SYSTEMS LABORATORY

RL crunch it connected

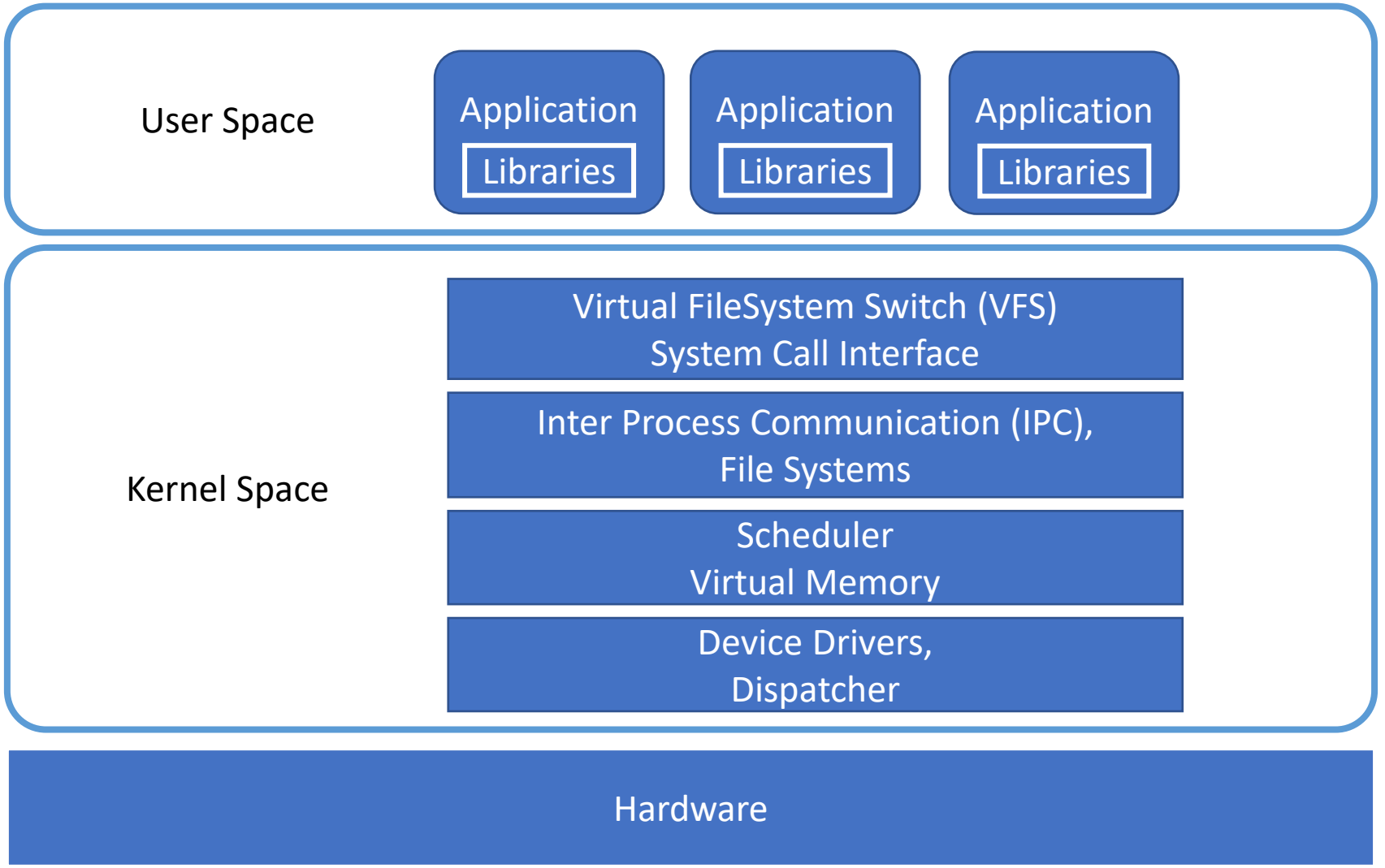
Tehnologii de containerizare | Docker

Ioan-Mihail STAN | ioan.stan@upb.ro

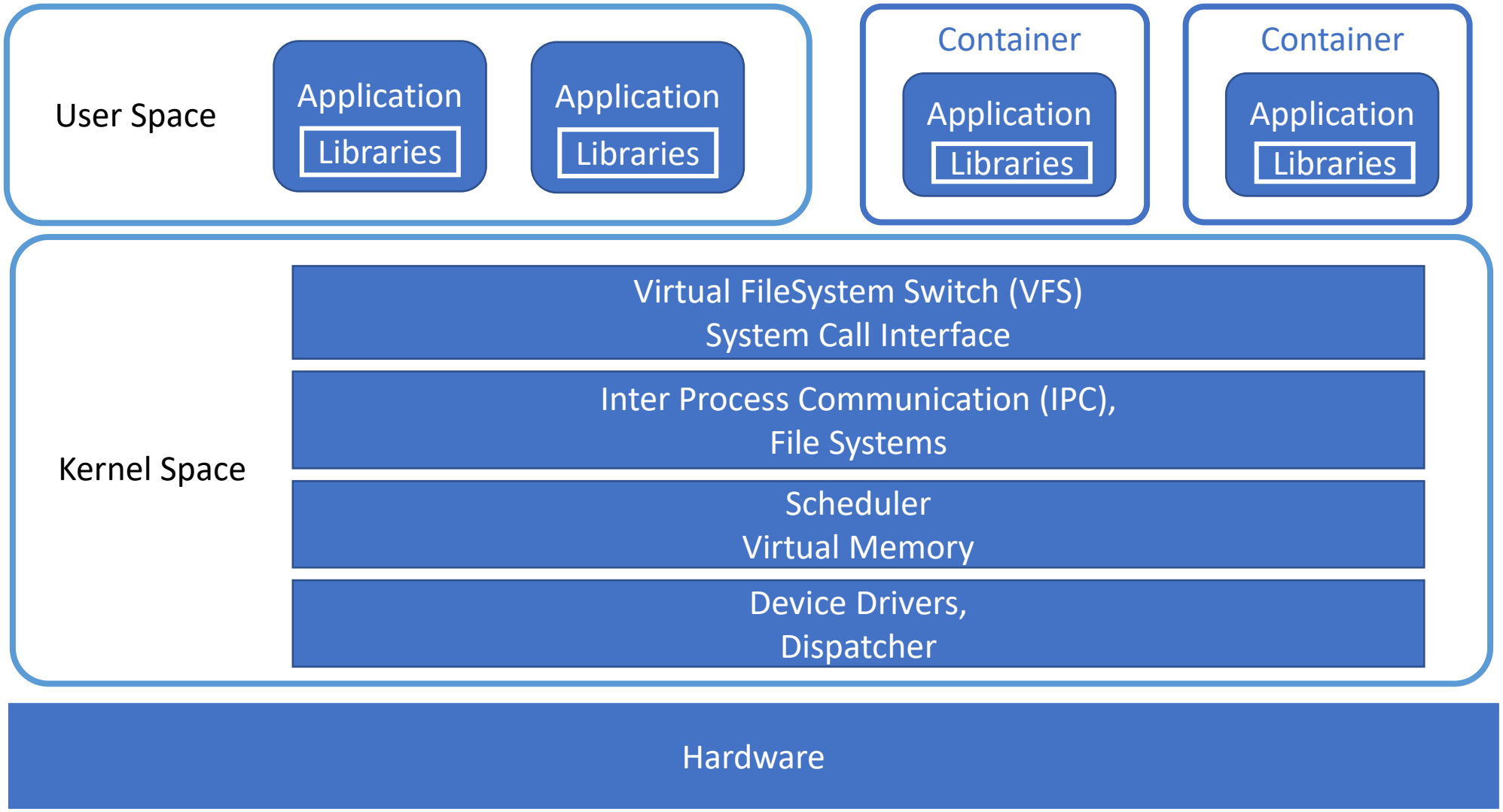
Cuprins

- Virtualizare bazată pe containere
- Container vs Mașină virtuală
- Tehnologii (runtimes) de containerizare
- Docker
- Izolare prin Namespace-uri
- Imagini Docker
- Optimizarea stocării datelor prin mecanismul de Copy-on-Write
- Docker Networking – terminologie
- Container Network Model
