

Sisteme distribuite

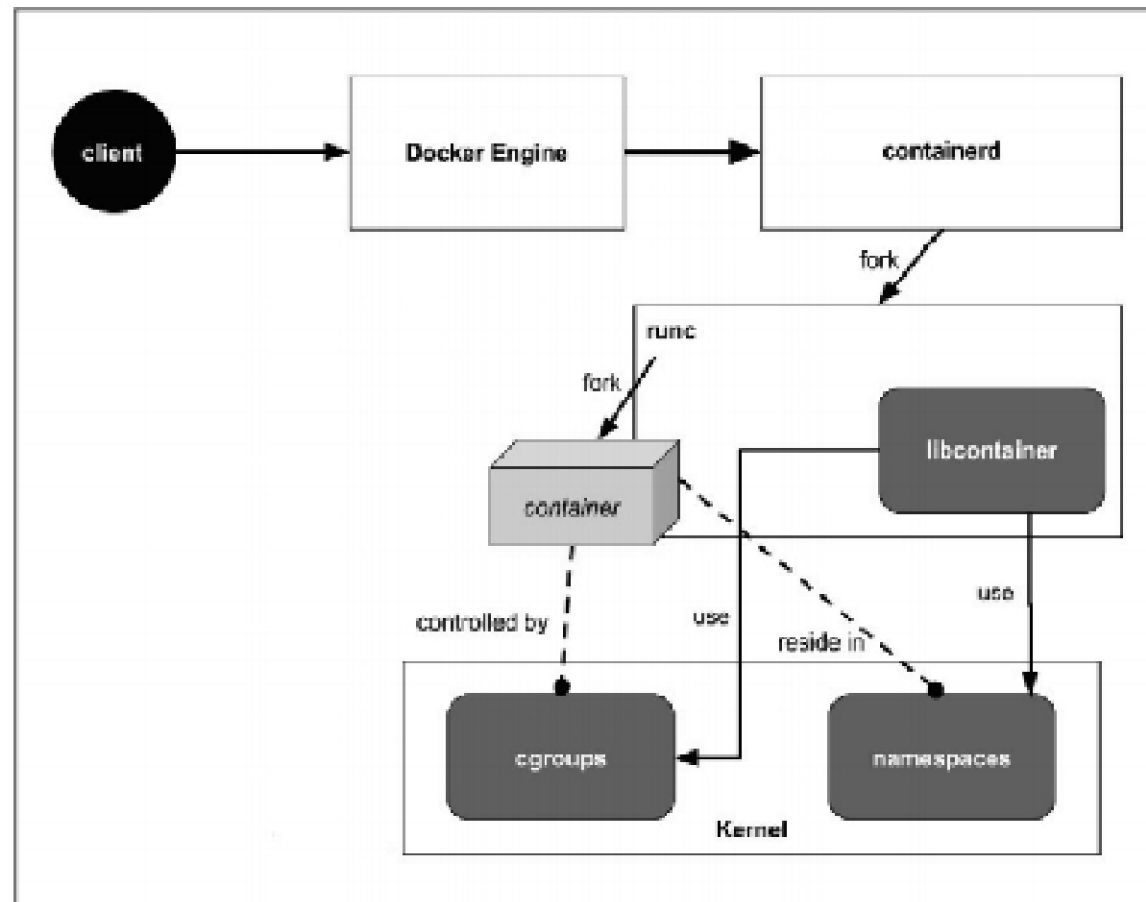
Mihai Zaharia

Cursul 8

Terminologie

- LXC
- LXD
- Docker

runC

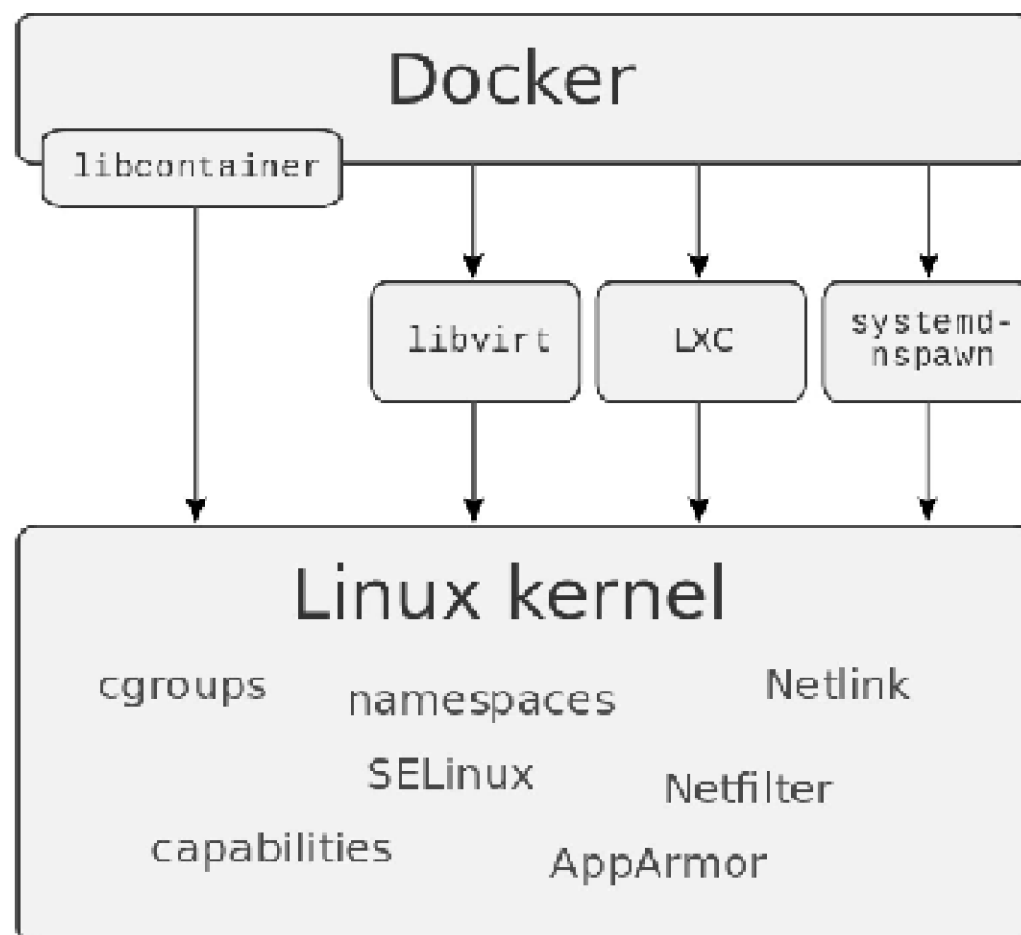


Ce oferă Linux pentru containerele docker

Detalii rețalie mașină Docker - Linux

Spații de lucru utilizate de mașina Docker

