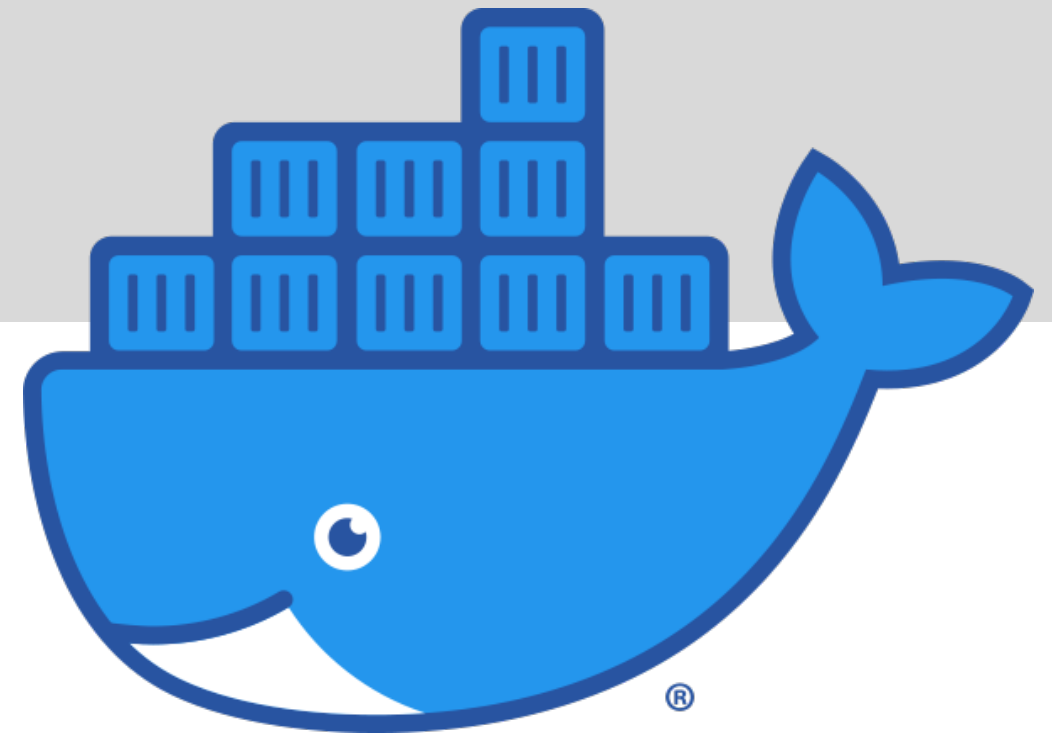


# Docker for IoT



<https://github.com/pmanzoni/applied-IoT-Summer-2021>

Pietro Manzoni

Universitat Politècnica de Valencia (UPV)

Valencia - SPAIN

pmanzoni@disca.upv.es



# Course outline

- ❑ This seminar will focus on the use of Docker for the quick Development of microservices for the IoT.
- ❑ I'll start with an overview of this tool and then I'll move to a practical part
- ❑ There is plenty of information about Docker on the Internet, but the main starting point is: <https://www.docker.com/get-started>

## Get Started with Docker

We have a complete container solution for you – no matter who you are and where you are on your containerization journey.



### Docker Desktop

Developer productivity tools and a local Kubernetes environment.

Download for  
Mac - Intel  
Chip



### Docker Hub

Cloud-based application registry and development team collaboration services.

