Welcome to

12. Cloud and Cloud integration

KEA System Integration F2020 10 ECTS

Henrik Kramselund Jereminsen hkj@zencurity.com @kramse

Slides are available as PDF, kramse@Github

12-cloud-and-integration-system-integration.tex in the repo security-courses

Goals for today

Todays goals:

- Finish the Camel book
- Talk about cloud systems, in particular Kubernetes
- Talk about cloud security, as time permits

Photo by Thomas Galler on Unsplash

Time schedule

- 08:15 - 09:00 and
- 09:15 - 10:00 2x sessions with 15min break
  Camel chapter 18: Microservices with Docker and Kubernetes
- 10:15 - 11:30 Kubernetes demo, discussion

- 12:15 - 13:45 Exercises

Plan for today

- Microservices with Docker and Kubernetes
- Cloud and Cloud integration
- Running Camel on Docker
- Getting more into Kubernetes

Exercises

- Running Java microservices on Docker
- Getting started with Kubernetes – run Minikube on the web
- Research how to run a few applications on Kubernetes

Part I

Microservices with Docker and Kubernetes

Docker

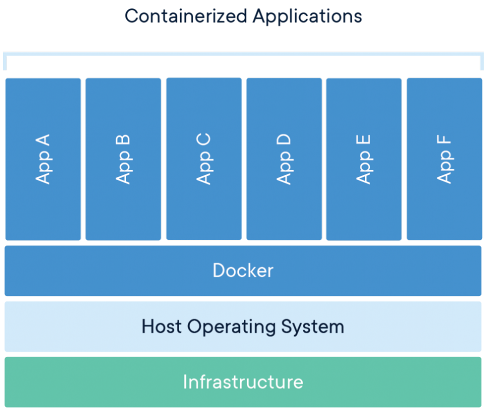
> Package Software into Standardized Units for Development, Shipment and Deployment A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Source:

`https://www.docker.com/resources/what-container`

- One of the most popular deployment methods today

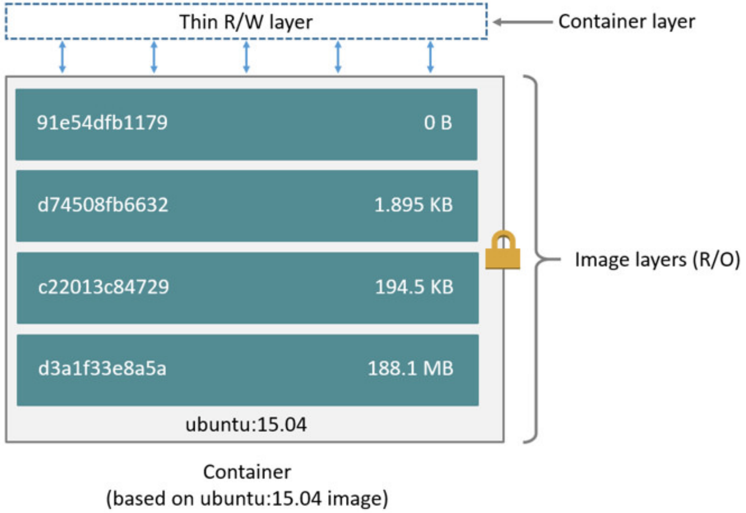
Containerized Applications



Source: `https://www.docker.com/resources/what-container`

- See also `https://en.wikipedia.org/wiki/Linux_namespaces`
  Various container software use Linux namespaces in combination with cgroups to isolate their processes, including Docker[11] and LXC.

Docker Images and layers



A Docker image is built up from a series of layers. Each layer represents an instruction in the image's Dockerfile. Each layer except the very last one is read-only.

Source: `https://docs.docker.com/storage/storagedriver/`

Kubernetes

> Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications. It groups containers that make up an application into logical units for easy management and discovery. Kubernetes builds upon 15 years of experience of running production workloads at Google, combined with best-of-breed ideas and practices from the community.