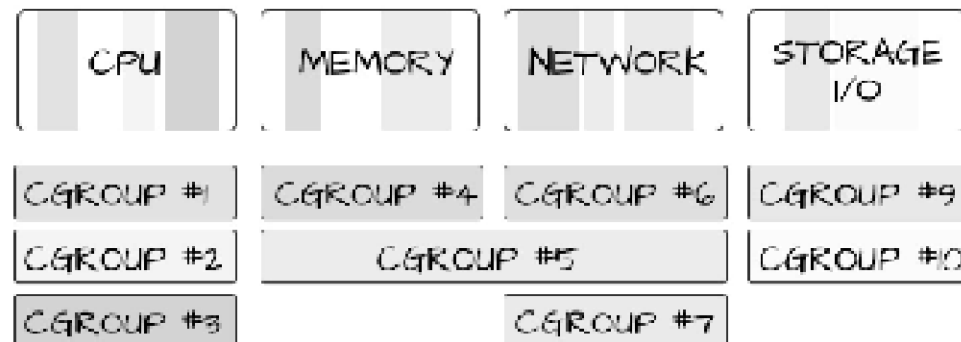
- PID
- NET
- IPC
- MNT
- UTS



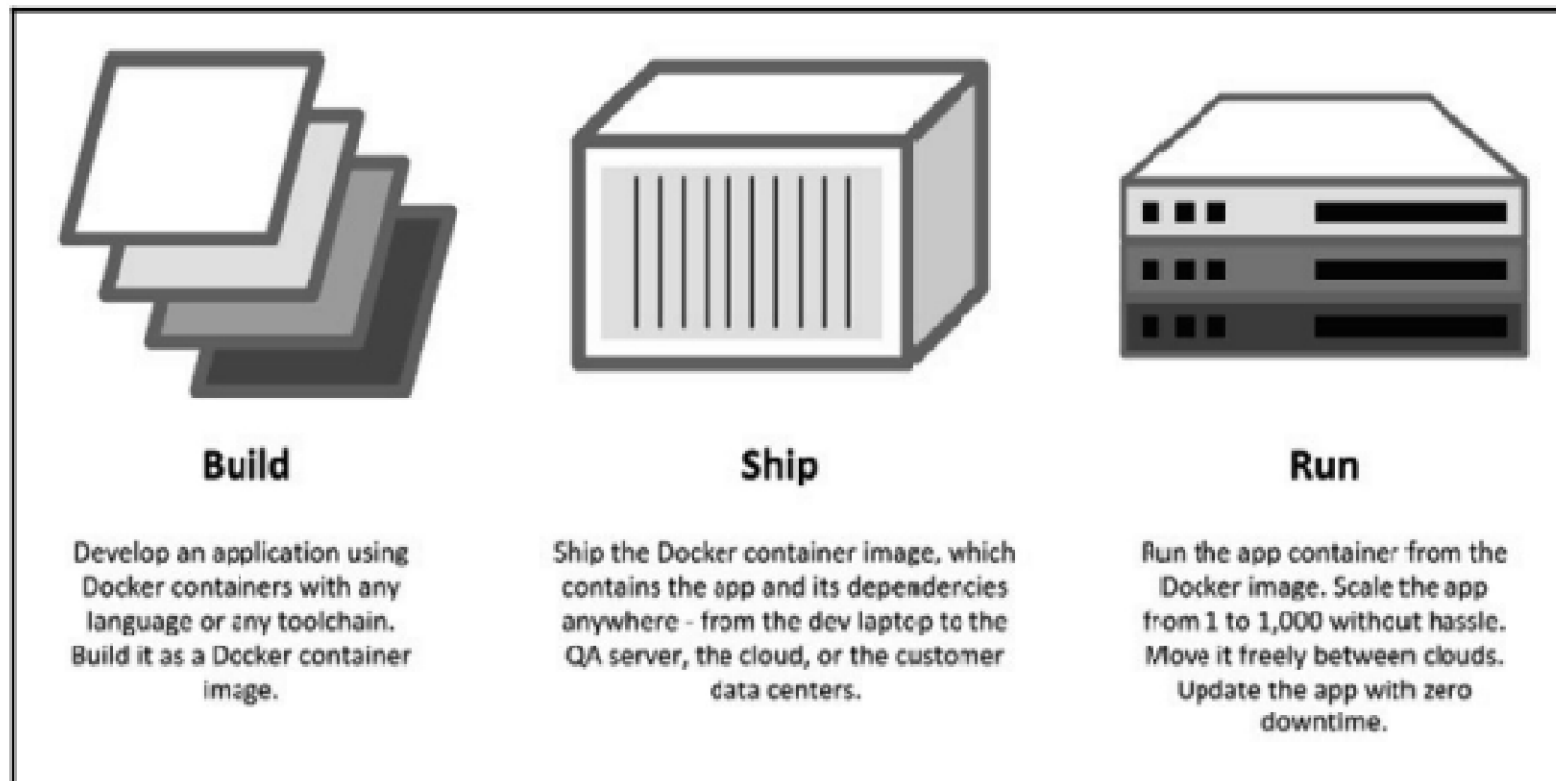
Izolarea resurselor - detalii

Cgroups - Izolare și enumerare resurse sistem

- cpu
- memory
- block i/o
- devices
- network
- numa
- freezer

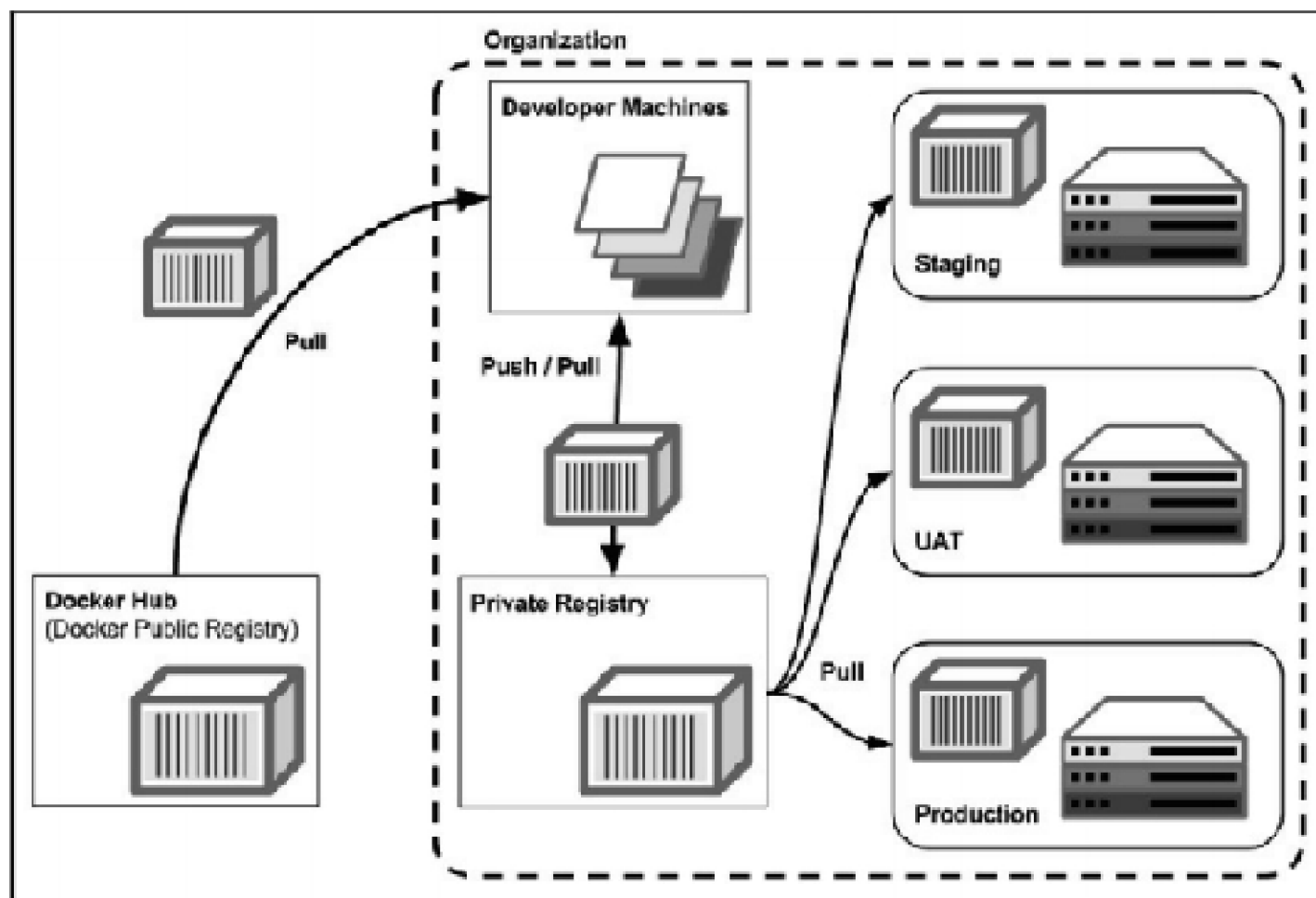