Signup



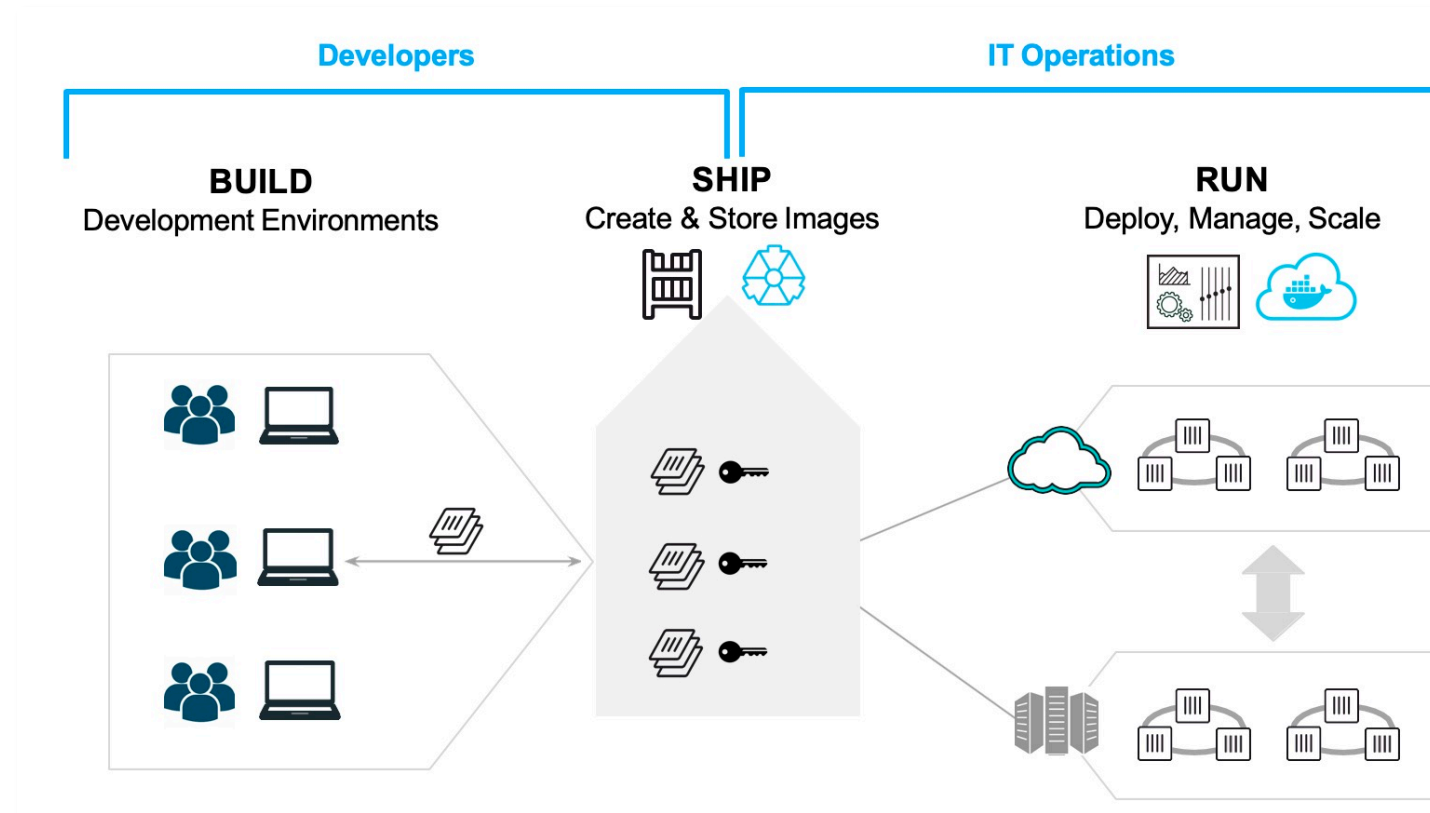
### Play with Docker

Cloud-based docker environment to try out docker and learn the ropes.