- Network Drivers
- CNM vs Container Network Interface
- Orchestrare și Clusterizare
- Docker Swarm
- Overlay Network & VXLAN
- Tehnologii & Concepte din Universul Docker

Kernel Monolitic



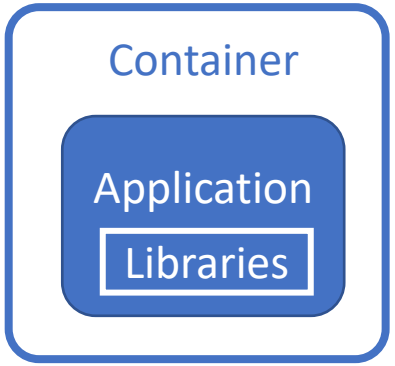
Virtualizare bazată pe containere



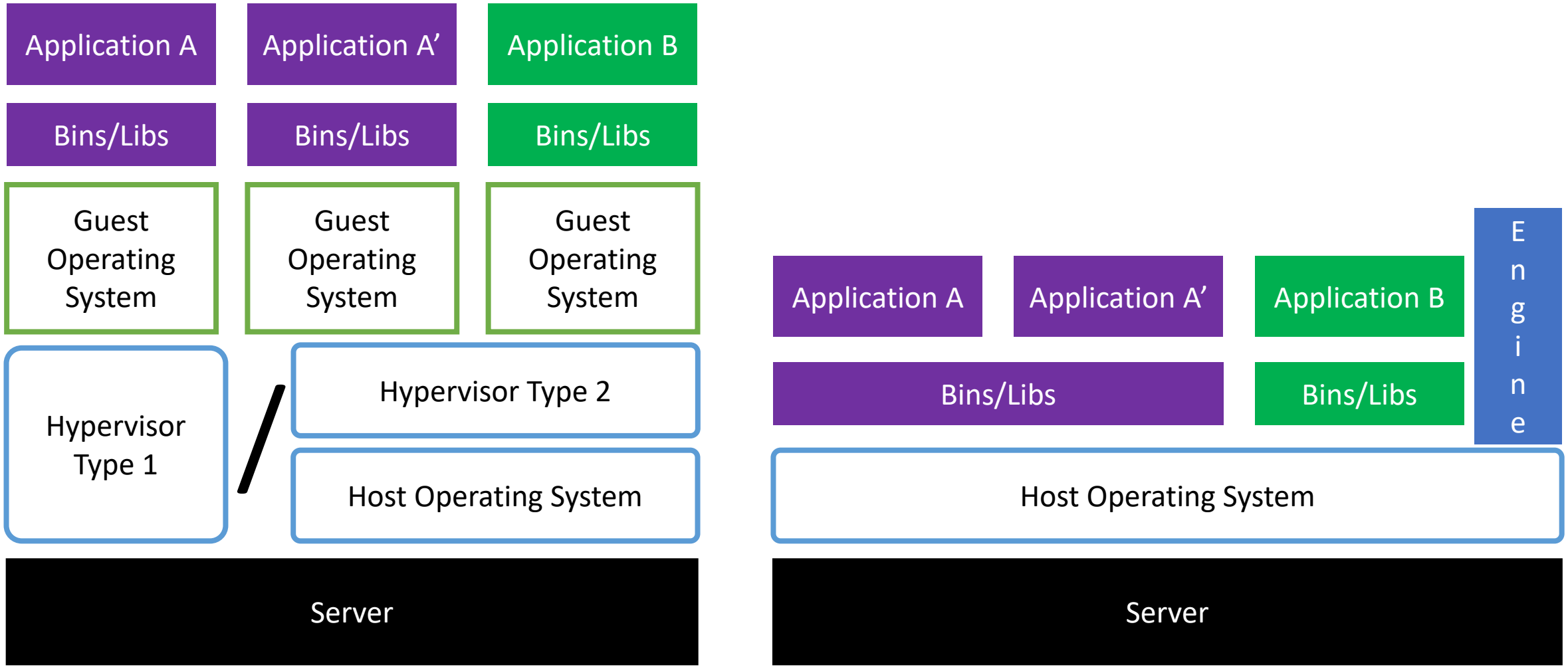
Container

- Metodă de virtualizare la nivelul sistemului de operare
- Obiect care poate împacheta o aplicație, cât și dependințele acesteia
- Mecanism de multiplicare a user-space-ului

