

Conteneurs: Notions de base (Usage)

Cycle de vie, volumes, réseaux, ressources, bonnes pratiques

Emmanuel BRUNO
Université de Toulon, LIS UMR CNRS 7020
emmanuel.bruno@univ-tln.fr

2026-10-14

Table des matières

1 Objectifs du cours	1	8.3 Exercice de persistance	22
2 Rappel rapide	1	9 Gestion des ressources (Avancé)	31
3 Les images	2	9.1 Modifier les limites d'un conteneur	
4 Cycle de vie d'un conteneur	2	existant	34
4.1 Conteneur en arrière-plan (<code>--detach</code> ,		10 Bonnes pratiques d'usage	34
<code>-d</code>)	3	11 Atelier final	34
4.2 Liste des conteneurs (<code>container ls</code> ,		12 Conclusion	35
<code>ps</code>)	5		
4.3 Arrêter (<code>stop</code>) / démarrer (<code>start</code>) /			
supprimer (<code>rm</code>)	6		
4.4 Exercice 1	6		
5 Logs, exec et monitoring	6		
5.1 Exécuter une commande dans un			
conteneur en cours	8		
5.2 Monitoring	8		
6 Réseau	8		
6.1 Réseau par défaut (bridge)	9		
6.2 Réseau custom (bonne pratique) . . .	9		
6.3 Vérifier le réseau (Avancé)	12		
6.4 Communication conteneur →			
conteneur	13		
6.5 Exercice 2 (Max 10 min)	16		
7 Port mapping	16		
7.1 Tester depuis l'hôte	19		
7.2 Accès conteneur -> hôte	20		
8 Volumes : persistance	20		
8.1 Bind mount (risky, dev)	20		
8.2 Volume nommé (better, prod)	21		

1 Objectifs du cours

À l'issue de ce cours, vous devez être capables de :

- Manipuler des conteneurs Docker (run, stop, start, rm)
- Utiliser les volumes pour la persistance
- Connecter des conteneurs via des réseaux
- Exposer des ports vers l'hôte
- Gérer les ressources d'un conteneur (CPU / RAM / PIDs)
- Comprendre et appliquer les bonnes pratiques d'usage

2 Rappel rapide

Un **conteneur** est une instance d'une **image**. L'image est un artefact immuable (voir le cours «Image»). Le conteneur est un **processus isolé** qui tourne sur le noyau de l'hôte.

💡 Astuce

- **Image** = template immuable
- **Conteneur** = exécution d'un processus isolé

3 Les images

- Une image contient tout le nécessaire pour exécuter une application (code, runtime, bibliothèques, dépendances, fichiers de configuration).
- Chaque couche est en lecture seule, sauf la dernière (couche de conteneur).
- Les images sont stockées dans des registres (Docker Hub, GitHub Container Registry, etc.).
- Les images sont identifiées par un nom : `[registre/]nom[:tag]` (ex: `docker.io/library/nginx:1.23`).
 - Si le registre n'est pas spécifié, Docker Hub est utilisé par défaut.
 - Si le tag n'est pas spécifié, `latest` est utilisé par défaut.
 - Une image peut être référencée par son **digest** (SHA256).
 - Une image peut avoir plusieurs tags.
 - Ex:
`nginx:1`, `nginx:1.29`, `nginx:mainline`, `nginx:latest`, `nginx:f3524ef8b874` peuvent référencer la même image.
- Une image est destinée à une plateforme spécifique (architecture CPU, OS).
 - Ex: `linux/amd64`, `linux/arm64/v8`, `windows/amd64`.
 - Il est possible de créer des images multi-plateformes (multi-arch).

4 Cycle de vie d'un conteneur

Exécution simple

- La commande `docker run` crée et démarre un conteneur à partir d'une image.

- Le conteneur s'arrête lorsque le processus principal se termine (`--rm` pour supprimer automatiquement après).
- Le format général est `docker run [OPTIONS] IMAGE [COMMAND] [ARG...]`.
- Si l'image n'est pas présente localement, Docker la télécharge depuis le registre.

```
#| label: lst-run-alpine
#| title: "Premier conteneur Docker"
#| caption: "Exécution d'un conteneur Alpine jetable affichant un message"
#| code-fold: true
#| code-summary: "Afficher la commande Docker"
#| collapse: true
docker run --label ebpro-render=true
--rm alpine:3.18 echo "Hello"
```

```
Unable to find image 'alpine:3.18'
locally
```

```
3.18: Pulling from library/alpine
```

```
44cf07d57ee4: Pulling fs layer
```

```
44cf07d57ee4:
Downloading 48.51kB/3.418MB
```

```
44cf07d57ee4: Verifying Checksum
```

```
44cf07d57ee4: Download complete
```

```
44cf07d57ee4:
Extracting 65.54kB/3.418MB
```

```
44cf07d57ee4:
Extracting 2.49MB/3.418MB
```

```
44cf07d57ee4:
Extracting 3.418MB/3.418MB
```

```
44cf07d57ee4: Pull complete
```

```
Digest: sha256:de0eb0b3f2a47baleb89389859:
Status: Downloaded newer image for
alpine:3.18
Hello
```

4.1 Conteneur en arrière-plan (--detach, -d)

- La commande `docker run -d` démarre un conteneur en arrière-plan (détaché --detach).

```
#| label: lst-run-nginx
#| title: "Conteneur Nginx en
arrière-plan"
#| caption: "Démarrage d'un conteneur
Nginx en arrière-plan"
docker rm -f my-nginx >/dev/null 2>&1
|| true
docker run --label ebpro-render=true
-d --name my-nginx nginx:1.23
```

```
Unable to find image 'nginx:1.23'
locally
1.23: Pulling from library/nginx
```

```
f03b40093957: Pulling fs layer
0972072e0e8a: Pulling fs layer
a85095acb896: Pulling fs layer
d24b987aa74e: Pulling fs layer
6c1a86118ade: Pulling fs layer
9989f7b33228: Pulling fs layer
d24b987aa74e: Waiting
6c1a86118ade: Waiting
```

```
9989f7b33228: Waiting
```

```
a85095acb896:
Downloading      624B/624B
```

```
a85095acb896: Verifying Checksum
```

```
a85095acb896: Download complete
```

```
0972072e0e8a:
Downloading    257.7kB/25.58MB
```

```
f03b40093957:
Downloading    321.6kB/31.4MB
```

```
0972072e0e8a:
Downloading    5.797MB/25.58MB
```

```
f03b40093957:
Downloading    4.805MB/31.4MB
```

```
0972072e0e8a:
Downloading    12.15MB/25.58MB
```

```
f03b40093957:
Downloading    9.944MB/31.4MB
```

```
d24b987aa74e:
Downloading      958B/958B
```

```
d24b987aa74e: Verifying Checksum
```

```
d24b987aa74e: Download complete
```

```
f03b40093957:
Downloading    15.09MB/31.4MB
```

```
0972072e0e8a:
Downloading    18.42MB/25.58MB
```

```
0972072e0e8a:
Downloading    24.47MB/25.58MB
```