Play with Docker

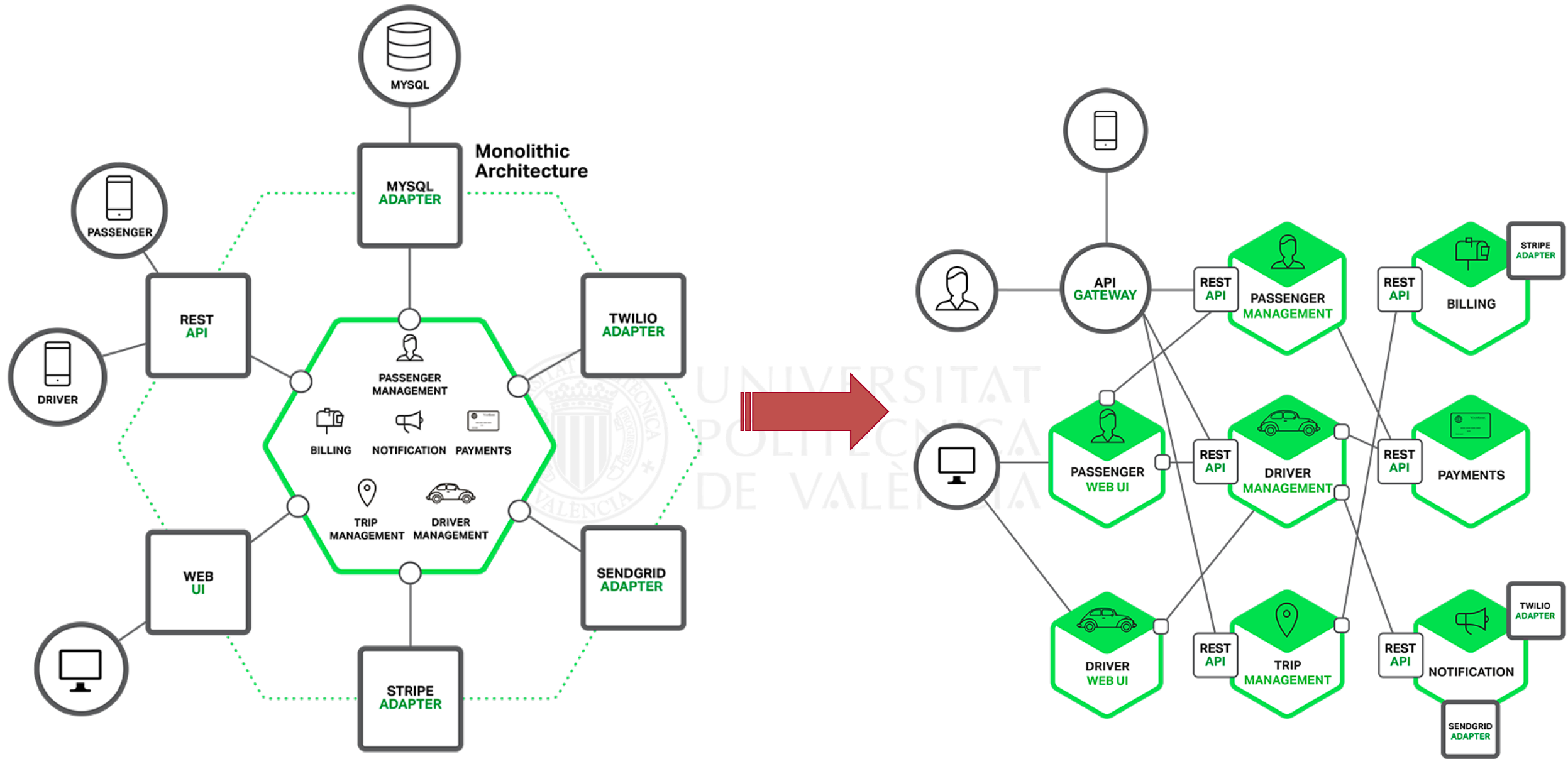
# Docker overview

- ❑ Docker is an open platform for developing, shipping, and running applications.

