



# docker



## Docker Demo

## Cemal Erdemir



# İÇERİK

- Nereden çıktı bu docker (LXC, dotCloud)
- Neden popüler oldu (containers, wfm, microservice, devops, CI, CD)
- Arkasındaki teknolojiler (go, LXC, aufs)
- Jargon
- Docker CLI Demo (pull, run, build)
- Demo (load balancer arkasında web uygulaması ayaklandırmaca)

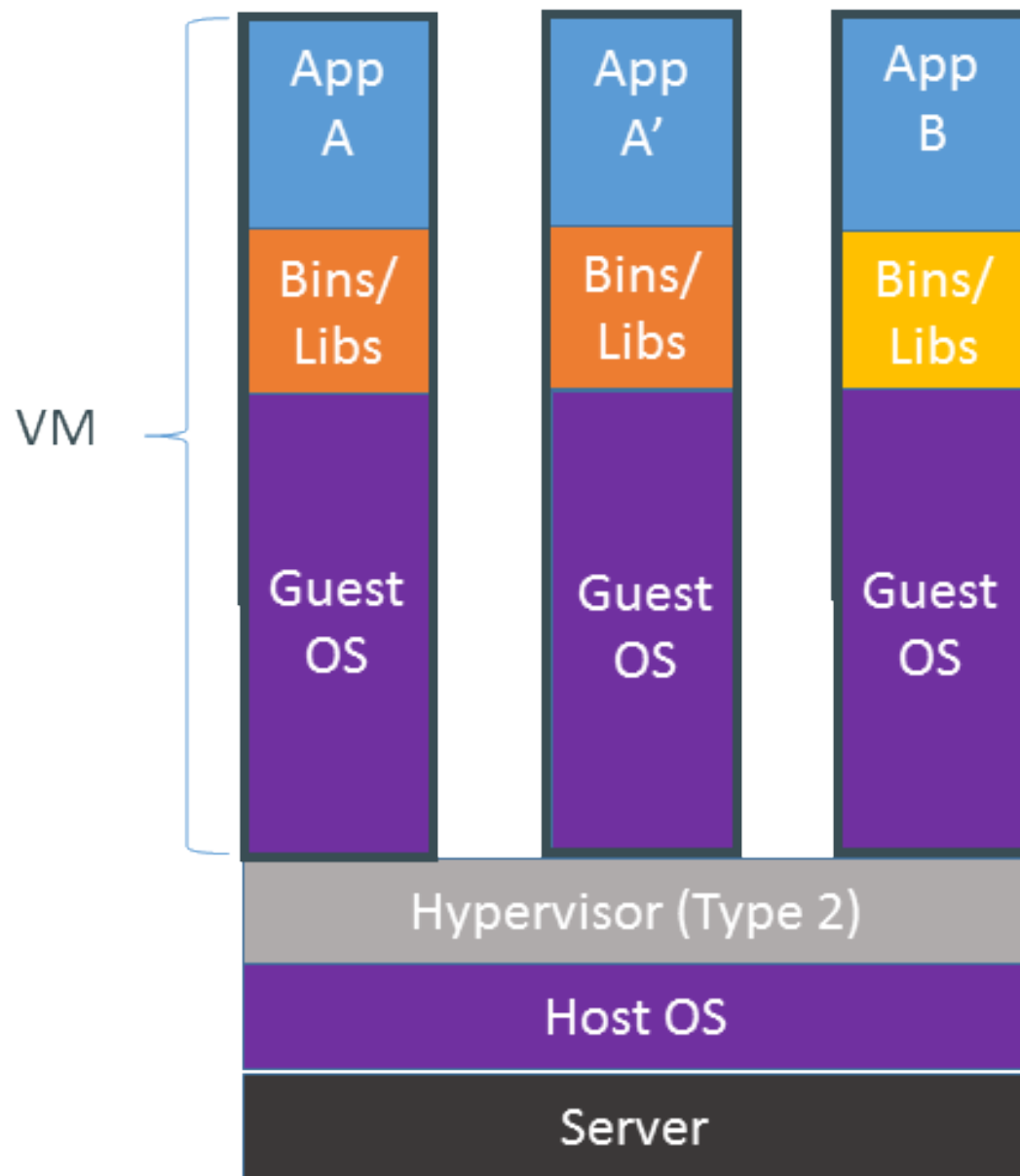
# NEREDEN ÇIKTI

- LXC
- dotCloud
- Solomon Hykes



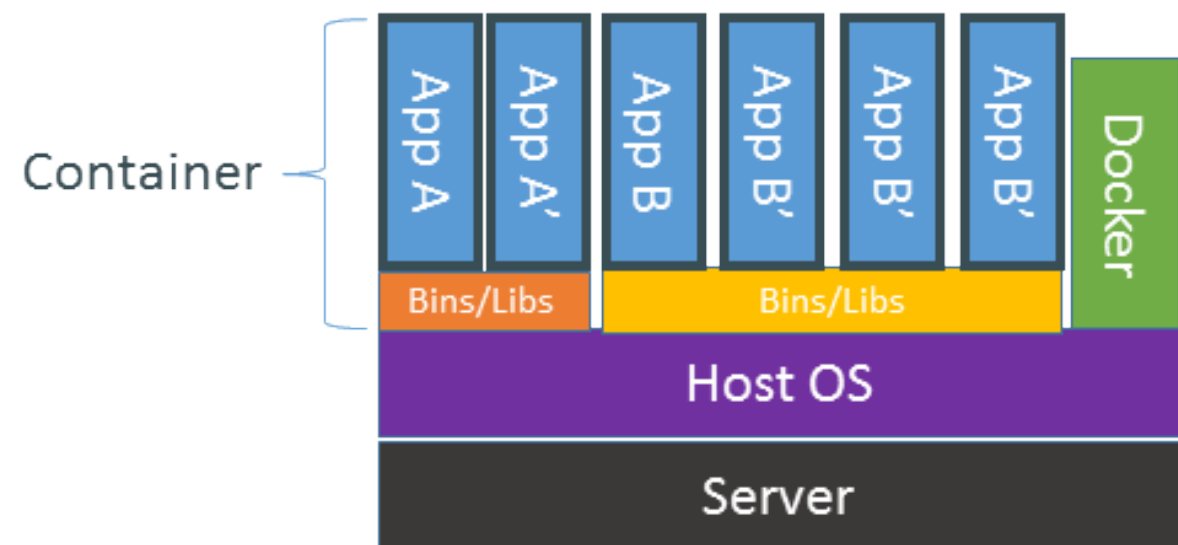
# NEDEN POPÜLER

## Containers



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

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

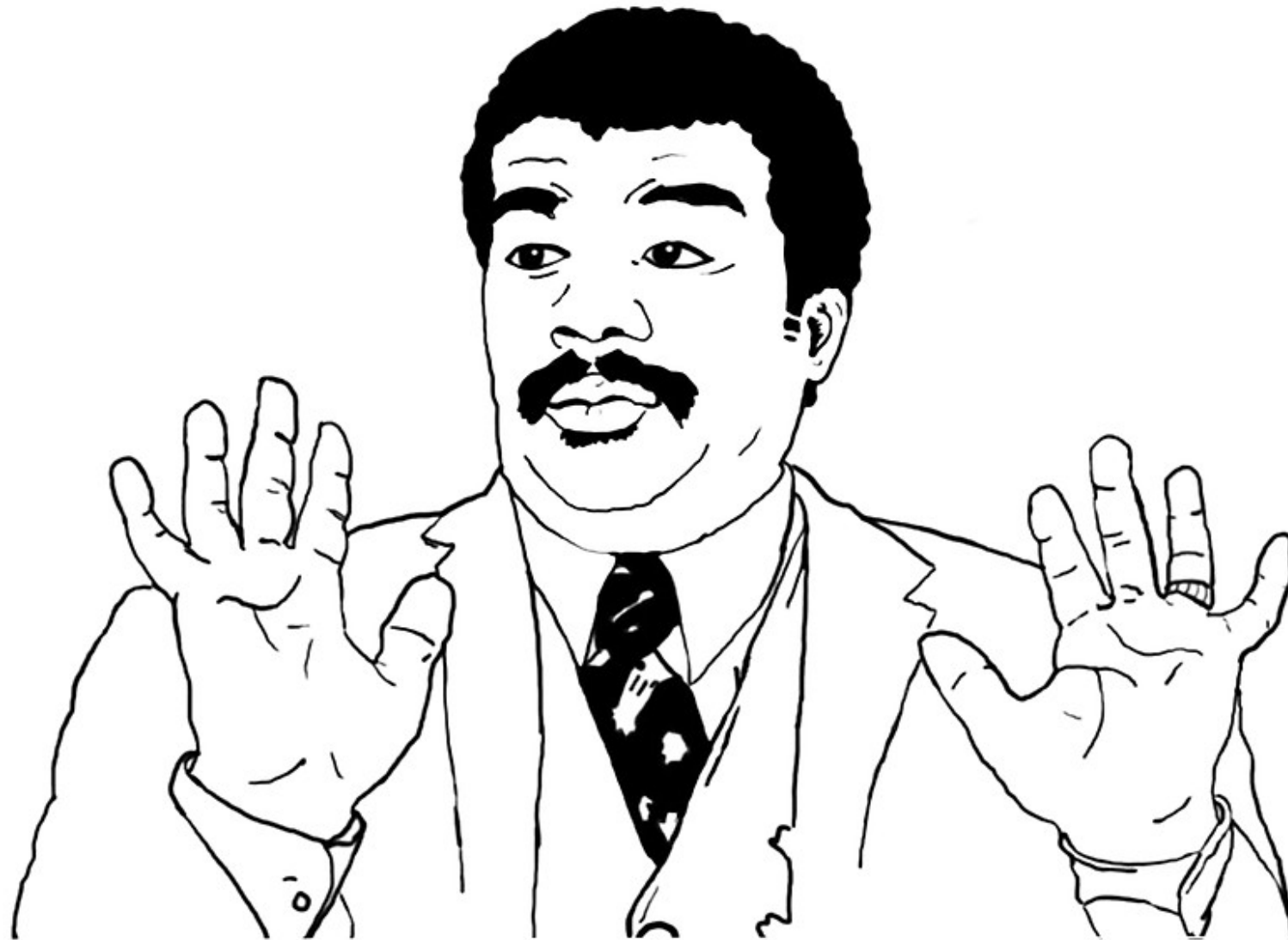


