Docker Swarm

Engels, Wedekind, Weigand

Agenda

- Docker Basics
- Swarm overview
- Competing products
- Swarm vs Kubernetes
- Quiz
- Practical task

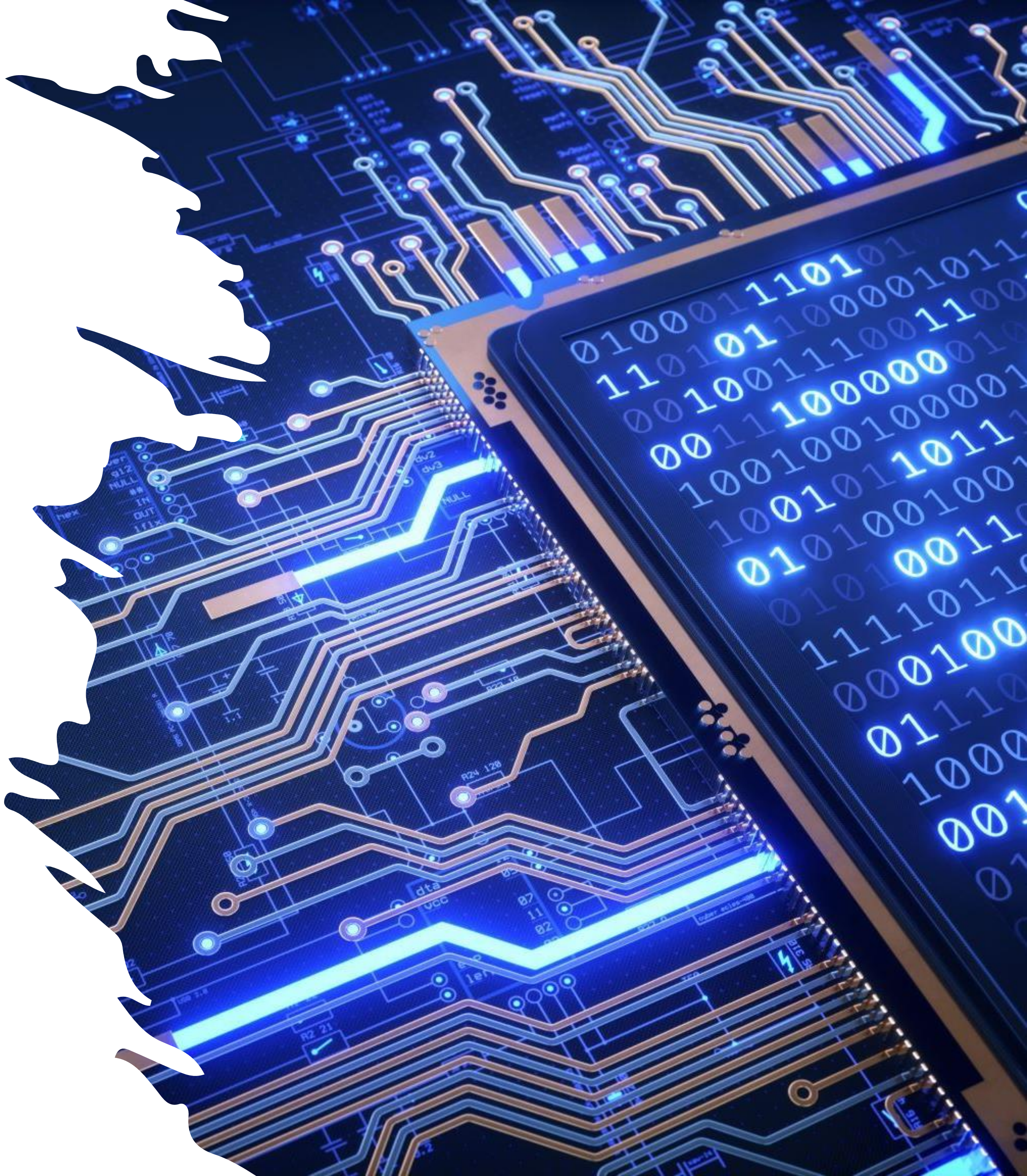
Docker Terminology

Environments -> container

Apps for the container -> image

Docker Basics

- What is Docker
- Worflow
- Docker files
- Docker Compose
- Docker vs VM



What is Docker

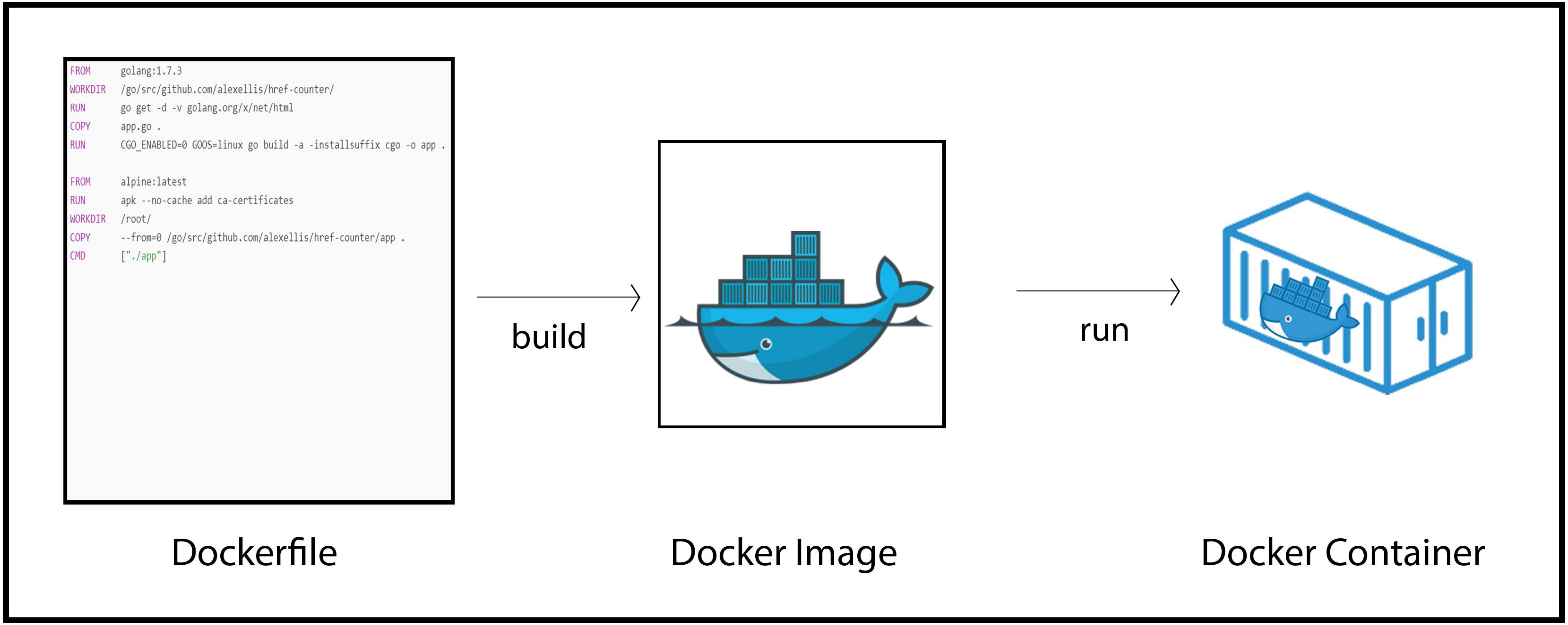
Allows to run apps in isolated container

Easy share of images

Easy share of containers

Can run multiple containers on one environment

Workflow



```
FROM      golang:1.7.3
WORKDIR   /go/src/github.com/alexellis/href-counter/
RUN       go get -d -v golang.org/x/net/html
COPY      app.go .
RUN       CGO_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -o app .

FROM      alpine:latest
RUN       apk --no-cache add ca-certificates
WORKDIR   /root/
COPY      --from=0 /go/src/github.com/alexellis/href-counter/app .
CMD       ["./app"]
```