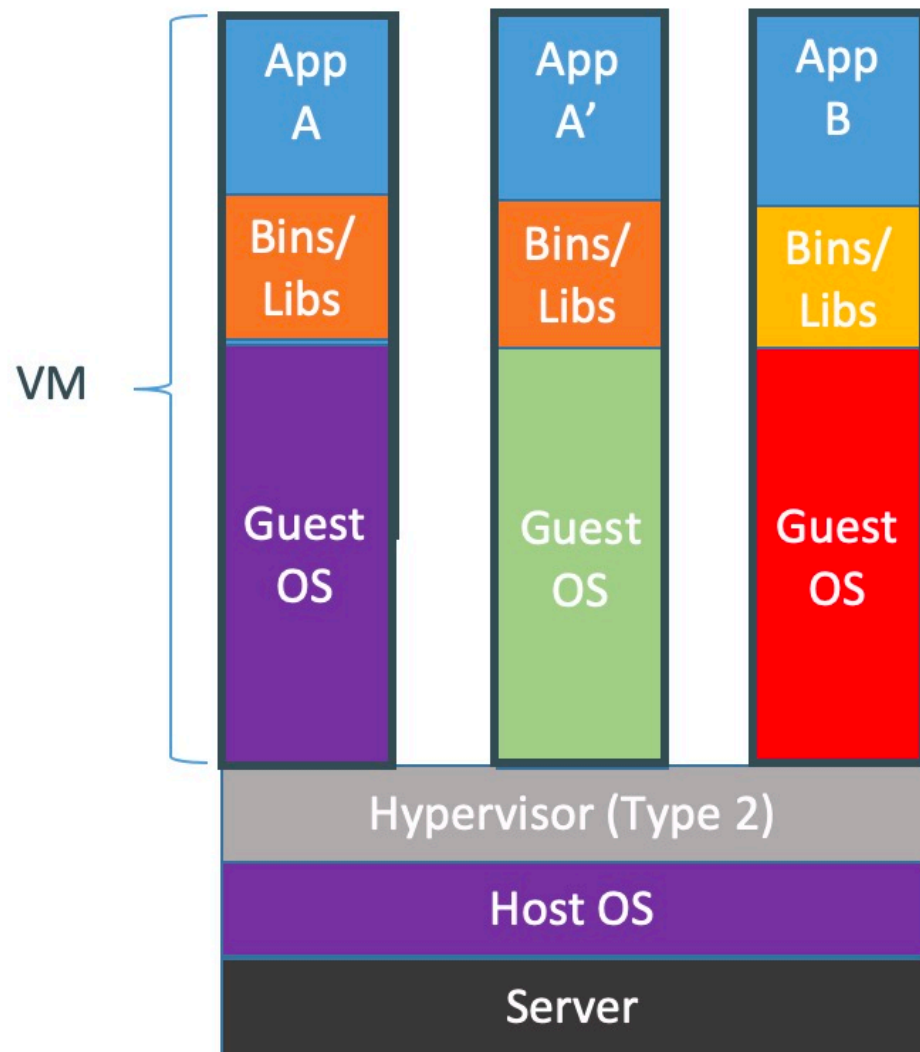


- ❑ Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.
- ❑ <https://docs.docker.com/get-started/overview/>

# Monolithic vs. Microservices

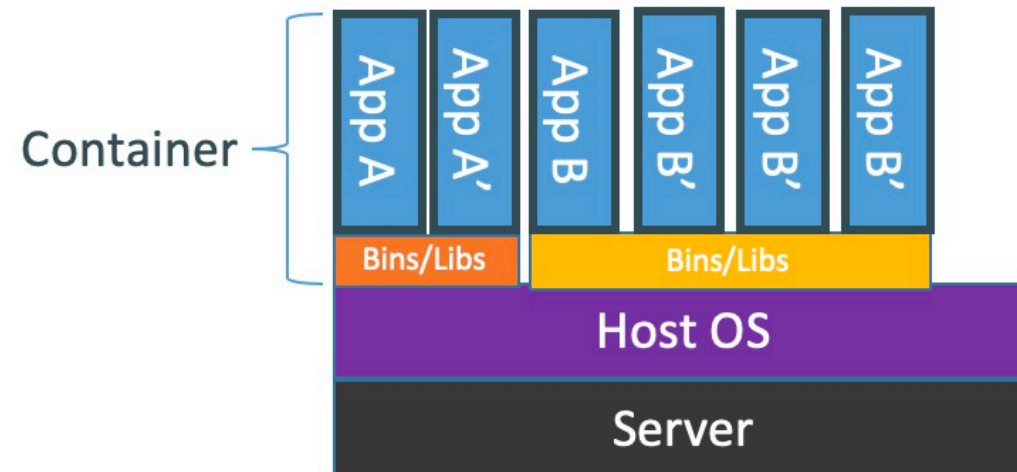


# Docker vs. Virtual Machine



Containers are isolated, but share OS kernel and, where appropriate, bins/libraries

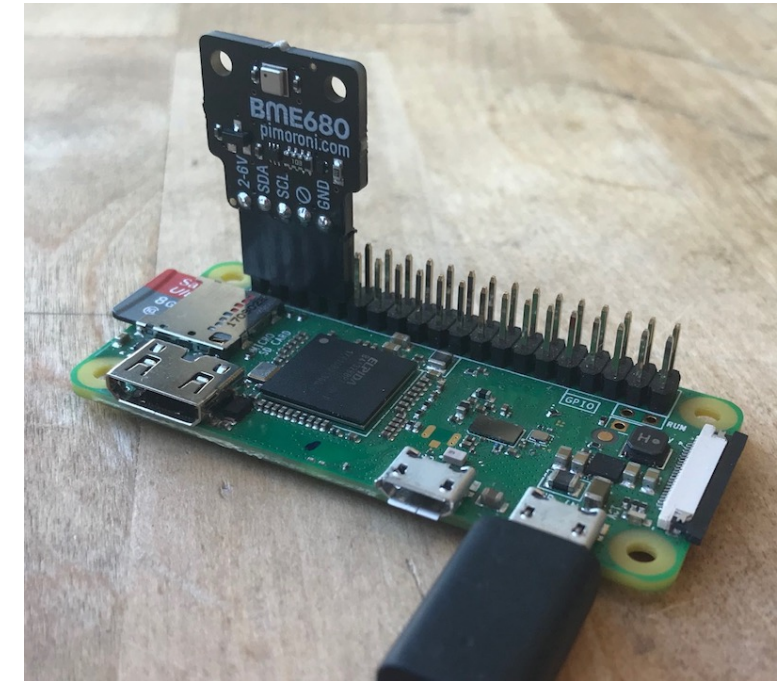
...result is significantly faster deployment, much less overhead, easier migration, faster restart





# Deployment and scaling

- ❑ Docker containers can run on a developer's local laptop, on physical or virtual machines in a data center, on cloud providers, or in a mixture of environments.
- ❑ Docker's portability and lightweight nature also make it easy to **dynamically manage workloads**, scaling up or tearing down applications and services as business needs dictate, in near real time.




