A cartoon illustration of a blue whale with a white eye and a red tongue, swimming in a dark blue sea. On its back, there are three blue Docker containers. The title text is overlaid on the whale's body.

# DOCKER GRUNDLAGEN

NICO MAAS

PYTHONCAMP 2018



# WHO AM I?

- Nico Maas
- Master of Science
- IT Systemelektroniker
- [mail@nico-maas.de](mailto:mail@nico-maas.de)
- [www.nico-maas.de](http://www.nico-maas.de)
- [@nmaas87](https://www.instagram.com/nmaas87)



# OVERVIEW (1/3)

- Introduction
  - VMs vs Container
  - Container Technology
  - Docker Architecture
  - Images / Container



# OVERVIEW (2/3)

- Workshop
  - Demo Docker Basics
  - Demo Dockerfile
  - docker-compose
  - Docker Swarm

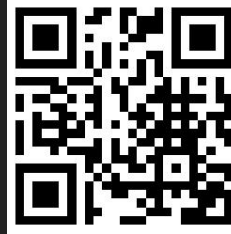


# OVERVIEW (3/3)

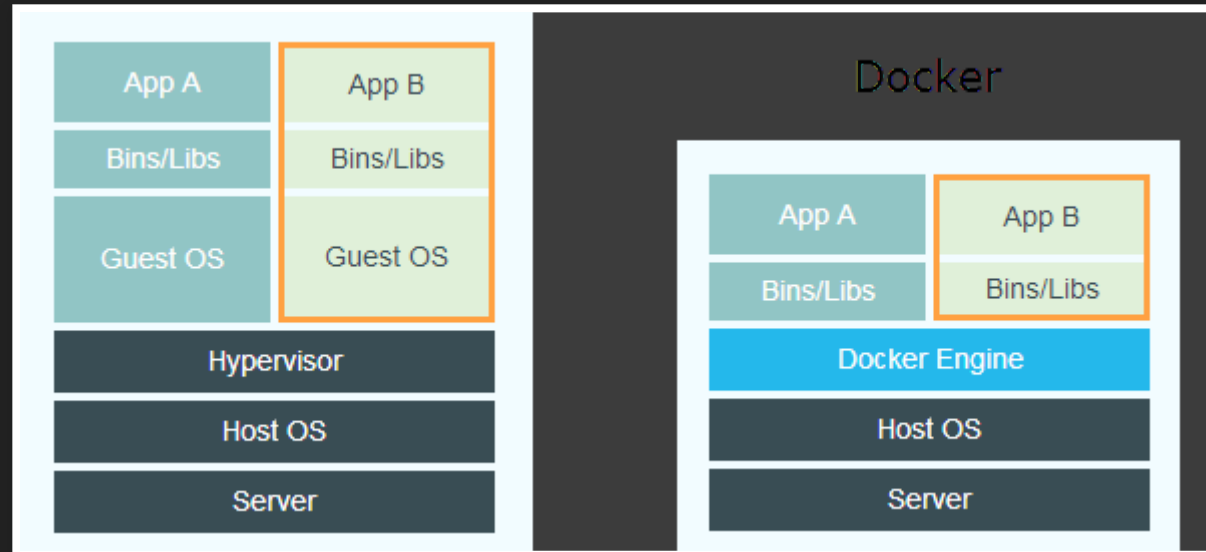
- Docker at Work
  - resin.io
  - gogs.io
  - drone.io