Source: `https://kubernetes.io/`

Key points:

- Open source originally from Google
- Scalable
- Uses containers inside
- Infrastructure as code

Infrastructure as code

> **Infrastructure as code (IaC)** is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.[1] The IT infrastructure managed by this comprises both physical equipment such as bare-metal servers as well as virtual machines and associated configuration resources. The definitions may be in a version control system. It can use either scripts or declarative definitions, rather than manual processes, but the term is more often used to promote declarative approaches.

Source: `https://en.wikipedia.org/wiki/Infrastructure_as_code`

- Has become the norm in many places

Camel chapter 18: Microservices with Docker and Kubernetes

This chapter covers

- Running Camel on Docker
- Getting started with Kubernetes
- Running and debugging Camel on Kubernetes
- Understanding essential Kubernetes concepts
- Building resilient microservices on Kubernetes
- Testing Camel microservices on Kubernetes
- Introducing fabric8, Kubernetes Helm, and OpenShift

Source:

Camel in action, Claus Ibsen and Jonathan Anstey, 2018, 2nd edition ISBN: 978-1-61729-293-4

18.1.2 Running Camel on Docker

the Dockerfile you'll use to run the Spring Boot client microservice contains just three lines of text:

```
FROM openjdk:latest
COPY maven /maven/
CMD java -jar maven/spring-docker-2.0.0.jar
```

A Docker image is a compressed TAR file that includes the Dockerfile in the root alongside other files you want to include in the image. The Spring Boot Docker image consists of only two files:

```
maven/spring-docker-2.0.0.jar
Dockerfile
```

- We have seen problems with various JDK versions
- Running on Docker might be simpler
- Help available: `https://docs.docker.com/develop/develop-images/dockerfile_best-practices/`

Getting started with Kubernetes: Minikube

Not running it now, but later

```
minikube start --cpus 2 --memory 2048 --disk-size 10g
```

The last parameter is important; it specifies which VM driver to use (see Minikube documentation for details). After the installation is complete, you can get the status of Minikube:

```
$ minikube status
minikubeVM: Running
localkube: Running
kubectl: Correctly Configured: pointing to minikube-vm at 192.168.64.2
```

This means the local Kubernetes cluster is up and running.

- We already saw Minikube in our browser

Running and debugging Camel on Kubernetes

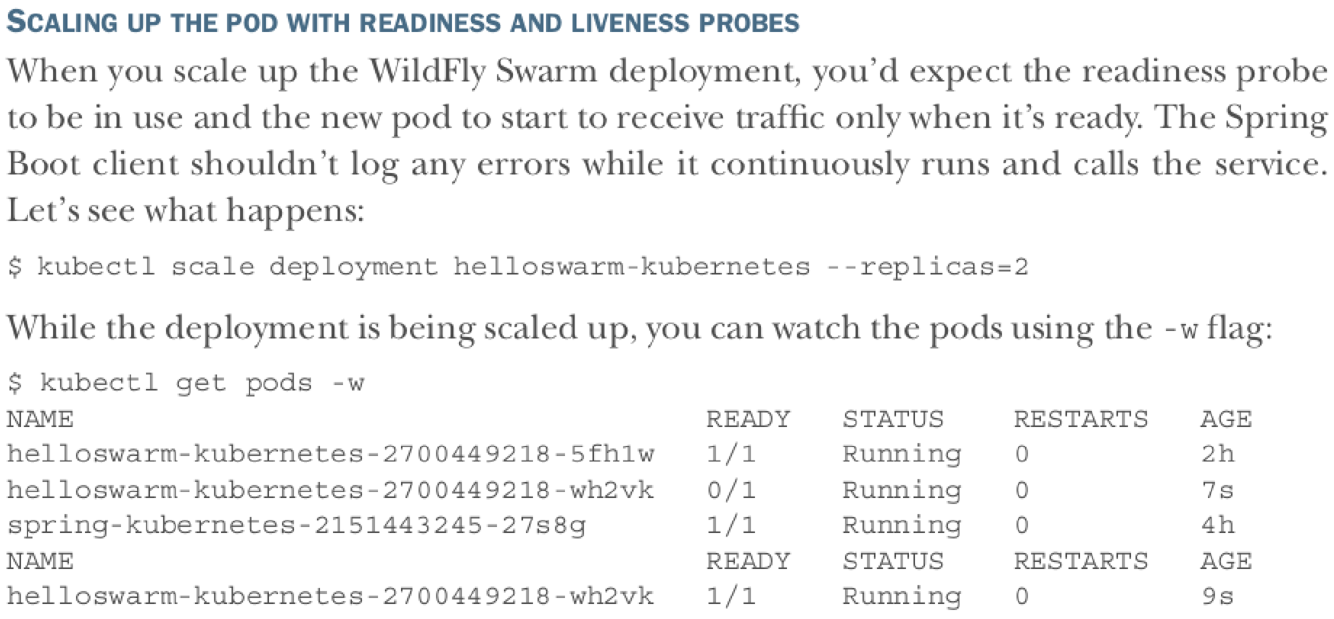
18.3 Running Camel and other applications in Kubernetes When you **run applications on Kubernetes, they run as containers loaded from Docker images.** The information you learned in the previous section about running Camel on Docker is required knowledge for working with Kubernetes.

