Experiments

In this lesson, we'll look at some fun experiments you can try.

WE'LL COVER THE FOLLOWING

Additional microservice

Supplement the Kubernetes system with an additional microservice.

- A microservice that is used by a call center agent to create notes for a call can be used as an example. The call center agent should be able to select the customer.
- For calling the customer microservice the hostname `customer` has to be used.
- Of course, you can copy and modify one of the existing microservices.
- Package the microservice in a Docker image and upload it into the Docker repository. This can be done by adapting the script `docker-build.sh`.
- Adapt `kubernetes-deploy.sh` in such a manner that the microservice is deployed.

- Adapt `kubernetes-remove.sh` in such a manner that the microservice is deleted.

Interactive tutorial

https://kubernetes.io/docs/getting-started-guides/ is an interactive tutorial that shows how to use Kubernetes. It complements this chapter well. Work through the tutorial to get an impression of the Kubernetes features.

Run a rolling update

Kubernetes supports rolling updates. A new version of a pod is rolled out in such a way that there are **no interruptions to the service**. See https://kubernetes.io/docs/tasks/run-application/rolling-update-replication-controller/.

Run a rolling update! To do this you need to create a new Docker image. The scripts for compiling and delivering to the Docker hub are included in the example.

Try a hosted solution

Cloud providers such as Google or Microsoft offer Kubernetes infrastructures, see https://kubernetes.io/docs/getting-started-guides/#hosted-solutions.

Make the example work in such an environment! The scripts can be used without changes because `kubectl` also supports these technologies.

Test load balancing

Test the load balancing in the example.

- `kubectl scale` alters the number of pods in a replica set. `kubectl scale -h` indicates which options there are. For example, scale the replica set `catalog`.
- `kubectl get deployments` shows how many pods are running in the respective deployment.
- Use the service. For example, `minikube service apache` opens the web page with links to all microservices. Select the order microservice and get the orders displayed.
- `kubectl describe pods -l run=catalog` displays the running pods. There

you can also find the IP address of the pods in a line which starts with `IP`.

- Log into the Kubernetes node with `minikube ssh`. To read out the metrics, you can use a command like `curl 172.17.0.8:8080/metrics`. You have to adapt the IP address. This way you can display the metrics of the catalog pods which Spring Boot creates. For example, the metrics contain the number of requests that have been responded with an HTTP 200 status code (OK). If you use the catalog microservice via the web page, each pod should process some of the requests and the metrics of all pods should go up.
- Use `minikube dashboard` for observing the information in the dashboard.

Deploy Docker registry examples

The example currently uses the public Docker hub. Install your own Docker registry. Save the Docker images of the example in the registry and deploy the example from this registry.

Kubernetes workshop

At https://github.com/GoogleCloudPlatform/kubernetes-workshops you can find material for a Kubernetes workshop to further familiarize yourself with this system.

Port asynchronous examples to Kubernetes

Port the Kafka example (see chapter 7) or the Atom example (see chapter 8) to Kubernetes.

This shows how asynchronous microservices can also run in a Kubernetes cluster. Kafka stores data, which can be difficult in a Kubernetes system. Explore how to run a Kafka cluster in a Kubernetes system in production.

Familiarize yourself with the log

Use `kubectl logs -help` for familiarizing yourself with the log administration in Kubernetes.

Take a look at the logs of at least two microservices.

Checkout logs of some pods

Checkout logs of some pods

Use kail for displaying the logs of some pods.

QUIZ

1 What does the command `kubectl scale --replicas=5 catalog` do?

COMPLETED 0%

1 of 2

We'll conclude this chapter with a quick summary in the next lesson.