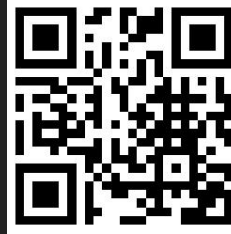
# INTRODUCTION



# VM VS CONTAINER



# MATRIX FROM HELL



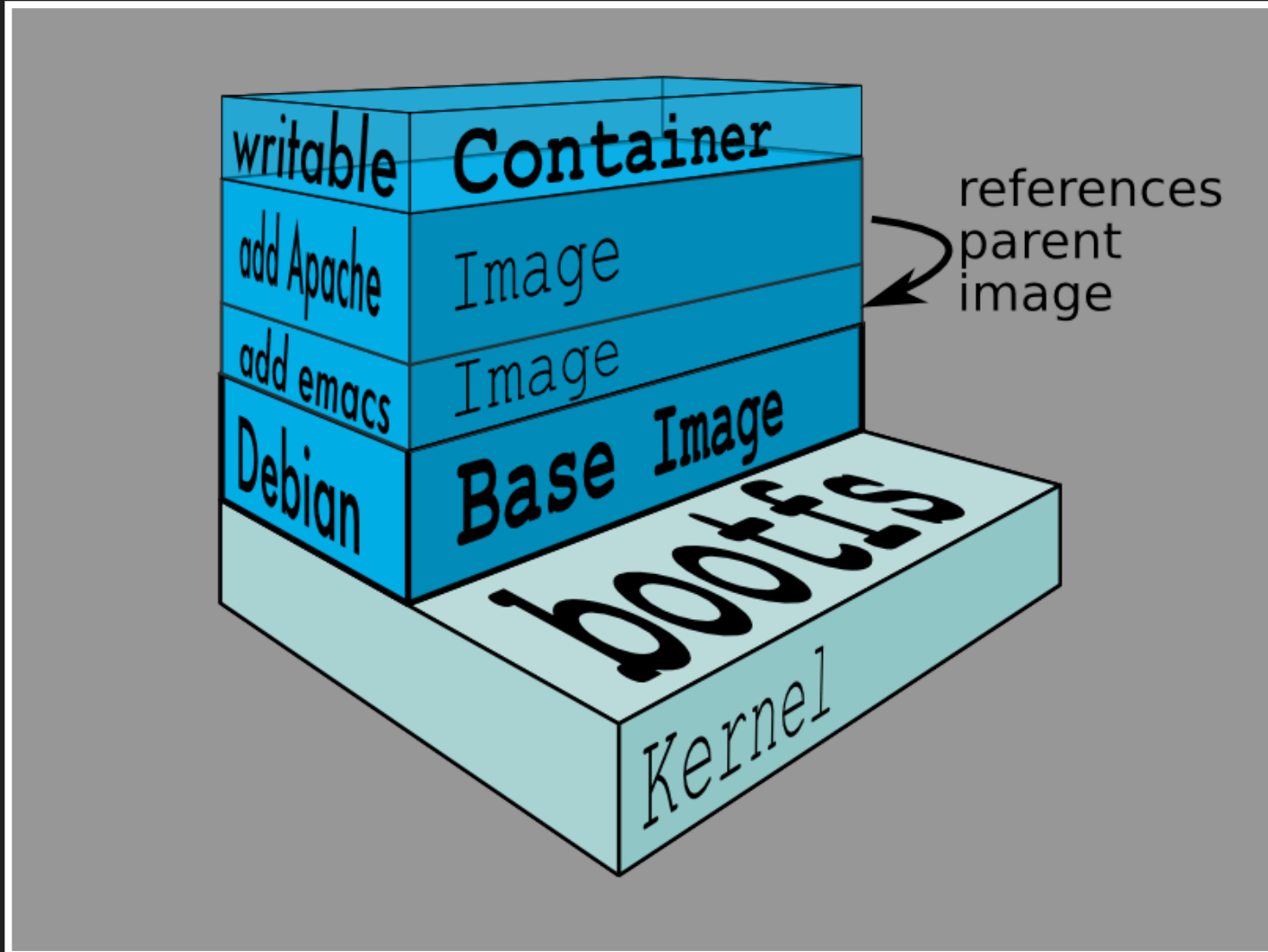
## The Matrix of Hell

|  |                    |                |           |                    |                |              |                      |                  |
|--|--------------------|----------------|-----------|--------------------|----------------|--------------|----------------------|------------------|
|  | Static website     | ?              | ?         | ?                  | ?              | ?            | ?                    | ?                |
|  | Web frontend       | ?              | ?         | ?                  | ?              | ?            | ?                    | ?                |
|  | Background workers | ?              | ?         | ?                  | ?              | ?            | ?                    | ?                |
|  | User DB            | ?              | ?         | ?                  | ?              | ?            | ?                    | ?                |
|  | Analytics DB       | ?              | ?         | ?                  | ?              | ?            | ?                    | ?                |
|  | Queue              | ?              | ?         | ?                  | ?              | ?            | ?                    | ?                |
|  |                    | Development VM | QA Server | Single Prod Server | Onsite Cluster | Public Cloud | Contributor's laptop | Customer Servers |
|  |                    |                |           |                    |                |              |                      |                  |