- Suport în Kernel:
  - Control Groups (cgroups)
    - Permite stabilirea limitărilor asupra resurselor hardware utilizate
  - Namespaces
    - Limitează accesul la resursele sistemului gazdă
    - Pid, Net, Mount, IPC, UTS, User



Mașini Virtuale vs Containere



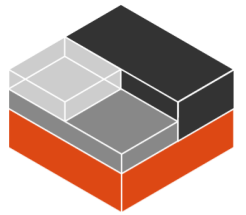
Tehnologii de containerizare

Containere de aplicații

- Docker
- RKT
- Podman + Buildah

Containere de sistem

- LXC/LXD
- OpenVZ
- Linux VServer



lxd

Developers bring their ideas to life with Docker

Download Docker Desktop | Sign Up for Docker Hub

Developing with Docker

Developing apps today requires so much more than writing code. Multiple languages, frameworks, architectures, and discontinuous interfaces between tools for each lifecycle stage creates enormous complexity. Docker simplifies and accelerates your workflow, while giving developers the freedom to innovate with their choice of tools, application stacks, and deployment environments for each project.

Namespace

pid	Process ID	• Creează o tabelă unică de procese. • Procesul cu process id-ul 1 din container va fi diferit de procesul init activ pe sistemul gazdă.
net	Networking	• Generează o stivă de rețea izolată față de cea a sistemului gazdă. • Un socket deschis pe portul 80 pe interfața virtuală a containerului nu va impacta funcționarea unui web server care rulează pe sistemul gazdă.
ipc	Inter Process Communication	• Asigură izolarea mecanismelor de partajare a informației inter-procese • Un semnal trimis către un proces care rulează în cadrul unui container nu va impacta procesele de pe sistemul gazdă.
mnt	Mount	• Asigură management izolat al montării sistemelor de fișiere. • Montarea și demontarea unui sistem de fișiere în interiorul unui container nu va impacta mount-urile existente pe sistemul gazdă.
uts	Unix Timesharing System	• Permite utilizarea unui hostname diferit la nivelul containerului fata de cel al sistemului gazdă.
user	User	• Crează o bază de date unică de utilizatori. • Utilizatorul cu UID 0 din container va fi diferit de utilizatorul root de pe sistemul gazdă.

