

KUBERNETES FROM ZERO · WORKLOADS & TROUBLESHOOTING 2 OF 14 · LESSON 14 · LESSON KIT · CHEAT SHEET

Kubernetes from Zero, part 14: StatefulSets, stable names and a disk per Pod

Stable names, one claim per Pod, and the commands from Lesson 14.

Every command ran on three-node kind v0.33.0 (node v1.37.0, kubectl v1.37.2), postgres:18-alpine, 10 Nov 2026. Cluster: k8s-ep13.

THE ONE RULE

Use a StatefulSet when each replica needs its own name and its own disk. Create the headless Service, keep claims on scale down, and back up the data.

What a StatefulSet keeps stable

from the recap and the cheat sheet

PIECE	WHAT THE LESSON SAYS	THE FIELD OR THE NAME
Headless Service	You create it. `clusterIP: None`.	`spec.serviceName` must match its name
Pod names	`web-0`, `web-1`, and so on. The set name plus the ordinal.	stable across restarts
Pod DNS	`web-0.web.k8s-from-zero.svc.cluster.local`. Short name `web-0.web` in the namespace.	needs the headless Service
Claim template	`volumeClaimTemplates` makes one claim per Pod. Claims are named `www-web-0` and so on.	the template is immutable
OrderedReady	Pods start in ordinal order and stop in reverse. One unready Pod blocks every higher ordinal.	default management
Parallel	Starts them together and keeps the identities.	`podManagementPolicy`
partition	Only ordinals at or above this number update.	`rollingUpdate.partition`
Claim retention	Claims survive scale-down and deletion by default. Both policies default to Retain.	`persistentVolumeClaimRetentionPolicy`

The learning Postgres

what the lesson says this is, and is not

1	**image**	The lesson tested `postgres:18-alpine`. Postgres 18 mounts at `/var/lib/postgresql`. Versions 17 and older use `/var/lib/postgresql/data`.
2	**order**	With OrderedReady, `web-0` becomes Ready before `web-1` is created. A 403 from a readiness probe left `web-0` unready for three minutes and `web-1` was never created.
3	**claims**	Scale-down stops the highest ordinal first and, by default, leaves its claim. List PVCs before you reuse a StatefulSet name.
4	**production**	Running a database this way works for learning. The lesson says production needs replication, backups and tested restores, which later episodes cover.

SCAN FOR THE LESSON

Lesson: **Kubernetes from Zero, part 14: StatefulSets, stable names and a disk per Pod**

https://aitamer.news/posts/k8s-from-zero-14-statefulsets/?utm_source=lessonkit&utm_medium=pdf&utm_campaign=k8s-14&utm_content=cheatsheet

NEXT IN THE SERIES **Next: Lesson 15, DaemonSets, Jobs and CronJobs.** Kubernetes from Zero, part 15, DaemonSets, Jobs and CronJobs. Lesson page

Free to share with credit: © aitamer.news

Key commands

commands verbatim from the lesson

Watch and scale

`kubectl get pods -l app.kubernetes.io/name=web -w`
each Pod waits for Ready

`kubectl rollout status statefulset/web`
works like Deployments

`kubectl scale statefulset web --replicas=1`
highest ordinal stops first

`kubectl get pods -L apps.kubernetes.io/pod-index`
the ordinal, stable since v1.32

`kubectl delete pvc -l app.kubernetes.io/name=web`
check the list first

SCAN FOR THE LESSON

Lesson: **Kubernetes from Zero, part 14: StatefulSets, stable names and a disk per Pod**

https://aitamer.news/posts/k8s-from-zero-14-statefulsets/?utm_source=lessonkit&utm_medium=pdf&utm_campaign=k8s-14&utm_content=cheatsheet

NEXT IN THE SERIES Next: Lesson 15, DaemonSets, Jobs and CronJobs. Kubernetes from Zero, part 15, DaemonSets, Jobs and CronJobs. Lesson page

Free to share with credit: © aitamer.news