Chapter Conclusion

In this lesson, we'll conclude this chapter with a quick summary of what we have learned.

WE'LL COVER THE FOLLOWING

Summary

Kubernetes solves the challenges of synchronous microservices as follows:

- DNS offers **service discovery**. Thanks to DNS, microservices can be used transparently in any programming language. However, DNS only provides the IP address, so the port must be known. No code is required to register the services. When you start the service, a DNS record is created automatically.
- **Load balancing** is ensured by Kubernetes by distributing the traffic for the IP address of the Kubernetes service to the individual pods on the IP level. This is transparent for callers and for the called microservice.
- **Routing** is covered by Kubernetes via the load balancer or node ports of the services. This is also transparent for the microservices.
- **Resilience** is offered by Kubernetes via the restarting of containers and load balancing.
    - In addition, a library like Hystrix can be useful for implementing timeouts or circuit breakers.
    - A proxy like Envoy can be an alternative to Hystrix. Envoy is also part of Istio and implements resilience for Istio.

Within one package, Kubernetes offers complete support for microservices in the cluster including:

- service discovery
- load balancing
- resilience
- scalability

In this way, Kubernetes **solves many challenges** that arise during the operation of a microservices environment. The code of the microservices remains free of these concerns. **No dependencies** on Kubernetes are introduced into the code.

This is attractive, but also represents a **fundamental change**. While Consul or the Netflix stack run on virtual machines or even bare metal, **Kubernetes requires everything to be packed in Docker containers**. This can be a fundamental change compared to an existing mode of operation and can make the migration to this environment harder.

Advantages

- Kubernetes solves most typical challenges of microservices (load balancing, routing, service discovery).
- The code has no dependencies on Kubernetes.
- Kubernetes covers operation and deployment.
- The Kubernetes platform enforces standards and is thereby the definition of a macro architecture.

Challenges

- A complete change of operation is required to use Kubernetes instead of other log or deployment technologies.
- Kubernetes is very powerful, but also very complex.

That’s it for this chapter! We’ll look at a new recipe with the next chapter.