Introduction to Docker

RES, Lecture 3

Olivier Liechti

heig-vd

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD

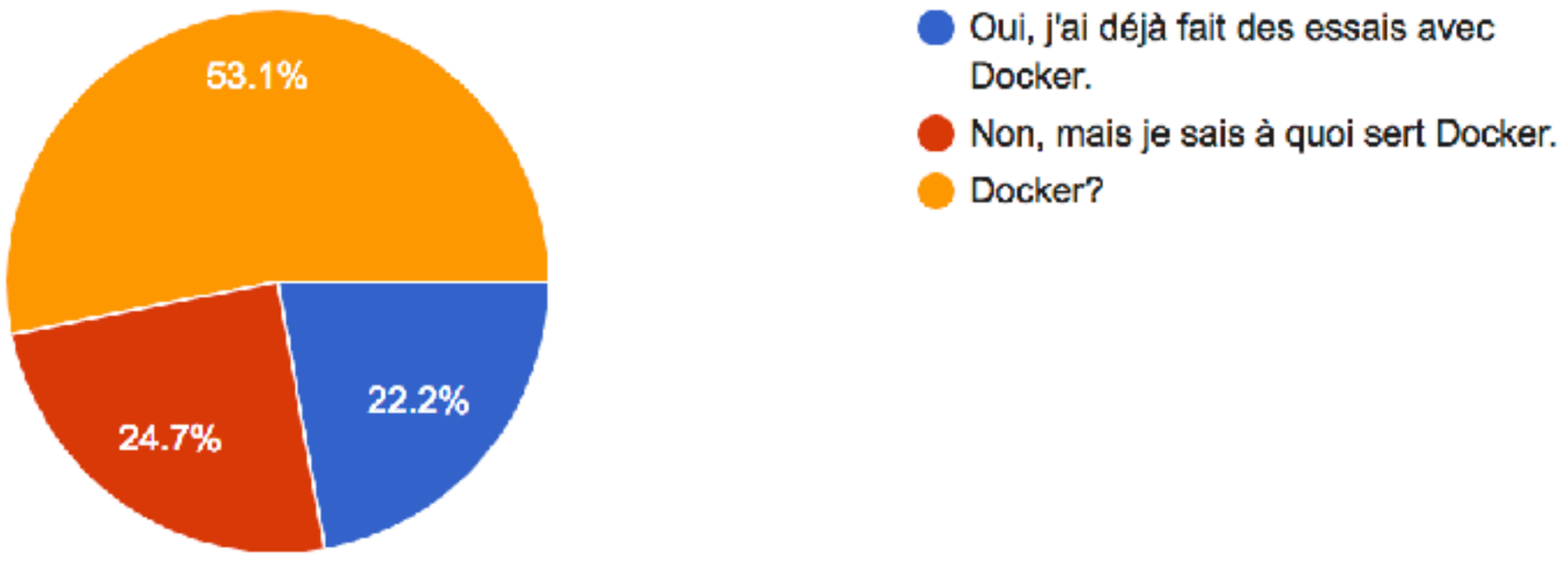
www.heig-vd.ch

heig-vd
HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Introduction to Docker

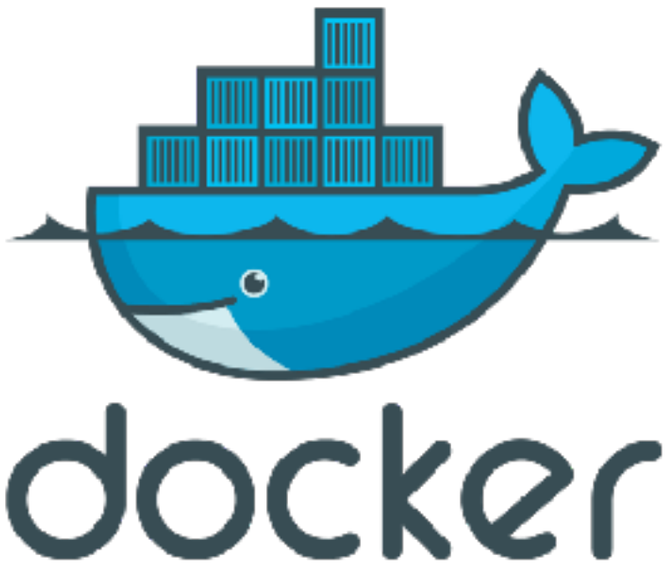
Est-ce que vous connaissez et avez utilisé la technologie Docker?

