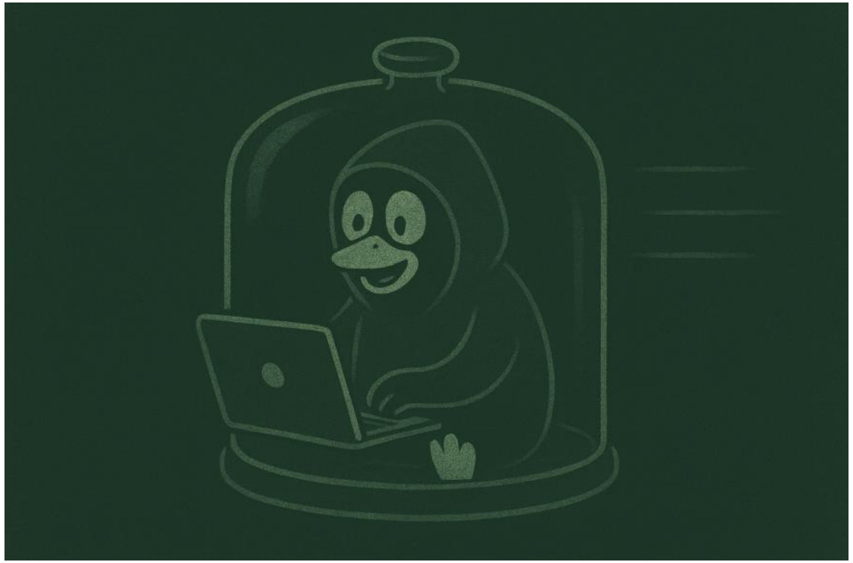
Operating System Fundamentals

Containers with Docker

Karel de Grote Hogeschool



Content

Course

- See PDF

Karel de Grote Hogeschool

Images & Containers

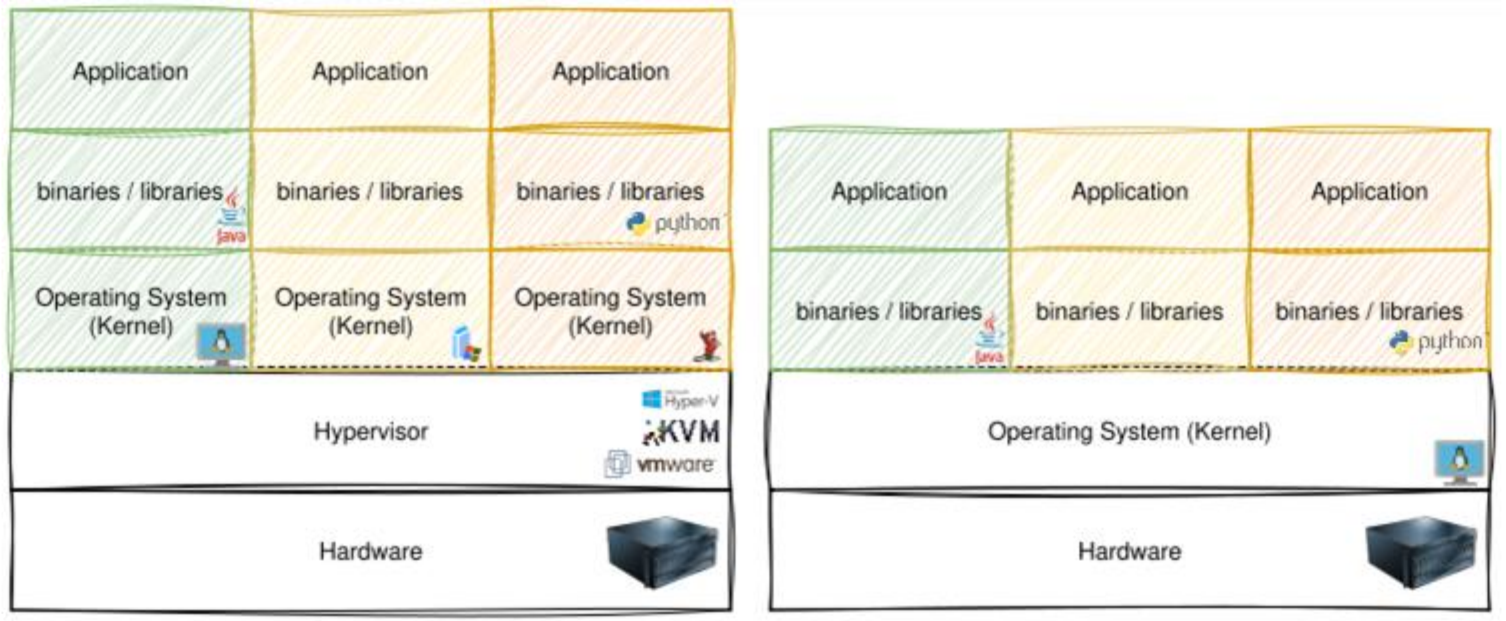
Karel de Grote Hogeschool

Images & Containers

- Package software in an "image"
  - 1 application with all necessary components
  - Download image from "registry"
- Run this image → "**container**"
  - Program is "locked" in container
  - Access to the rest of the system is "shielded"
    - File system isolation: only access to own files
    - Network isolation: only access to own network
    - Process isolation: only sees its own programs

Karel de Grote Hogeschool

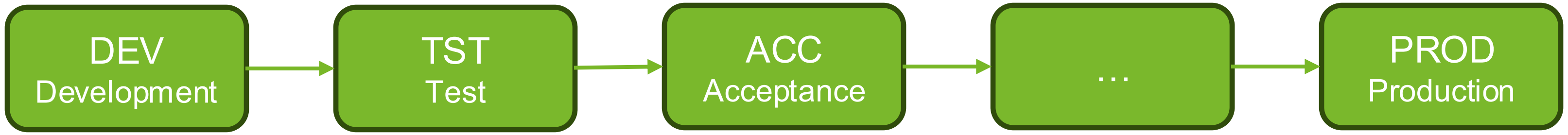
Containers vs. Virtual Machines



Karel de Grote Hogeschool

Images & Containers

- Image: package software with all its dependencies
  - “Everything is in the box”
- Container: run image with its program(s)
  - In a closed part of the OS
  - All necessary software is included in the image
  - No need to install anything extra
- Major advantage: same image in every environment



Docker

dot Sc
dotClo
"The automation for carrying coffee across the world is better and more reliable than the kind of tools we use to ship software between computers. That's pretty embarrassing."
Solomon Hykes
dotScale 2013
Watch this talk and more at http://dotscale.eu

https://youtu.be/3N3n9FzebAA?feature=shared&t=50

0:50 → 11:50

Karel de Grote Hogeschool

Docker

- Container technology has been around for much longer
  - Jails, Solaris Zones, etc.
  - Linux namespaces & cgroups
- Docker made technology "accessible"
  - Easy to use

    (Now there are alternatives/competitors, e.g., podman)
- All necessary software neatly packaged in one "image"
  - Run as a container
  - Ensures that exactly the same software and configuration is used everywhere

Karel de Grote Hogeschool

Docker Desktop

2

Karel de Grote Hogeschool

Installation

- See https://www.docker.com/products/docker-desktop/
  - Docker Desktop for "simplicity"
  - Docker command line in 2nd year

  See course

- But containers weren't virtual machines, were they?
  - On Windows: using a virtual machine on WSL
  - On Mac: also uses light-weight VM
  - On Linux: "native"
    - Use docker command line (docker-ce)
    - No Docker Desktop

