Kubernetes Intro Exercises

Namespaces, Nodes, and Pods

Setup

- Register at https://killercoda.com
- Open the Kubernetes playground: https://killercoda.com/playgrounds/scenario/kubernetes
- Check whether `kubectl` can talk to the cluster:

```
kubectl get nodes
```

- Apply the exercise resources:

```
kubectl apply -f https://georg-schwarz.com/2026-ohm-lectures/ex1.yaml
```

Exercise 1: Explore the Cluster

- a) How many non-`control-plane` nodes are there in the cluster?
- b) Which Kubernetes version is running on the nodes of the cluster?
- c) What is the IP address of the control plane?
- d) What container runtime is used?
- e) How many pods can be deployed on the control plane node?

```
# Useful commands
kubectl get nodes
kubectl get nodes -o wide
kubectl describe node <node-name>
```

Exercise 2: Namespaces

- a) How many pods are there on the overall cluster?
- b) How many pods are there in the `shop-dev` namespace?
- c) Which namespaces have been added recently?

```
# Useful commands:
kubectl get namespaces
kubectl get pods
kubectl get pods -A
kubectl get pods -n <namespace>
```

Exercise 3: Inspect Pods

- a) Which node does the pod `web` in the `shop-dev` namespace run on?
- b) What is the ip address of pod `web` in the `shop-dev` namespace?
- c) Which container images does the pod `web` in the `shop-dev` namespace use?
- d) How much RAM can the pod `web` in the `shop-dev` namespace use?

```
# Useful commands:
kubectl get pods -n <namespace> -o wide
kubectl describe pod <pod>
```

Exercise 4: Logs and Exec

- a) Check the logs of the `broken` pod in the `project-x` namespace. What is the issue?
- b) Check if you can reach the `working` pod from the `toolbox` pod in the `project-x` namespace. Use `curl` to check the connection.

```
# Useful commands:
kubectl logs <pod>
kubectl exec -it <pod> -- sh
```

Exercise 5: Multi-Container Pod

- a) Inspect the `multi-tool` pod in namespace `project-x`. How many containers are currently running?
- b) Check the logs of the containers in the `multi-tool` pod. Explain what you see.

Exercise 6: Debug Broken Pods

- a) Diagnose what is wrong with pod `webapp1` in namespace `superapp` and fix it.
- b) Diagnose what is wrong with pod `webapp2` in namespace `superapp` and fix it.

```
# Useful commands:
kubectl describe pod <pod>
kubectl logs <pod>
kubectl edit pod <pod>
kubectl get pod <pod> -o yaml > <local-file>
kubectl delete pod <pod>
kubectl apply -f <yaml-manifest>
```

Cleanup

Close the KillerCoda playground.