<https://www.balena.io/blog/build-an-environment-and-air-quality-monitor-with-raspberry-pi/>




<https://blog.alexellis.io/getting-started-with-docker-on-raspberry-pi/>

# Docker containers on embedded devices

<https://www.balena.io/os/>

An open source project by  **balena** | [More products](#) ▾


 **balenaOS**

[Docs](#) [Getting started](#) [Download](#) [Features](#) [Forums](#)

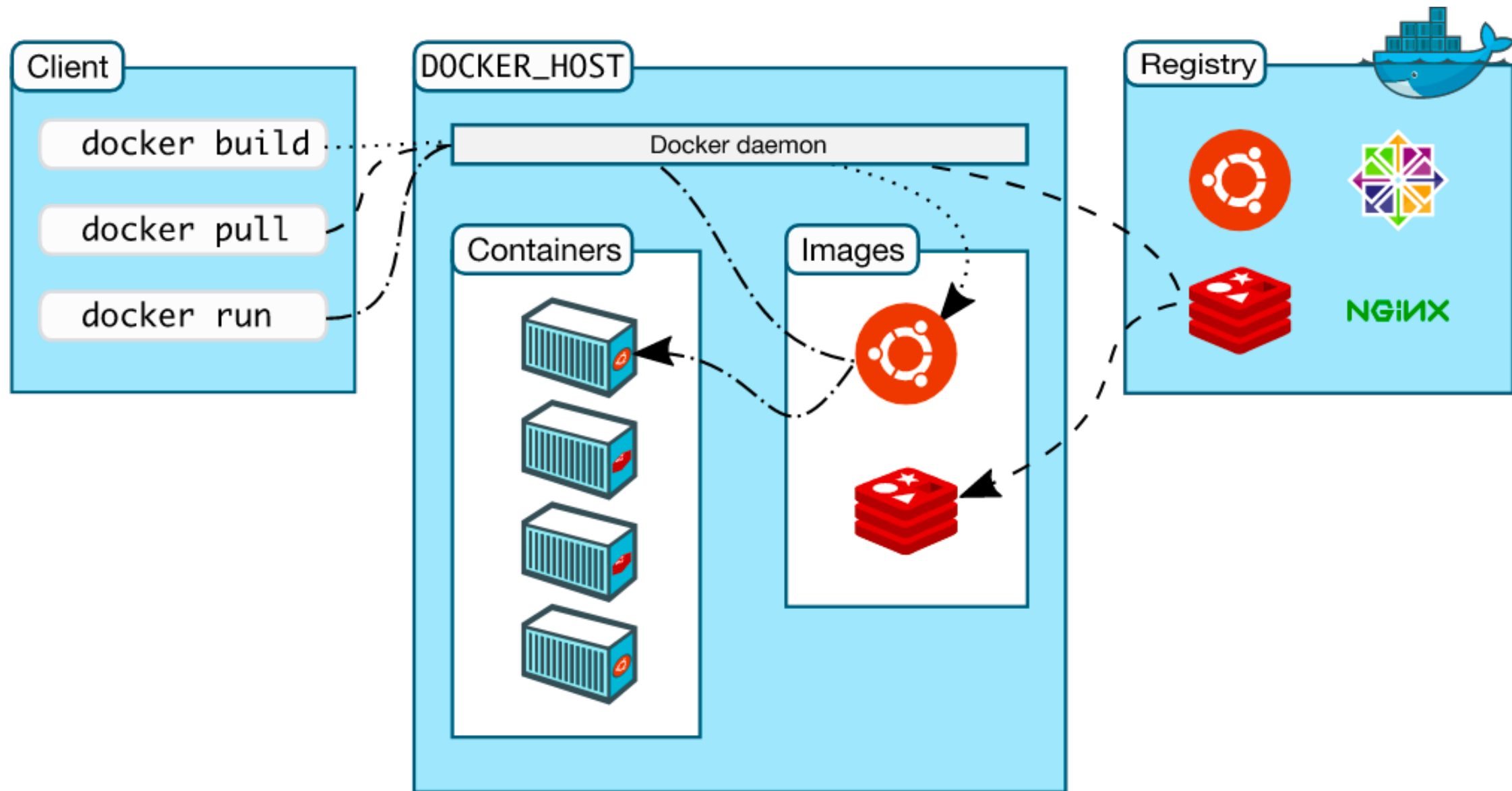
## Run Docker containers on embedded devices

A host OS tailored for containers, designed for reliability, proven in production.