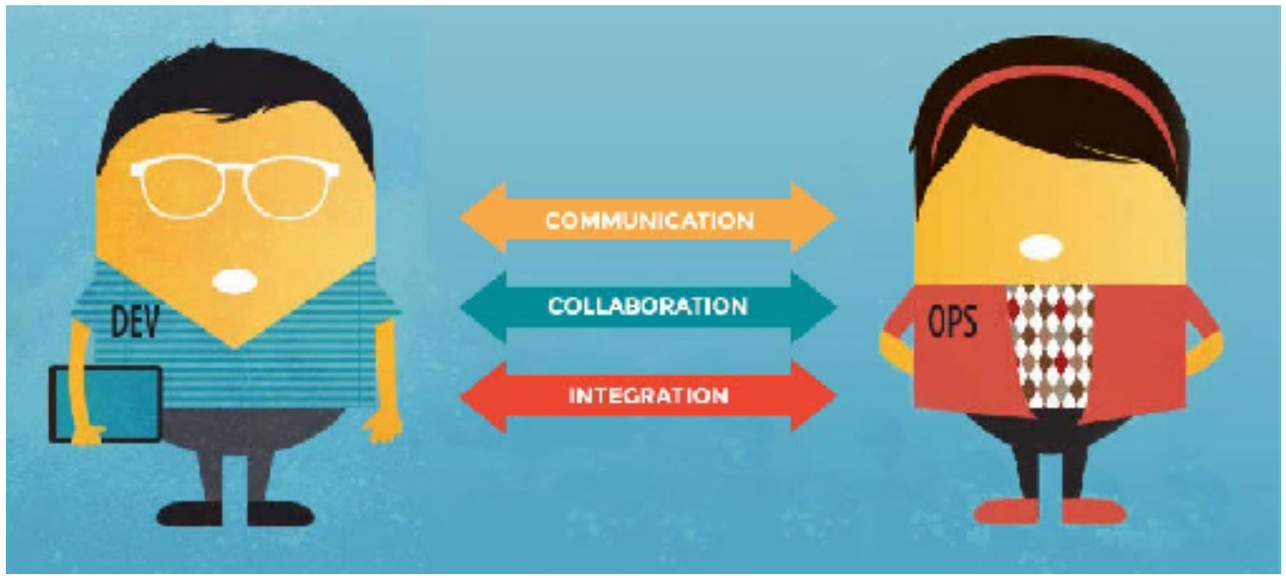
81 responses



heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

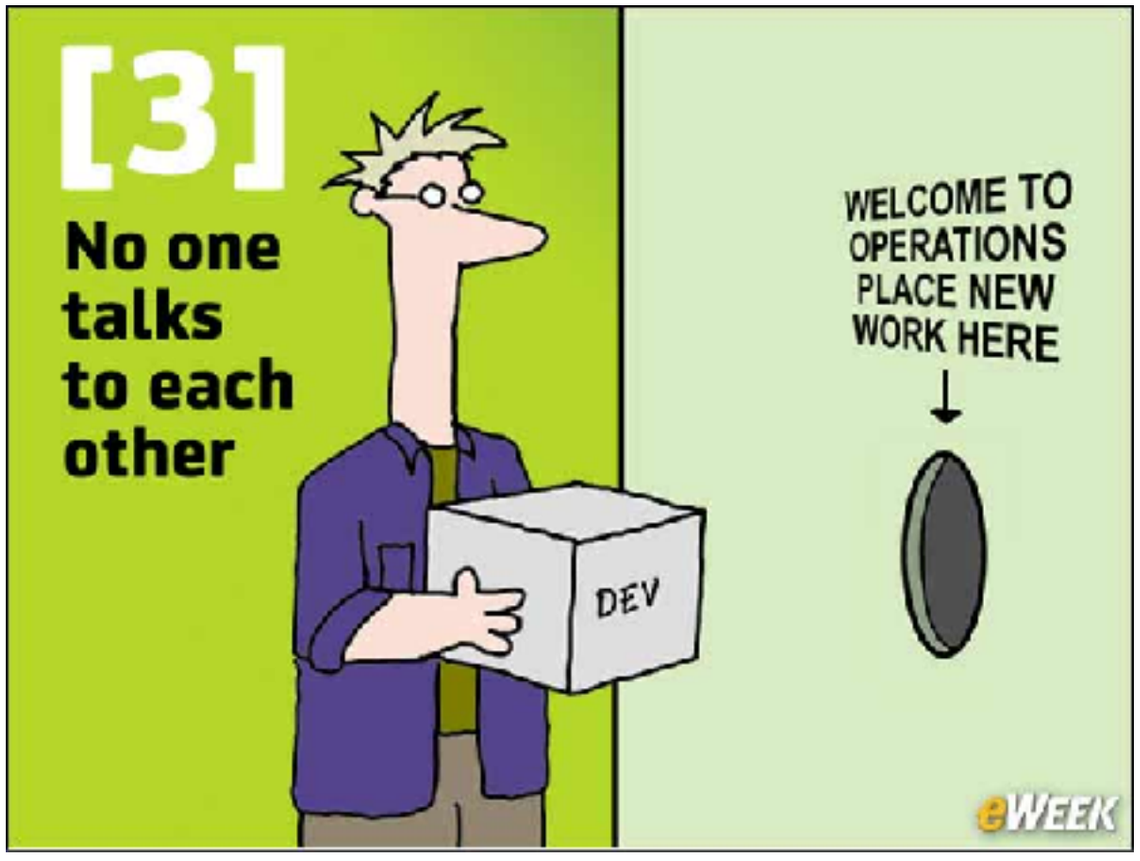