Understanding essential Kubernetes concepts

```
$ kubectl get deployment -o yaml hello-world
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: 2017-11-04T20:33:36Z
  generation: 1
  labels:
    foo: bar
  name: hello-world
  ...
```

- Example YAML file from Kubernetes

Building resilient microservices on Kubernetes



Scaling up the pod with readiness and liveness probes

When you scale up the WildFly Swarm deployment, you'd expect the readiness probe to be in use and the new pod to start to receive traffic only when it's ready. The Spring Boot client shouldn't log any errors while it continuously runs and calls the service. Let's see what happens:

```
$ kubectl scale deployment helloswarm-kubernetes --replicas=2
```

While the deployment is being scaled up, you can watch the pods using the `-w` flag:

```
$ kubectl get pods -w
NAME                                READY   STATUS    RESTARTS   AGE
helloswarm-kubernetes-2700449218-5fh1w   1/1     Running   0          2h
helloswarm-kubernetes-2700449218-wh2vk   0/1     Running   0          7s
spring-kubernetes-2151443245-27s8g       1/1     Running   0          4h
NAME                                READY   STATUS    RESTARTS   AGE
helloswarm-kubernetes-2700449218-wh2vk   1/1     Running   0          9s
```

- Kubernetes can be told to create more pods/containers
- AND can check if it alive and good

Introducing fabric8, Kubernetes Helm, and OpenShift

Book lists multiple tools that can help making Java applications Kubernetes-ready:

- Docker Maven plugin
- Kubernetes-ready fabric8 Maven plugin
- Kubernetes Helm
- OpenShift

We wont go into detail with these, and check if better tools are available before use

Securing Kubernetes

> Attacking and Defending Kubernetes, with Ian Coldwater
> Ian Coldwater specializes in breaking and hardening Kubernetes, containers, and cloud native infrastructure. A pre-eminent voice in the Kubernetes security community, Ian is currently a Lead Platform Security Engineer at Heroku. Ian joins Adam Glick and Craig Box to talk about the offensive and defensive arts.

`https://www.heroku.com/podcasts/kubernetes-podcast-from-google/attacking-and-defending-kubernetes-with-ian-coldwater`

- Securing Kubernetes can be hard work
-
- follow Ian Coldwater, @IanColdwater `https://twitter.com/iancoldwater`

Helm Database

- Book uses Helm to deploy a database
- Easier than running Postgresql yourself?
- Do you want your database inside Kubernetes? why/why not

Similar thoughts about load balancing

- Do we run everything inside the Kubernetes cluster?
- Do we want/need hardware acceleration for things like load balancing and HTTPS/TLS termination