# VM Docker

- Değiştirilebilir
- HW seviyesi sanallaştırma
- Taşınması zor
- Bakımı zor
- Değiştirilemez
- OS seviyesi sanallaştırma
- Taşınması kolay
- Bakımı kolay

# NEDEN POPÜLER

It works on my  
machine



# NEDEN POPÜLER

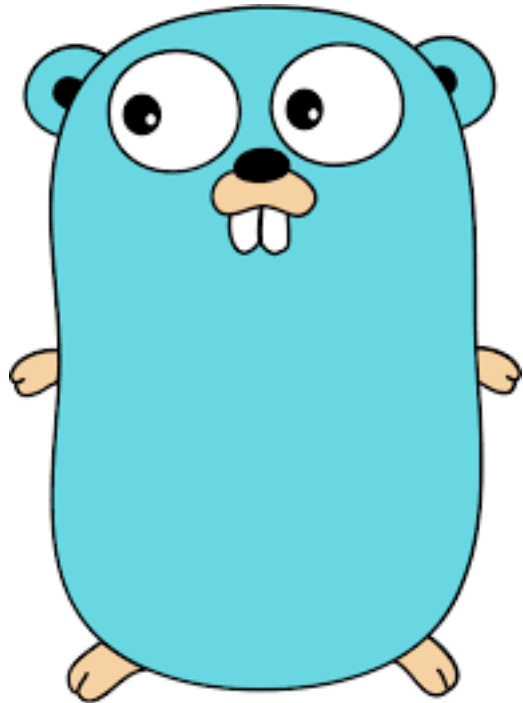
**Microservices**

**DevOps**

**CI**

**CD**

# TEKNOLOJI



**aufs**



# JARGON

**Docker Engine**

**Docker CLI, Kitematics**

**Run, build, pull**

**Container**

**Image**

**Dockerfile**

**Docker Compose**

**Docker Swarm**

# Dockerfile

```
FROM node:boron
```

```
WORKDIR /usr/src/app
```

```
COPY package.json .
```

```
RUN npm install
```

```
COPY . .
```

```
EXPOSE 8080
```

```
CMD [ "npm", "start" ]
```

# docker-compose

```
node1:
  environment:
    - SERVERNAME=node1
  build: ./node
  links:
    - redis
  ports:
    - "8080"
node2: ...
node3: ...
redis:
  image: redis
  ports:
    - "6379"
nginx:
  build: ./nginx
  links:
    - node1:node1
    - node2:node2
    - node3:node3
  ports:
    - "80:80"
```



# Docker CLI Demo

**docker pull hello-world**

**docker run hello-world**

**docker ps**

**docker ps -a**

**docker run 0aa160bf367a**

**docker start 0aa160bf367a**

**docker attach c8cc0b874dd2**

**docker exec -it c8cc0b874dd2 bash**

**docker images**

**docker inspect kibana**

**docker inspect 70e73a80c440**



# Demo