build

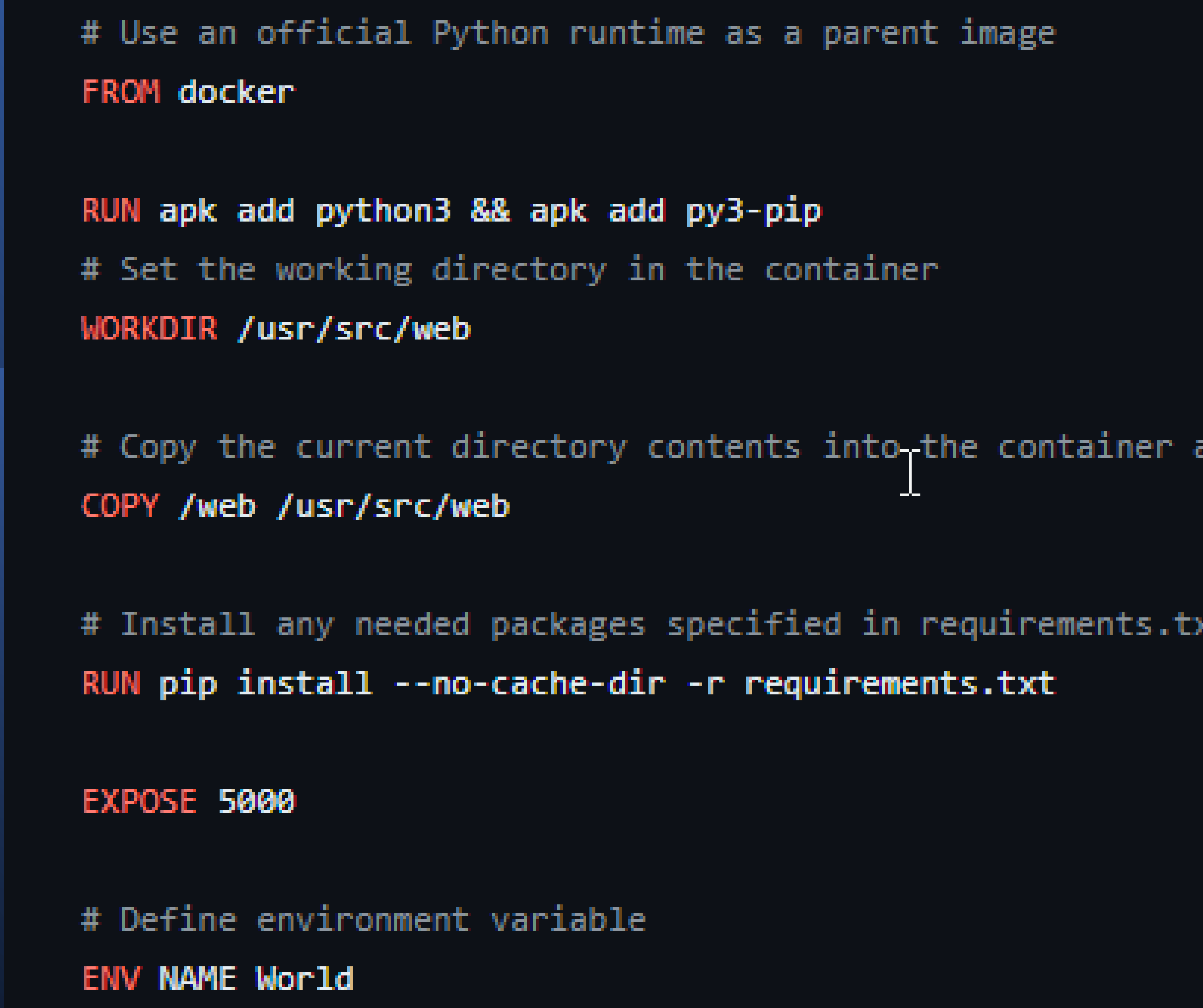
run

Dockerfile

Docker Image

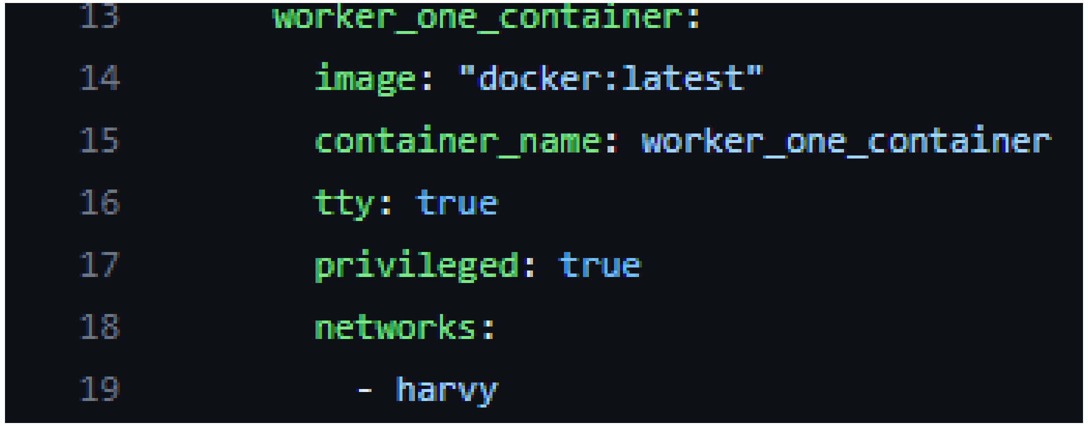
Docker Container

Docker Files

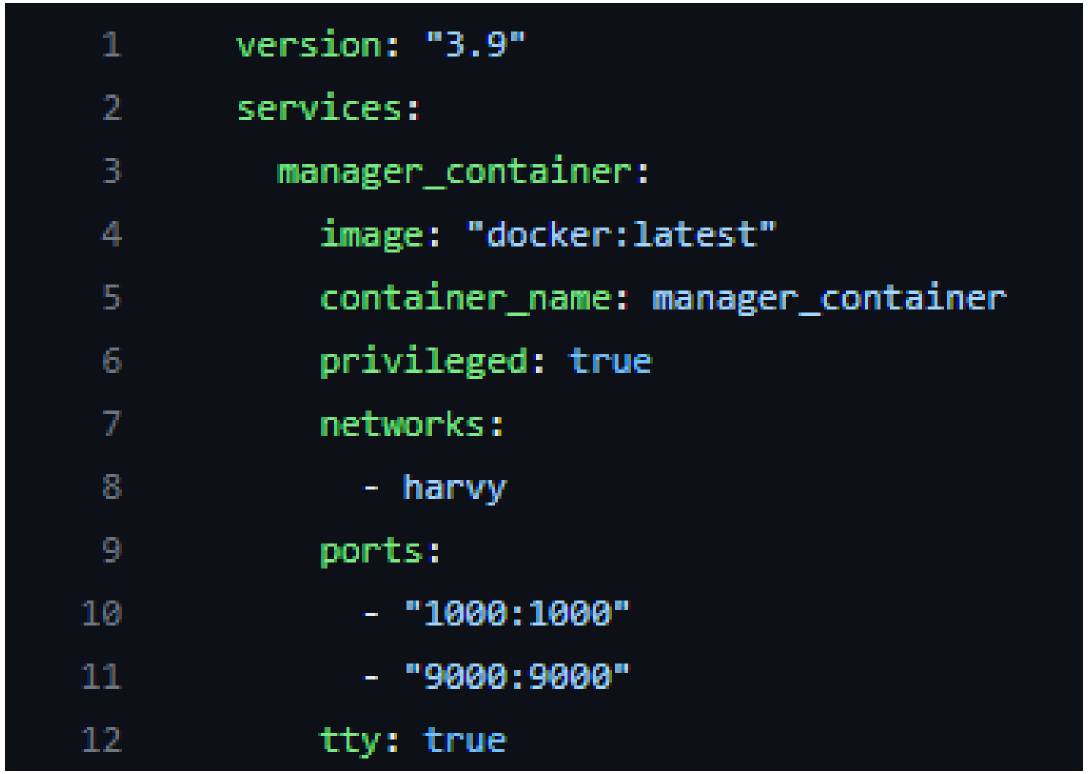


```
# Use an official Python runtime as a parent image
FROM docker

RUN apk add python3 && apk add py3-pip
# Set the working directory in the container
WORKDIR /usr/src/web

# Copy the current directory contents into the container a
COPY /web /usr/src/web

# Install any needed packages specified in requirements.tx
RUN pip install --no-cache-dir -r requirements.txt

EXPOSE 5000

# Define environment variable
ENV NAME World
```

