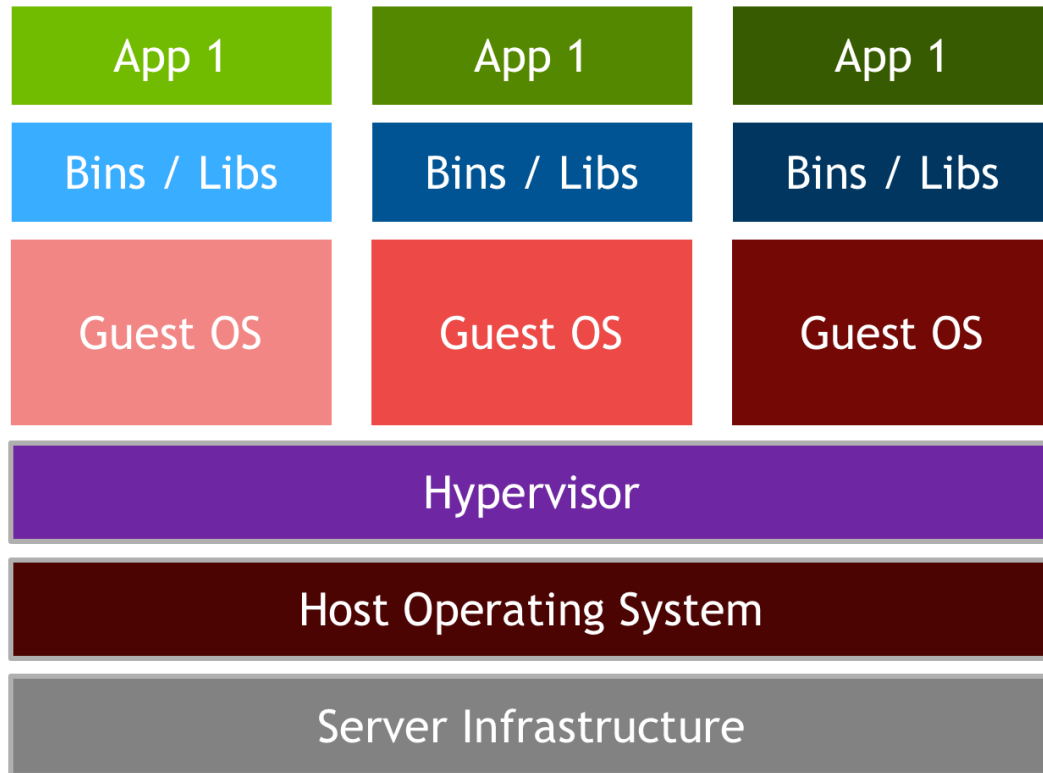


Docker Einführung

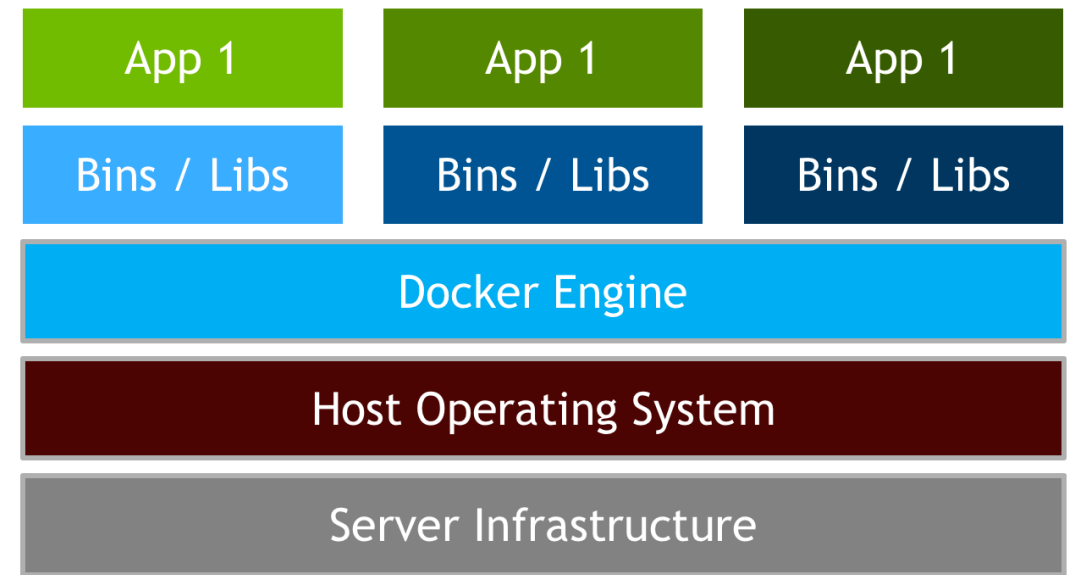
Compass Security
ivan.buetler@compass-security.com