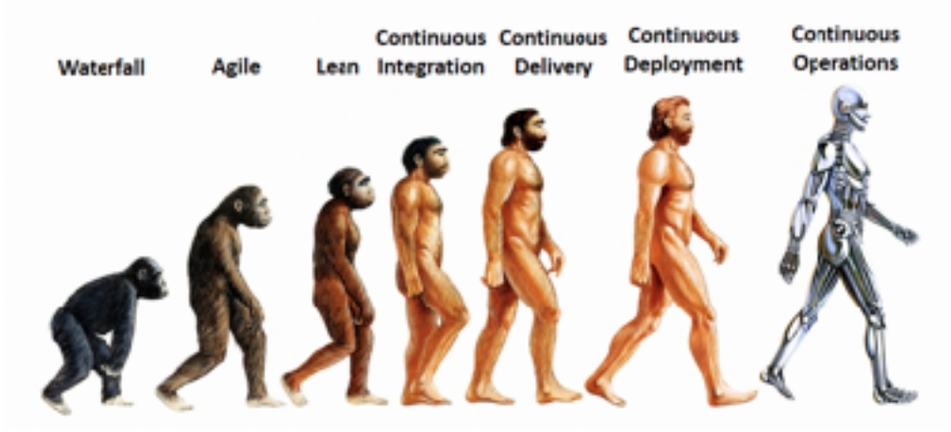


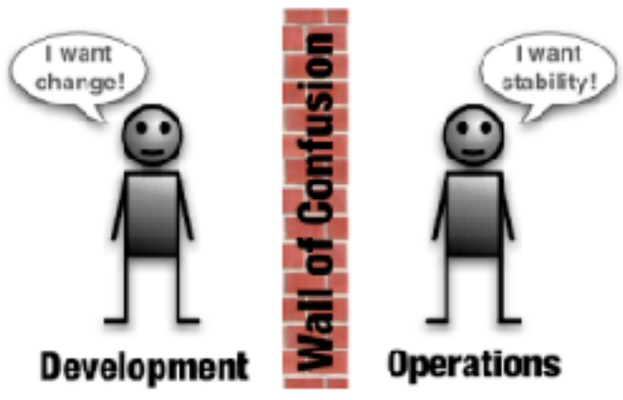
heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

