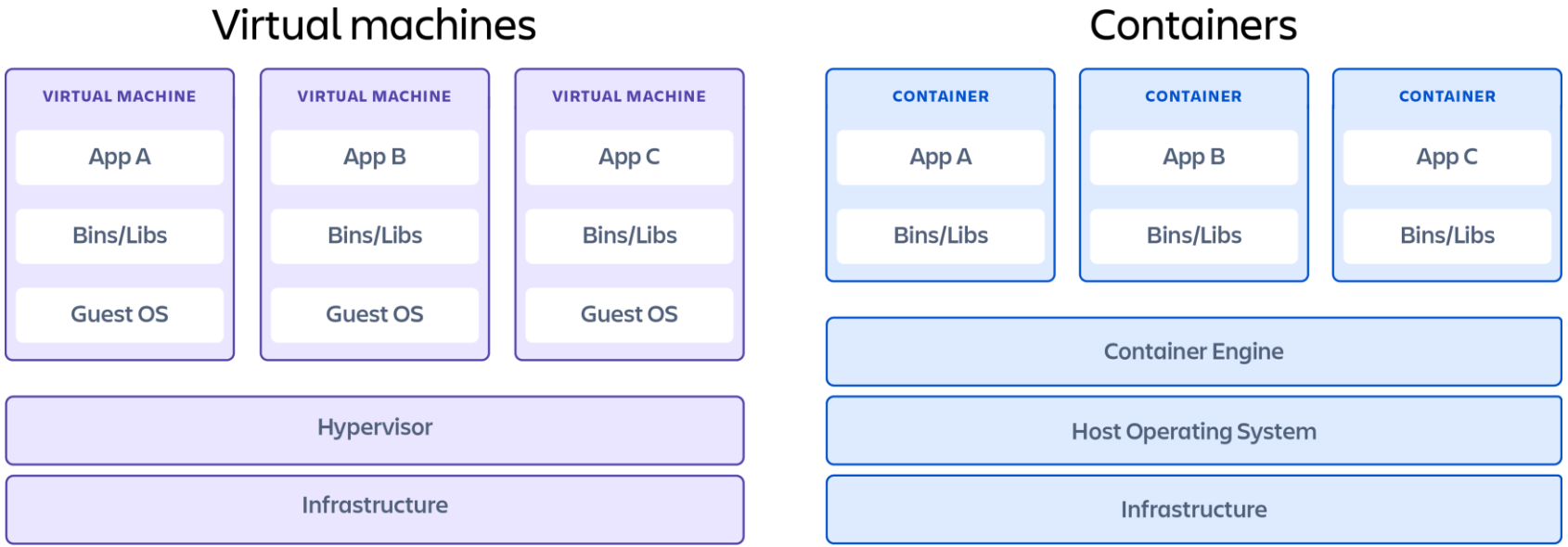
DevOps

Containers

AP HOGESCHOOL ANTWERPEN

AP.BE

Containers vs VM's



Containers

- Pro:
  - Ontwikkelsnelheid
  - Portable
  - Makelijk te automatiseren
  - Robuust
- Con:
  - Host compromised == alles compromised

Containers

- Providers:
  - Docker
  - RKT
  - LXC (Linux Containers)
  - CRI-O

Benodigde software

- Docker https://www.docker.com/get-started
  - Kan lastig zijn op Windows
  - Windows subsystem for Linux of VM aan te raden
  - M1 mac's kunnen problemen geven

  --> wij werken op een Linux machine

Vereisten Linux VM

- Bereikbaar via SSH vanop je host
- Bij voorkeur geen GUI
- Bij voorkeur Debian
- 1G ram + 1vCPU is voldoende (it's Linux after all)
- Installeer de docker engine: https://docs.docker.com/engine/install/debian/

Docker

The original

Belangrijke termen

- **Image** = een package dat alles bevat om een applicatie te draaien. (code, runtime, libraries, config, variabelen, etc.)
- **Container** = een draaiende versie van een image.

Waar vind je docker-images

- Docker-hub https://hub.docker.com/ of `docker search`
- Bij verschillende leveranciers van software in de repo
- Zelfbouw met een dockerfile (volgend labo)

De docker lifecycle

Code(Dockerfile) ==> image ==> container(s)

`docker build` `docker run`

Docker commands

`docker search <zoekterm>`

Doorzoekt docker hub voor docker images die binnen de zoekterm vallen.

Docker commands

`docker pull <imgage>`

Haalt een image binnen met de gegeven naam van docker hub.

Docker commands

`docker run <imgage>`

Start een container op basis van de gegeven image.

Opties:

`-d detatch` draait de container in de achtergrond

`-p <externepoort>:<internepoort>` poort forwarding naar host

`--name <naam>` geeft de container een leesbare naam

`-v <src>:<dst>` mapt een directory op de host naar een plek in de container

`--link` zorgt ervoor dat alles beschikbaar is vanop een andere container

`-e <var>` geeft een environment variabele mee met de container

Docker commands

`docker ps`

Toont alle draaiende containers

`-a` toont alle containers

Docker commands

```
docker images
```

Toont alle locale images

Docker commands

```
docker stop <container>
docker start <container>
docker restart <container>
```

Containers beheren.

Docker commands

`docker rm <container>`

Verwijdert een container.

Docker commands

`docker rmi <image>`

Verwijdert een image.

Docker commands

`docker inspect <container>`

Geeft de configuratie van een container terug als een JSON object.

Docker commands

```
docker exec –it <container> /bin/bash
```

Opent een bash shell in de container.

Verlaat de shell met CTRL-P of CTRL-Q of exit

Docker commands

`docker logs <container>`

Geeft de logs van een container weer.

Opties:

`-f` follow, volgt de logs live in de terminal

`--tail` toont enkel de laatste logs

Docker commands

```
docker commit <container> <image>:<versie>
```

Slaat een container op als image.

Docker commands

`docker save <image> > <bestandsnaam>.tar`

Slaat een image op als een tar bestand.

(zie lessen Linux voor uitleg tar)

Opdracht

It's go time

AP HOGESCHOOL ANTWERPEN

AP.BE

Opdracht docker basic

1. Download de officiele nginx docker image van docker hub
2. Maak er een container van en zorg dat poort 3000 op de host geforward word naar poort 80 op de container.
3. Open een shell in de container
4. Installeer een editor in de container (vim,nano,emacs,…)
5. Pas de html in **/usr/share/nginx/html** aan zodat er een h1 tag met je naam staat.
6. Kijk na of je de website kan zien op **http://<vm-ip>:3000**
7. Sla deze container op als een image met als naam je voornaam
8. Sla deze image op als <voornaam>.tar en stuur dit in