

KUBERNETES FROM ZERO · WORKLOADS & TROUBLESHOOTING 3 OF 14 · LESSON 15 · LESSON KIT · CHEAT SHEET

Kubernetes from Zero, part 15: DaemonSets, Jobs and CronJobs

One Pod per node, work that finishes, and the commands from Lesson 15.

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

THE ONE RULE

A DaemonSet keeps one Pod on each matching node. A Job runs Pods until enough succeed or retries run out. A CronJob creates Jobs on a schedule.

Three controllers from the recap

CONTROLLER	WHAT THE LESSON SAYS	THE FIELD TO REMEMBER
DaemonSet	One Pod on every qualifying node. There is no replicas field. DESIRED is that node count.	taints and a node selector
Job	Runs Pods until enough succeed, or retries run out. Failed Pods stay when restartPolicy is Never.	completions and parallelism
Indexed Job	Each Pod gets an index from 0. The hostname is the Job name plus the index.	JOB_COMPLETION_INDEX
CronJob	Creates a Job on a five-field schedule. Set timeZone or the controller manager's zone is used.	concurrencyPolicy

Defaults the lab used as the lesson states them

FIELD	WHAT THE LESSON SAYS	DEFAULT
completions	How many Pods must succeed. The batch Job set this to 6.	1
parallelism	How many Pods may run at once. The batch Job set this to 2.	1
backoffLimit	Retries before the Job is Failed. Delay starts at 10s and doubles, capped at ten minutes.	6
restartPolicy	Never or OnFailure. Never keeps each failed Pod so you can read its logs.	Never while debugging
TTL	ttlSecondsAfterFinished deletes the finished Job and its Pods. Stable since v1.23.	unset means they stay
History	The report CronJob kept 2 successful Jobs and 1 failed Job.	successfulJobsHistoryLimit
Overlap	Forbid skips a tick while a Job is still running. Replace stops the old one. Allow overlaps.	Allow

What the lab showed on the three-node kind cluster k8s-ep13

1	DaemonSet	node-agent was DESIRED 2 until a toleration for node-role.kubernetes.io/control-plane:NoSchedule was added. DESIRED then became 3.
2	Indexed Job	batch finished 6/6 in 18 seconds, two Pods at a time. completedIndexes was 0-5.
3	Retries	flaky used backoffLimit 2, failed three times, and ended Failed with reason BackoffLimitExceeded.
4	CronJob	report ran every minute in Etc/UTC. kubectl create job --from=cronjob/report runs the template now. suspend: true pauses new Jobs.

SCAN FOR THE LESSON

Lesson: **Kubernetes from Zero, part 15: DaemonSets, Jobs and CronJobs**

https://aitamer.news/posts/k8s-from-zero-15-daemonsets-jobs-cronjobs/?utm_source=lessonkit&utm_medium=pdf&utm_campaign=k8s-15&utm_content=cheatsheet

NEXT IN THE SERIES **Next lesson**. The next lesson in Kubernetes from Zero. Series page

Free to share with credit: © aitamer.news

Key commands

commands verbatim from the lesson

DaemonSets

```
kubectl get daemonsets -A
```
DESIRED is the number of qualifying nodes

```
kubectl describe node k8s-ep13-control-plane | grep Taints
```
control-plane:NoSchedule on kind

```
kubectl rollout status daemonset/node-agent
```
rolls one node at a time

CronJobs

```
kubectl create job report-now --from=cronjob/report
```
runs the template without waiting

```
kubectl patch cronjob report -p '{"spec":{"suspend":true}}'
```
running Jobs continue

```
kubectl get events --field-selector \
  involvedObject.kind=CronJob
```
history limits show up as deletes

Jobs

```
kubectl wait --for=condition=complete job/batch --timeout=120s
```
or condition=failed

```
kubectl logs -l job-name=batch
```
the Job adds the job-name label

```
kubectl get job batch -o jsonpath='{.status.completedIndexes} {.status.succeeded}{"\n"}'
```
Indexed mode records the indexes

SCAN FOR THE LESSON

Lesson: **Kubernetes from Zero, part 15: DaemonSets, Jobs and CronJobs**

https://aitamer.news/posts/k8s-from-zero-15-daemonsets-jobs-cronjobs/?utm_source=lessonkit&utm_medium=pdf&utm_campaign=k8s-15&utm_content=cheatsheet

NEXT IN THE SERIES **Next lesson**. The next lesson in Kubernetes from Zero. Series page

Free to share with credit: © aitamer.news