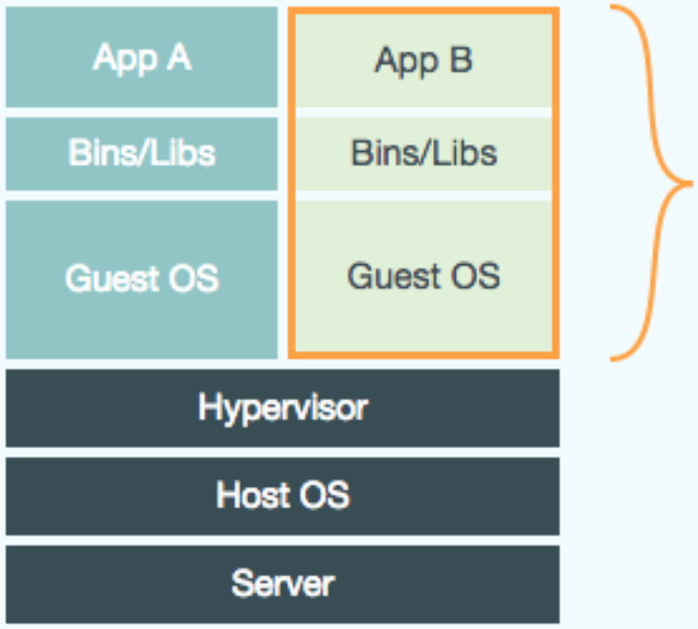
DevOps

DevOps Movement



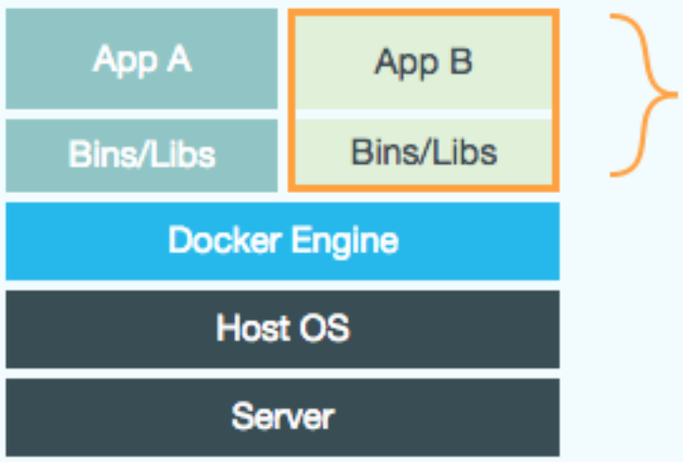






Virtual Machines

Each virtualized application includes not only the application - which may be only 10s of MB - and the necessary binaries and libraries, but also an entire guest operating system - which may weigh 10s of GB.



Docker

The Docker Engine container comprises just the application and its dependencies. It runs as an isolated process in userspace on the host operating system, sharing the kernel with other containers. Thus, it enjoys the resource isolation and allocation benefits of VMs but is much more portable and efficient.

https://www.docker.com/whatisdocker/

Installing Docker

heig-vd
HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

- Docker is based on the **Linux Containers** (LXC) technology. If you run a linux distribution on your laptop, you will use Docker natively. Just be careful with the version you are using.

- If you run **Windows** or **Mac OS X**, you have different options. Behind the scenes, you will need a **Linux VM**. The tools that you will install will hide this VM (more or less), but it will still be here.