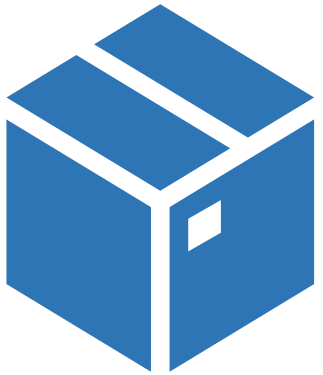
RL
crunch it connected

Imagini Docker

IMAGINE

Build time

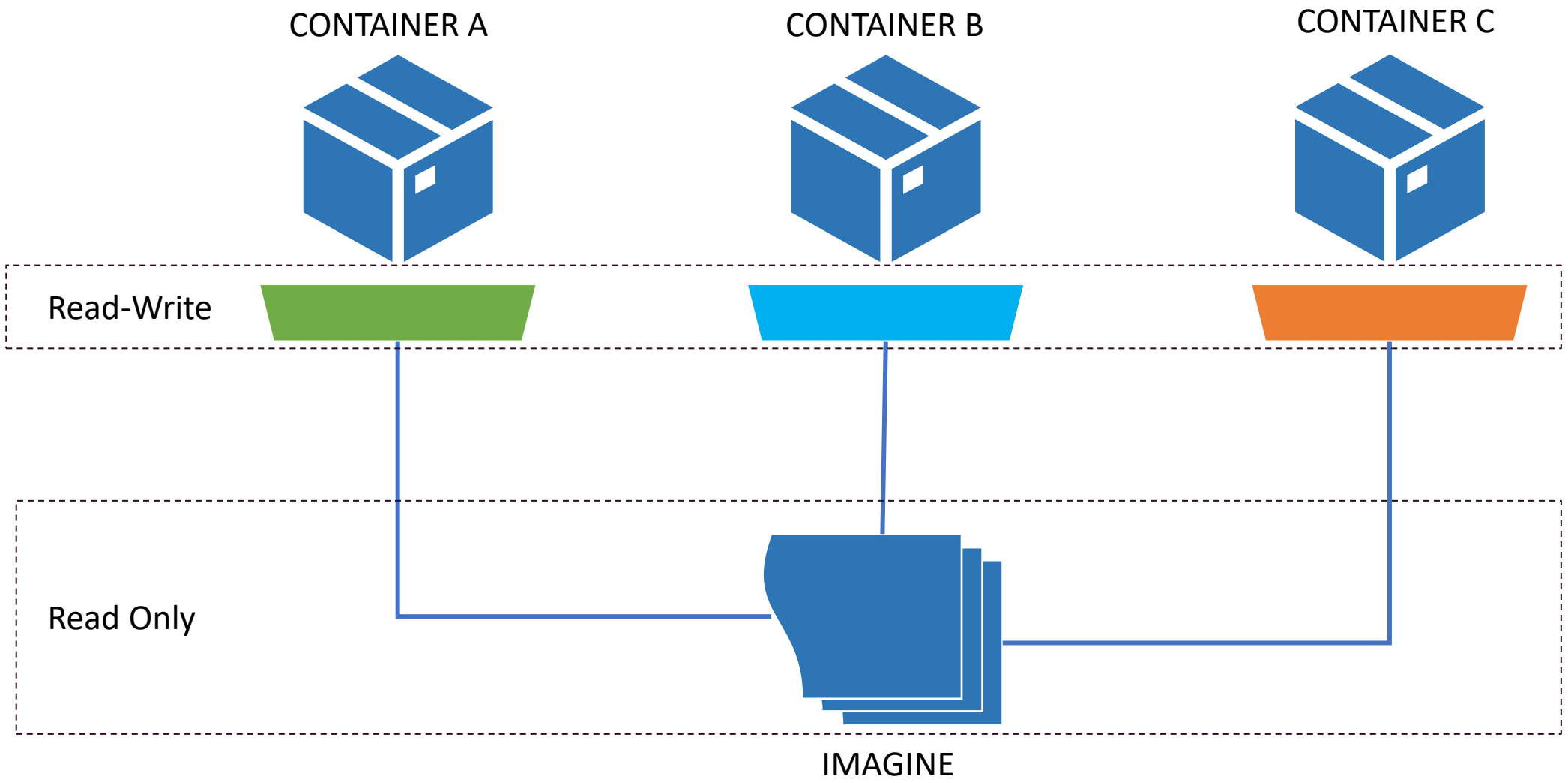
CONTAINER



Run time

RL
crunch it connected

Imagini Docker - Copy-on-Write



Imagini Docker – Layers (Straturi)

Imagini Docker – Layers (Straturi) (2)

```
john@john-[redacted]:~/Workspace/dockerdemo$ docker image inspect --format "{{json .RootFS}}" rlteam/hellorl:v1.0 | python -mjson.tool
{
    "Layers": [
        "sha256:6cebf3abed5fac58d2e792ce8461454e92c245d5312c42118f02e231a73b317f",
        "sha256:f7eae43028b334123c3a1d778f7bdf9783bbe651c8b15371df0120fd13ec35c5",
        "sha256:7beb13bce073c21c9ee608acb13c7e851845245dc76ce81b418fdf580c45076b",
        "sha256:122be11ab4a29e554786b4a1ec4764dd55656b59d6228a0a3de78eaf5c1f226c",
        "sha256:47fc3e6b81ce681f4b95a2ff5b543f0cab3a146752faed8276ac42f55f312b2e",
        "sha256:16192b28ad02a58aa63ed1d250f65db73d0ae0b8b5383768d064b7de3d03d6ad"
    ],
    "Type": "layers"
}
john@john-[redacted]:~/Workspace/dockerdemo$ docker image history rlteam/hellorl:v1.0
IMAGE               CREATED             CREATED BY                                      SIZE                COMMENT
e24c7d3e5329        About an hour ago   /bin/sh -c #(nop)  CMD ["/bin/sh" "-c" "pyth…   0B
0872e24cafb2        About an hour ago   /bin/sh -c apt update && apt -y install pyth…   62.5MB
574f9848642b        About an hour ago   /bin/sh -c #(nop) COPY dir:cddba28a60a9469e3…   51B
ea4cd63ea1cb        About an hour ago   /bin/sh -c #(nop)  MAINTAINER rlteam            0B
a2a15febcdf3        6 days ago          /bin/sh -c #(nop)  CMD ["/bin/bash"]            0B
<missing>           6 days ago          /bin/sh -c mkdir -p /run/systemd && echo 'do…   7B
<missing>           6 days ago          /bin/sh -c set -xe   && echo '#!/bin/sh' > /…   745B
<missing>           6 days ago          /bin/sh -c [ -z "$(apt-get indextargets)" ]     987kB
<missing>           6 days ago          /bin/sh -c #(nop) ADD file:c477cb0e95c56b51e…   63.2MB
john@john-[redacted]:~/Workspace/dockerdemo$ cat Dockerfile
FROM ubuntu:18.04
MAINTAINER rlteam


COPY app /opt/app
RUN apt update && apt -y install python3


CMD python3 /opt/app/app.py
john@john-[redacted]:~/Workspace/dockerdemo$
```