f03b40093957:
Downloading 20.91MB/31.4MB

0972072e0e8a: Verifying Checksum

0972072e0e8a: Download complete

f03b40093957:
Downloading 28.34MB/31.4MB

f03b40093957: Verifying Checksum

f03b40093957: Download complete

f03b40093957:
Extracting 327.7kB/31.4MB

6c1a86118ade:
Downloading 772B/772B

6c1a86118ade: Verifying Checksum

6c1a86118ade: Download complete

9989f7b33228:
Downloading 1.405kB/1.405kB

9989f7b33228: Verifying Checksum

f03b40093957:
Extracting 2.949MB/31.4MB

f03b40093957:
Extracting 5.898MB/31.4MB

f03b40093957:
Extracting 8.192MB/31.4MB

f03b40093957:
Extracting 10.81MB/31.4MB

f03b40093957:
Extracting 13.43MB/31.4MB

f03b40093957:
Extracting 15.4MB/31.4MB

f03b40093957:
Extracting 17.04MB/31.4MB

f03b40093957:
Extracting 19.66MB/31.4MB

f03b40093957:
Extracting 23.27MB/31.4MB

f03b40093957:
Extracting 26.21MB/31.4MB

f03b40093957:
Extracting 28.51MB/31.4MB

f03b40093957:
Extracting 28.84MB/31.4MB

f03b40093957:
Extracting 29.82MB/31.4MB

f03b40093957:
Extracting 30.15MB/31.4MB

f03b40093957:
Extracting 30.47MB/31.4MB

f03b40093957:
Extracting 31.13MB/31.4MB

f03b40093957:
Extracting 31.4MB/31.4MB

f03b40093957: Pull complete

0972072e0e8a:
Extracting 262.1kB/25.58MB

0972072e0e8a:

```
Extracting 2.621MB/25.58MB
```

```
0972072e0e8a:
```

```
Extracting 6.029MB/25.58MB
```

```
0972072e0e8a:
```

```
Extracting 9.961MB/25.58MB
```

```
0972072e0e8a:
```

```
Extracting 13.63MB/25.58MB
```

```
0972072e0e8a:
```

```
Extracting 16.25MB/25.58MB
```

```
0972072e0e8a:
```

```
Extracting 19.4MB/25.58MB
```

```
0972072e0e8a:
```

```
Extracting 22.54MB/25.58MB
```

```
0972072e0e8a:
```

```
Extracting 23.33MB/25.58MB
```

```
0972072e0e8a:
```

```
Extracting 23.59MB/25.58MB
```

```
0972072e0e8a:
```

```
Extracting 25.58MB/25.58MB
```

```
0972072e0e8a: Pull complete
```

```
a85095acb896: Extracting 624B/624B
```

```
a85095acb896: Extracting 624B/624B
```

```
a85095acb896: Pull complete
```

```
d24b987aa74e: Extracting 958B/958B
```

```
d24b987aa74e: Extracting 958B/958B
```

```
d24b987aa74e: Pull complete
```

```
6c1a86118ade: Extracting 772B/772B
```

```
6c1a86118ade: Extracting 772B/772B
```

```
6c1a86118ade: Pull complete
```

```
9989f7b33228:
```

```
Extracting 1.405kB/1.405kB
```

```
9989f7b33228:
```

```
Extracting 1.405kB/1.405kB
```

```
9989f7b33228: Pull complete
```

```
Digest: sha256:f5747a42e3adcb3168049d63278d7251d91185bb
```

```
Status: Downloaded newer image for  
nginx:1.23
```

```
a4787417cff55ed719dc2570ccfa7ed8107156b34076ce1fb7a06d0
```

4.2 Liste des conteneurs (container ls, ps)

- La commande `docker ps` (docker container ls) liste les conteneurs en cours d'exécution.
- L'option `-a` liste tous les conteneurs (y compris arrêtés).
- Chaque conteneur a un ID unique, un nom, un statut, des ports exposés, etc.

```
#| label: lst-docker-ps  
#| title: "Liste des conteneurs en  
cours d'exécution"  
#| caption: "Affichage des conteneurs  
Docker en cours d'exécution"  
docker ps
```

CONTAINER	ID	IMAGE
COMMAND		CREATED
STATUS	PORTS	NAMES
a4787417cff5		nginx:1.23
"/docker-entrypoint..."	2 seconds	
ago Up Less than a second	80/tcp	
my-nginx		
7202cb0885be		ghcr.io/ebpro/

```
solen:quarto-
full-build-20261010T172146Z    "bash
-lc 'source /h..." 3 minutes ago Up
3 minutes                      quarto-
render-38521027713
```

4.3 Arrêter (stop) / démarrer (start) / supprimer (rm)

- Le cycle de vie d'un conteneur inclut les états : démarré, arrêté, supprimé.
- Lorsqu'un conteneur est arrêté, il reste sur le système jusqu'à sa suppression (rm).

```
#| label: lst-stop-nginx
docker stop my-nginx
```

```
my-nginx
```

```
#| label: lst-docker-ps-all
docker ps -a
```

CONTAINER	ID	IMAGE
COMMAND		CREATED
STATUS		PORTS
NAMES		
a4787417cff5		nginx:1.23
"/docker-entrypoint..."	3 seconds ago	Exited (0) Less than a second ago
	my-nginx	
7202cb0885be		ghcr.io/ebpro/solen:quarto-
full-build-20261010T172146Z		"bash
-lc 'source /h..."	3 minutes ago	Up 3 minutes
quarto-render-38521027713		

```
#| label: lst-start-nginx
docker start my-nginx
```

```
my-nginx
```

```
#| label: lst-rm-nginx
docker stop my-nginx
```

```
my-nginx
```

```
#| label: lst-rm-nginx
docker rm my-nginx
```

```
my-nginx
```

4.4 Exercice 1

Lancer un conteneur `alpine`, exécuter `uname -a`, puis le supprimer.

```
#| label: lst-run-alpine-uname
docker rm -f my-alpine >/dev/null 2>&1 || true
docker run --label ebpro-render=true --name my-alpine alpine:3.18 uname -a
docker rm my-alpine
```

```
Linux 3f35bf4b79d8 6.8.0-142-generic
#142-Ubuntu SMP PREEMPT_DYNAMIC Wed
Sep 2 14:24:27 UTC 2026 x86_64 Linux
my-alpine
```

- L'option `--rm` peut être utilisée avec `docker run` pour supprimer automatiquement le conteneur après son arrêt.