Docker Eco System

Virtual Machines versus Containers

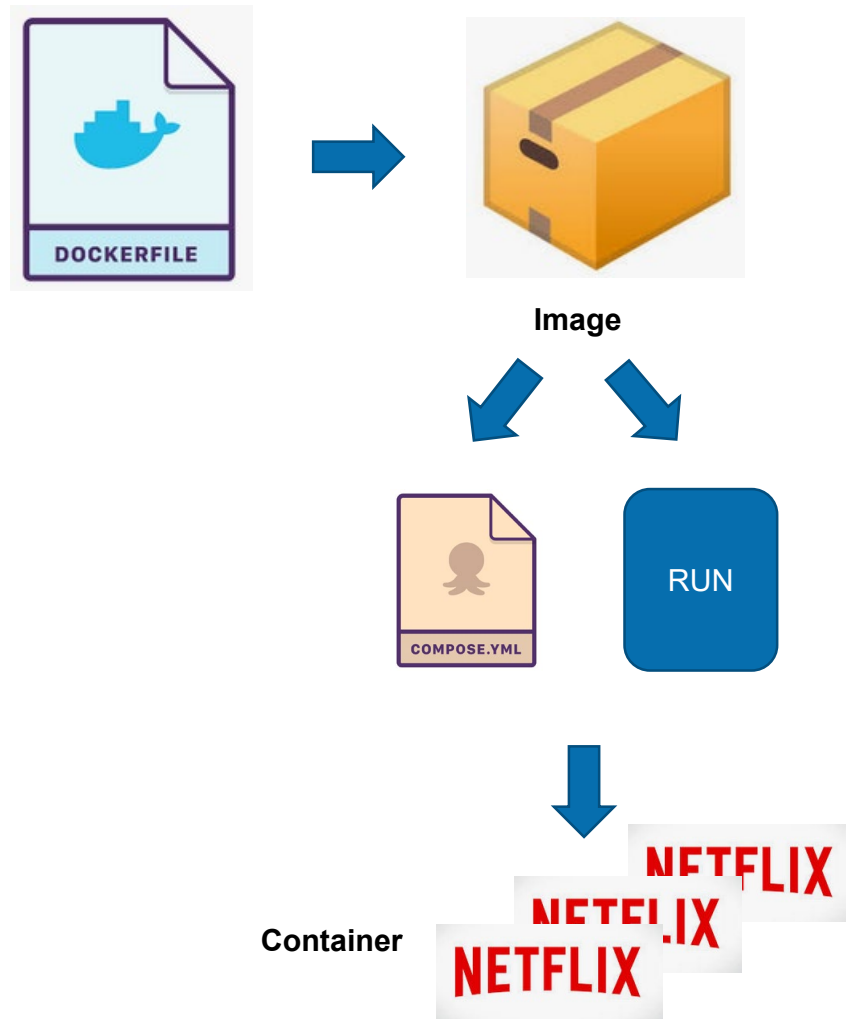


VIRTUAL MACHINES



CONTAINERS

Dockerfile => Docker Image => Docker Container



Entwickler erstellt ein Dockerfile (oftmals Dockerfile und einige dazugehörige Dateien)

