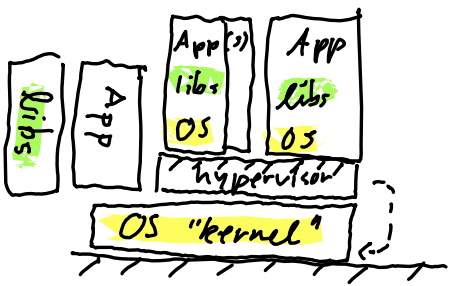
Docker and Træfik

Docker & træfik

1. Containers and their alternatives
2. What containers are made of
3. Docker Compose
4. Ports, Networks Routing
5. Traefik
6. Demo time

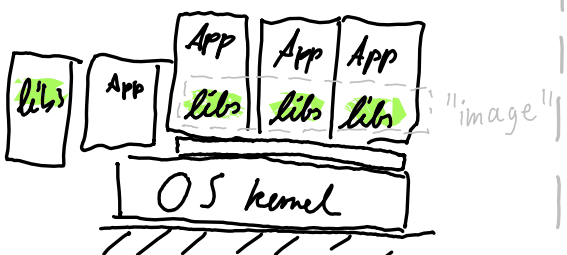
Virtualization

a) Virtual Machines



- uses more resources, boot time
- network access configuration
+ can use kernel features
(+) no need for compatible OS kernel
(usecase: hosted server)

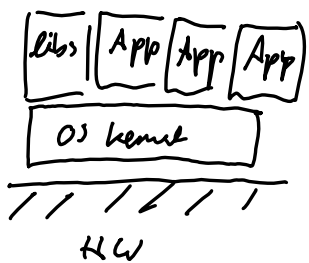
b) Container



-/+ encapsulation

*+: podman lets you run containers as unprivileged user

c) direct install

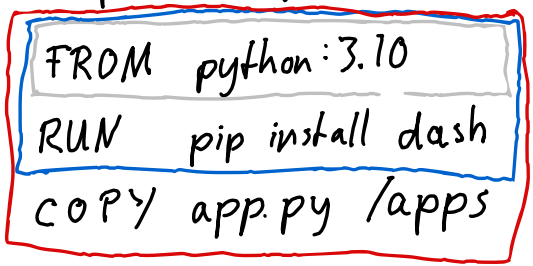


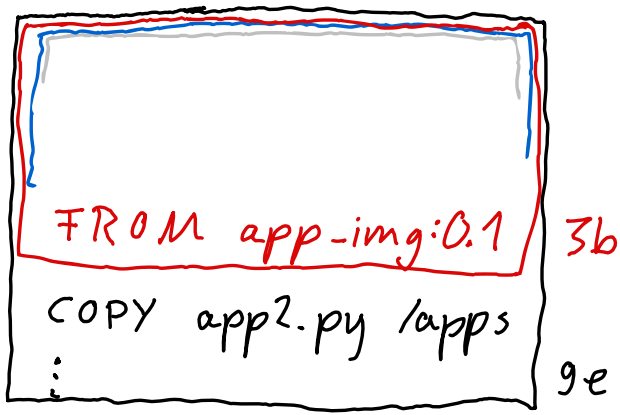
Security: - run apps as unprivileged users
- use firewalls for internet access

Images & Containers

Images

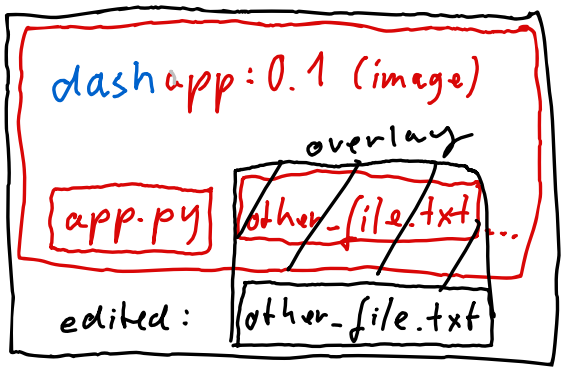
Example Dockerfile:





↖ persistant / static

Containers



← non-persistant

- How to save changing data? By mount..., volumes

Example: Dashboard app

possible aproaches:

1. $ docker run -it python:3.10 /bin/bash
   container $ pip install dash
   $ vim app.py #...write app
   $ python app.py

2. $ docker build -f Dockerfile --tag dashapp:0.1
   $ docker run -d --rm --port 80:8000 dash

Dockerfile

```
FROM python:3.10
RUN pip install dash
COPY ./app.py:app.py.
CMD python3 /app.py
```

3. $ docker-compose -up

docker-compose.yml

```
services:
  dashboard:
    image:
  volume
  network
```

how to draw an owl

1.