6 Straturi (Layers)

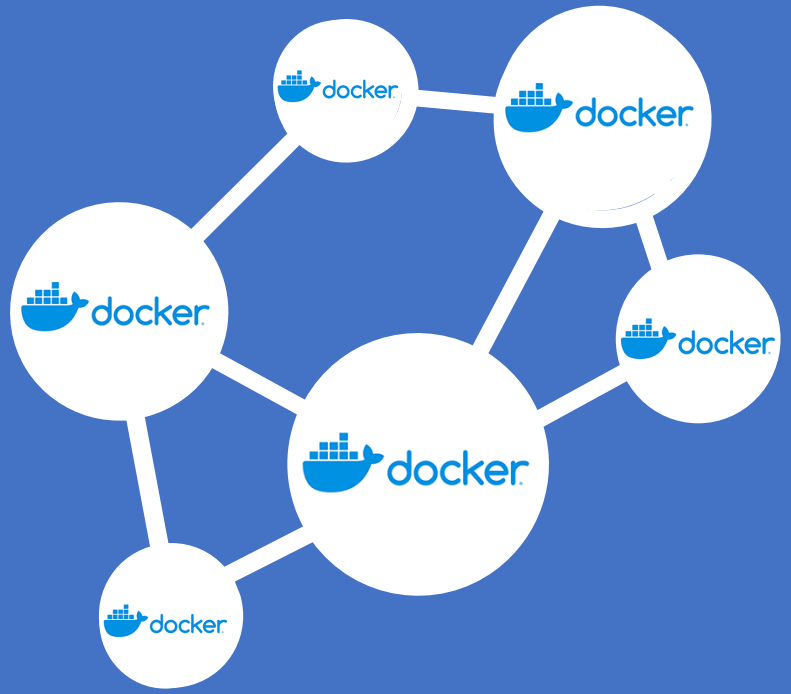
2 Straturi nou create

4 Straturi moștenite de la imaginea de bază Ubuntu 18.04

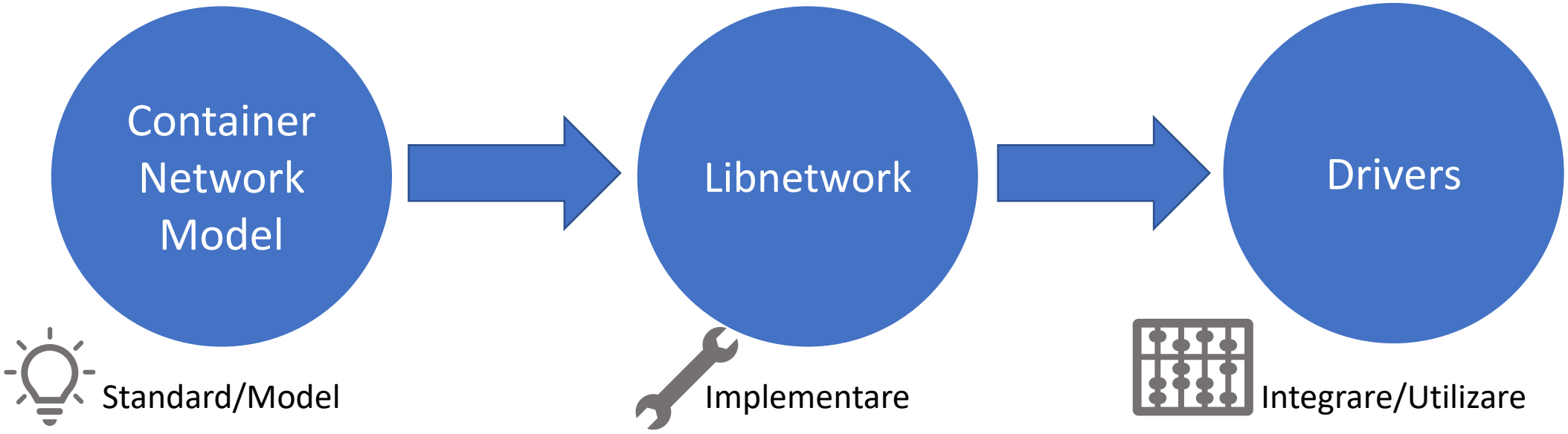
SYSTEMS LABORATORY

RL crunch it connected

Docker Networking

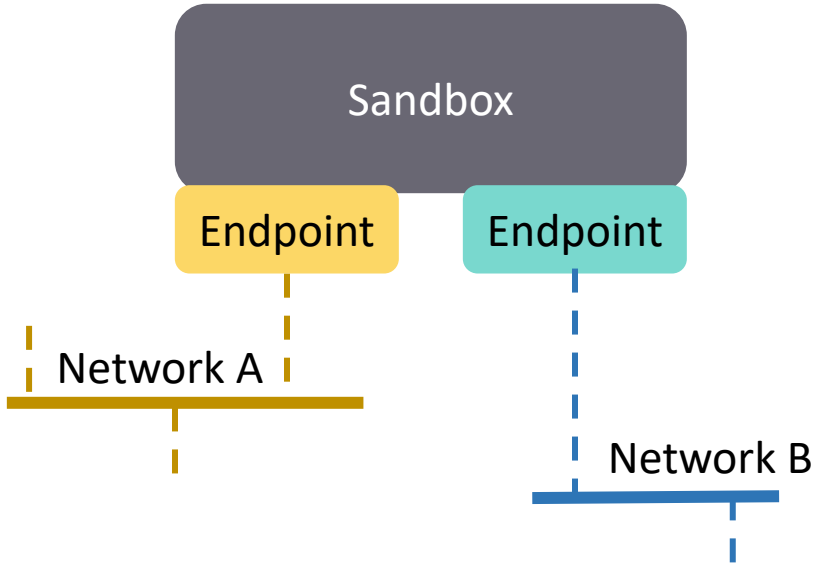


Docker Networking

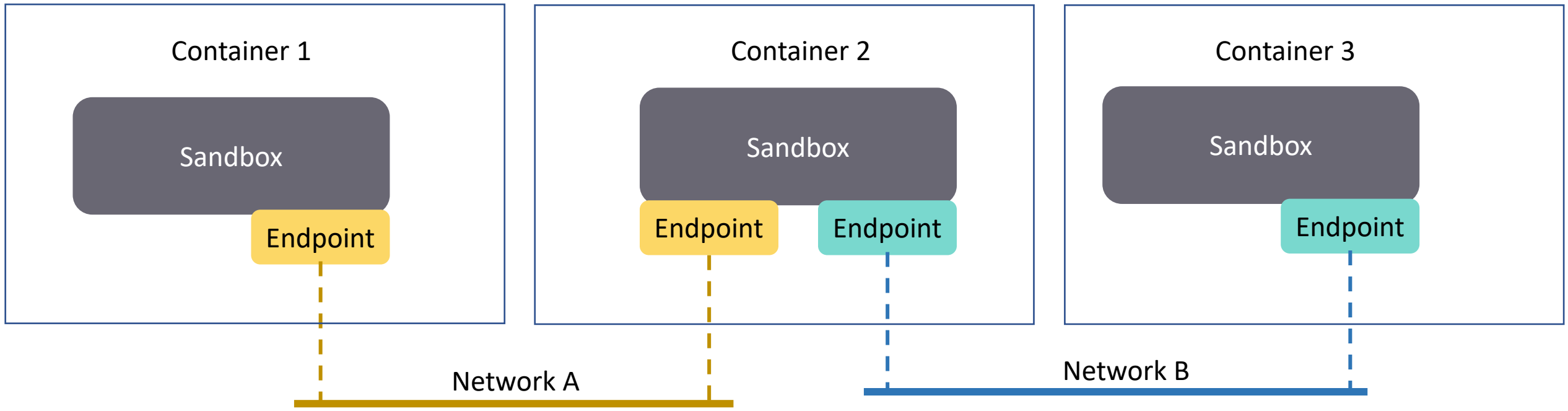


Container Network Model

- Sandbox
  - Network Namespace în sistemele Linux
- Endpoint
  - Interfața de rețea
  - en0, eth0
- Network
  - Rețea stabilită între mai multe endpoint-uri



Container Network Model(2)



Drivers

bridge	Facilitează comunicația pe același sistem gazdă.
host	Partajează același network namespace cu sistemul gazdă.
overlay	Facilitează comunicarea între containere de pe noduri diferite.
macvlan	Asigură contectarea directă la rețeaua fizică și alocă o adresă MAC interfeței container-ului.
none	Dezactivează contectivitatea la rețea. Utilizat în special împreună cu custom drivers.
3rd party plugins	Permite integrarea Docker Engine în implementări complexe, specializate de rețea.