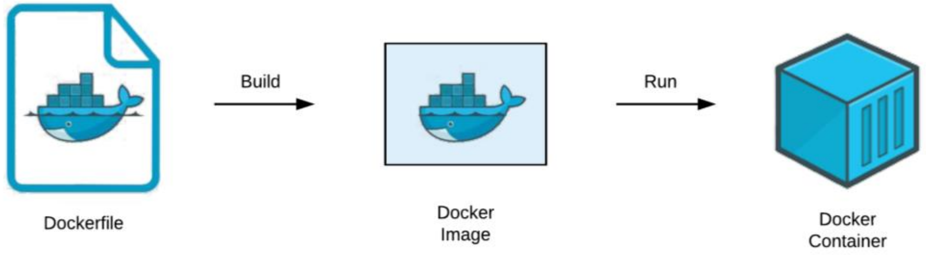
Images

Karel de Grote Hogeschool

Image

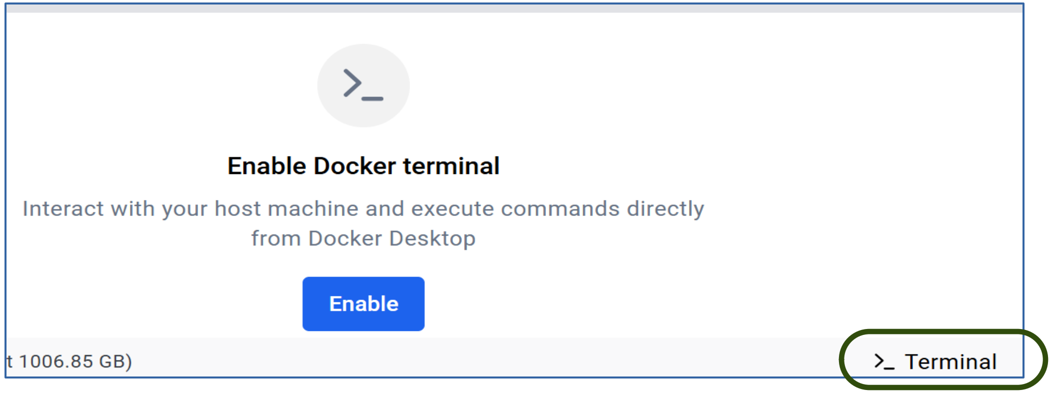
- Docker Hub: countless images available
- Download image: `docker pull image`

- Want to create your own image? Next year
  - With Dockerfile



Exercise

- Start terminal: button in the right corner



Interact with your host machine and execute commands directly from Docker Desktop

- Check if everything works: `docker info`

Exercise

- `docker pull hello-world`
- `docker run hello-world`

- `docker run rancher/cowsay "Hello"`
  - Image automatically downloaded
- `docker run rancher/cowsay "Hello class"`
  - Image no longer downloaded, cached
- `docker run rancher/cowsay "Bye"`

Exercise

- `docker container ls –a`
  - All containers that have been terminated are still visible
- `docker images`
  - All downloaded and cached images

Karel de Grote Hogeschool

4

Registries

Karel de Grote Hogeschool

Registries

- Images are hosted in registries
- Docker's default registry is Docker Hub
    - docker run docker.io/hello-world = docker run hello-world
    - Images from Docker Hub: docker.io is optional

      (FYI: docker.io is short for registry.hub.docker.com)

- Image from another registry?
    - E.g. docker pull **docker.elastic.co/**elasticsearch/elasticsearch

More image registries

Quay.io (Red Hat/IBM)

GitLab Container Registry

GitHub Container Registry (GHCR)

JFrog Artifactory

Amazon Elastic Container (ECR)

Google Artifact Registry

Azure Container Registry (ACR)

Oracle Cloud Infrastructure Registry (OCIR)

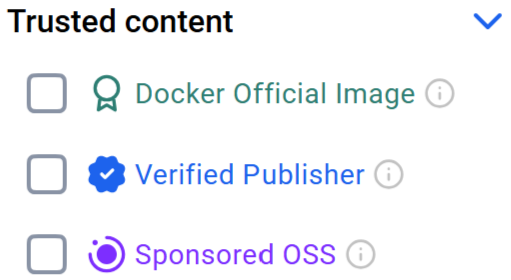
Alibaba Cloud Container Registry

…

Registries

- Images also have versions, "tags"
  - docker.io/mongo:**7.0.26**
  - docker.io/postgresql:**18**
  - docker.io/php:**8**
- Images can have multiple tags
- If you do not specify anything, the default tag ":latest" is used

- Security!
  - Only use "trusted" images
  - From well-known companies


Trusted content

- [ ] Docker Official Image
- [ ] Verified Publisher
- [ ] Sponsored OSS

Interactive containers

- Interactive linux containers:
  - `docker run -it --rm almalinux:9 /bin/bash`
  - `docker run -it --rm debian:13 /bin/bash`
  - `docker run -it --rm alpine /bin/ash`

Linux on Windows
Linux on Mac
Linux distro on other distro

- Options
  - `-i` interactive
  - `-t` terminal
  - `--rm`: remove container when finished
  - `/bin/(b)ash`: command to start within container
    - Often optional
    - Try `docker run --it --rm <image> echo Hello world`

Karel de Grote Hogeschool

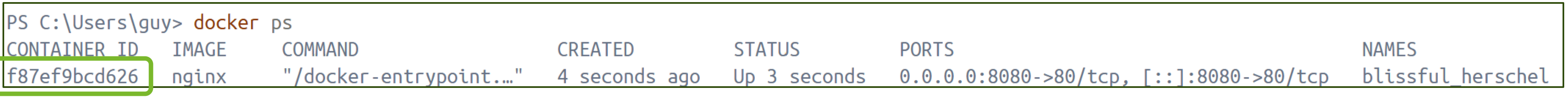
Server applications

5

Karel de Grote Hogeschool

Server programs

- Server applications: container in the background
  - Daemons
  - `docker run --rm` **`-d`** `<image>` → detached
- Stop+remove container
  - `docker ps`