Docker compose

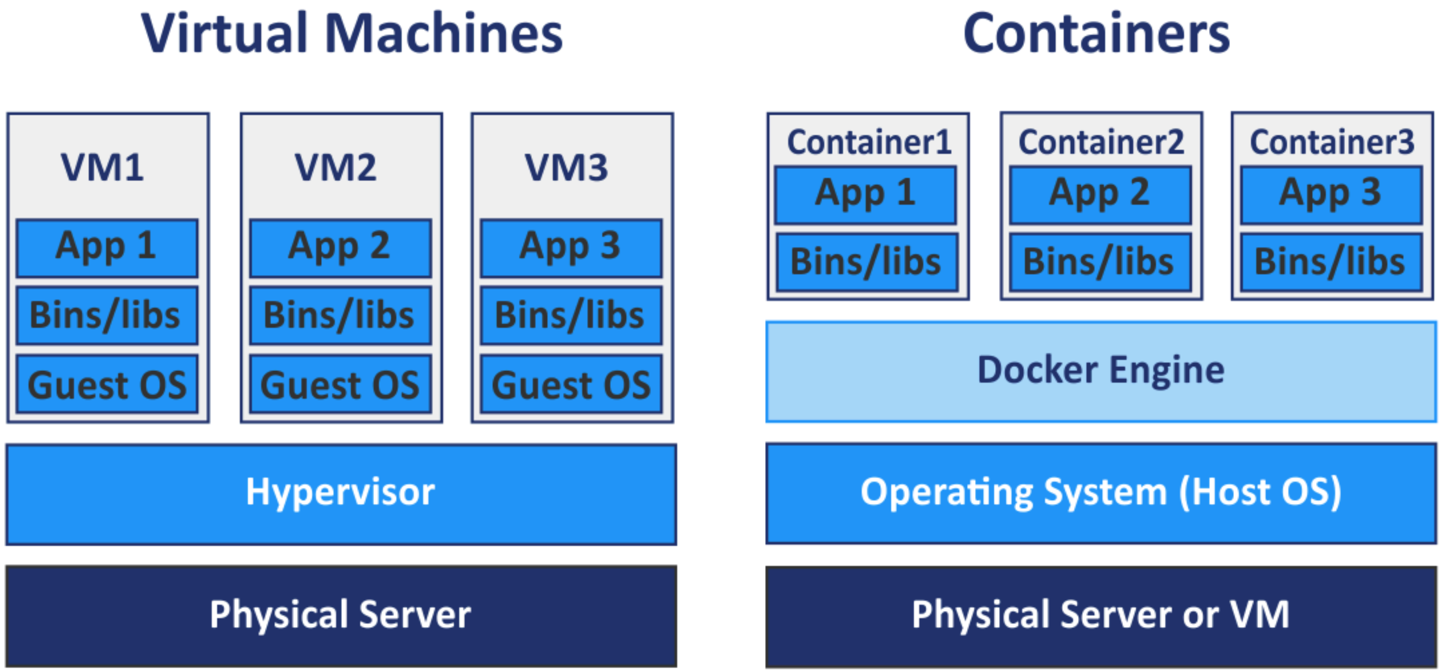


```yaml
  worker_one_container:
    image: "docker:latest"
    container_name: worker_one_container
    tty: true
    privileged: true
    networks:
      - harvy
```