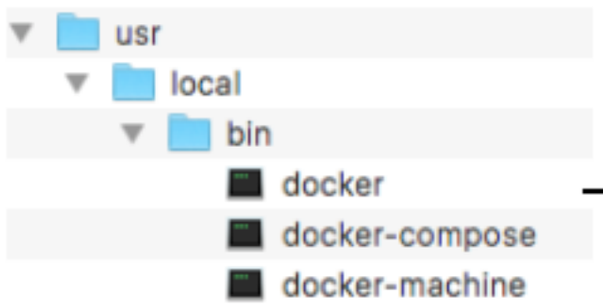
- On the Docker website, you will be encouraged to install **Docker for Windows** or **Docker for Mac**. This might not be the best choice, and you might want to go for the (legacy) **Docker Toolbox** and **Docker Machine** toolset.

- If you run Windows, be aware that you cannot run **Docker for Windows** and **VirtualBox** at the same time. You will need to enable/disable **Hyper V** and reboot to switch between environments. Docker Toolbox does not have this issue.

- Personally, I still use **Docker Machine** and not Docker for Mac. Reasons: performance (file system), ability to have different environments, the Linux VM is less hidden (has its own IP address). In my demos / webcasts, you will see 192.168.99.100 and not localhost.

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Docker Toolbox installation




```
DOCKER_HOST=tcp://xxx.xxx.xxx.xxx:2376
DOCKER_MACHINE_NAME=default
DOCKER_TLS_VERIFY=1
DOCKER_CERT_PATH=$HOME/.docker/machine/machines/default
```

manages

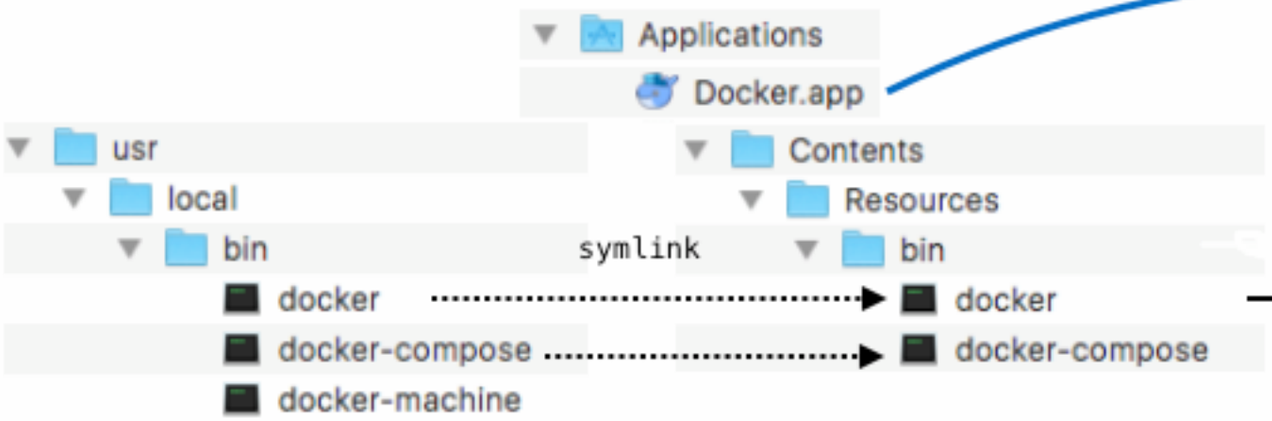
default

IP xxx.xxx.xxx.xxx

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD

Docker for Mac installation

manages




```
unset ${!DOCKER_*}
```

```
/var/run/docker.sock
```

docker.local

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Docker image

Docker container

Docker containers (1)

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- You can think of a container as a **lightweight** virtual machine. Each container is isolated from the others and has its own IP address.
- Each container is created from a docker image (one could say that a container is a **running instance of an image**).
- There are **commands** to start, list, stop and delete containers.

```
# Start a container (more on this later)
$ docker run

# List running containers
$ docker ps

# List all containers
$ docker ps -a

# Delete a container
$ docker rm

# Display logs produced by a container
$ docker logs
```