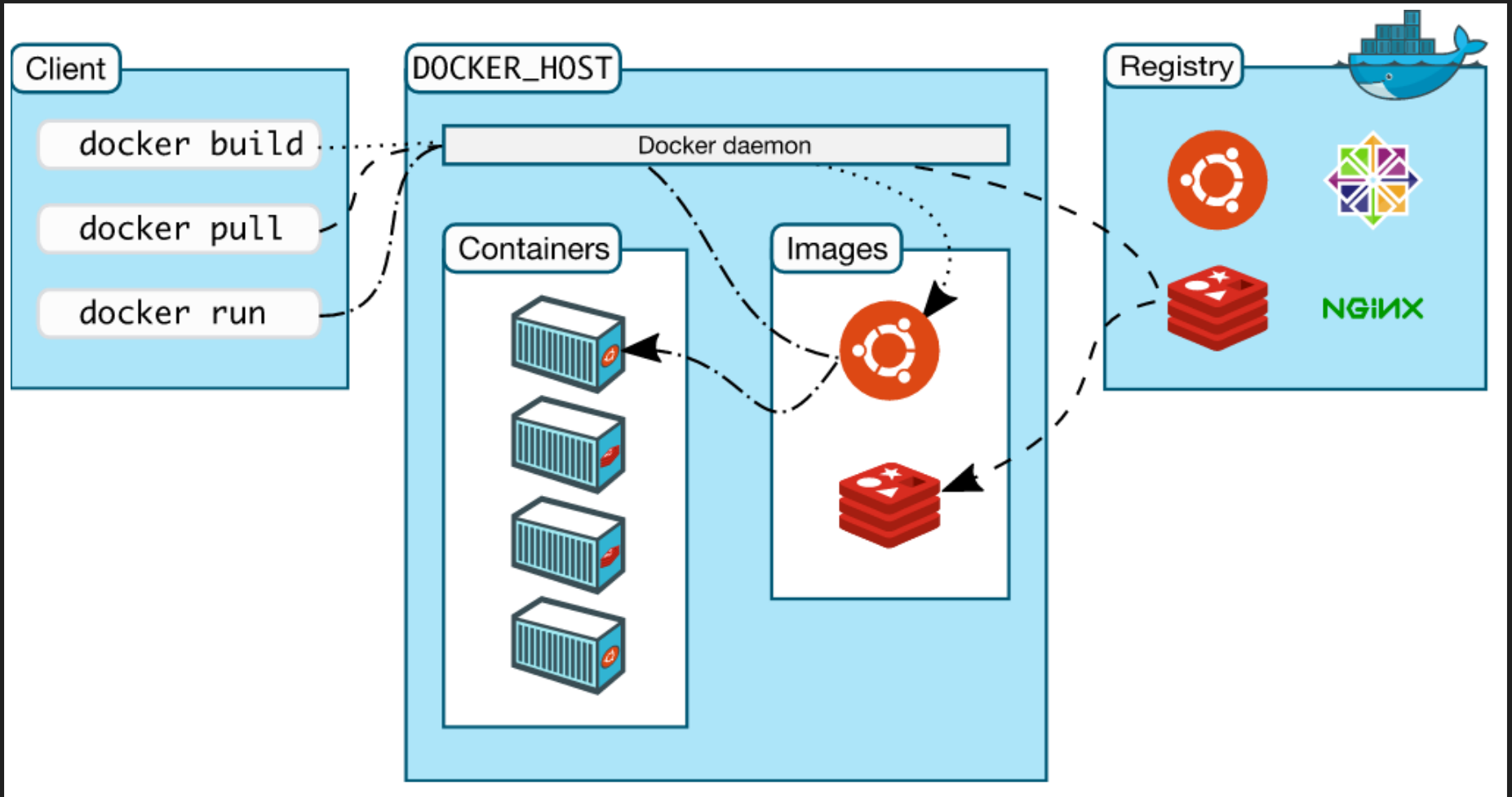




# CONTAINER



# DOCKER ARCHITECTURE

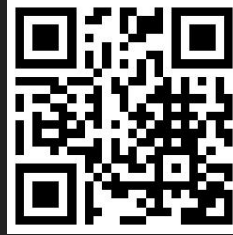


# IMAGES / CONTAINER



- Images
  - Filesystem
  - R/O
  - A whole repo is available on [hub.docker.com](https://hub.docker.com)
- Container
  - running instance of a image
  - R/W
  - changes to the container must be "saved" with `docker commit` - which creates a new image with the changes as a new layer

# IMAGE ARCHITECTURE

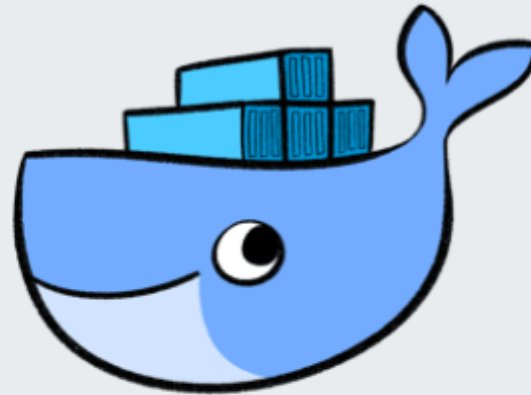


- Images consists of Layers
- Do not build Images by hand (docker commit) - use a Dockerfile
- Every CMD in a Dockerfile creates a new Layer
- Layers are cached - also on Image creation - parts that change often should go to the end of the Dockerfile - so that Docker does not need to rebuild everything from 0
- Example for the layers: <https://imagelayers.io/?images=microscaling%2Fimagelayers-api:latest,microscaling%2Fimagelayers-web:latest>



# WORKSHOP

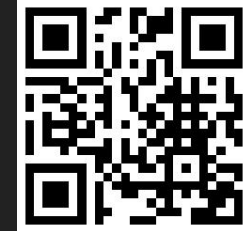
# BEFORE WE START...



## Play with Docker

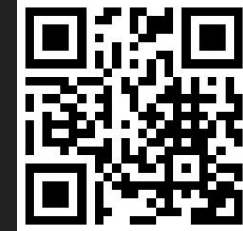
A simple, interactive and fun playground to learn Docker

Start



# DEMO DOCKER BASICS

- docker
  - Docker Help
- docker version
  - Versioninformation
- docker container ps
  - Running containers
- docker container ps -a
  - Running and stopped containers



# DEMO DOCKER BASICS

- docker container run nmaas87/docker-openwrt echo "Hello Pythoncamp 2018!"