Filozofia Docker



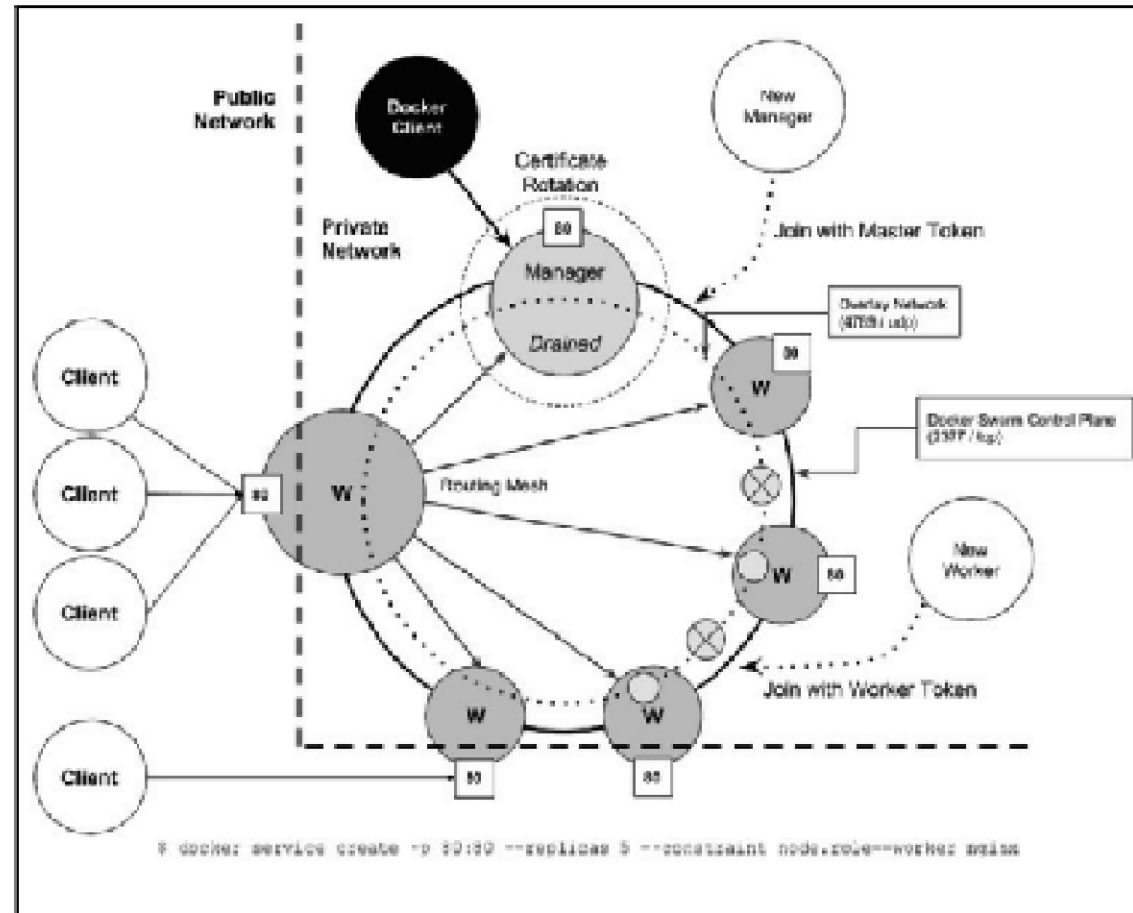
Build-Ship-Run

Gestiunea imaginilor Docker



procesul de push/pull

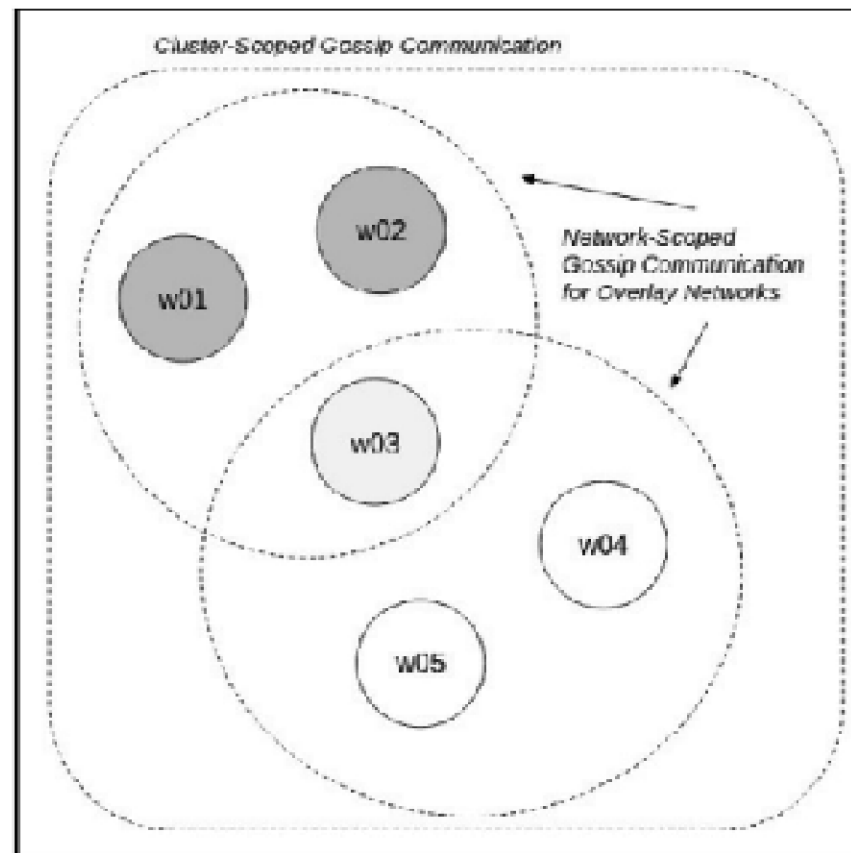
Servicii și sarcini



- `--replicas`
- `--name`
- *ingress*
- `docker node update --availability drain mg0`