Docker containers (2)

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- With Docker, the philosophy is to have **one application-level service per container** (it is not a strict rule).
- With Docker, the philosophy is also to **run several (many) containers on the same machine**.
- If you think of a typical **Web Application infrastructure**, you would have one or more containers for the apache web server, one container for the database, one container for the reverse proxy, etc.
- With Docker, containers tend also to be **short-lived**. Each container has an **entry point**, i.e. a command that is executed at startup. When this command returns, the container is stopped (and will typically be removed).
- **If a container dies, it should not be a big deal**. Instead of trying to fix it, one will create a new one (from the same image).

Docker images (1)

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- **A Docker image is a template**, which is used to create containers.
- Every image is **built from a base image** and adds its own configuration and content.
- With Vagrant, we use a file named Vagrantfile to configure and provision a Vagrant box. With Docker, we use a file name **Dockerfile** to create an image. The file contains statements (FROM, RUN, COPY, ADD, EXPOSE, CMD, VOLUME, etc.)
- Just like the community is sharing Vagrant boxes, **the community is sharing Docker images**. This happens on the Docker Hub registry (https://registry.hub.docker.com/).

Docker images (2)

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

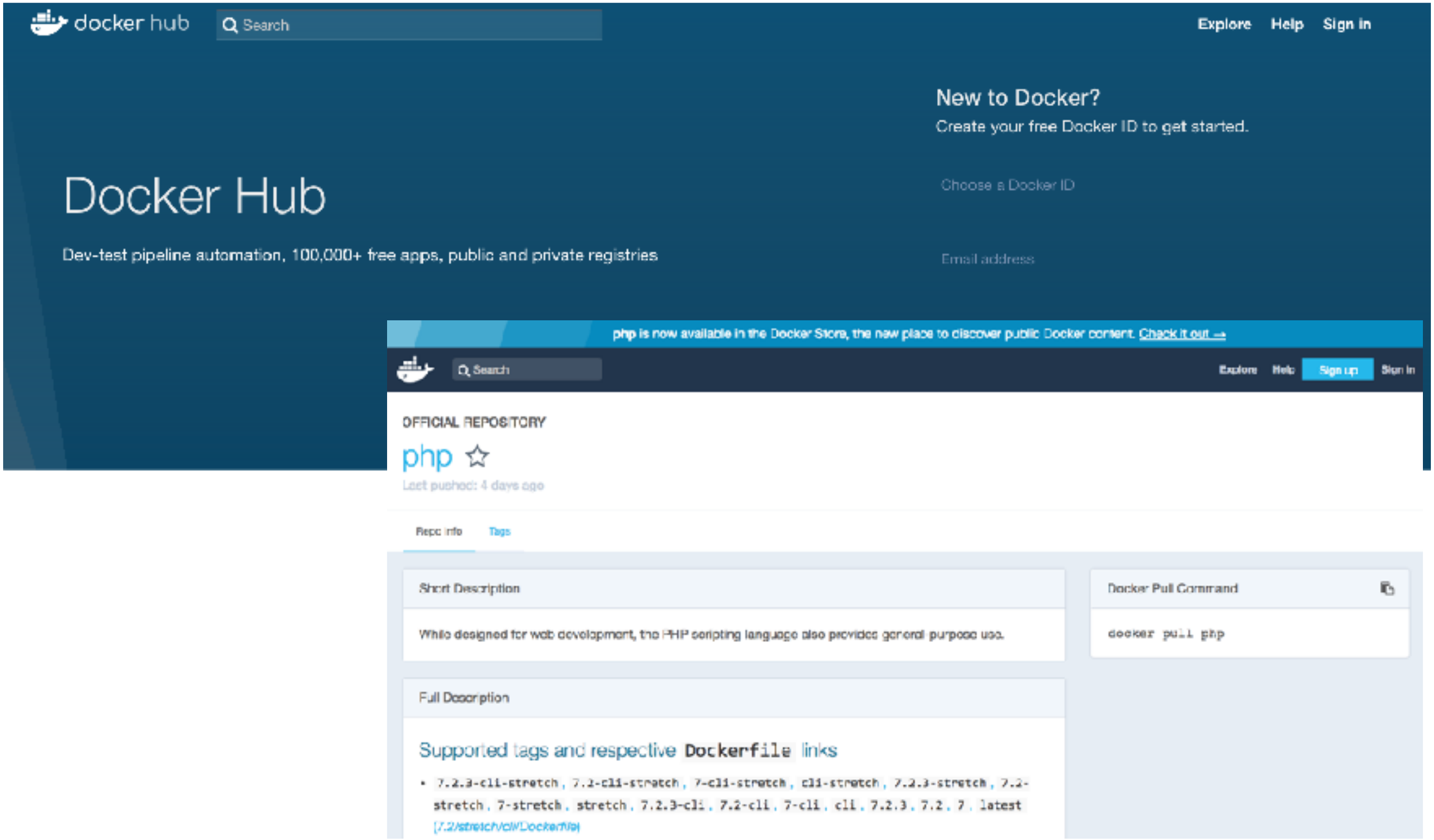
- Here is an example for a Dockerfile (used for first experiments, does not

```
# This image is based on another image
FROM node:8

# When we create the image, we copy files from the host into
# the image file system. This is NOT a shared folder!

COPY file_system /opt/res/

# With RUN, we can execute commands when we create the image. Here,
# we install the PM2 process manager

RUN npm install -g pm2@0.12.9
```

```
# Create an image from this Dockerfile
$ docker build -t heigvd/res-demo .

# Execute /bin/bash in a new container, created from the image
$ docker run -i -t heigvd/res-demo /bin/bash
```