  - Download image nmaas87/docker-openwert on "latest" Tag from [hub.docker.com](https://hub.docker.com), spin up new container and execute command echo "Hello Pythoncamp 2018!"



# DEMO DOCKER BASICS



- `docker container run -d nmaas87/docker-openwrt ping 127.0.0.1 -c 50`
  - Run ping cmd in deattached Mode
- `docker container ps`
  - Running containers

| CONTAINER ID | IMAGE                  | COMMAND          | CREATED        | STATUS        | PORTS | NAMES         |
|--------------|------------------------|------------------|----------------|---------------|-------|---------------|
| a4f4424741a4 | nmaas87/docker-openwrt | "ping 127.0.0.1" | 18 seconds ago | Up 16 seconds |       | nifty_murdock |

- `docker container attach a4f`
  - Attach to running container
- Container "dies" as soon as the running root process (PID 1, here ping) ends

# DEMO DOCKER BASICS



- `docker container run -it nmaas87/docker-openwrt /bin/sh`
  - Run interactive shell in container
- `CTRL + P + Q`
  - Disconnect from shell
- `docker container ps`
  - Running containers

| CONTAINER ID | IMAGE                  | COMMAND   | CREATED        | STATUS        | PORTS | NAMES                |
|--------------|------------------------|-----------|----------------|---------------|-------|----------------------|
| 53620b420ca0 | nmaas87/docker-openwrt | "/bin/sh" | 16 seconds ago | Up 15 seconds |       | confident_chatterjee |