2.

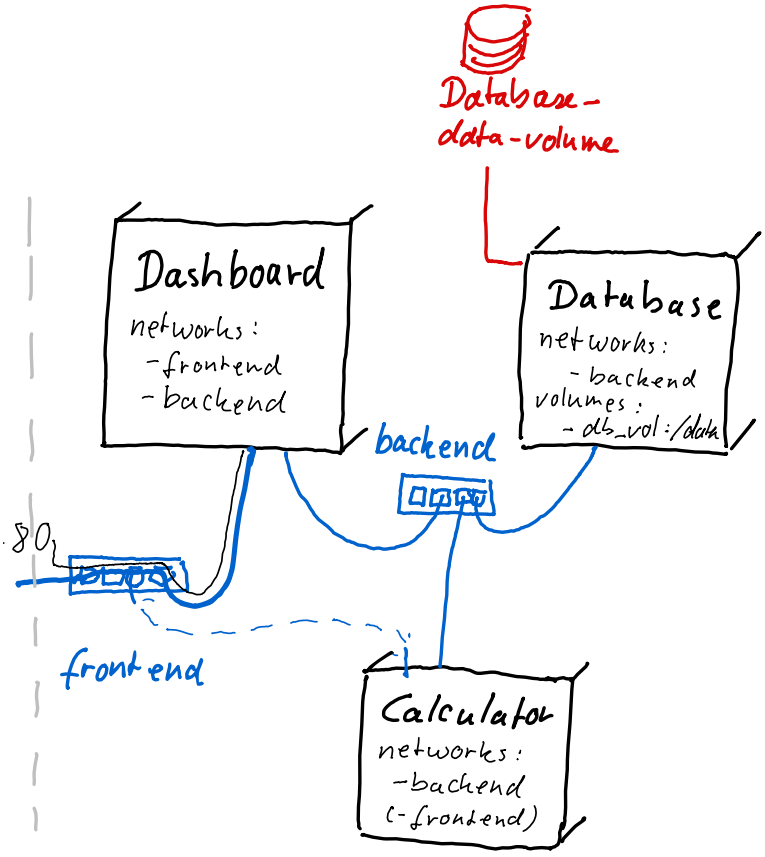
1. Draw some circles
2. Draw the rest of the fucking owl