5 Logs, exec et monitoring

- Les logs d'un conteneur sont accessibles via `docker logs`.
- L'option `-f` permet de suivre les logs en temps réel.

```
#| label: lst-run-nginx-logs
docker rm -f my-nginx >/dev/null 2>&1
|| true
docker run --label ebpro-render=true
--name my-nginx -d nginx:1.23
```

```
ba571fc084930c47274f79e1d2b12fa57e6e6410d
```

```
#| label: lst-docker-logs
docker logs my-nginx
# docker logs -f my-nginx
```

```
/docker-entrypoint.sh: /docker-
entrypoint.d/ is not empty, will
attempt to perform configuration
/docker-entrypoint.sh: Looking for
shell scripts in /docker-
entrypoint.d/
/docker-entrypoint.sh: Launching /
docker-entrypoint.d/10-listen-on-
ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh:
info: Getting the checksum of /etc/
nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh:
info: Enabled listen on IPv6 in /etc/
nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /
docker-entrypoint.d/20-envsubst-on-
templates.sh
/docker-entrypoint.sh: Launching /
docker-entrypoint.d/30-tune-worker-
processes.sh
/docker-entrypoint.sh: Configuration
complete; ready for start up
2026/10/14 06:23:25 [notice] 1#1:
using the "epoll" event method
2026/10/14 06:23:25 [notice] 1#1:
nginx/1.23.4
2026/10/14 06:23:25 [notice] 1#1:
built by gcc 10.2.1 20210110 (Debian
```

```
10.2.1-6)
2026/10/14 06:23:25 [notice] 1#1: OS:
Linux 6.8.0-142-generic
2026/10/14 06:23:25 [notice]
1#1: getrlimit(RLIMIT_NOFILE):
1048576:1048576
2026/10/14 06:23:25 [notice] 1#1:
start worker processes
2026/10/14 06:23:25 [notice] 1#1:
start worker process 29
2026/10/14 06:23:25 [notice] 1#1:
start worker process 30
2026/10/14 06:23:25 [notice] 1#1:
start worker process 31
2026/10/14 06:23:25 [notice] 1#1:
start worker process 32
2026/10/14 06:23:25 [notice] 1#1:
start worker process 33
2026/10/14 06:23:25 [notice] 1#1:
start worker process 34
2026/10/14 06:23:25 [notice] 1#1:
start worker process 35
2026/10/14 06:23:25 [notice] 1#1:
start worker process 36
2026/10/14 06:23:25 [notice] 1#1:
start worker process 37
2026/10/14 06:23:25 [notice] 1#1:
start worker process 38
2026/10/14 06:23:25 [notice] 1#1:
start worker process 39
2026/10/14 06:23:25 [notice] 1#1:
start worker process 40
2026/10/14 06:23:25 [notice] 1#1:
start worker process 41
2026/10/14 06:23:25 [notice] 1#1:
start worker process 42
2026/10/14 06:23:25 [notice] 1#1:
start worker process 43
2026/10/14 06:23:25 [notice] 1#1:
start worker process 44
2026/10/14 06:23:25 [notice] 1#1:
start worker process 45
2026/10/14 06:23:25 [notice] 1#1:
```

```

start worker process 46
2026/10/14 06:23:25 [notice] 1#1:
start worker process 47
2026/10/14 06:23:25 [notice] 1#1:
start worker process 48
2026/10/14 06:23:25 [notice] 1#1:
start worker process 49
2026/10/14 06:23:25 [notice] 1#1:
start worker process 50
2026/10/14 06:23:25 [notice] 1#1:
start worker process 51
2026/10/14 06:23:25 [notice] 1#1:
start worker process 52
2026/10/14 06:23:25 [notice] 1#1:
start worker process 53
2026/10/14 06:23:25 [notice] 1#1:
start worker process 54
2026/10/14 06:23:25 [notice] 1#1:
start worker process 55
2026/10/14 06:23:25 [notice] 1#1:
start worker process 56
2026/10/14 06:23:25 [notice] 1#1:
start worker process 57
2026/10/14 06:23:25 [notice] 1#1:
start worker process 58
2026/10/14 06:23:25 [notice] 1#1:
start worker process 59
2026/10/14 06:23:25 [notice] 1#1:
start worker process 60

```

5.1 Exécuter une commande dans un conteneur en cours

- La commande `docker exec` permet d'exécuter des commandes dans un conteneur en cours d'exécution.
- Lance un nouveau processus dans le conteneur
 - `docker exec container <cmd>` → non interactif
 - `docker exec -it container sh` → interactif (ou un autre shell ou programme interactif)

```

#| label: lst-run-nginx-exec
docker exec my-nginx ls /usr/share/nginx/html

```

```

50x.html
index.html

```

```

#|      label:      lst-run-nginx-exec-interactive
docker stop my-nginx
docker rm my-nginx

```

```

my-nginx
my-nginx

```

5.2 Monitoring

- La commande `docker stats` affiche l'utilisation des ressources (CPU, mémoire, réseau, disques) des conteneurs en cours d'exécution.
- L'option `--no-stream` affiche une seule fois les statistiques.

```

#| label: lst-docker-stats
docker stats --no-stream

```

CONTAINER	ID	NAME
CPU %	MEM USAGE / LIMIT	MEM
%	NET I/O	BLOCK I/
0	PIDS	
7202cb0885be		quarto-
render-38521027713		0.14%
857.4MiB / 62.66GiB		1.34%
16.2kB / 4.28kB	1.02GB / 60.8MB	56

6 Réseau

- Chaque conteneur a une interface réseau virtuelle et une adresse IP.
- Par défaut, les conteneurs sont connectés à un réseau bridge par défaut.