- `docker build -t hackinglab/alpine-gotty-root`

Mit «docker build» wird aus einem Dockerfile ein Image

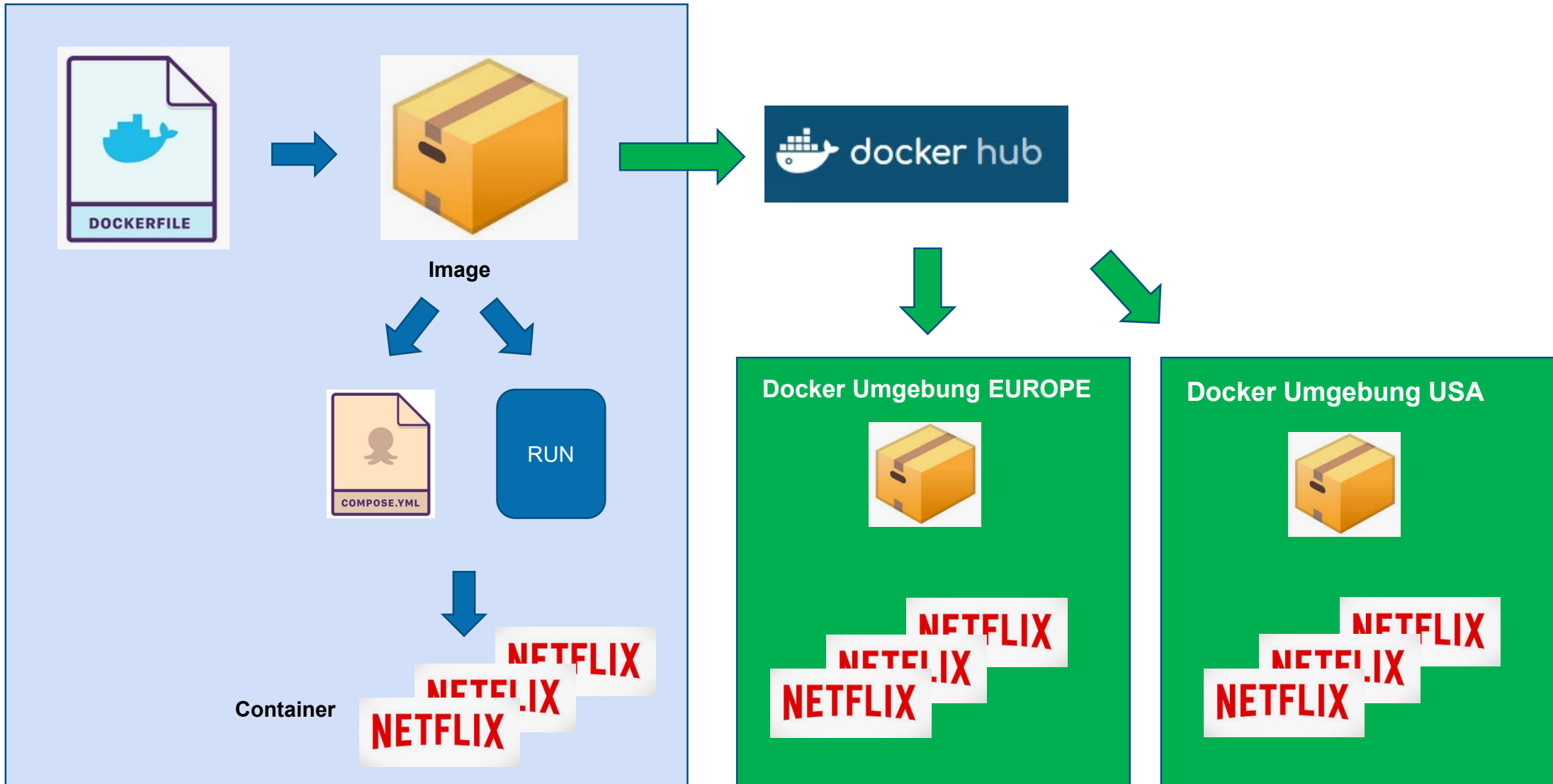
Ein Image das «gebootet» wird, nennt man Container