- `docker container attach a4f`
  - Attach to running container



# DEMO DOCKERFILE

- Used to build a new Docker Image / "recipe"
- <https://github.com/nmaas87/docker-demo>

```
FROM nmaas87/docker-openwrt:15.05.1_x86

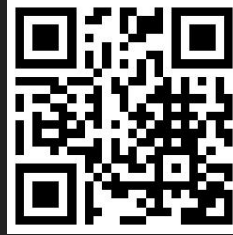
# Update opkg Package List, Install python 2.7, pip, Flask and create directories
RUN opkg update && \
    opkg install wget ca-certificates libffi libopenssl python-openssl python && \
    cd /tmp && \
    wget https://bootstrap.pypa.io/get-pip.py && python get-pip.py && \
    pip install --no-cache-dir Flask && \
    mkdir /app/ /app/templates

# copy files required for the app to run
COPY app.py /app/
COPY templates/index.html /app/templates/

# tell the port number the container should expose
EXPOSE 80

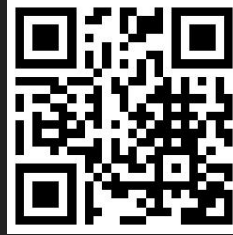
# run the application
#CMD /sbin/init
CMD ["python", "/app/app.py"]
```

# DEMO DOCKERFILE



- git clone <https://github.com/nmaas87/docker-demo.git>
- cd docker-demo
- docker image build -t app .
  - Create a Docker Image "app" from Dockerfile in the current directory (.)
- docker container run -p80:80 app
  - Start new container from "app", expose container port 80 to port 80 of the PC
- Access <http://127.0.0.1> via browser
- Change app.py, rebuild the image and restart

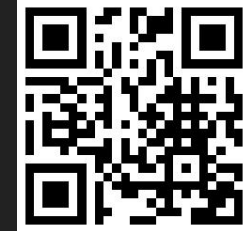
# DOCKER-COMPOSE



Used to execute and build images more easily

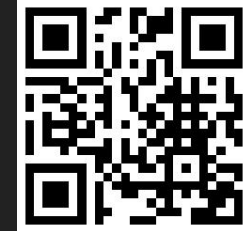
```
flaskapp:  
  restart: unless-stopped  
  image: app  
  ports:  
    - "80:80"
```

- docker-compose.yml
  - Run container "flaskapp", restart even if service / Docker dies, use image "app", bind pc port 80 to container port 80 (HOST:Container)
  - Start with docker-compose up (-d with deattached Mode)



# DOCKER SWARM

- Redundant Cluster of Docker Nodes, self-healing, routing overlay network
- Create swarm
  - Create Swarm on "Master/Manager" node:
  - `docker swarm init --advertise-addr`
  - Prints a CMD similar to:
    - `docker swarm join --token :2377`
  - Execute on other Docker nodes
  - Done



# DOCKER SWARM

- Create service
  - `docker service create --replicas 1 --name pinger nmaas87/docker-openwrt:trunk_x86 ping docker.com`
- `docker service ls`
- `docker service inspect --pretty pinger`
- `docker service ps pinger`
- `docker service scale pinger=3`
- `docker service rm pinger`