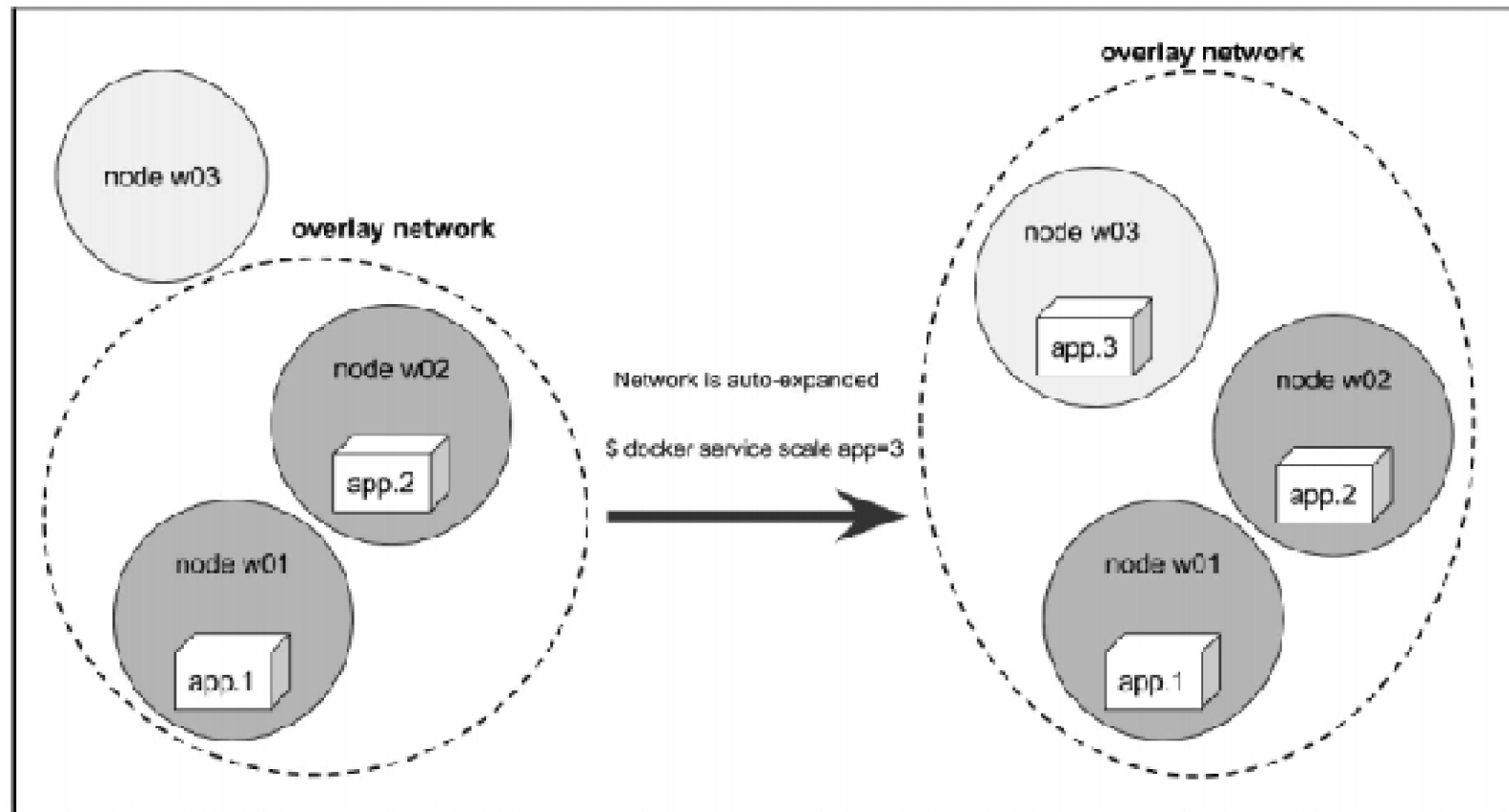
Bârfa în Docker (nu numai la Români)

- `docker inspect web`



mecanismul de bârfă (gossip) utilizat
pentru comunicarea inter-rețea în roi (swarm)

Multiplicarea/scalarea unui serviciu



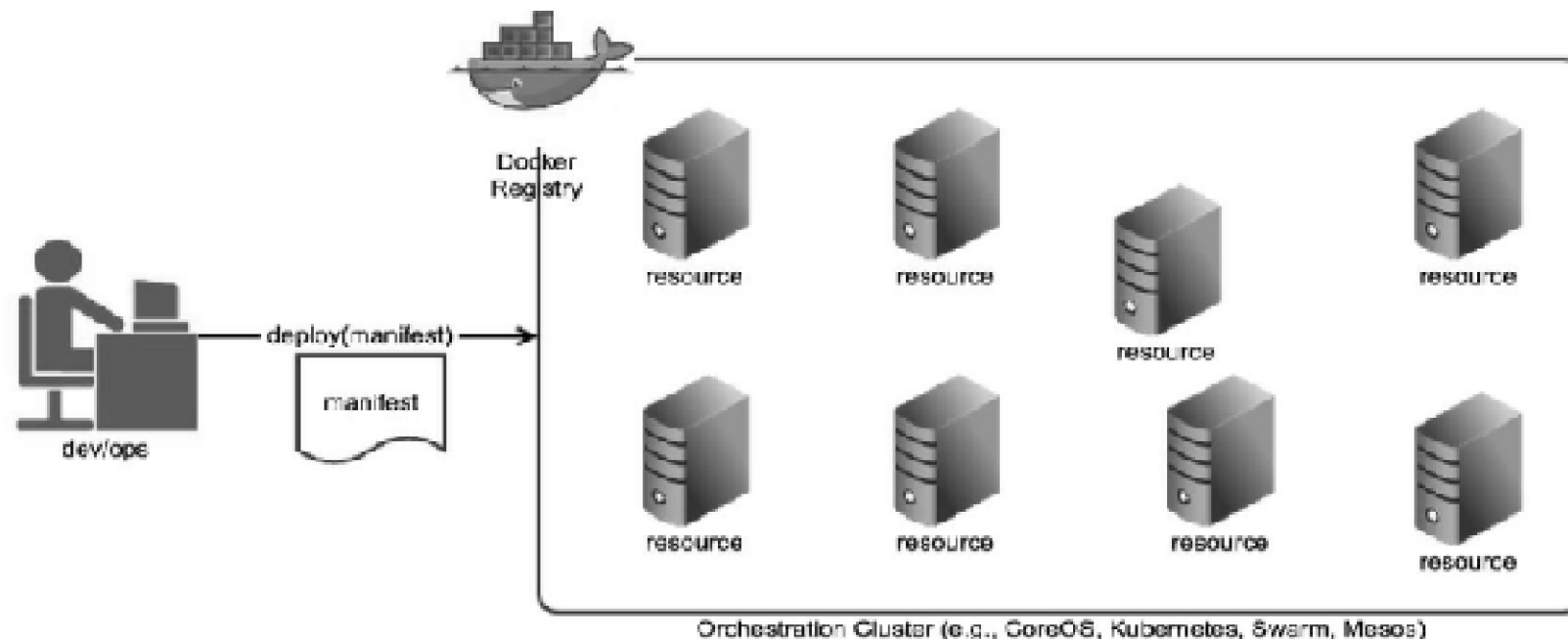
replicare în swarm
docker service scale

Terminologie

- POD
- POD

Chiar este la modă virtualizarea?