Es gibt viele Wege um eine Image zu booten

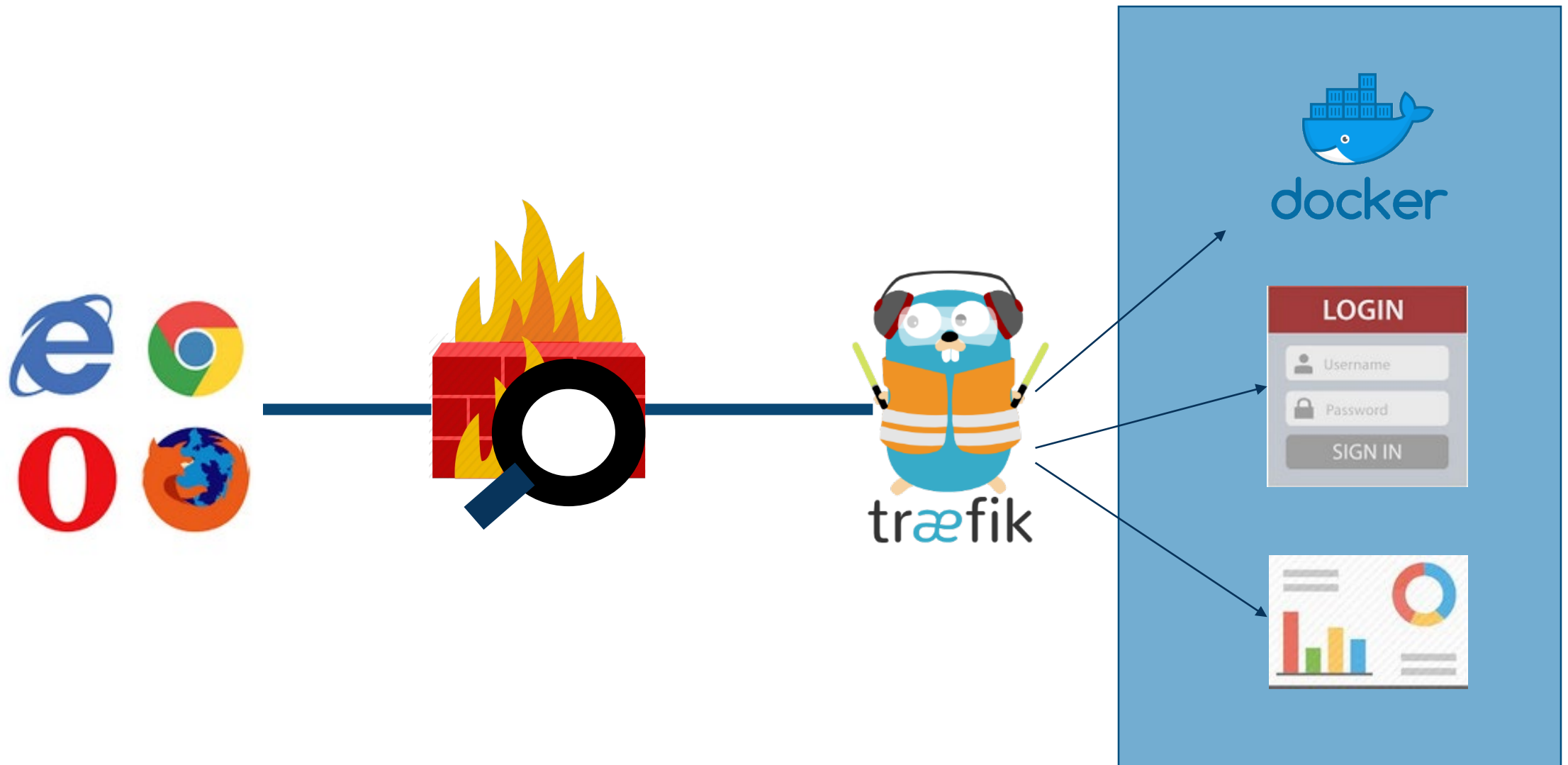
Die einfachsten 2 Varianten für das «booten»