# DOCKER SWARM

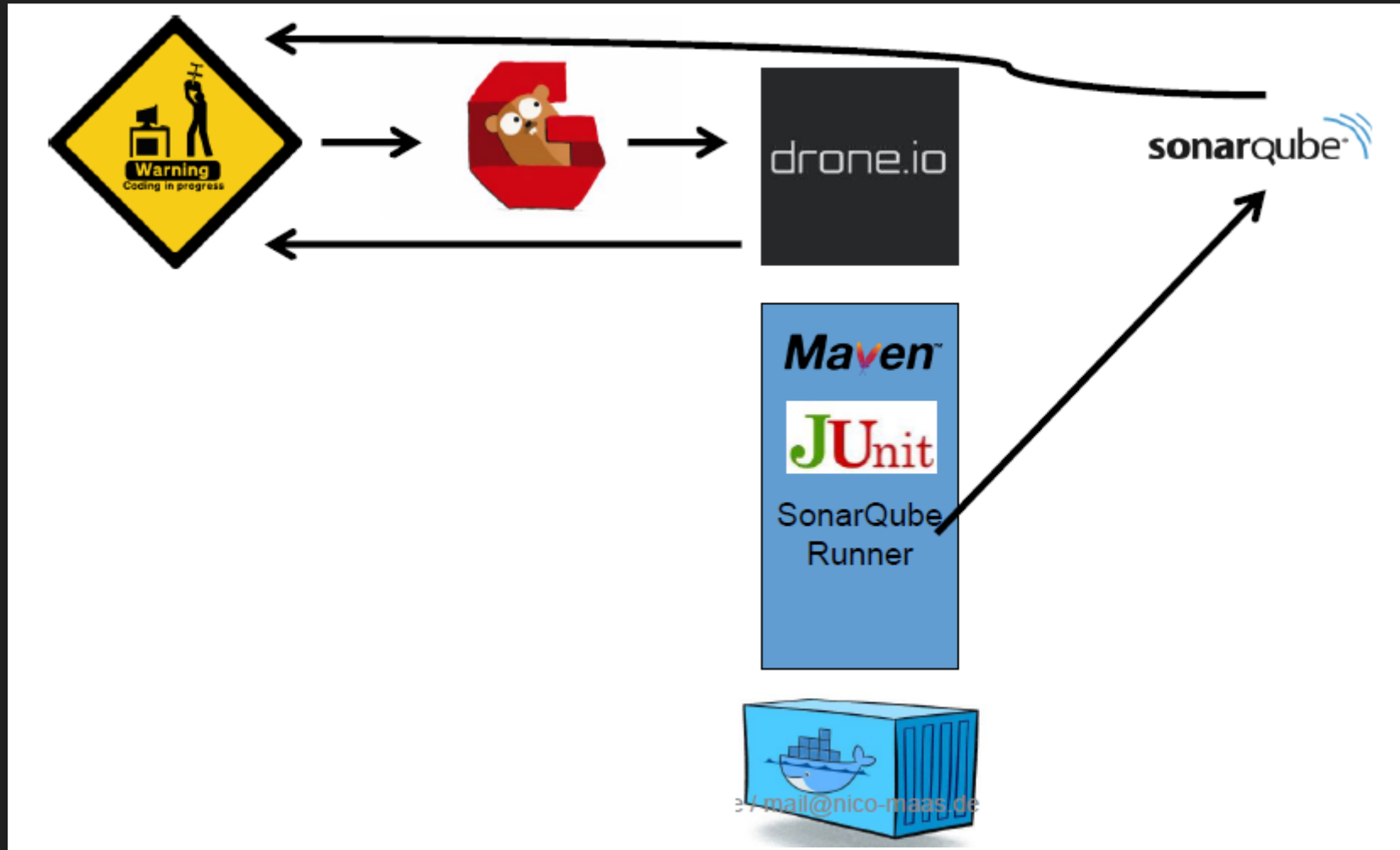
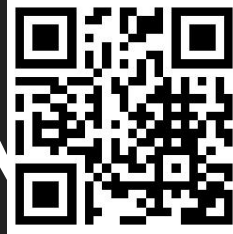
- Useful CMDs (on Manager!)
  - `docker node ls`
  - `docker info`





# REALLIFE

# GOGS / DRONE / MAVEN / JUN



# SYSTEMS



- [www.resin.io](http://www.resin.io)
  - Uses Docker as Deployment Method for Apps on i.e. Raspberry Pi or other Platforms
- [www.gogs.io](http://www.gogs.io)
  - Gitlab clone written in Go, packed as Docker container
- [www.drone.io](http://www.drone.io)
  - CI System, packed as Docker container, uses plugins and addons also packed as Docker container, builds code in Docker containers and works great with Gogs.



# QUESTIONS?

Thank you for your attention :)

[www.docker.com](http://www.docker.com)

[www.play-with-docker.com](http://www.play-with-docker.com)

[www.nico-maas.de](http://www.nico-maas.de)