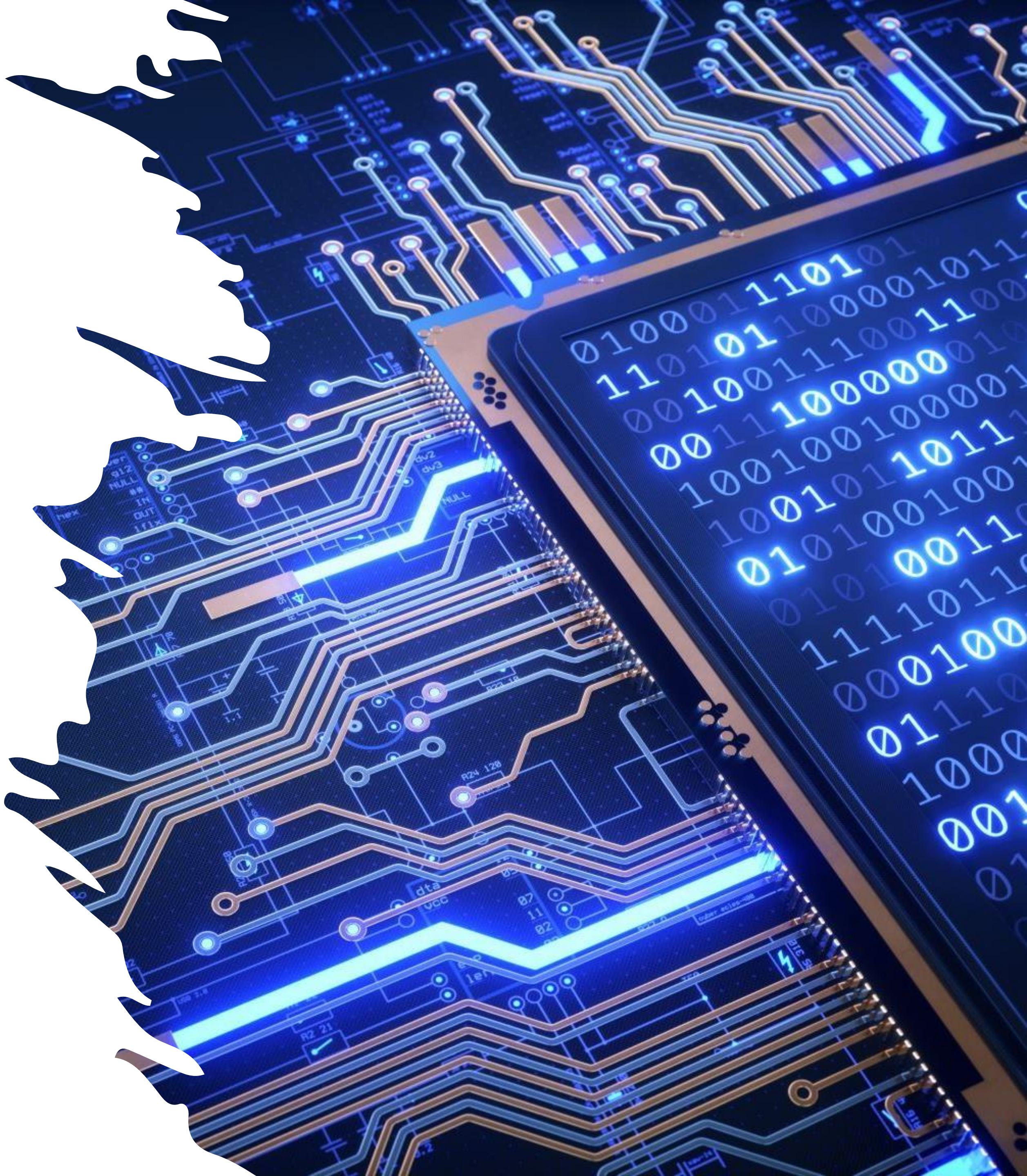
```yaml
version: "3.9"
services:
  manager_container:
    image: "docker:latest"
    container_name: manager_container
    privileged: true
    networks:
      - harvy
    ports:
      - "1000:1000"
      - "9000:9000"
    tty: true
```