[Try balenaOS](#)



# Docker architecture



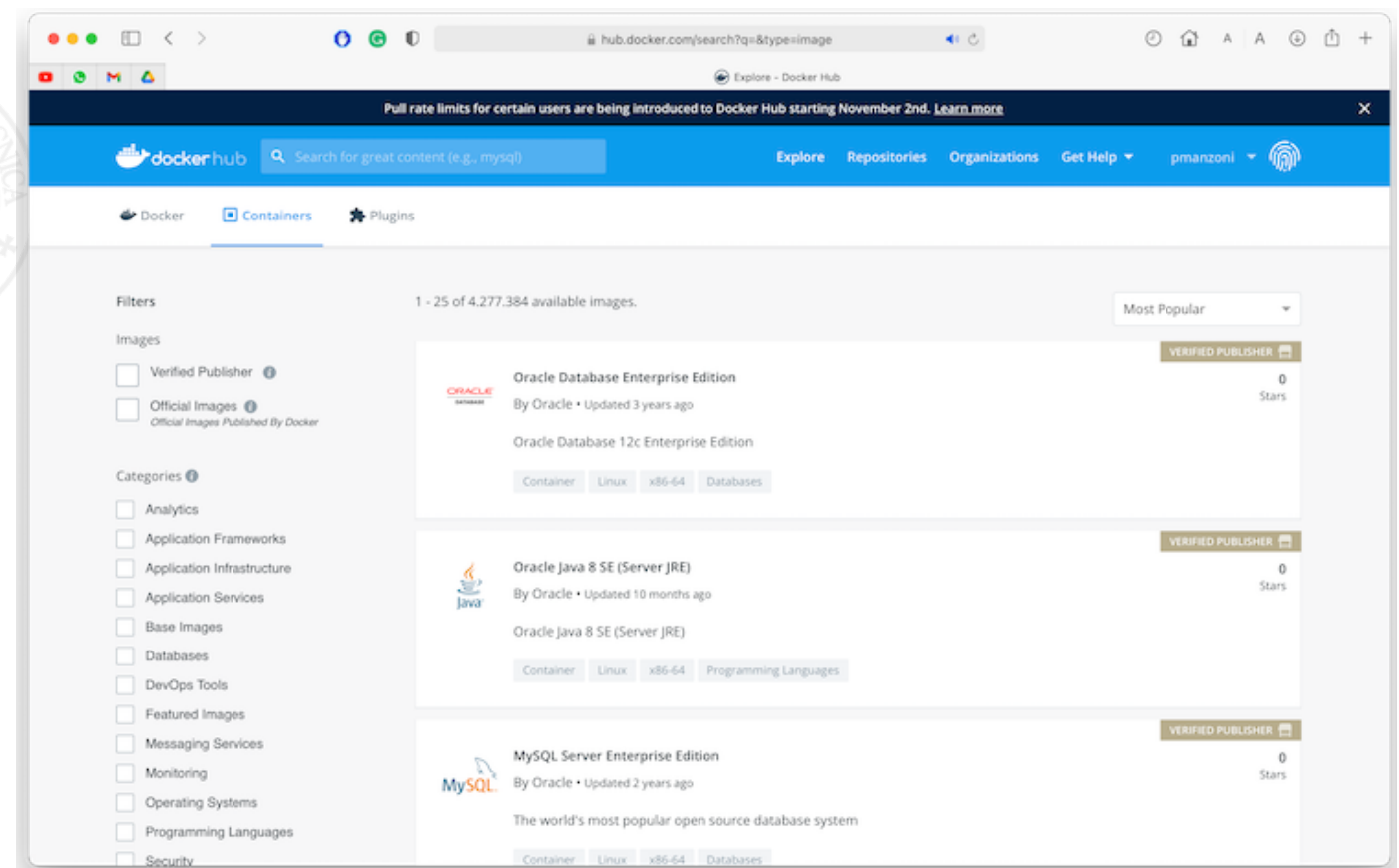


# Docker registries

- A Docker registry stores Docker images. Docker Hub (<https://hub.docker.com/>) is a public registry that anyone can use, and Docker is configured to look for images on Docker Hub by default.
  - You can run your own private registry, too, e.g., <https://docs.github.com/en/packages/guides/about-github-container-registry>

But also:

- Amazon EC2 Container Service.
- Microsoft Azure Container Service.
- Google Container Engine for Docker Containers.