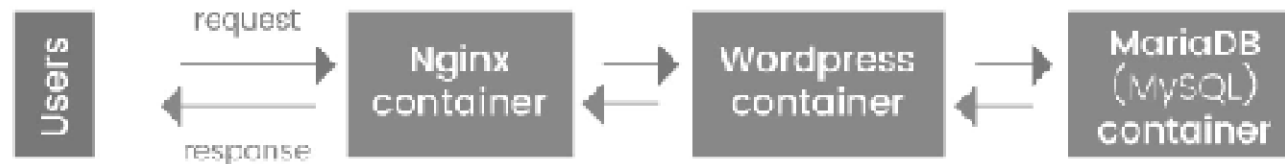
Lansarea unui container Docker



pe baza instrucțiunilor din manifest

Cum se pot containeriza microservicii

Dockerized App (microservice)



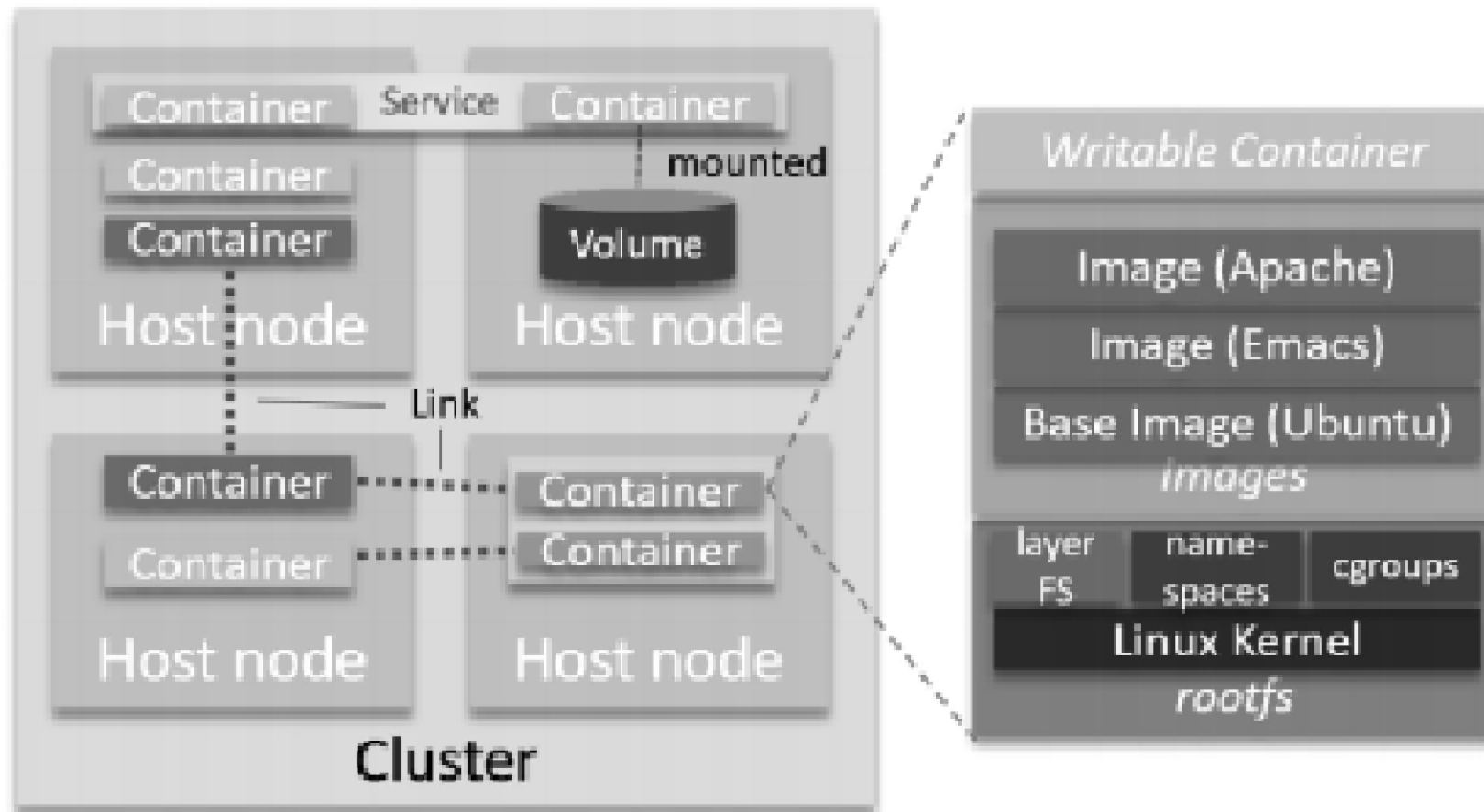
docker-compose.yml

- version: '2'
- services:
- nginx:
- build: nginx
- restart: always
- ports:
- - 8080:80
- volumes_from:
- - wordpress
- wordpress:
- image: wordpress:php7.1-fpm-alpine
- environment:
- WORDPRESS_DB_HOST: mysql
- WORDPRESS_DB_PASSWORD: example
- mysql:
- image: mariadb
- environment:
- MYSQL_ROOT_PASSWORD: example
- volumes:
- - ./demo-db:/var/lib/mysql

deci în esență cum fac o aplicație

1. creez un microserviciu într-un limbaj/tehnologie
2. îl testez
3. creez o imagine de container
4. creez o compoziție de servicii conform proiectării anterioare (model arhitectural + combinații de modele de proiectare specifice)

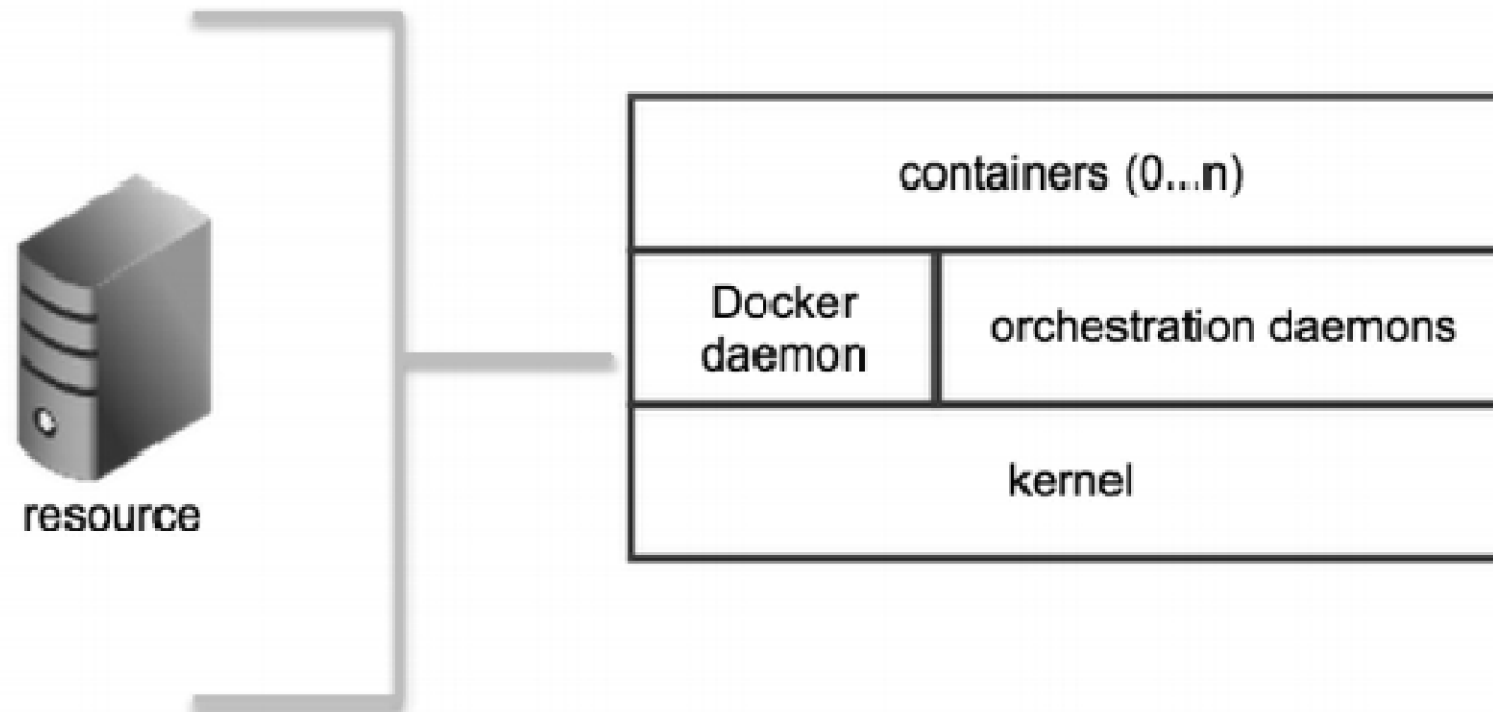
Cluster de containere



Containere în nor

- gestiune dependențe
- agenți pe ciclul de afaceri
- PaaS & containere

Cum intervine orchestrarea?