Container Network Model vs Container Network Interface

Container Network Model	Container Network Interface
Propus de Docker	Propus de CoreOS
Adoptat de proiectul libnetwork	Adoptat de proiecte precum Kubernetes, Cloud Foundry, Apache Mesos sau rkt
Plugin-uri dezvoltate de către proiecte precum Weave, Project Calico, VMware, OVN sau Cisco Contiv	Plugin-uri dezvoltate de către proiecte precum Weave, Project Calico sau Cisco Contiv
Suportă doar runtime-ul de containerizare Docker	Este suportat la scară largă și adoptat de Cloud Native Computing Foundation

SYSTEMS LABORATORY

RL crunch it connected

Orchestrarea Docker

Orchestrarea containerelor

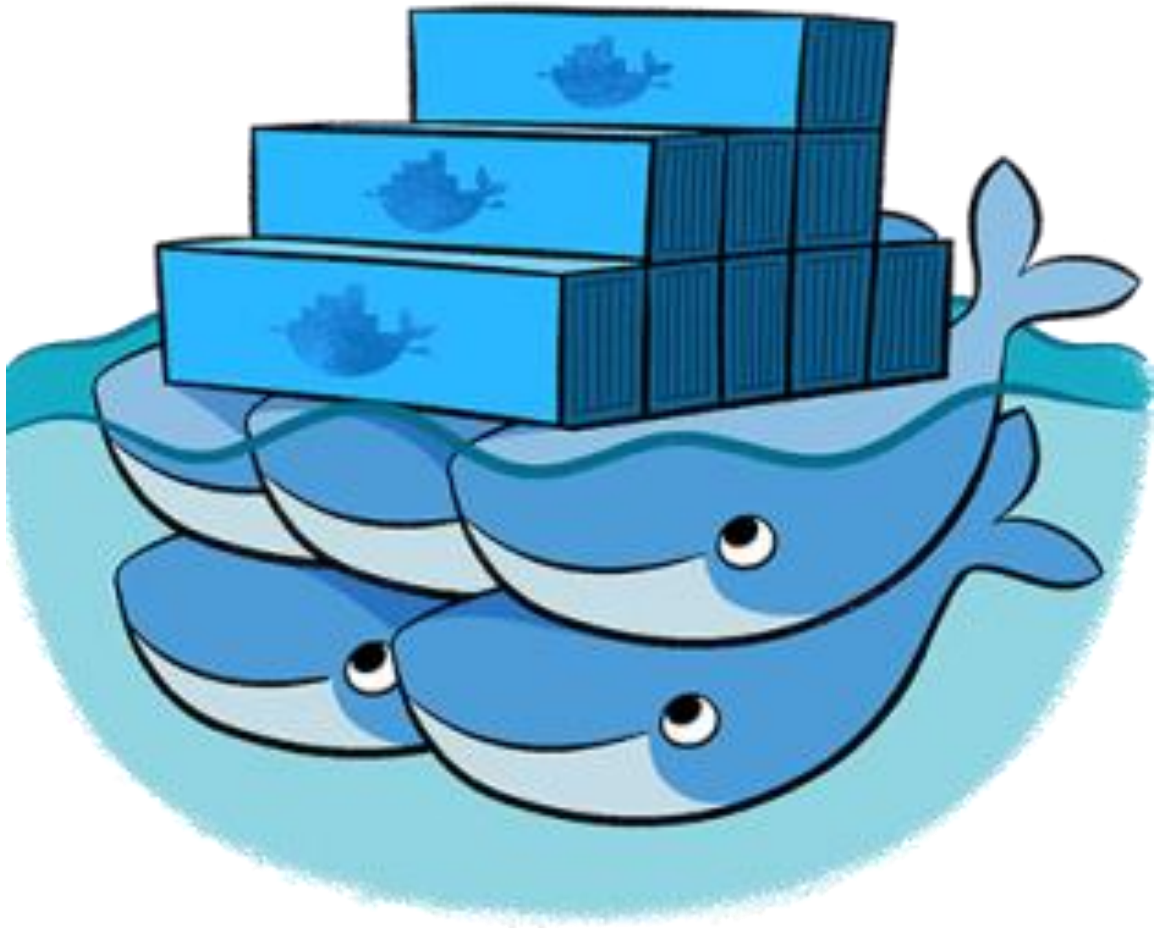
- Soluțiile de orchestrare aduc un nivel de complexitate superior față de funcțiile standard ale unui engine de containerizare

- În general se adresează topologiilor multi-nod (clusters, cloud)
  - Mai multe mașini fizice sau virtuale lucrează împreună pentru o mai bună gestiune a aplicațiilor containerizate