# Images

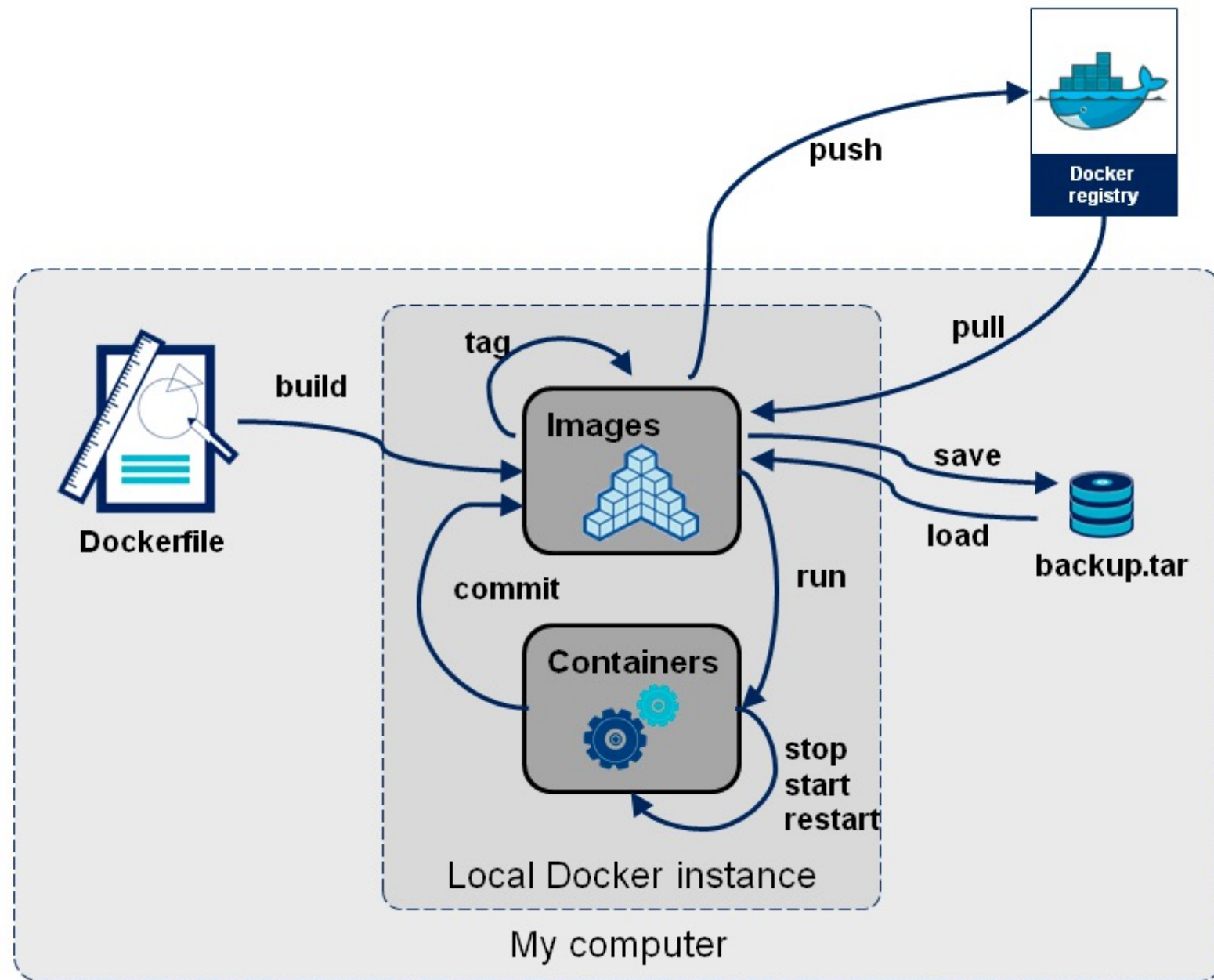
- ❑ An image is a read-only template with instructions for creating a Docker container. Often, an image is based on another image, with some additional customization.
  - ❑ For example, you may build an image which is based on the ubuntu image, but installs the Apache web server and your application, as well as the configuration details needed to make your application run.
- ❑ You might create your own images or you might only use those created by others and published in a registry.
- ❑ To build your own image, you create a Dockerfile with a simple syntax for defining the steps needed to create the image and run it.
  - ❑ ... we will see an example in the hands-on part 😊

```
FROM alpine:3.5
RUN apk add --update py2-pip
RUN pip install paho-mqtt
COPY sisub.py /home/
CMD ["python", "/home/sisub.py"]
```