- Il est possible de créer des réseaux custom pour isoler les conteneurs.
- Les conteneurs sur le même réseau custom peuvent communiquer entre eux par **nom** (DNS interne).

6.1 Réseau par défaut (bridge)

- Par défaut, Docker crée un réseau bridge nommé `bridge`.

```
#| label: lst-docker-network-ls
docker rm -f c1 >/dev/null 2>&1 || true
docker run --label ebro-render=true -d --name c1 alpine:3.18 sleep 3600
docker rm -f c2 >/dev/null 2>&1 || true
docker run --label ebro-render=true -d --name c2 alpine:3.18 sleep 3600
```

```
bea399dc5c6605add839099bc1fe7821160922f44f09d775b434f73fc1f58a2b62cb2cec02118e51312ef67f9083633bfb7e3d03c8cf467980657884ca353301
```

- Les conteneurs ont une adresse IP dans le réseau bridge par défaut.
 - La commande `docker inspect` permet de voir les détails d'un conteneur et permet de définir un format personnalisé (ici l'adresse IP).

```
#| label: lst-docker-inspect-ip
# Robustesse du rendu : si un conteneur est absent (échec transitoire de `docker run`), # on le recrée avant l'inspect pour ne pas faire échouer tout le rendu.
for c in c1 c2; do
  docker inspect "$c" >/dev/null 2>&1 || \
    docker run --label ebro-render=true -d --name "$c" alpine:3.18 sleep 3600
done
```

```
docker inspect -f '{{.Name}} - {{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' c1 c2
```

```
/c1 - 172.17.0.3
/c2 - 172.17.0.4
```

✗ Sur le réseau bridge par défaut :

- les conteneurs **n'ont pas de DNS interne**
- ils ne peuvent pas se résoudre par nom

```
#| label: lst-docker-ping-fail
docker exec c1 ping -c 1 c2
```

```
ping: bad address 'c2'
```

```
: 1
```

6.2 Réseau custom (bonne pratique)

- Créer un réseau custom avec `docker network create`.
- Lancer des conteneurs (un serveur web, une base de données) sur ce réseau avec `--network`.

```
#| label: lst-docker-network-create
docker network inspect app-net >/dev/null 2>&1 || docker network create app-net
```

```
docker rm -f web >/dev/null 2>&1 || true
docker run --label ebro-render=true --detach --name web \
  --network app-net \
  nginx:1.23
```

```
# Robustesse du rendu : on attend que le conteneur web soit réellement en
```

```

cours
# d'exécution (limite 30 s) avant les
tests DNS/HTTP suivants, pour éviter
les
# échecs transitoires de résolution
(`curl: (6) Could not resolve host:
web`).
timeout 30 sh -c 'until [ "$(docker
inspect -f "{{.State.Running}}" web
2>/dev/null)" = "true" ]; do sleep
1; done'

docker rm -f db >/dev/null 2>&1 ||
true
docker run --label ebpro-render=true
--detach --name db \
  --network app-net \
  redis:7.0.9

```

```

a9ba21eedc24c69148c61c7d290f5f2b13dc89b0e
b737f62clef26bfb3e32cbae9eeecfbd38778c05
Unable to find image 'redis:7.0.9'
locally
7.0.9: Pulling from library/redis

```

```

3f9582a2cbe7: Pulling fs layer
241c2d338588: Pulling fs layer
89515d93a23e: Pulling fs layer
65e8ba9473fe: Pulling fs layer
585124038cab: Pulling fs layer
b483de716a47: Pulling fs layer
65e8ba9473fe: Waiting
585124038cab: Waiting

```

```

b483de716a47: Waiting
241c2d338588:
Downloading 1.732kB/1.732kB
241c2d338588: Verifying Checksum
241c2d338588: Download complete
89515d93a23e:
Downloading 16.38kB/1.463MB
3f9582a2cbe7:
Downloading 323.2kB/31.41MB
89515d93a23e: Verifying Checksum
89515d93a23e: Download complete
3f9582a2cbe7:
Downloading 6.171MB/31.41MB
3f9582a2cbe7:
Downloading 13.33MB/31.41MB
65e8ba9473fe:
Downloading 97.66kB/9.579MB
3f9582a2cbe7:
Downloading 20.15MB/31.41MB
65e8ba9473fe:
Downloading 5.504MB/9.579MB
3f9582a2cbe7:
Downloading 26.01MB/31.41MB
65e8ba9473fe: Verifying Checksum
65e8ba9473fe: Download complete
3f9582a2cbe7: Verifying Checksum

```

3f9582a2cbe7: Download complete

3f9582a2cbe7:

Extracting 327.7kB/31.41MB

3f9582a2cbe7:

Extracting 2.949MB/31.41MB

585124038cab:

Downloading 133B/133B

585124038cab: Verifying Checksum

585124038cab: Download complete

b483de716a47:

Downloading 574B/574B

b483de716a47: Verifying Checksum

b483de716a47: Download complete

3f9582a2cbe7:

Extracting 6.226MB/31.41MB

3f9582a2cbe7:

Extracting 8.52MB/31.41MB

3f9582a2cbe7:

Extracting 11.14MB/31.41MB

3f9582a2cbe7:

Extracting 13.76MB/31.41MB

3f9582a2cbe7:

Extracting 15.4MB/31.41MB

3f9582a2cbe7:

Extracting 17.37MB/31.41MB

3f9582a2cbe7:

Extracting 20.97MB/31.41MB

3f9582a2cbe7:

Extracting 24.58MB/31.41MB

3f9582a2cbe7:

Extracting 27.2MB/31.41MB

3f9582a2cbe7:

Extracting 28.84MB/31.41MB

3f9582a2cbe7:

Extracting 29.82MB/31.41MB

3f9582a2cbe7:

Extracting 30.15MB/31.41MB

3f9582a2cbe7:

Extracting 30.47MB/31.41MB

3f9582a2cbe7:

Extracting 31.13MB/31.41MB

3f9582a2cbe7:

Extracting 31.41MB/31.41MB

3f9582a2cbe7: Pull complete

241c2d338588:

Extracting 1.732kB/1.732kB

241c2d338588:

Extracting 1.732kB/1.732kB

241c2d338588: Pull complete

89515d93a23e:

Extracting 32.77kB/1.463MB

89515d93a23e:

Extracting 1.463MB/1.463MB

89515d93a23e:

Extracting 1.463MB/1.463MB

```
89515d93a23e: Pull complete
```

```
65e8ba9473fe:
```

```
Extracting 98.3kB/9.579MB
```

```
65e8ba9473fe:
```

```
Extracting 1.769MB/9.579MB
```

```
65e8ba9473fe:
```

```
Extracting 3.932MB/9.579MB
```

```
65e8ba9473fe:
```

```
Extracting 6.783MB/9.579MB
```

```
65e8ba9473fe:
```

```
Extracting 8.946MB/9.579MB
```

```
65e8ba9473fe:
```

```
Extracting 9.579MB/9.579MB
```

```
65e8ba9473fe: Pull complete
```

```
585124038cab: Extracting 133B/133B
```

```
585124038cab: Extracting 133B/133B
```

```
585124038cab: Pull complete
```

```
b483de716a47: Extracting 574B/574B
```

```
b483de716a47: Extracting 574B/574B
```

```
b483de716a47: Pull complete
```

```
Digest: sha256:e50c7e23f79ae81351beacb20e
```

```
Status: Downloaded newer image for  
redis:7.0.9
```

```
d6548aa7f7a5a5454bb8f330d9e2bb200eb58a95f
```

6.3 Vérifier le réseau (Avancé)

```
#| label: lst-docker-network-inspect  
docker network inspect app-net
```

```
[  
  {  
    "Name": "app-net",  
    "Id": "a9ba21eedc24c69148c61c7d290f5f2b13dc89b0e71",  
    "Created":  
      "2026-10-14T06:23:53.832692352Z",  
    "Scope": "local",  
    "Driver": "bridge",  
    "EnableIPv4": true,  
    "EnableIPv6": false,  
    "IPAM": {  
      "Driver": "default",  
      "Options": {},  
      "Config": [  
        {  
          "Subnet":  
            "172.18.0.0/16",  
          "Gateway": "172.18.0.1"  
        }  
      ]  
    },  
    "Internal": false,  
    "Attachable": false,  
    "Ingress": false,  
    "ConfigFrom": {  
      "Network": ""  
    },  
    "ConfigOnly": false,  
    "Containers": {  
      "b737f62c1ef26bfb3e32cbae9eeecfdbd38778c054c3c9"  
    }  
    {  
      "Name": "web",  
      "EndpointID": "d57574c587c4edc2d3bcebe16501a",  
      "MacAddress":  
        "e2:95:9d:14:ab:62",  
      "IPv4Address":  
        "172.18.0.2/16",  
      "IPv6Address": ""  
    },  
    "d6548aa7f7a5a5454bb8f330d9e2bb200eb58a95f813a4"  
  }  
  {  
    "Name": "db",  
    "EndpointID": "af15157095ccdc426be3078550cd4"
```

```

        "MacAddress":
"62:e3:1d:67:d0:05",
        "IPv4Address":
"172.18.0.3/16",
        "IPv6Address": ""
    }
},
"Options": {},
"Labels": {}
}
]

```

6.4 Communication conteneur → conteneur

```

#| label: lst-docker-ping-success
docker run --label ebpro-render=true
--rm --network app-net curlimages/
curl:7.88.1 \
    --retry 5 --retry-connrefused --
retry-delay 1 http://web

```

```

Unable to find image 'curlimages/
curl:7.88.1' locally
7.88.1: Pulling from curlimages/curl

```

```

213ec9aee27d: Pulling fs layer
ea634e3b33ec: Pulling fs layer
55bfd993f83a: Pulling fs layer
df0b84b7230e: Pulling fs layer
8d68097e7e08: Pulling fs layer
4c26da78c210: Pulling fs layer
659101a913e8: Pulling fs layer
473ceed980f8: Pulling fs layer

```

```

b0calde0cc3e: Pulling fs layer
45477c99a790: Pulling fs layer
8d68097e7e08: Waiting
df0b84b7230e: Waiting
d4c655b444ee: Pulling fs layer
473ceed980f8: Waiting
b0calde0cc3e: Waiting
4c26da78c210: Waiting
659101a913e8: Waiting
45477c99a790: Waiting
d4c655b444ee: Waiting
55bfd993f83a:
Downloading 16.38kB/608.7kB
213ec9aee27d:
Downloading 32.77kB/2.806MB
ea634e3b33ec:
Downloading 48.51kB/4.396MB
213ec9aee27d:
Downloading 2.806MB/2.806MB
213ec9aee27d: Download complete
213ec9aee27d:
Extracting 32.77kB/2.806MB
ea634e3b33ec:
Downloading 4.396MB/4.396MB

```

ea634e3b33ec: Verifying Checksum

ea634e3b33ec: Download complete

213ec9aee27d:
Extracting 1.671MB/2.806MB

213ec9aee27d:
Extracting 2.806MB/2.806MB

55bfd993f83a:
Downloading 555.7kB/608.7kB

55bfd993f83a: Verifying Checksum

55bfd993f83a: Download complete

213ec9aee27d: Pull complete

ea634e3b33ec:
Extracting 65.54kB/4.396MB

8d68097e7e08:
Downloading 16.38kB/123.2kB

df0b84b7230e:
Downloading 1.265kB/1.265kB

df0b84b7230e: Verifying Checksum

df0b84b7230e: Download complete

8d68097e7e08:
Downloading 123.2kB/123.2kB

8d68097e7e08: Verifying Checksum

8d68097e7e08: Download complete

ea634e3b33ec:
Extracting 2.294MB/4.396MB

ea634e3b33ec:

Extracting 4.396MB/4.396MB

4c26da78c210:
Downloading 16.38kB/347.4kB

ea634e3b33ec: Pull complete

4c26da78c210: Verifying Checksum

4c26da78c210: Download complete

55bfd993f83a:
Extracting 32.77kB/608.7kB

55bfd993f83a:
Extracting 608.7kB/608.7kB

55bfd993f83a:
Extracting 608.7kB/608.7kB

473ceed980f8:
Downloading 3.5kB/52.34kB

659101a913e8:
Downloading 3.457kB/162.4kB

473ceed980f8: Download complete

55bfd993f83a: Pull complete

659101a913e8:
Downloading 162.4kB/162.4kB

659101a913e8: Verifying Checksum

659101a913e8: Download complete

df0b84b7230e:
Extracting 1.265kB/1.265kB

df0b84b7230e:
Extracting 1.265kB/1.265kB

df0b84b7230e: Pull complete

8d68097e7e08:

Extracting 32.77kB/123.2kB

8d68097e7e08:

Extracting 123.2kB/123.2kB

8d68097e7e08:

Extracting 123.2kB/123.2kB

b0calde0cc3e:

Downloading 161B/161B

b0calde0cc3e: Verifying Checksum

b0calde0cc3e: Download complete

8d68097e7e08: Pull complete

4c26da78c210:

Extracting 32.77kB/347.4kB

4c26da78c210:

Extracting 347.4kB/347.4kB

4c26da78c210:

Extracting 347.4kB/347.4kB

45477c99a790:

Downloading 159B/159B

45477c99a790: Verifying Checksum

45477c99a790: Download complete

d4c655b444ee:

Downloading 291B/291B

d4c655b444ee: Verifying Checksum

d4c655b444ee: Download complete

4c26da78c210: Pull complete

659101a913e8:

Extracting 32.77kB/162.4kB

659101a913e8:

Extracting 162.4kB/162.4kB

659101a913e8:

Extracting 162.4kB/162.4kB

659101a913e8: Pull complete

473ceed980f8:

Extracting 32.77kB/52.34kB

473ceed980f8:

Extracting 52.34kB/52.34kB

473ceed980f8: Pull complete

b0calde0cc3e: Extracting 161B/161B

b0calde0cc3e: Extracting 161B/161B

b0calde0cc3e: Pull complete

45477c99a790: Extracting 159B/159B

45477c99a790: Extracting 159B/159B

45477c99a790: Pull complete

d4c655b444ee: Extracting 291B/291B

d4c655b444ee: Extracting 291B/291B

d4c655b444ee: Pull complete

Digest: sha256:48318407b8d98e8c7d5bd4741c88e8e1a5442de6

Status: Downloaded newer image for
curlimages/curl:7.88.1

% Total	% Received	% Xferd	Average
Speed	Time	Time	Time Current

```

Upload    Total    Spent    Left    Dload
Speed

0         0         0         0         0         0
0 ---:--:-- ---:--:-- ---:--:-- 0
100 615 100 615 0 0 69327
0 ---:--:-- ---:--:-- ---:--:-- 76875
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial,
sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web
server is successfully installed and
working. Further configuration is
required.</p>

<p>For online documentation and
support please refer to
<a href="http://nginx.
org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.
com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</
em></p>
</body>
</html>

```

6.5 Exercice 2 (Max 10 min)

Créer un réseau `myenet`, lancer deux conte-
neurs `alpine` nommés `a1` et `a2` sur ce
réseau, puis vérifier qu'ils se ping par nom

en exécutant une commande dans un
conteneur en mode **non-interactif**.

7 Port mapping

- Chaque conteneur a son propre espace ré-
seau isolé.
- Ils peuvent donc utiliser les mêmes ports en
interne sans conflit.
 - Ex: plusieurs conteneurs `postgres` peuvent
écouter sur **leur** port 5432.
- Pour exposer un service d'un conteneur vers
l'hôte, utiliser `--publish hôte:conteneur`.
- Cela mappe un port de l'hôte vers un port du
conteneur.
 - Ex: `--publish 8080:80` mappe le port
8080 de l'hôte vers le port 80 du conteneur.
 - Attention aux conflits de ports sur l'hôte.
 - Attention si docker tourne dans une VM
(WSL, Docker Desktop) : le port est exposé
sur la VM, pas directement sur l'hôte.
 - Le port mapping est une règle de NAT
entre l'hôte et le réseau bridge Docker.

```

#| label: lst-docker-port-mapping
docker rm -f web-port-1 >/dev/null
2>&1 || true
docker run --label ebpro-render=true
--detach --name web-port-1 --publish
8080:80 nginx:1.28

docker rm -f web-port-2 >/dev/null
2>&1 || true
docker run --label ebpro-render=true
--detach --name web-port-2 --publish
8081:80 nginx:1.23

```

```

Unable to find image 'nginx:1.28'
locally
1.28: Pulling from library/nginx

```

```

5435b2dcdf5c: Pulling fs layer

```

6a7080264fc9: Pulling fs layer

f176d5d8a1c1: Pulling fs layer

63ef37274310: Pulling fs layer

4744f2344932: Pulling fs layer

39a24e95a1e9: Pulling fs layer

400f8ac367e0: Pulling fs layer

63ef37274310: Waiting

4744f2344932: Waiting

39a24e95a1e9: Waiting

400f8ac367e0: Waiting

f176d5d8a1c1:
Downloading 627B/627B

f176d5d8a1c1: Verifying Checksum

f176d5d8a1c1: Download complete

5435b2dcdf5c:
Downloading 310.6kB/29.78MB

6a7080264fc9:
Downloading 341.2kB/33.11MB

5435b2dcdf5c:
Downloading 5.291MB/29.78MB

6a7080264fc9:
Downloading 5.123MB/33.11MB

5435b2dcdf5c:
Downloading 11.83MB/29.78MB

6a7080264fc9:
Downloading 10.57MB/33.11MB

63ef37274310:
Downloading 955B/955B

63ef37274310: Verifying Checksum

63ef37274310: Download complete

5435b2dcdf5c:
Downloading 19.61MB/29.78MB

6a7080264fc9:
Downloading 16.06MB/33.11MB

5435b2dcdf5c:
Downloading 25.21MB/29.78MB

6a7080264fc9:
Downloading 20.49MB/33.11MB

5435b2dcdf5c: Download complete

6a7080264fc9:
Downloading 26.28MB/33.11MB

5435b2dcdf5c:
Extracting 327.7kB/29.78MB

6a7080264fc9: Verifying Checksum

6a7080264fc9: Download complete

4744f2344932:
Downloading 402B/402B

4744f2344932: Download complete

5435b2dcdf5c:
Extracting 3.277MB/29.78MB

5435b2dcdf5c:

Extracting 5.898MB/29.78MB

39a24e95a1e9:
Downloading 1.208kB/1.208kB

39a24e95a1e9: Verifying Checksum

39a24e95a1e9: Download complete

5435b2dcdf5c:
Extracting 8.52MB/29.78MB

400f8ac367e0:
Downloading 1.397kB/1.397kB

400f8ac367e0: Verifying Checksum

400f8ac367e0: Download complete

5435b2dcdf5c:
Extracting 10.49MB/29.78MB

5435b2dcdf5c:
Extracting 12.12MB/29.78MB

5435b2dcdf5c:
Extracting 15.4MB/29.78MB

5435b2dcdf5c:
Extracting 18.35MB/29.78MB

5435b2dcdf5c:
Extracting 21.63MB/29.78MB

5435b2dcdf5c:
Extracting 24.9MB/29.78MB

5435b2dcdf5c:
Extracting 26.21MB/29.78MB

5435b2dcdf5c:
Extracting 28.51MB/29.78MB

5435b2dcdf5c:
Extracting 28.84MB/29.78MB

5435b2dcdf5c:
Extracting 29.16MB/29.78MB

5435b2dcdf5c:
Extracting 29.78MB/29.78MB

5435b2dcdf5c: Pull complete

6a7080264fc9:
Extracting 360.4kB/33.11MB

6a7080264fc9:
Extracting 2.884MB/33.11MB

6a7080264fc9:
Extracting 6.128MB/33.11MB

6a7080264fc9:
Extracting 9.372MB/33.11MB

6a7080264fc9:
Extracting 12.98MB/33.11MB

6a7080264fc9:
Extracting 14.78MB/33.11MB

6a7080264fc9:
Extracting 18.38MB/33.11MB

6a7080264fc9:
Extracting 21.63MB/33.11MB

6a7080264fc9:
Extracting 25.23MB/33.11MB

6a7080264fc9:
Extracting 30.28MB/33.11MB

6a7080264fc9:
Extracting 30.64MB/33.11MB

```

6a7080264fc9:
Extracting 32.08MB/33.11MB

6a7080264fc9:
Extracting 33.11MB/33.11MB

6a7080264fc9: Pull complete

f176d5d8a1c1: Extracting 627B/627B

f176d5d8a1c1: Extracting 627B/627B

f176d5d8a1c1: Pull complete

63ef37274310: Extracting 955B/955B

63ef37274310: Extracting 955B/955B

63ef37274310: Pull complete

4744f2344932: Extracting 402B/402B

4744f2344932: Extracting 402B/402B

4744f2344932: Pull complete

39a24e95a1e9:
Extracting 1.208kB/1.208kB

39a24e95a1e9:
Extracting 1.208kB/1.208kB

39a24e95a1e9: Pull complete

400f8ac367e0:
Extracting 1.397kB/1.397kB

400f8ac367e0:
Extracting 1.397kB/1.397kB

400f8ac367e0: Pull complete
Digest: sha256:146adea4768b83c607d0bdfa41

```

```

Status: Downloaded newer image for
nginx:1.28
e952461c8c9bba71860cd749354cee4ba67c3e4695bdd9b72c336f3
5d35ed14e1b2eee176b1645070cf9d21b9562a51b0b6d5963df6916

```

7.1 Tester depuis l'hôte

- Il est possible de tester l'accès au service exposé depuis l'hôte avec curl ou un navigateur.
- Avec l'adresse `http://localhost:8080` si Docker est natif.
- Avec l'adresse IP de la VM si Docker tourne dans une VM (ex: WSL, Docker Desktop).
- A moins que Docker soit configuré pour exposer les ports vers l'hôte directement.

```
curl --max-time 30 -I http://localhost:8080
```

```
curl --max-time 30 -I http://localhost:8081
```

```
#| label: lst-docker-port-mapping-curl
curl --max-time 30 -I http://dind:8080
```

```
curl --max-time 30 -I http://dind:8081
```

```

HTTP/1.1 200 OK
Server: nginx/1.28.3
Date: Wed, 14 Oct 2026 06:24:22 GMT
Content-Type: text/html
Content-Length: 615
Last-Modified: Tue, 24 Mar 2026 18:33:23 GMT
Connection: keep-alive
ETag: "69c2d8f3-267"
Accept-Ranges: bytes

```

```

HTTP/1.1 200 OK
Server: nginx/1.23.4
Date: Wed, 14 Oct 2026 06:24:22 GMT
Content-Type: text/html
Content-Length: 615
Last-Modified: Tue, 28 Mar 2023
15:01:54 GMT
Connection: keep-alive
ETag: "64230162-267"
Accept-Ranges: bytes

```

7.2 Accès conteneur -> hôte

- Il est possible d'accéder à un service exposé sur l'hôte depuis un conteneur.
- Utiliser `host.docker.internal` comme nom d'hôte.
- ATTENTION : `host.docker.internal` fonctionne nativement sur Docker Desktop (Windows/Mac) et Docker sous Linux (depuis Docker 20.10) mais n'est standardisé que sur Docker Desktop.

```

docker run --label ebpro-render=true
--rm \
    curlimages/curl:7.88.1 http://
host.docker.internal:8080

```

8 Volumes : persistance

- Les données dans un conteneur sont éphémères : elles disparaissent à l'arrêt/suppression du conteneur.
- Pour persister les données, utiliser des volumes.
- Un volume est un espace de stockage géré par Docker, indépendant du cycle de vie des conteneurs.
- Les volumes peuvent être montés dans un conteneur avec `--volume volume:chemin_dans_conteneur`.
- Il existe deux types de volumes :
 - **Bind mount** : mappe un répertoire de l'hôte vers un répertoire du conteneur.

- **Volume nommé** : volume géré par Docker, stocké dans l'espace de stockage Docker.

💡 Astuce

- bind mount = couplage fort hôte ↔ conteneur
- volume nommé = abstraction portable

8.1 Bind mount (risky, dev)

- montage avec `--volume /chemin/host:/chemin/conteneur`

```

#| output: true
#| echo: true

#| label: lst-bind-mount-nginx
echo "hello" > ${TP_DIR}/www/
index.html

docker run --label ebpro-render=true
--rm -v ${TP_DIR}/www:/usr/share/
nginx/html \
    -p :80 nginx:1.23

```

- Utile en développement pour monter du code source.
- Permet d'éditer les fichiers sur l'hôte et de les voir dans le conteneur.
- Dangers :
 - les permissions peuvent poser problème selon l'OS hôte
 - les performances peuvent être dégradées (ex: Docker Desktop sur Windows/Mac)
 - le conteneur ne peut pas être déplacé facilement (dépendance à l'hôte)