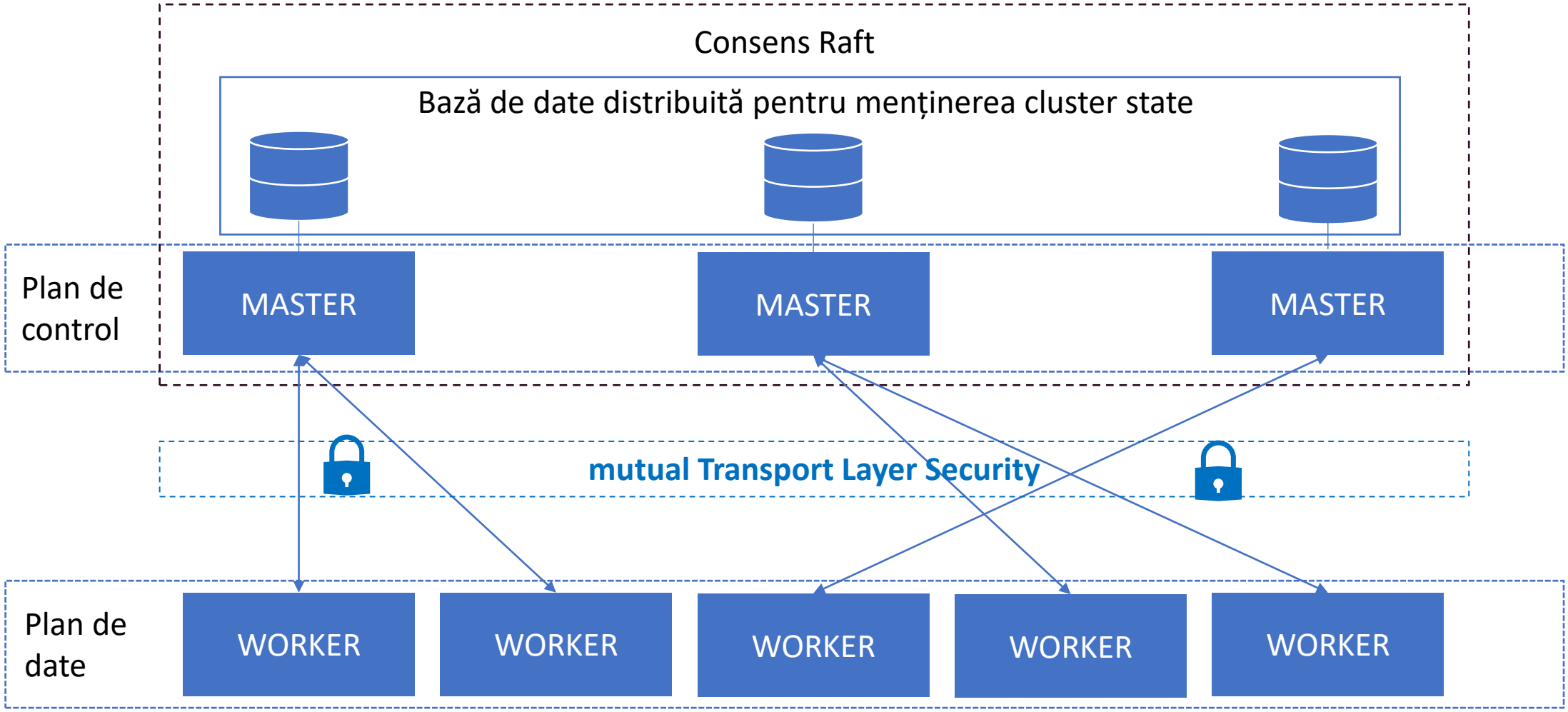
- Printre mecanismele clasice regăsim:
  - Monitorizarea resurselor nodului și deployment inteligent în funcție de încărcare
  - Self-healing pentru containerele care intră într-un stadiu de eroare
  - Funcția de balansare echitabilă a traficului pentru implementari multi-instanță ale unei aplicații

Docker Swarm

- Tehnologie nativă Docker
- Rețea Docker multi-nod
- Oferă clusterizarea și orchestrare soluțiilor containerizate cu Docker
- Alte caracteristici importante:
  - Oferă scalabilitatea aplicațiilor containerizate
  - Prezintă Load balancing nativ
  - Permite Rolling updates

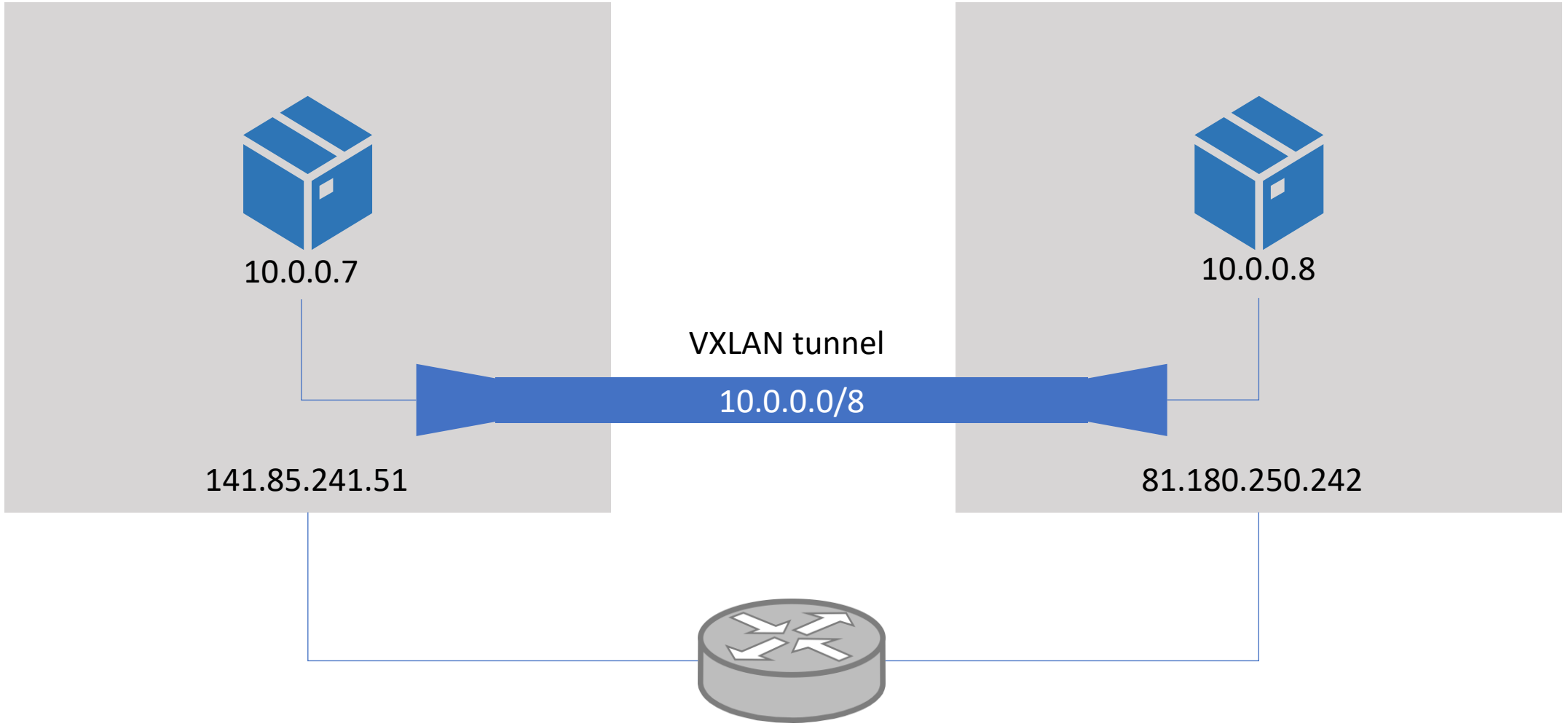


Docker Swarm(2) – Docker Engines cluster



RL
crunch it connected

Docker Overlay Network



VXLAN

- Virtual Extensible LAN
- Cadre de nivel 2 sunt transportate peste o infrastructură existentă de nivel 3
- Proiectat pentru scalabilitate în infrastructuri Cloud
  - Contectivitate la nivel de LAN între mașini aflate în locații geografice diferite.
- VXLAN Network Identifier (VNI)
  - 24 biți ≈ 16 milioane rețele virtuale
  - VLAN ID 12 biți = 4094 rețele virtuale