Docker Hub

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch



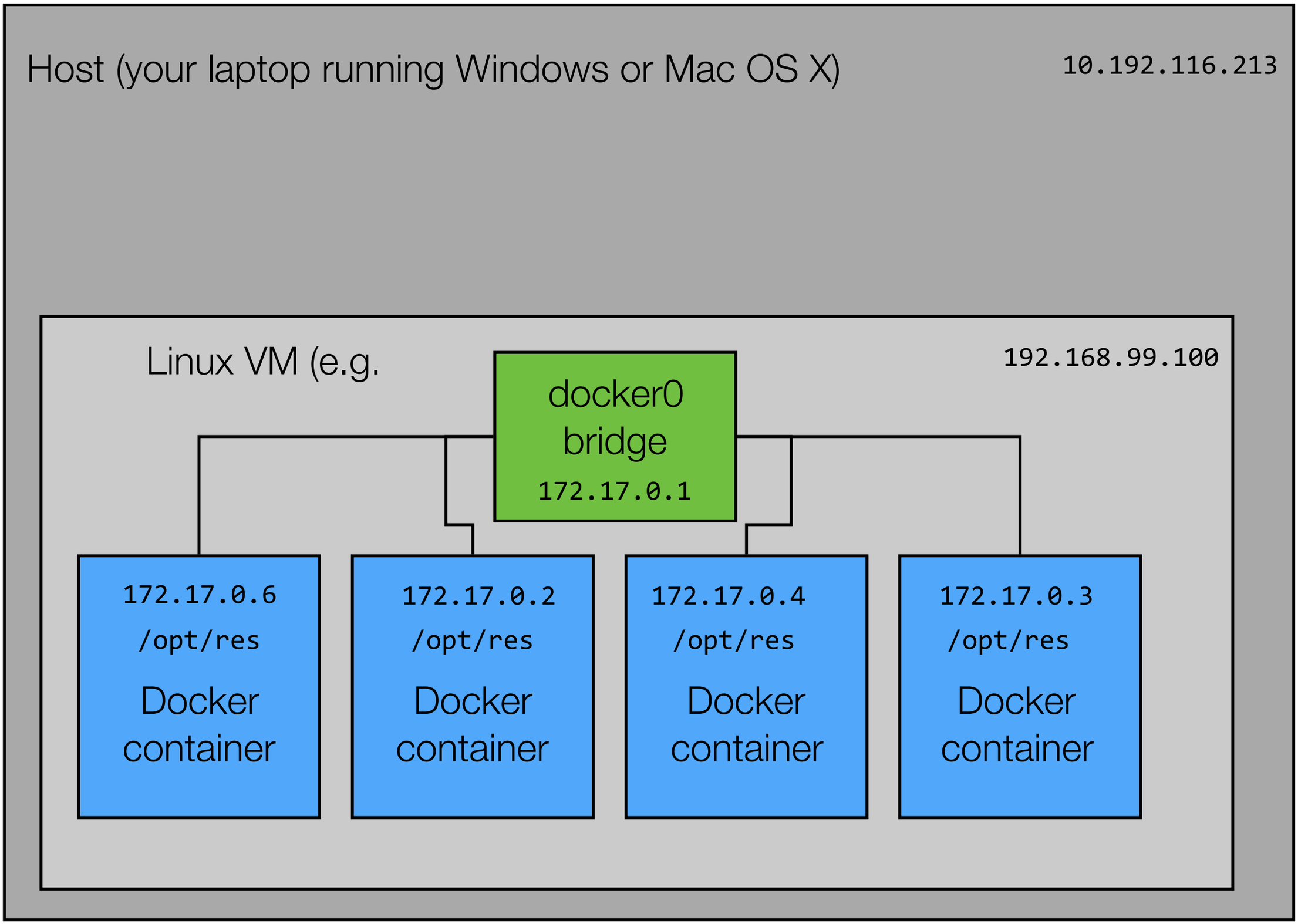
Some interesting images

https://hub.docker.com/_/httpd

https://hub.docker.com/_/wordpress/

https://hub.docker.com/r/gophernet/netcat/

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



Sniffing UDP traffic

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Using tcpdump

```
$ sudo tcpdump -i docker0 udp -A
```

```
E..J..@...E.........q&..6.e{"timestamp":1427800010160,"temperature":null}
11:06:50.362733 IP 172.17.0.8.58225 > 239.255.22.5.9907: UDP, length 46
E..J..@...E.........q&..6.e{"timestamp":1427800010362,"temperature":null}
11:06:50.565816 IP 172.17.0.8.58225 > 239.255.22.5.9907: UDP, length 46
E..J..@...E.........q&..6.e{"timestamp":1427800010565,"temperature":null}
11:06:50.768966 IP 172.17.0.8.58225 > 239.255.22.5.9907: UDP, length 46
E..J..@...E.........q&..6.e{"timestamp":1427800010768,"temperature":null}
11:06:50.970691 IP 172.17.0.8.58225 > 239.255.22.5.9907: UDP, length 46
E..J..@...E.........q&..6.e{"timestamp":1427800010970,"temperature":null}
11:06:51.172537 IP 172.17.0.8.58225 > 239.255.22.5.9907: UDP, length 46
E..J..@...E.........q&..6.e{"timestamp":1427800011172,"temperature":null}
11:06:51.374546 IP 172.17.0.8.58225 > 239.255.22.5.9907: UDP, length 46
E..J..@...E.........q&..6.e{"timestamp":1427800011374,"temperature":null}
11:06:51.578663 IP 172.17.0.8.58225 > 239.255.22.5.9907: UDP, length 46
E..J..@...E.........q&..6.e{"timestamp":1427800011578,"temperature":null}
```

Demo 1

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

https://github.com/SoftEng-HEIGVD/Teaching-Docker-SimpleJavaServer

Protocol design exercise

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

https://github.com/SoftEng-HEIGVD/Teaching-HEIGVD-RES-2019-Exercise-Calculator