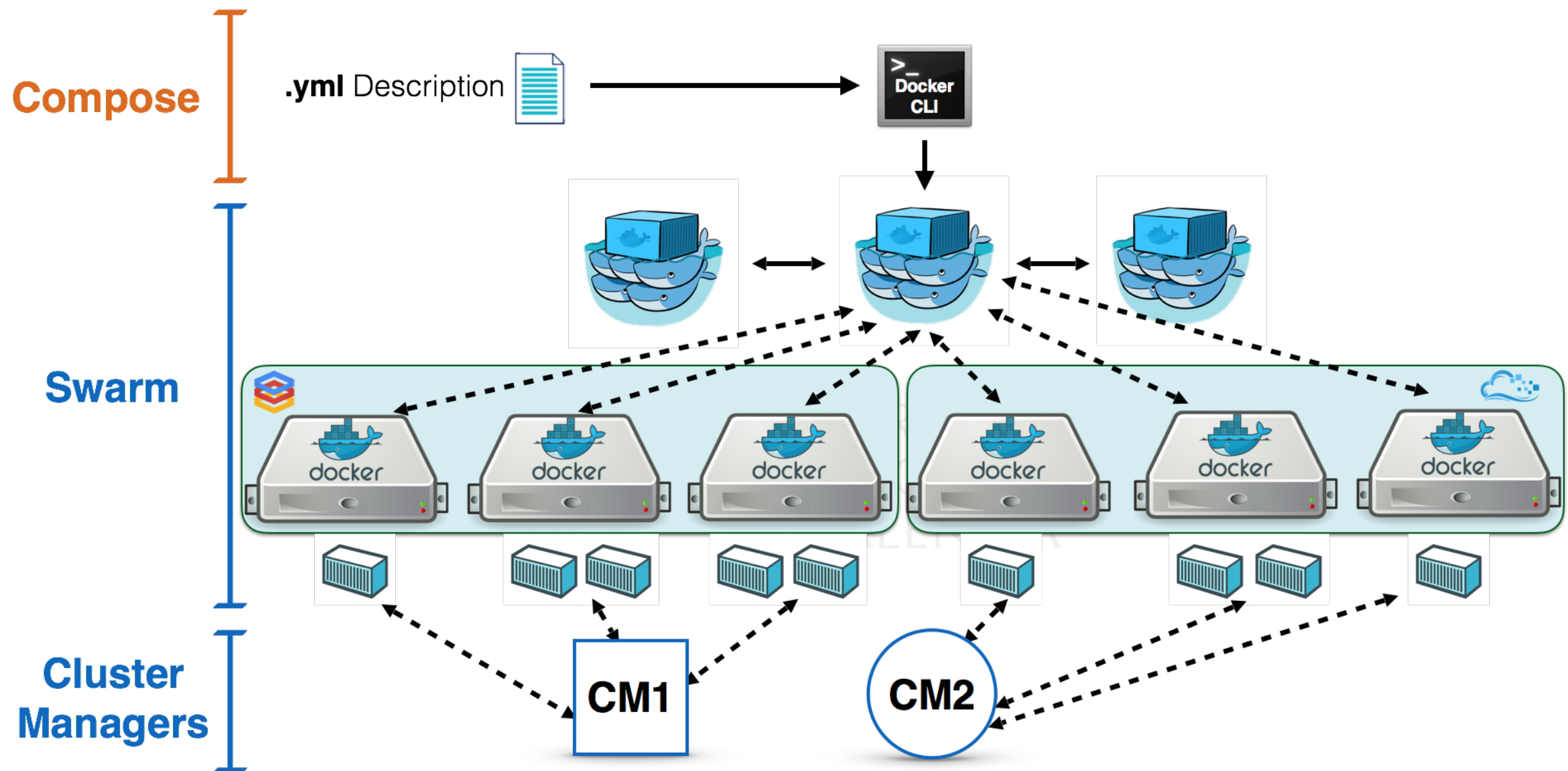
# Containers

- ❑ A container is **a runnable instance of an image**. You can create, start, stop, move, or delete a container using the Docker API or CLI. You can connect a container to one or more networks, attach storage to it, or even create a new image based on its current state.
- ❑ By default, **a container is relatively well isolated from other containers and its host machine**.
- ❑ A container is defined by its image as well as any configuration options you provide to it when you create or start it. **When a container is removed, any changes to its state that are not stored in persistent storage disappear.**

# Manage the lifecycle of your containers



# Docker Swarm





# Kubernetes

- ❑ Developed by Google and introduced in the year 2014.
- ❑ An IT management tool that has been specifically designed to simplify the scalability of workloads using containers.
- ❑ It has the ability to automate deployment, scaling, and operating application containers.

