

KUBERNETES FROM ZERO · FOUNDATIONS 1 OF 12 · LESSON 1 · LESSON KIT · ANSWER KEY

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

The answers to the homework sheet, matched by exercise number. Every answer comes from the lesson. Names with random suffixes will differ on your cluster.

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

Recall

1. What is a container, and what is an image?

   A container is a process that runs with its own filesystem, its own share of CPU and memory and its own view of the system, while sharing the host's kernel. An image is the packaged filesystem a container starts from.

2. Name the operational questions the lesson says containers left open.

   Where each container runs. Who restarts it when it crashes. How other services find it. How you ship version 2 without downtime, and go back to version 1 if it breaks. How you run ten copies instead of two when traffic grows.

3. What is desired state? What do `spec` and `status` hold, and who writes each?

   You tell Kubernetes what you want, such as three copies of a web server, and it keeps that true. `spec` is what you want, and you write it. `status` is what is true right now, and Kubernetes writes it. An object is a record of intent.

4. What is a controller? What everyday thing does the lesson compare it to?

   A loop that watches objects of one kind and keeps nudging the real world toward their spec. The controllers page compares it to a thermostat: you set a temperature, and it keeps checking until the room matches.

5. In one line each, what is a Pod, a Deployment, a ReplicaSet, and kubectl?

   A Pod is the smallest thing Kubernetes runs, one or more containers that share a network address. A Deployment says keep N identical Pods of this template running, and update them gradually. A ReplicaSet is the helper a Deployment creates to hold the exact Pod count. kubectl is the command-line tool you use to talk to a cluster.

6. Name four things the lesson says Kubernetes does not do.

   It does not build your code or deploy source code. It does not ship databases, caches or message queues as built-in services. It does not dictate logging, monitoring or alerting. It does not configure, patch or repair the machines themselves.

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=answer-key

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

Run this and explain

7 Run `kind create cluster --name k8s-from-zero`, then `kubectl cluster-info` and `kubectl get nodes`. What context does kind set, and what STATUS and VERSION does the lesson show?

> The lesson's output says `Set kubectl context to "kind-k8s-from-zero"`. The node `k8s-from-zero-control-plane` is `Ready`, version v1.37.0. Your port number will differ.

8 Run `kubectl create deployment hello --image=nginx:1.29-alpine --replicas=3` and `kubectl rollout status deployment/hello`. What did you declare, and when does rollout status return?

> Three copies of `nginx:1.29-alpine`. It returns when the deployment has successfully rolled out. The lesson's last line is `deployment "hello" successfully rolled out`.

9 Delete one `hello` Pod, using a name from your own `kubectl get pods`, then list the Pods again. How many are there, and why?

> Still three. The deleted Pod is gone and a new one takes its place. The ReplicaSet controller saw want 3, have 2, and created one. Nobody ran a restart command.

10 Run `kubectl scale deployment hello --replicas=5` and `kubectl get deployment hello -o yaml`. What do `spec.replicas` and `status.readyReplicas` say in the lesson?

> Both are 5. You wrote `spec.replicas` through `kubectl scale`. Kubernetes reports `status.readyReplicas`. When they match, the loop is at rest.

11 Run `kubectl get events --field-selector involvedObject.kind=ReplicaSet --sort-by=.lastTimestamp`. How many `SuccessfulCreate` events does the lesson show, and what are they?

> Six. The first three Pods, the replacement for the one you deleted, and the two extra for scaling to five.

12 Run `kubectl port-forward deployment/hello 8080:80` and, in a second terminal, `curl -s localhost:8080 | grep -i title`. What does the lesson print?

> `<title>Welcome to nginx!</title>`. Press Ctrl+C in the first terminal to stop the forward.

Predict the outcome

13 You delete a Pod that a Deployment owns, and a new Pod appears. Is that a bug, and how do you remove the Pods for good?

> It is the Deployment doing its job. To remove Pods for good, delete or scale down the Deployment that owns them.

14 A Pod stays in `ErrImagePull` or `ImagePullBackOff`. What does the lesson tell you to read, and what did the misspelt tag `nginx:1.29-alpinee` do?

> `kubectl describe pod <name>` shows the reason in Events. The misspelt tag stayed in `ErrImagePull`. On a machine with internet the message says the image or tag was not found. The lab said `i/o timeout` because the cluster had no internet route. Fix the tag with `kubectl set image`, or delete the Deployment.

15 The shell says `kind: command not found`. What does the lesson tell you to check?

> The binary is not in your PATH. Check with `ls -l /usr/local/bin/kind /usr/local/bin/kubectl`, and run the `sudo mv` line again.

16 kind says `Cannot connect to the Docker daemon`. What has to be true before `kind create cluster`?

> kind needs a running Docker (or Podman). Start Docker Desktop or the Docker service, then run `docker ps` to confirm.

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=answer-key

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