Docker vs VM

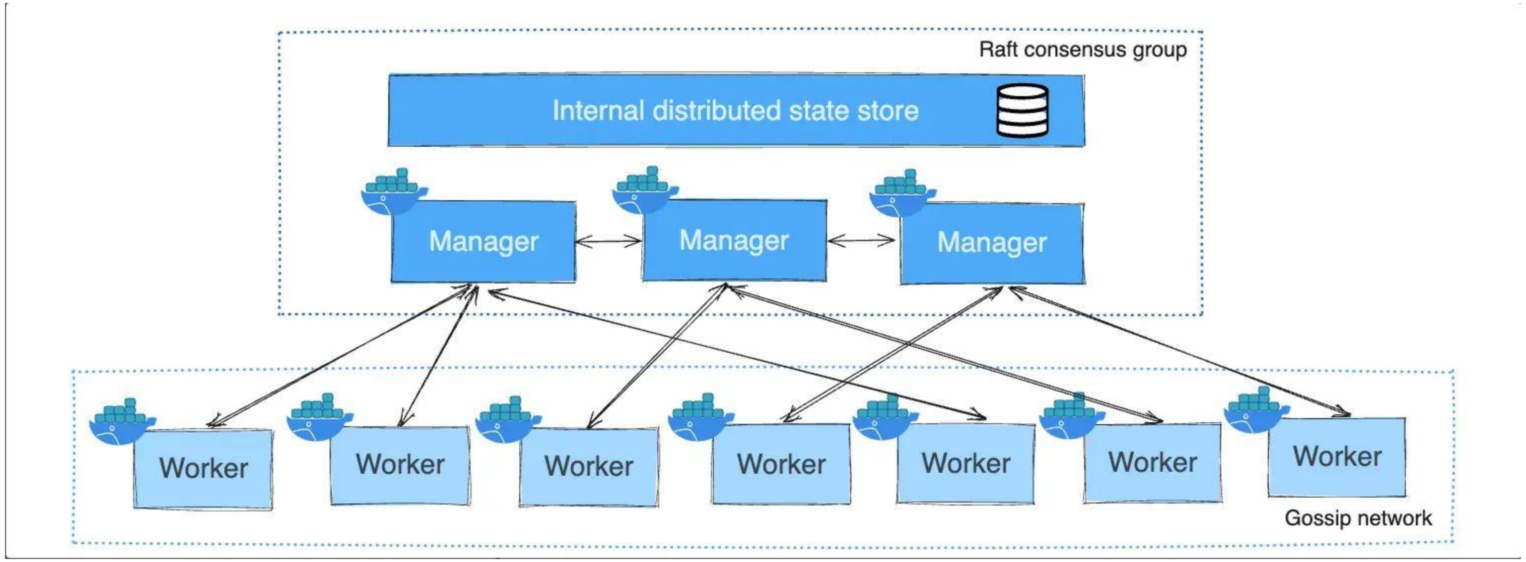

Virtual Machines
VM1
App 1
Bins/libs
Guest OS
VM2
App 2
Bins/libs
Guest OS
VM3
App 3
Bins/libs
Guest OS
Hypervisor
Physical Server
Containers
Container1
App 1
Bins/libs
Container2
App 2
Bins/libs
Container3
App 3
Bins/libs
Docker Engine
Operating System (Host OS)
Physical Server or VM

Swarm overview

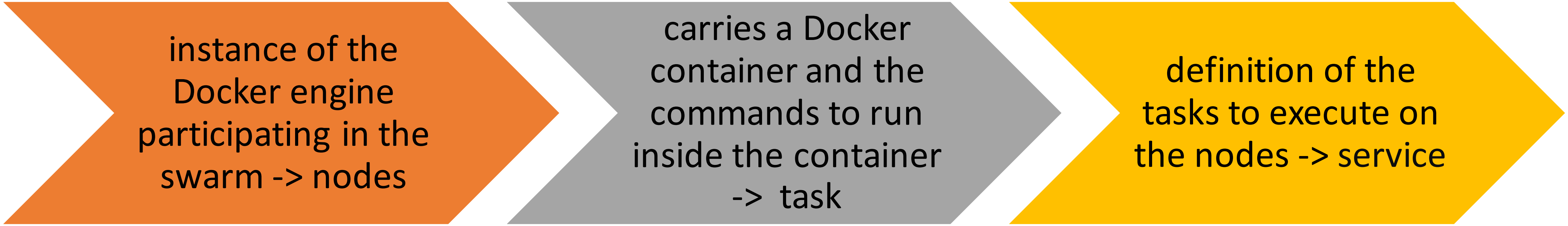
- Mode for Docker
- Based Nodes
- Services
- Tasks



Nodes


Raft consensus group
Internal distributed state store
Manager
Manager
Manager
Worker
Worker
Worker
Worker
Worker
Worker
Worker
Gossip network