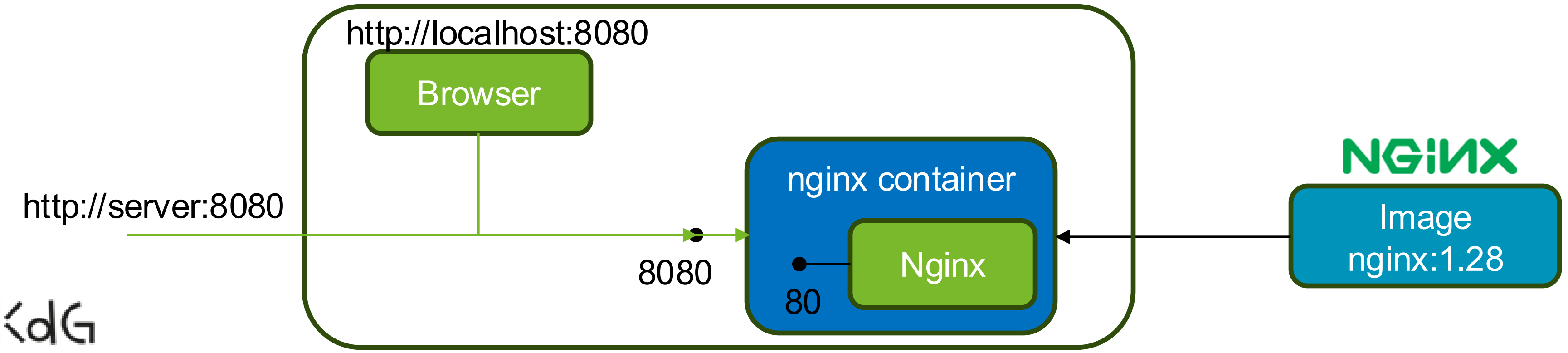


```
PS C:\Users\guy> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                                   NAMES
f87ef9bcd626   nginx     "/docker-entrypoint.…"   4 seconds ago   Up 3 seconds   0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   blissful_herschel
```

  - `docker stop` **`f87…`**
  - `docker rm` **`f87…`**

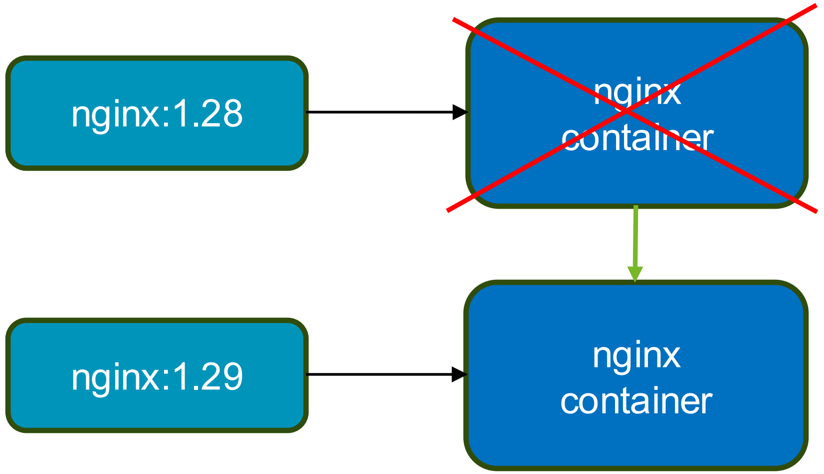
Port mapping

- Application in container listens on port
  - "Map" this port to port on host machine
  - `docker run --rm -d -p 8080:80 nginx:latest`
  - Go to http://localhost:8080 on your laptop
    - Port 80 in the container, port 8080 on your laptop