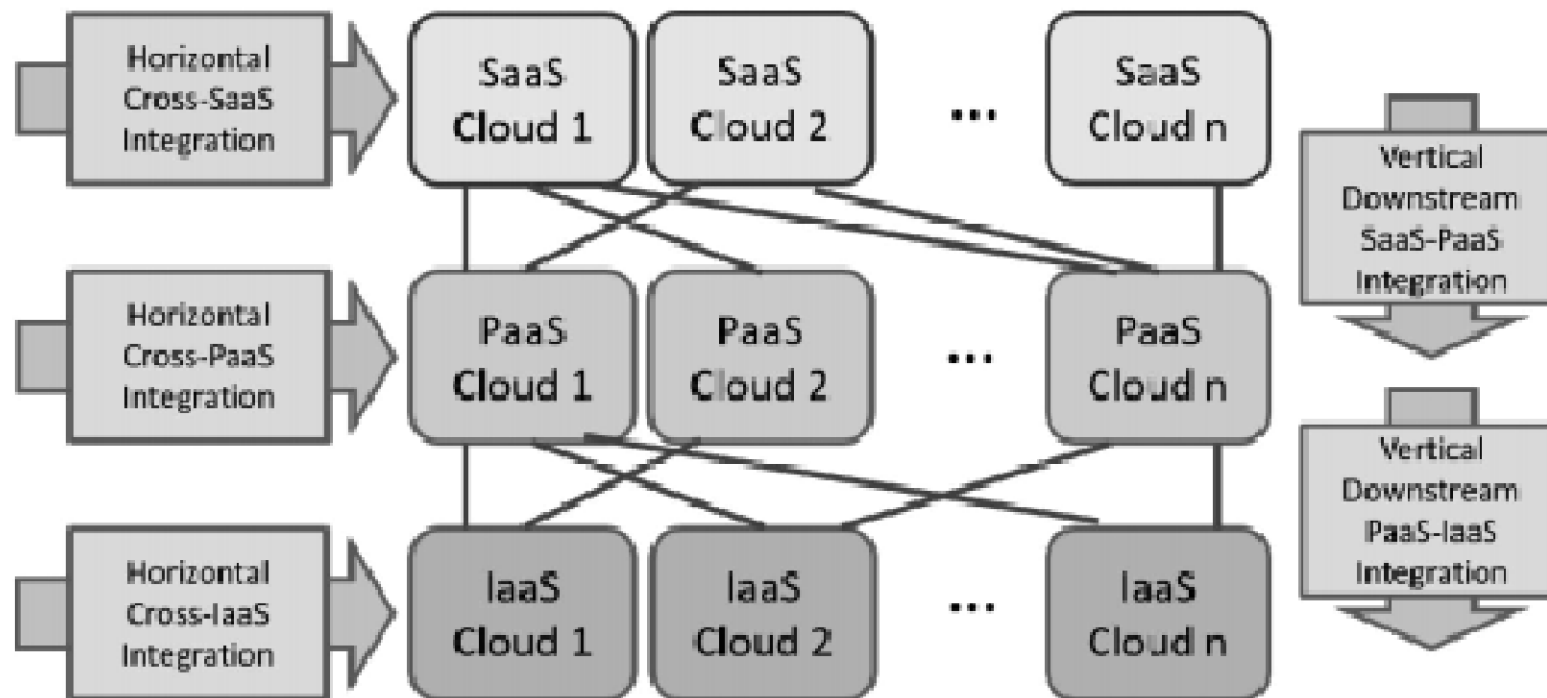


- X

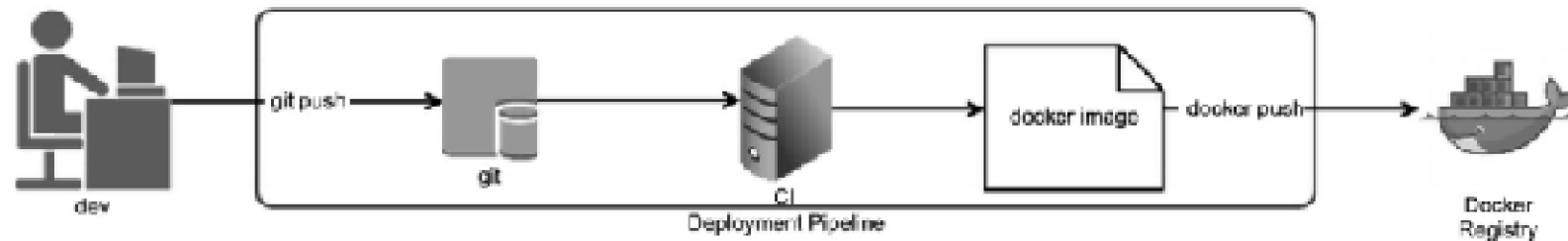
Orchestrarea

- asigurarea resurse
- instanțele
- replanificări
- relația cu interfața
- expunere servicii
- Mesos, Kubernetes, CorCos Tectonic si Docker Swarm

Arhitectura containerelor în nor



Dezvoltare continuă în Docker



Linie de productie (pipeline deployment) bazată pe containere

Docker SWOT - Pro

Configurare simplificată

Docker SWOT - Pro

Livrare continuă (Pipeline)

Docker SWOT - Pro

Productivitatea dezvoltatorilor

Docker SWOT - Pro

Izolarea Aplicațiilor

Docker SWOT - Pro

Gruparea/Consolidarea Serverelor

Docker SWOT - Pro

Abilități de depanare

Docker SWOT - Pro

Aplicații cu mai multi clienți diferiți
(multi-tenant applications)

Docker SWOT - Pro

Dezvoltare rapidă

Docker SWOT - Contra

Nu utilizați docker dacă doriți o creștere de performanță

Docker SWOT - Contra

Nu utilizați docker dacă aplicația are constrangeri serioase de securitate

Docker SWOT - Contra

Nu utilizati docker daca se dezvoltă o aplicație desktop cu
GUI

Docker SWOT - Contra

Nu utilizați docker dacă doriți un mediu simplu de dezvoltare și care să aibă și facilități de depanare pas cu pas

Docker SWOT - Contra

Nu utilizați docker dacă se dorește ca țintă alt sistem de operare sau kernel decât cel de bază (acum pe win a început să meargă cumva)

Docker SWOT - Contra

Nu utilizați docker dacă trebuie salvate multe date
importante

Docker SWOT - Contra

- Nu utilizați docker dacă aveți nevoie de o tehnologie ușor de controlat

Compozitorul Docker

- YAML - descriere
- integrare DevOps sau DevSecOps

Utilizarea Docker Compose

- definesc mediul
- definesc serviciile
- reguli suplimentare
- se lanseaza în execuție

docker-compose.yml

1. version: '3'
2. services:
3. web:
4. build: .
5. ports:
6. "5000:5000"
7. volumes:
8. ./code
9. - logvolume01:/var/log
10. inks:
11. - redis
12. redis:
13. image: redis
14. volumes:

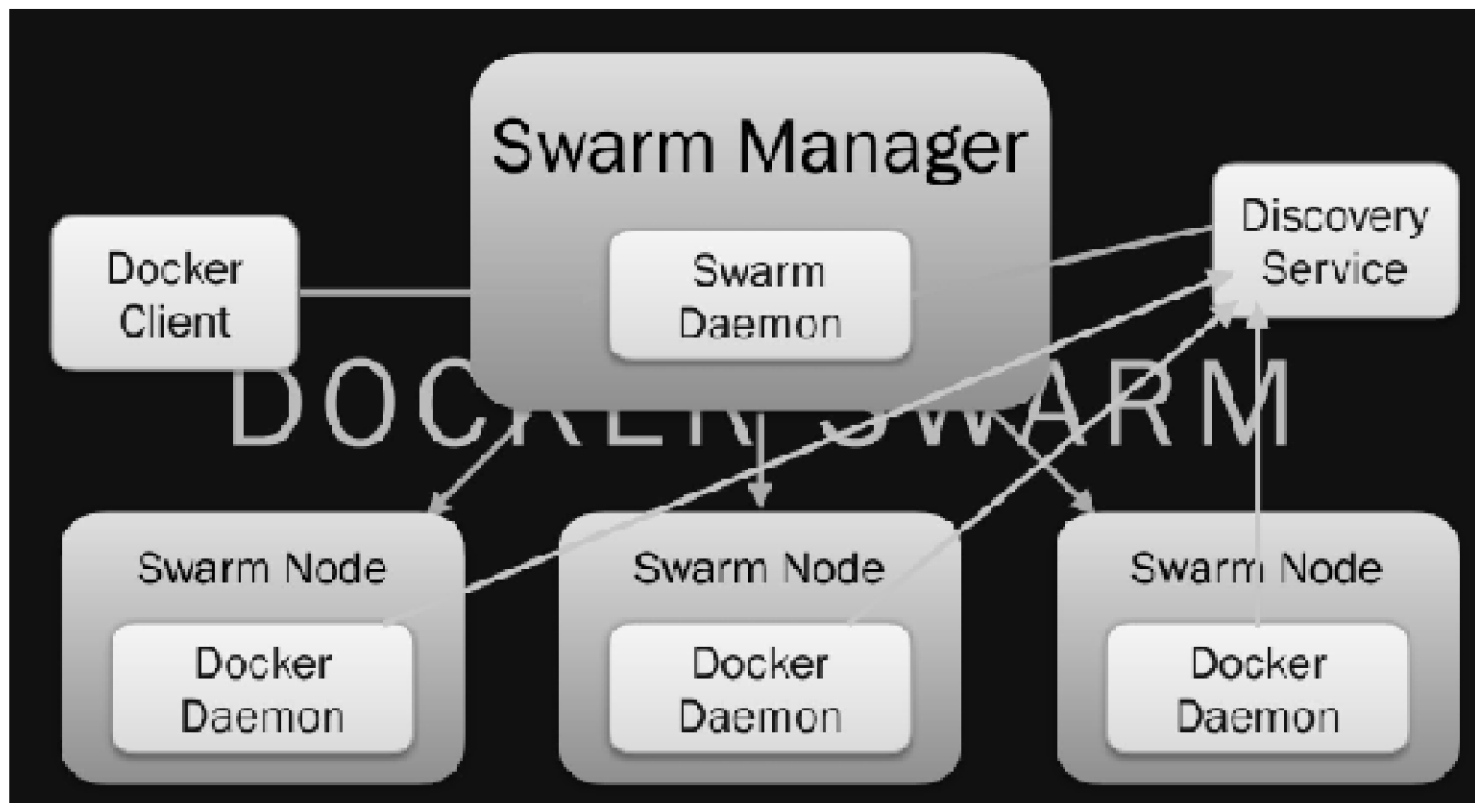
Docker Compose

- Permite multiple medii de lucru izolate pe o singura masina gazda
- Pastreaza volumele de date in momentul crearii containerelor
- Recreaza numai containerele care s-au modificat
- Variabile si posibilitatea schimbarii compozitiilor dintr-un mediu intr-altul

Cand se foloseste Compose?

- dezvoltare
- testare automată
 - \$ docker-compose up -d
 - \$./run_tests
 - \$ docker-compose down
- gazdă unică

Orchestrarea containerelor



Docker Swarm

Ce este un roi/cluster în docker

cluster

roi

Facilități oferite de Docker Swarm

- Management
- Scalarea
- Rețea între gazde
- Descoperirea serviciilor
- Echilibrarea încărcării - Load balancing
- Nodul din roi

Servicii și sarcini in Docker swarm

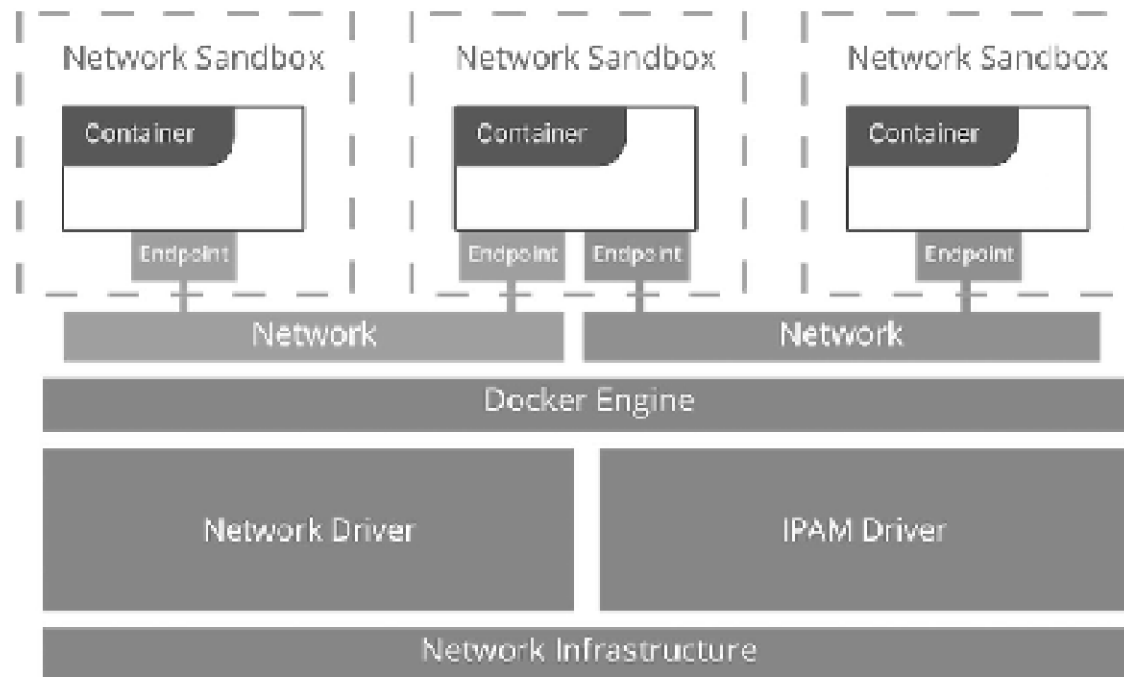
- serviciu
- sarcină/task
- planificare statică