- `docker run hackinglab/alpine-gotty-root`
- `docker-compose up -d`

Dockerfile => Docker Image => Docker Container

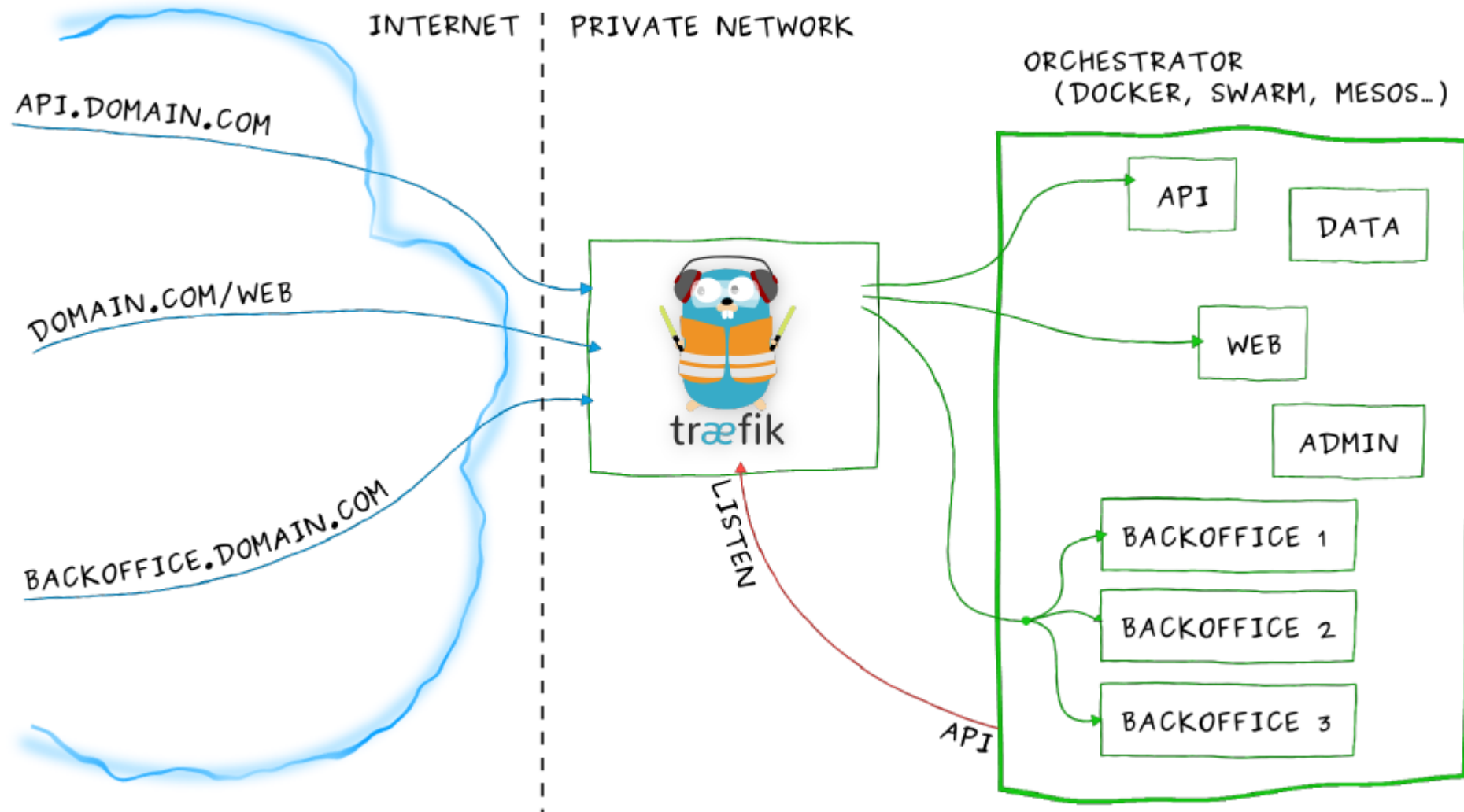


OnDemand Docker Services



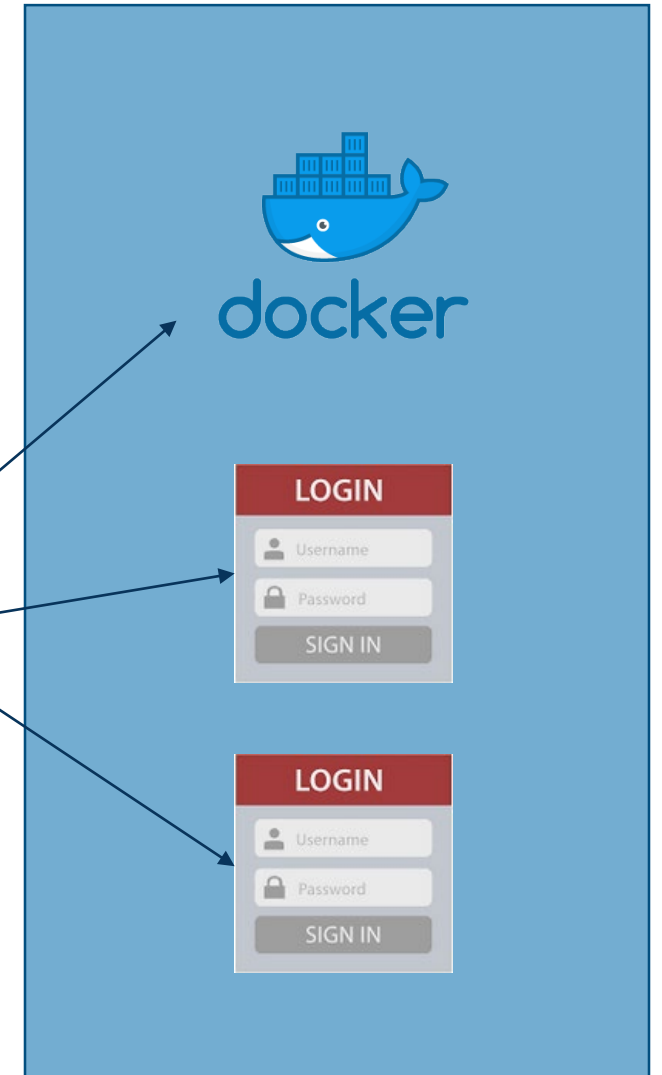
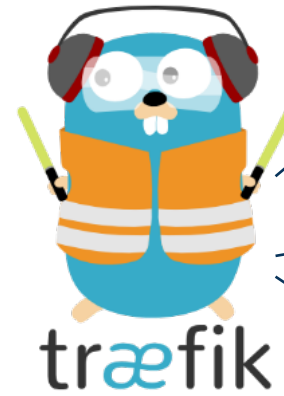
Traefik Load Balancer

Microservice Architecture



Hacking-Lab Test Suite

Wildcard SSL Certificate *.idocker.hacking-lab.com
New Routes without restarting the service



Traefik Status Page - DOCKER

The screenshot shows the Traefik Status Page in a web browser. The browser's address bar shows the URL `idocker.hacking-lab.com:8080/dashboard/`. The page header includes the Traefik logo, navigation links for 'PROVIDERS' and 'HEALTH', and version information 'V1.7.4 / MAROILLES' and 'DOCUMENTATION'.

The main content area is divided into two columns: 'FRONTENDS' and 'BACKENDS'.

FRONTENDS

- frontend-Host-auth-idocker-hacking-lab-com-7**
 - Route Rule: `Host:auth.idocker.hacking-lab.com`
 - Entry Points: `http`, `https`
 - Backend: `backend-keycloak-authidockerhackinglabcom`
- frontend-Host-devgit-idocker-hacking-lab-com-8**
 - Route Rule: `Host:devgit.idocker.hacking-lab.com`
 - Entry Points: `http`, `https`
 - Backend: `backend-devgit-devgitidockerhackinglabcom`

BACKENDS

- backend-devgit-devgitidockerhackinglabcom**
 - Server: `http://172.26.0.5:80`
 - Weight: 1
- backend-globalgit-globalgitidockerhackinglabcom**
 - Server: `http://172.29.0.2:80`
 - Weight: 1
- backend-keycloak-authidockerhackinglabcom**
 - Server: (empty)
 - Weight: (empty)

Traefik Status Page - FILE

The screenshot shows the Traefik status page in a web browser. The browser's address bar shows the URL `idocker.hacking-lab.com:8080/dashboard/`. The page has a dark blue header with the Traefik logo, navigation links for **PROVIDERS** and **HEALTH**, and version information **V1.7.4 / MAROILLES** and **DOCUMENTATION**.

Below the header is a search bar with the placeholder text "Filter by name or id ...". Below the search bar are two tabs: **docker** and **file**, with **file** being the active tab.

The main content area is divided into two columns. The left column is titled **3 FRONTENDS** and contains two entries: **frontend1** and **frontend2**. The right column is titled **2 BACKENDS** and contains two entries: **backend1** and **backend2**.