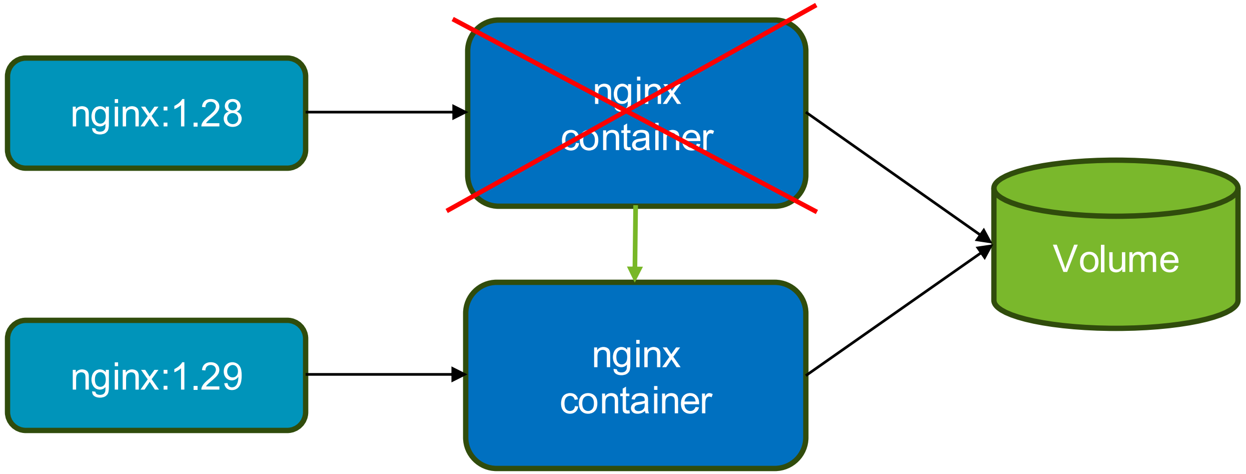
Replace instead of update

- Never update or upgrade containers
- Download a more recent image, "cattle vs. pets"



Volumes

- Upgrade application using new image
- Data remains stored outside the container in **volume**
    - Volume on host machine



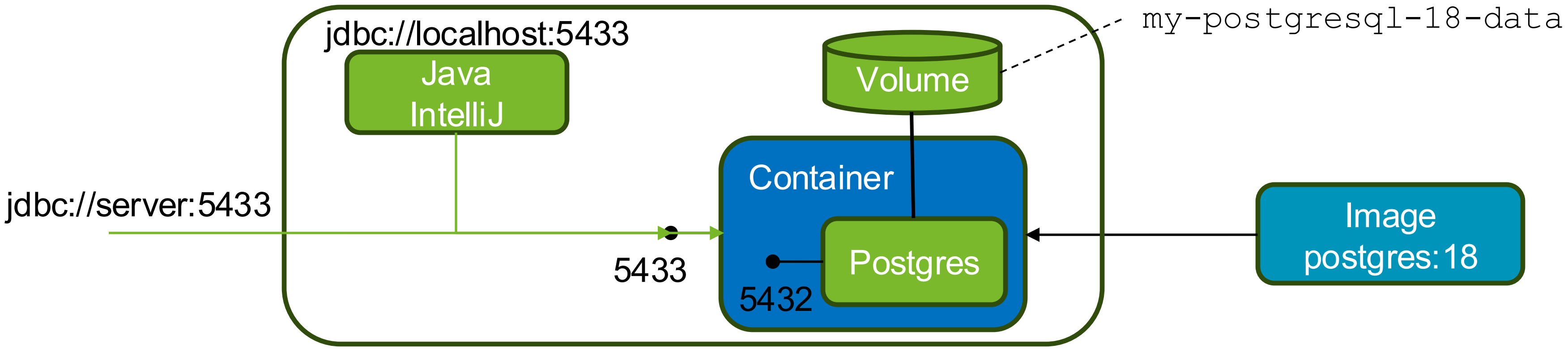
Exercise

- Run a container with PostgreSQL 18
- Ensure that the database is accessible via port 543**3**

- Additional, see course:
  - Use volume my-postgresql-18-data (-v option)
  - Configure the environment variable POSTGRES_PASSWORD with the value "supersecret" (-e option)

Exercise

- Postgres in container listens on port 543**3**
  - To avoid any conflicts with other postgres maybe listening on 5432



Containers everywhere

6

Karel de Grote Hogeschool

Images & containers everywhere

- De facto standard for software delivery
- Image formats & software *standardized*

- Kubernetes (*standardized*)
  - Many containers spread across multiple/many servers
  - "Container orchestration," invented by Google

- Containers in the cloud (proprietary)
  - AWS Fargate, Google Cloud Run, Azure Container Apps, etc.
    - Serverless = underlying infrastructure or OS completely hidden

Karel de Grote Hogeschool