Echilibrarea încărcării în Docker swarm

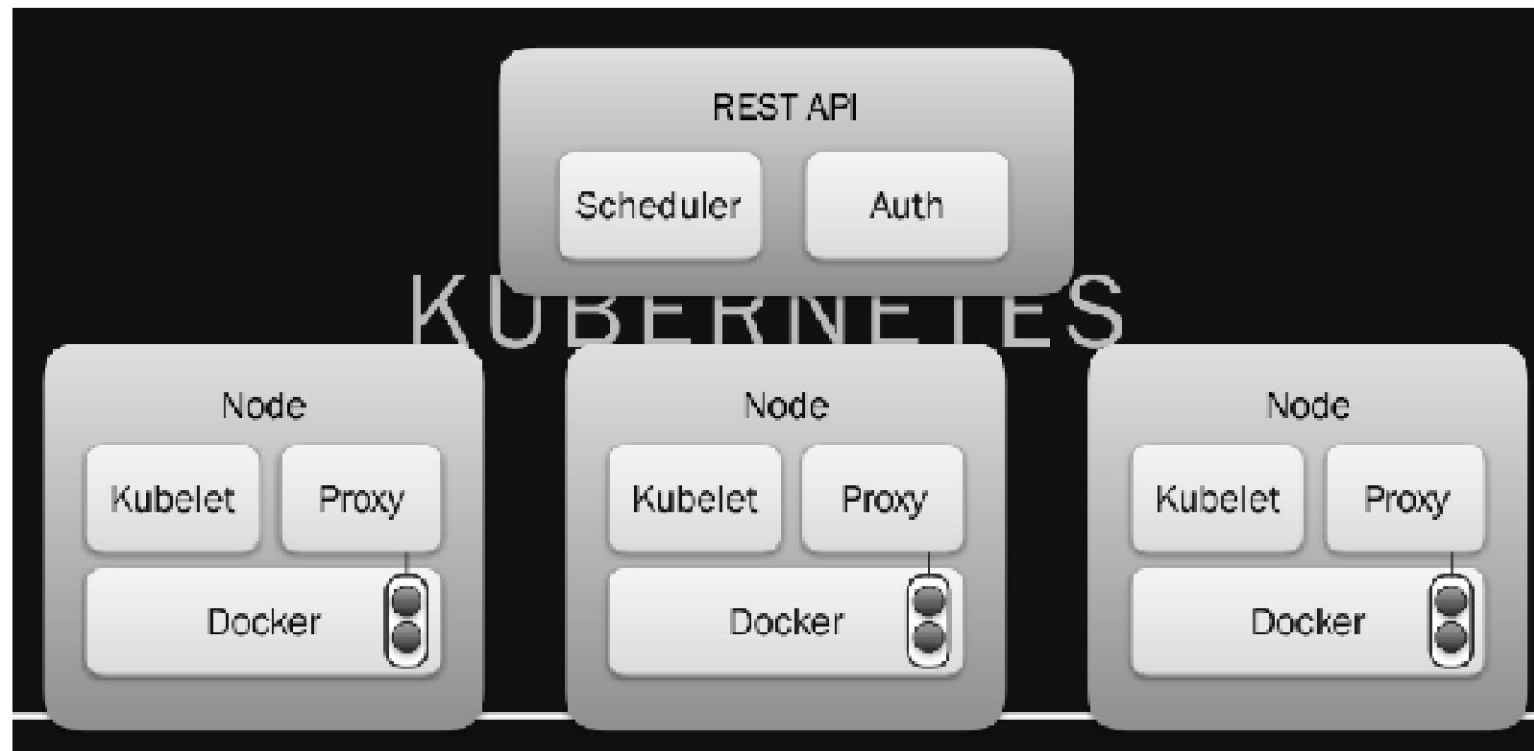
- port
 - manual
 - automat 30000-32767
- DNS local

Filtrele Docker swarm

- Contrangere sau marcaj nod
- Afinitate
- Portul
- Dependența
- Sănătatea

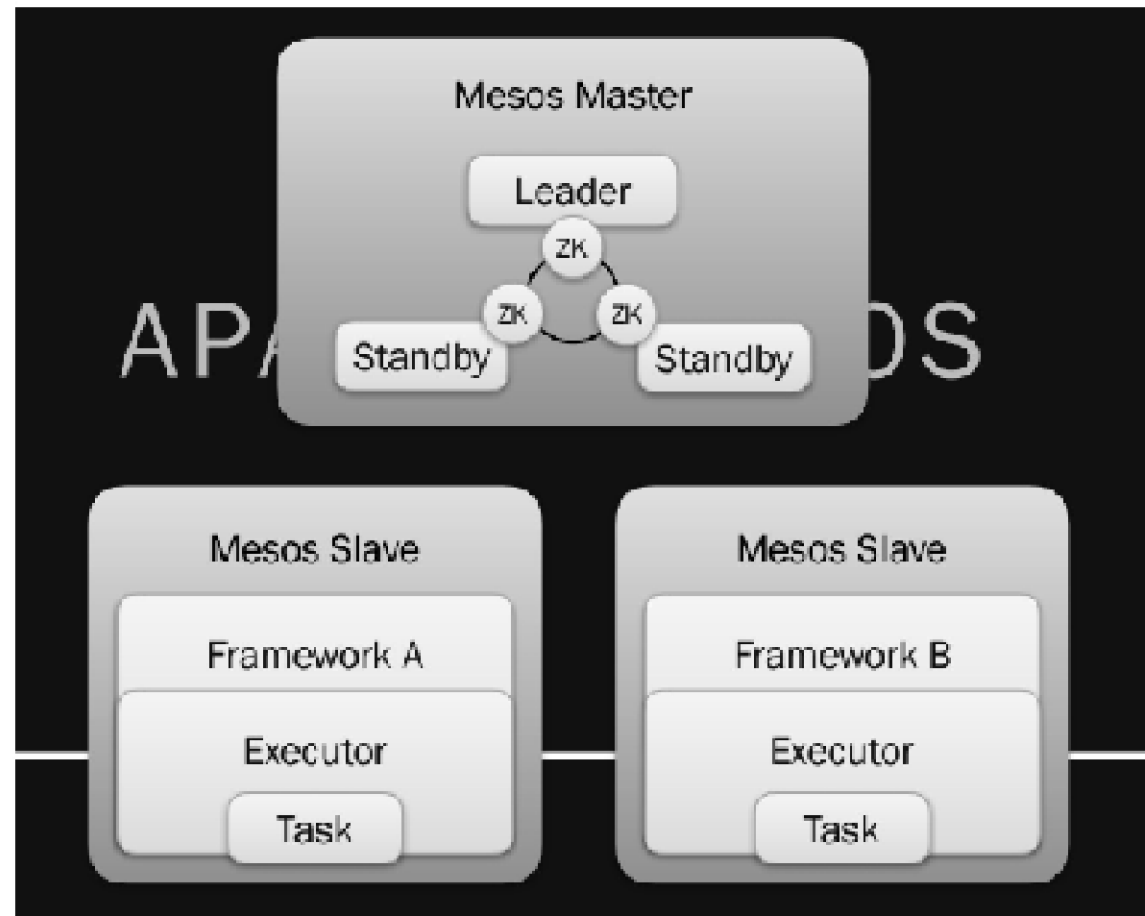


Alte soluții pentru orchestrarea containerelor



Kubernetes

Alte soluții pentru orchestrarea containerelor



Apache Mesos