Docker Terminology

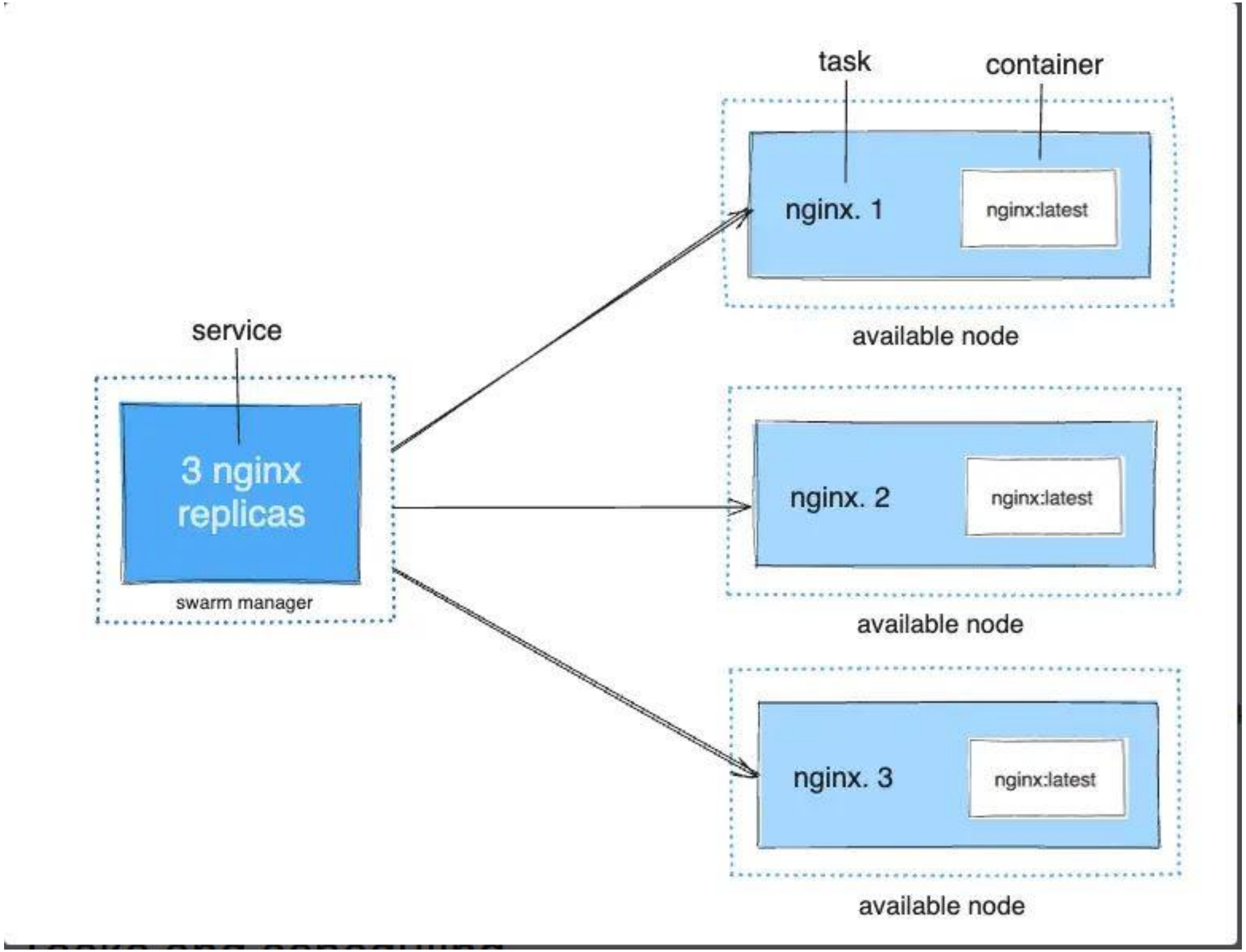


instance of the Docker engine participating in the swarm -> nodes

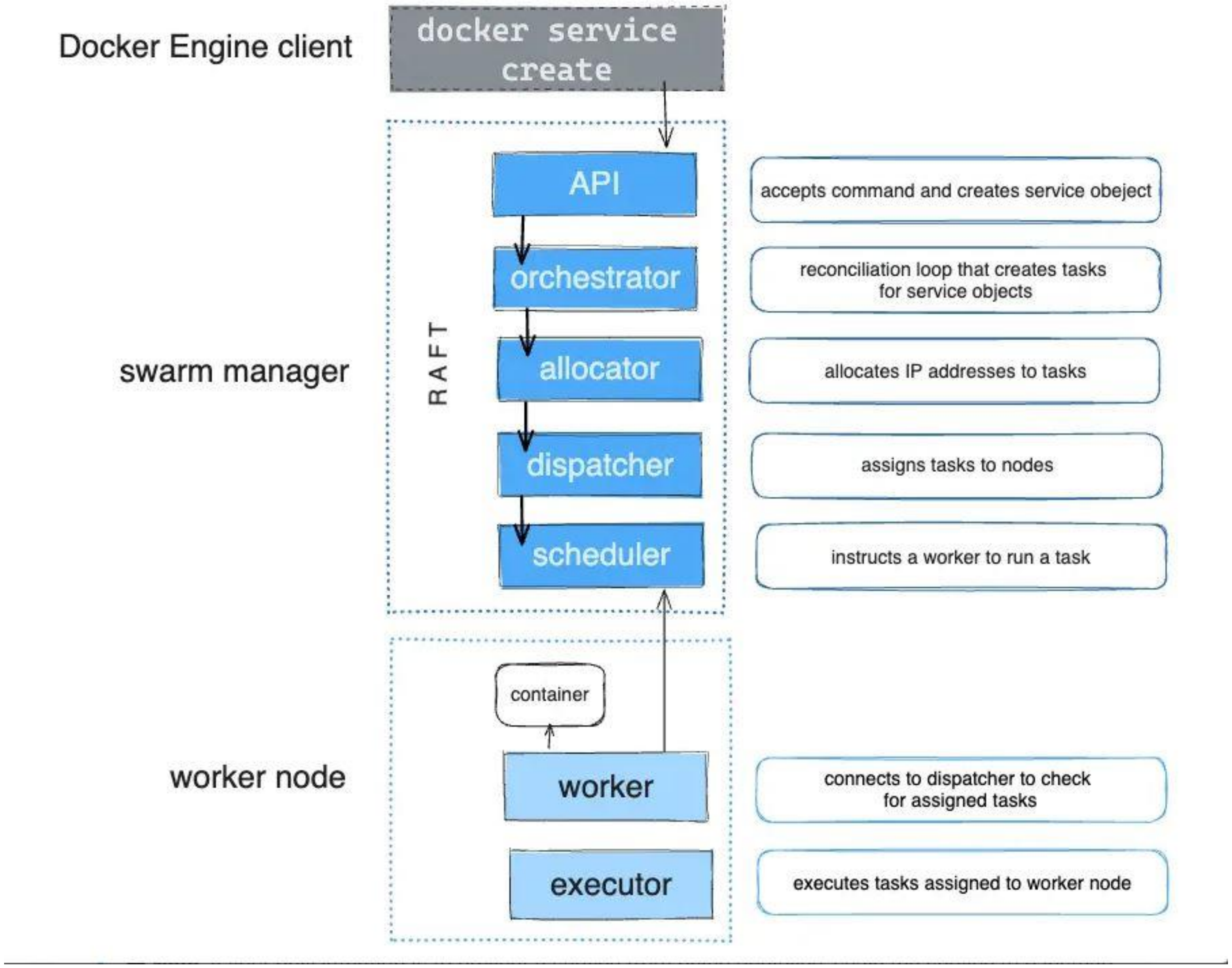
carries a Docker container and the commands to run inside the container -> task