SYSTEMS LABORATORY

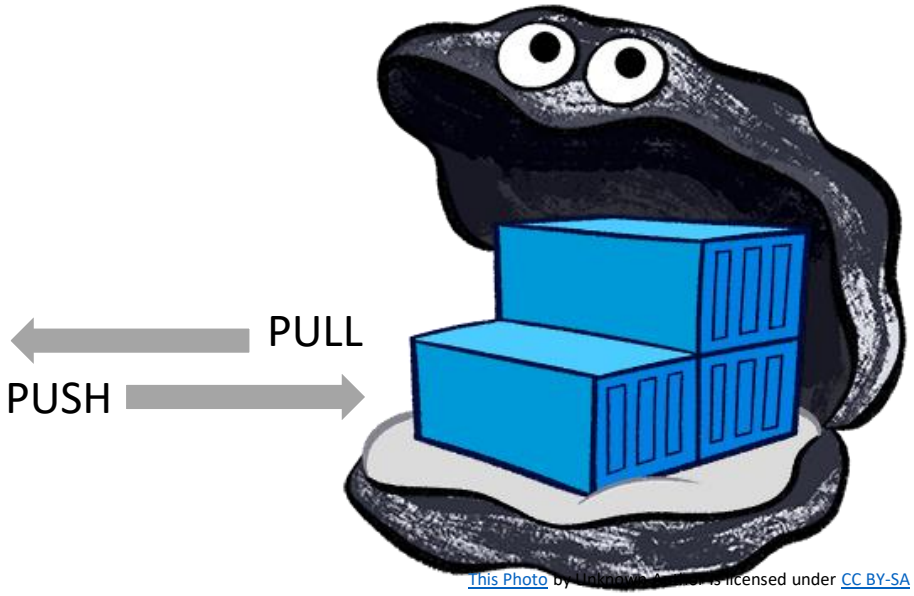
RL crunch it connected

Universul Docker

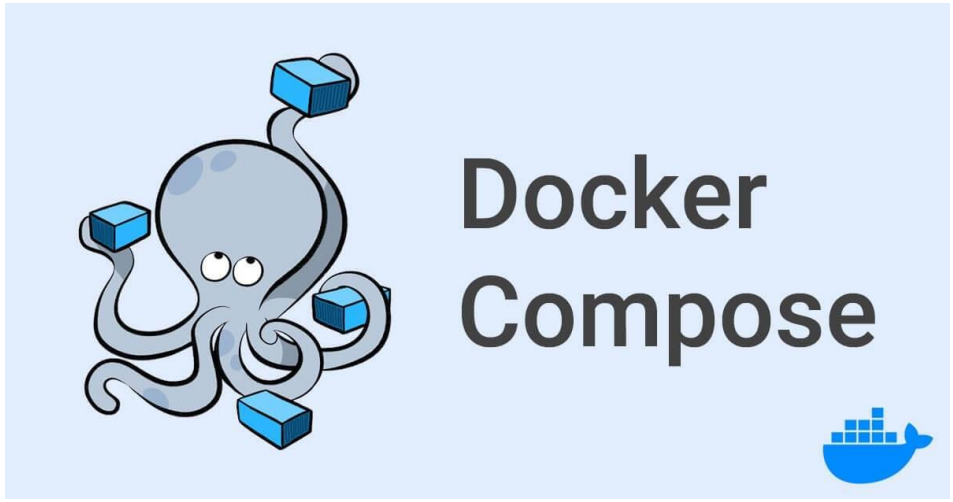
Docker Hub & Docker Registry

- Docker Hub echivalent-ul GitHub pentru stocarea de imagini Docker
- Sistem de stocare al artefactelor rezultate din procesul de BUILD al imaginilor Docker
- Docker Hub – platformă publică
- Docker Registry – platformă on-premise
- Implementări 3rd party:

- Harbor by Cloud Native Computing Foundation
  - Proiect inițiat de VMWare
  - Open Source



Docker Compose



- Tool utilizat pentru definirea și rularea aplicațiilor multi-container
- Limbaj declarativ – YAML
- Definește:
  - Serviciile – metodă de expunere a aplicației
  - Componenta de rețea
  - Volumele - metodă de stocare persistentă a datelor

RL
crunch it connected

Kubernetes

- Orchestrator pentru diverse tehnologii de containerizare
- Inițial creat de Google, actual întreținut de Cloud Native Computing Foundation
- Facilitează:
  - Deployment-ul automat de aplicații containerizate
  - Scalabilitate
  - Monitorizare și self-healing pentru aplicațiile gestionate
  - Load Balancing
  - etc.

kubernetes

RL
crunch it connected

Moby Project

- Proiect inițiat de Docker
- Open Source
- Oferă un framework pentru dezolvarea de noi soluții bazate pe containere
- Expune o bibliotecă de componente și un framework de asamblare

Cuvinte cheie

- Kernel Monolitic
- Container
- User Space
- cgroups
- namespaces
- Docker
- Imagini
- Runtime
- Build time
- Copy-on-Write(COW)
- Layers
- CNM
- CNI
- Libnetwork
- Drivers
- Sandbox
- Endpoint
- Swarm
- Orchestrare
- Cloud
- Cluster
- Overlay
- VXLAN
- DockerHub
- Docker Registry
- Harbor
- Compose
- Kubernetes
- Moby