Frontend1 Configuration:

- Route Rule:**
 - Host: `api.idocker.hacking-lab.com`
 - PathPrefix: `/api`
- Entry Points:** `http` and `https`
- Backend:** `backend1`

Frontend2 Configuration:

- Route Rule:**
 - Host: `idocker.hacking-lab.com`
 - PathPrefix: `/`
- Entry Points:** `http` and `https`

Backend1 Configuration:

Server	Weight
<code>http://localhost:81</code>	10

Backend2 Configuration:

Server	Weight
<code>http://localhost:82</code>	10

Conclusion

Paradigm shift into the direction of “Microservice Architectures”

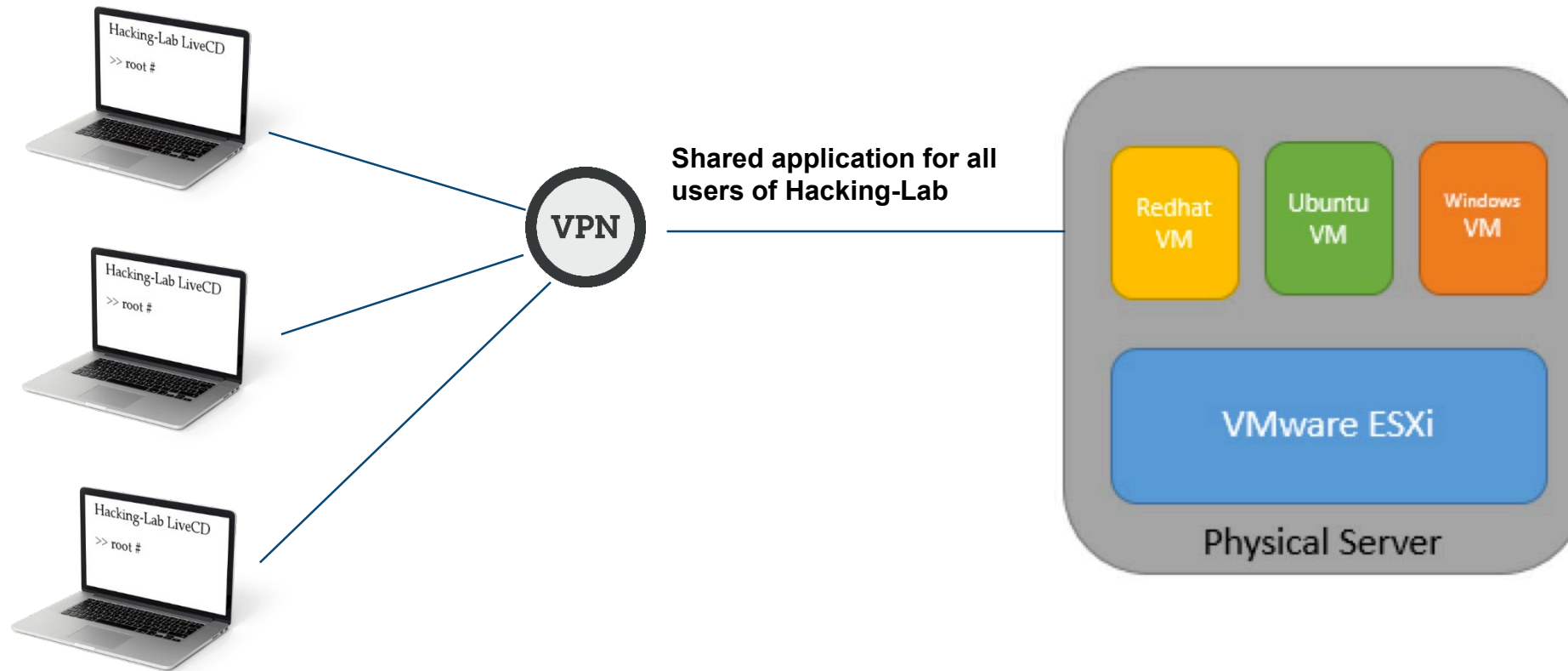
Small apps together shape the big application

We will face more and more RESTful service architectures

Docker & Hacking-Lab 1.0

Appendix

How we deployed vulnerable services in the past (and still do...)



How we deploy vulnerable services today