definition of the tasks to execute on the nodes -> service

Services, tasks and container



Task and scheduling



Docker Engine client

```
docker service create
```

swarm manager

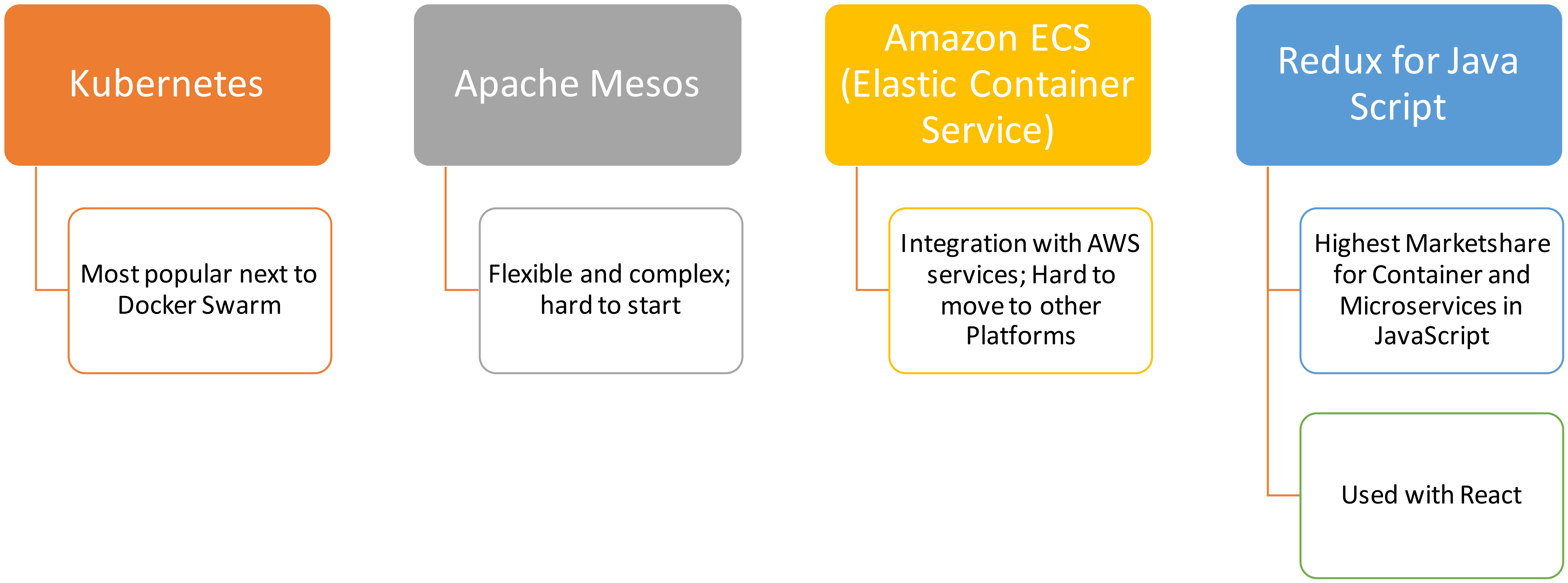
RAFT

- API: accepts command and creates service obeject
- orchestrator: reconciliation loop that creates tasks for service objects
- allocator: allocates IP addresses to tasks
- dispatcher: assigns tasks to nodes
- scheduler: instructs a worker to run a task

worker node

- container
- worker: connects to dispatcher to check for assigned tasks
- executor: executes tasks assigned to worker node

Competing Products



Kubernetes

Most popular next to Docker Swarm

Apache Mesos

Flexible and complex; hard to start

Amazon ECS (Elastic Container Service)

Integration with AWS services; Hard to move to other Platforms

Redux for Java Script

Highest Marketshare for Container and Microservices in JavaScript

Used with React

Swarm vs Kubernetes

Docker Swarm

- Designed for ease of use
- Manual scaling
- Easy deployments through swarm files
- Creates a network over all nodes
- Optimal for quick and easy deployment of simple apps

Kubernetes Cluster

- Complex processes and high learning curve
- Automated scaling
- Declarative and detailed definition for desired state
- Flat network model for peer to peer communication
- Optimal for high demand apps

Quiz

Quiz Link

Sources

- https://mesos.apache.org/documentation/latest/architecture/
- https://docs.aws.amazon.com/AmazonECS/latest/developerguide/Welcome.html
- https://redux.js.org/
- https://phoenixnap.com/blog/kubernetes-vs-docker-swarm
- https://www.freecodecamp.org/news/kubernetes-vs-docker-swarm-what-is-the-difference/#:~:text=The%20major%20difference%20between%20the,and%20Kubernetes%3A%20Installation%20and%20setup
- https://www.sumologic.com/blog/kubernetes-vs-mesos-vs-swarm/
- https://www.oodlestechnologies.com/dev-blog/ecs-vs-docker-swarm/