Part II: Running Kubernetes

- I will start up Kubernetes and demonstrate the setup
- I will act as if this is part of my job, and I am new to Kubernetes

Getting started with Kubernetes: Minikube

Not running it now, but later

```
minikube start --cpus 2 --memory 2048 --disk-size 10g
```

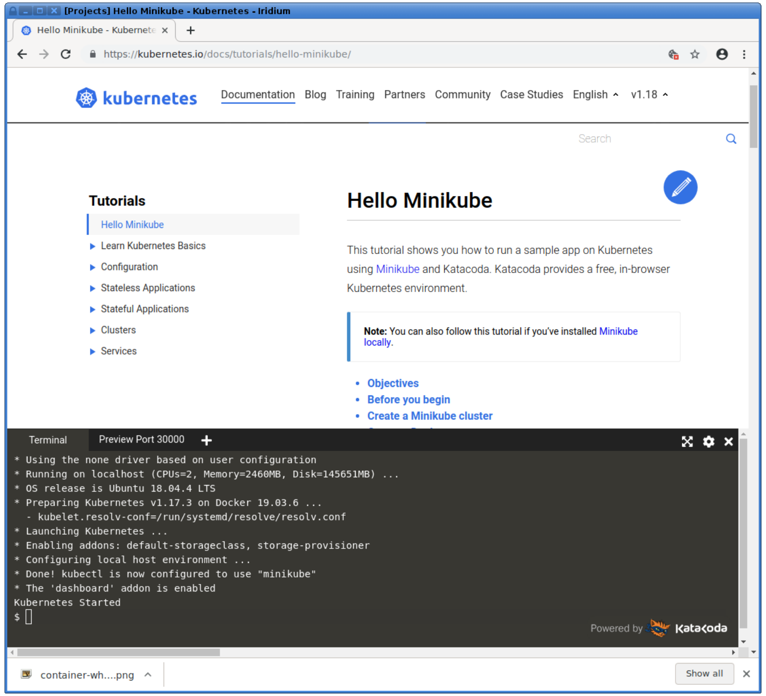
The last parameter is important; it specifies which VM driver to use (see Minikube documentation for details). After the installation is complete, you can get the status of Minikube:

```
$ minikube status
minikubeVM: Running
localkube: Running
kubectl: Correctly Configured: pointing to minikube-vm at 192.168.64.2
```

This means the local Kubernetes cluster is up and running.

- We already saw Minikube in our browser
- I will run a local version of Minukube on my Debian
- We will go through a bit of details with regards to Kubernetes

Exercise: Lets run Kubernetes



Tutorials

- Hello Minikube
- Learn Kubernetes Basics
- Configuration
- Stateless Applications
- Stateful Applications
- Clusters
- Services

Hello Minikube

This tutorial shows you how to run a sample app on Kubernetes using Minikube and Katacoda. Katacoda provides a free, in-browser Kubernetes environment.

Note: You can also follow this tutorial if you've installed Minikube locally.

- Objectives
- Before you begin
- Create a Minikube cluster

```
* Using the none driver based on user configuration
* Running on localhost (CPUs=2, Memory=2460MB, Disk=145651MB) ...
* OS release is Ubuntu 18.04.4 LTS
* Preparing Kubernetes v1.17.3 on Docker 19.03.6 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Launching Kubernetes ...
* Enabling addons: default-storageclass, storage-provisioner
* Configuring local host environment ...
* Done! kubectl is now configured to use "minikube"
* The 'dashboard' addon is enabled
Kubernetes Started
$ 
```

- https://kubernetes.io/docs/tutorials/hello-minikube/

Exercise

Now lets do the exercise

Running Minikube on the web – 45 min

which is number **21** in the exercise PDF.

Exercise

Now lets do the exercise

Nginx in Kubernetes load balancing – 30 min

which is number **22** in the exercise PDF.

Exercise

Now lets do the exercise

PostgreSQL in Kubernetes – 30 min

which is number **23** in the exercise PDF.