. You rarely touch it directly.	The events in step 6.
kubectl	The command-line tool you use to talk to a cluster.	Every lab command after kind.

What you still bring

the overview's own list, as the lesson states it

YOU STILL BRING	WHAT THE LESSON SAYS KUBERNETES DOES NOT DO	WHERE THE SERIES PICKS IT UP
Build and CI/CD	It does not build your code or deploy source code. You still need a pipeline that produces an image.	A later arc covers CI/CD.
Data services	It does not ship databases, caches or message queues as built-in services. You can run them as applications.	They are applications like any other.
Signals	It does not dictate logging, monitoring or alerting. It gives you hooks. You pick the tools.	Your own choice of tools.
The machines	It does not configure, patch or repair the machines themselves.	Outside the cluster.

SCAN FOR THE LESSON

Lesson: **Kubernetes from Zero, part 1: what Kubernetes is and why it exists**

https://aitamer.news/posts/k8s-from-zero-01-what-kubernetes-is/?utm_source=lessonkit&utm_medium=pdf&utm_campaign=k8s-01&utm_content=cheatsheet

NEXT IN THE SERIES **Next: Lesson 2, a tour of the control plane and the nodes**. A guided tour of the control plane and the nodes, following one kubectl apply until a container runs. Lesson page

Free to share with credit: © aitamer.news

What it does, in the overview's own list: service discovery and load balancing, storage orchestration, automated rollouts and rollbacks, automatic bin packing, self-healing, secret and configuration management, batch execution, horizontal scaling, and IPv4/IPv6 dual-stack.

The loop you ran

three copies, delete one, ask for five

1	**kind**	`kind create cluster --name k8s-from-zero` starts the cluster. The lesson's node is `k8s-from-zero-control-plane`, `Ready`, Kubernetes v1.37.0. Your port number will differ.
2	**kubectl create**	Deployment `hello`, image `nginx:1.29-alpine`, 3 replicas. `kubectl rollout status deployment/hello` returns when it has successfully rolled out.
3	**ReplicaSet**	You delete one Pod. Still three. The lesson says the ReplicaSet controller saw want 3, have 2, and created one. Nobody ran a restart command.
4	**kubectl scale**	`--replicas=5`. In the lesson, `spec.replicas` is 5 and `status.readyReplicas` is 5. When they match, the loop is at rest.

The events list six `SuccessfulCreate` lines: the first three Pods, the replacement, and the two extra for scaling to five. On the lab machine the image was preloaded with `kind load docker-image`. On a normal laptop it pulls by itself.

Key commands

commands verbatim from the lesson

Install

```
curl -LO \
  "https://dl.k8s.io/release/v1.37.2/bin/linux/amd64/kubectl"
```
pins kubectl v1.37.2. `curl -L -s https://dl.k8s.io/release/stable.txt` prints the latest

```
curl -Lo ./kind \
  https://kind.sigs.k8s.io/dl/v0.33.0/kind-linux-amd64
```
kind v0.33.0, from the quick start

```
chmod +x kubectl kind
```
mark both binaries executable

```
sudo mv kubectl kind /usr/local/bin/
```
the lesson's install line

Cluster

```
kind create cluster --name k8s-from-zero
```
needs Docker running

```
kind delete cluster --name k8s-from-zero
```
removes the node containers

```
kubectl cluster-info
```
prints the API server address

```
kubectl get nodes
```
`STATUS` should be `Ready`

Desired state

```
kubectl create deployment hello --image=nginx:1.29-alpine \
  --replicas=3
```
pin a tag, never `latest`

```
kubectl rollout status deployment/hello
```
returns when all copies are available

```
kubectl get pods -o wide
```
add `-A` for all namespaces

```
kubectl scale deployment hello --replicas=5
```
the controller adds or removes Pods

```
kubectl get deployment hello -o yaml
```
compare `spec` with `status`

```
kubectl get events --sort-by=.lastTimestamp
```
events expire after a while

```
kubectl describe pod <name>
```
read the Events section at the bottom

```
kubectl delete deployment hello
```
its Pods go with it

The lab's extra commands

```
kubectl delete pod hello-6bb4847c46-b8jpw
```
use a name from your own list

```
kubectl get events --field-selector \
  involvedObject.kind=ReplicaSet --sort-by=.lastTimestamp
```
the ReplicaSet controller's diary

```
kubectl port-forward deployment/hello 8080:80
```
then curl from a second terminal

```
curl -s localhost:8080 | grep -i title
```
the lesson prints the nginx title

SCAN FOR THE LESSON

Lesson: **Kubernetes from Zero, part 1: what Kubernetes is and why it exists**

https://aitamer.news/posts/k8s-from-zero-01-what-kubernetes-is/?utm_source=lessonkit&utm_medium=pdf&utm_campaign=k8s-01&utm_content=cheatsheet

NEXT IN THE SERIES **Next: Lesson 2, a tour of the control plane and the nodes**. A guided tour of the control plane and the nodes, following one kubectl apply until a container runs. Lesson page

Free to share with credit: © aitamer.news