Docker compose

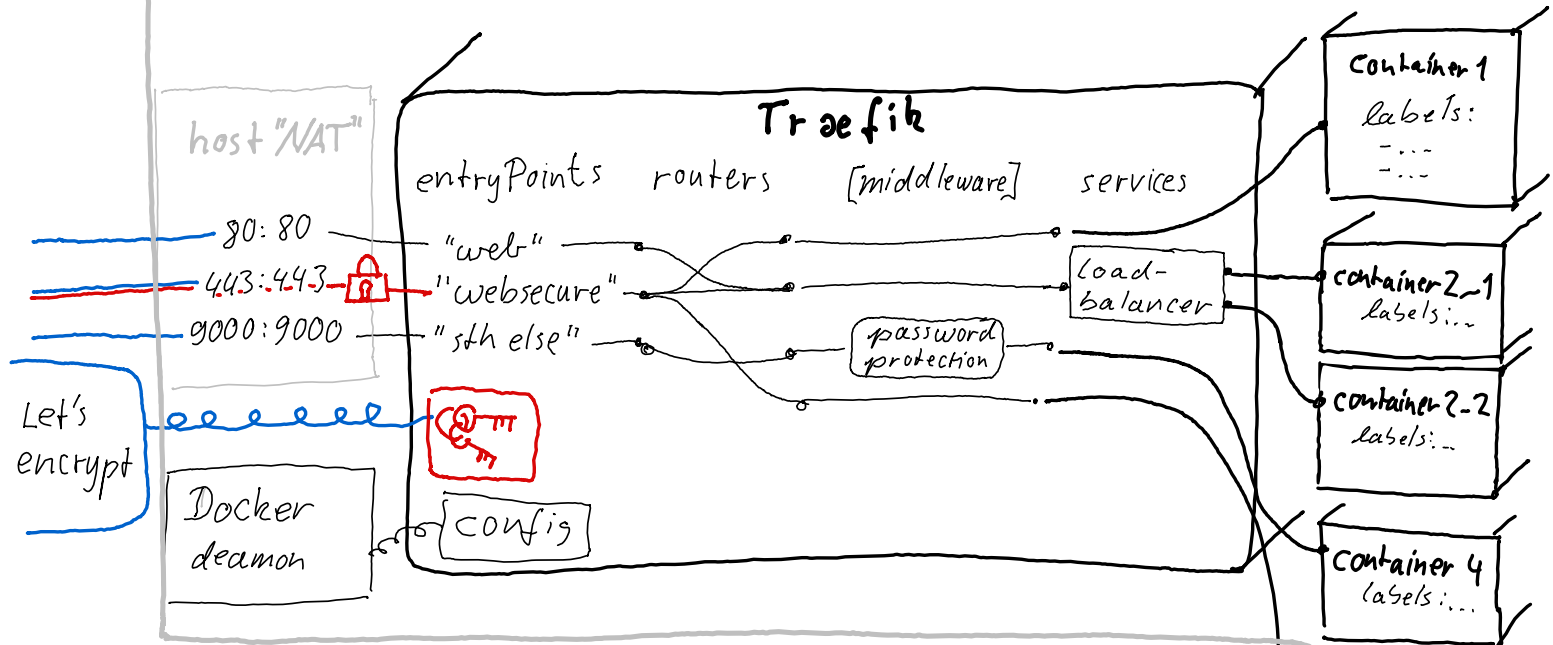
```yaml
services:
  db:
    image: postgres:14.1-alpine
    environment:
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=postgrestopsecretzzz
    ports:
      - '5432:5432'
    volumes:
      - db:/var/lib/postgresql/data
  calculator:
    build: ./calculator
    [...]
  viz:
    build: ./graph-dashboard
    volumes:
      - ./graph-dashboard/source_files/:/sources
    ports:
      - '8200:5000'
    command: python3 /sources/run_dash.py
    labels:
      - "traefik.http.routers.vizrouter.rule=
           Host(`tr2.sidechannel.de`)
           && PathPrefix(`/dash/`)"
      - "traefik.http.routers.vizrouter=dash-stripprefix"
      - "traefik.http.middlewares.dash-stripprefix
           .stripprefix.prefixes=/dash"

volumes:
  db:
    driver: local

networks:
  frontend:
  backend:
```



→ networks and ports, domain names




```yaml
entryPoints:
  web:
    address: ":80"
  other:
    address: ":9080"
```

```yaml
http:
  routers:
    router0:
      entryPoints:
        - "web"
      rule: "Host(`tr.sidechannel.de`)"
      service: "myservice0"
  services:
    myservice0:
      loadBalancer:
        servers:
          - url: "http://172.17.0.1:8000/"
          - url: "http://172.17.0.1:8001/"
```

```yaml
labels:
  - "traefik.http.routers.vizrouter.rule= Host(`tr2.sidechannel.de`) && PathPrefix(`/dash/`)"
  - "traefik.http.routers.vizrouter=dash-stripprefix"
  - "traefik.http.middlewares.dash-stripprefix.stripprefix.prefixes=/dash"
```

Secure communication over the internet

- http : not encrypted
- https : encrypted communication between browser and server
  - Encryption : privacy
    - self-signed certificate sufficient
  - Certificate : authentication of server
    "Let's encrypt" issues server certificates

Example : sexy.example.com / beautiful_shoes?sort_by=price

visible* | encrypted

- ssh : encrypted + key exchange (fingerprint) for authentication
- other (tcp / udp) protocols : depends

easy solution : ssh-tunneling
1. ssh -L localhost:5432:localhost:5433 -N destination.de
2. connect to DB at localhost:5432


WHERE DO BOATS GO WHEN THEY FEEL SICK?
TO THE DOCK.
imgflip.com