8.2 Volume nommé (better, prod)

- montage avec --volume nom_volume:chemin/conteneur

```
#| label: lst-named-volume-postgres
IMAGE="postgres:15.2"
VOLUME="pgdata"
PW="secret"
```

```
docker volume create "$VOLUME"
```

```
# Initialisation de la base et
création d'une table
docker rm -f db >/dev/null 2>&1 ||
true
docker run --label ebpro-render=true
-d --name db \
    -v "$VOLUME":/var/lib/postgresql/
data \
    -e POSTGRES_PASSWORD="$PW" \
    "$IMAGE"
```

```
# Attendre que la base soit prête
timeout 60 sh -c 'until docker exec
db pg_isready -U postgres >/dev/null
2>&1; do echo -n '.'; sleep 1; done'
|| (echo "Postgres not ready after
60s" >&2; exit 1)
```

```
# Créer une table et insérer des
données depuis le conteneur de la base
docker exec db psql -U postgres -c
"CREATE TABLE test (id SERIAL PRIMARY
KEY, name VARCHAR(50));"
docker exec db psql -U postgres -
c "INSERT INTO test (name) VALUES
('Alice'), ('Bob');"
```

```
# Arrêter et supprimer le conteneur
mais conserver le volume
docker stop db && docker rm db
```

```
pgdata
91c33293c9d7104b12b1d184325b763d924c871bab4ca01e3b077e4
..
CREATE TABLE
INSERT 0 2
db
db
```

```
#| label: lst-verify-named-volume
# Redémarrer un nouveau conteneur
avec le même volume
docker rm -f db2 >/dev/null 2>&1 ||
true
docker run --label ebpro-render=true
-d --name db2 \
    -v "$VOLUME":/var/lib/postgresql/
data \
    -e POSTGRES_PASSWORD="$PW" \
    "$IMAGE"
```

```
# Attendre que la base soit prête
timeout 60 sh -c 'until docker exec
db2 pg_isready -U postgres >/dev/null
2>&1; do echo -n '.'; sleep 1; done'
|| (echo "Postgres not ready after
60s" >&2; exit 1)
```

```
# Vérifier que les données persistent
docker exec db2 psql -U postgres -c
"SELECT * FROM test;"
```

```
# Nettoyage
docker stop db2 && docker rm db2
docker volume rm "$VOLUME"
```

```
eab6c544a65825a78a045935cdae961ccf1dd437f6efcd7cdba1b35
id | name
----+-----
 1 | Alice
 2 | Bob
(2 rows)
```

```
db2
db2
pgdata
```

8.3 Exercice de persistance

- lancer MySQL sans volume → perdre les données
- relancer avec volume → conserver les données

Hint: l'image officielle MySQL stocke les données dans `/var/lib/mysql`. Voir la page Docker Hub : https://hub.docker.com/_/mysql

Exercice 3 (10 min)

Démarrer MySQL au premier plan, observer l'initialisation (création de la base), arrêter avec Ctrl-C, puis relancer le même volume et constater qu'il n'y a plus d'initialisation.

Consignes :

- Lisez la page Docker Hub de MySQL pour comprendre
 - les variables d'environnement essentielles (notamment `MYSQL_ROOT_PASSWORD`, `MYSQL_DATABASE`) : https://hub.docker.com/_/mysql
 - le chemin de stockage des données dans le conteneur (`/var/lib/mysql`)
- Créez un volume nommé `mysql-data` pour stocker les données persistantes.
- Démarrez MySQL en mode **attaché** (sans `--detach`)

```
#| label: lst-mysql-persistence

# create a named volume for MySQL data
(one-time)
```

```
docker volume create mysql-data

export MYSQL_ROOT_PASSWORD=secretpw
export MYSQL_DATABASE=demo

# Run MySQL in the foreground so you
see initialization logs
docker rm -f mysql-db >/dev/null 2>&1
|| true
docker run --label ebpro-render=true
--name mysql-db --detach\
--env MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASSWORD} \
--env
MYSQL_DATABASE=${MYSQL_DATABASE} \
--volume mysql-data:/var/lib/mysql \
mysql:8.0

# Wait for MySQL to be ready
timeout 60 sh -c 'until docker
exec mysql-db mysqladmin ping -h
"localhost" --silent >/dev/null 2>&1;
do echo -n "."; sleep 1; done' ||
(echo "MySQL not ready after 60s" >&2;
exit 1)

# Stop and remove the container to
simulate a restart
docker stop mysql-db && docker rm
mysql-db

# Start a new container with the same
volume to see data persistence.
docker rm -f mysql-db >/dev/null 2>&1
|| true
docker run --label ebpro-render=true
--name mysql-db --detach\
--env MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASSWORD} \
--env
MYSQL_DATABASE=${MYSQL_DATABASE} \
--volume mysql-data:/var/lib/mysql
```

```

\
mysql:8.0

# Wait for MySQL to be ready again
timeout 60 sh -c 'until docker
exec mysql-db mysqladmin ping -h
"localhost" --silent >/dev/null 2>&1;
do echo -n "."; sleep 1; done' ||
(echo "MySQL not ready after 60s" >&2;
exit 1)

# Check the logs to confirm no re-
initialization
docker logs mysql-db|head -n 20

```

```

mysql-data
Unable to find image 'mysql:8.0'
locally
8.0: Pulling from library/mysql

edf85873f64e: Pulling fs layer
6ef6c7b50a93: Pulling fs layer
e3e5d1ac74c1: Pulling fs layer
0d74d296605b: Pulling fs layer
297d04cfe470: Pulling fs layer
4c8a3e0d4e4b: Pulling fs layer
a63160a5eda1: Pulling fs layer
7534d1db9f8d: Pulling fs layer
49ec2dab01d9: Pulling fs layer
0d74d296605b: Waiting
a63160a5eda1: Waiting

```

```

297d04cfe470: Waiting
4c8a3e0d4e4b: Waiting
7534d1db9f8d: Waiting
ab24264a27e9: Pulling fs layer
96d30d9fbee8: Pulling fs layer
ab24264a27e9: Waiting
49ec2dab01d9: Waiting
96d30d9fbee8: Waiting
6ef6c7b50a93:
Downloading      884B/884B
6ef6c7b50a93: Verifying Checksum
6ef6c7b50a93: Download complete
e3e5d1ac74c1:
Downloading  15.74kB/783.6kB
edf85873f64e:
Downloading  488.6kB/47.31MB
e3e5d1ac74c1:
Downloading  783.6kB/783.6kB
e3e5d1ac74c1: Verifying Checksum
e3e5d1ac74c1: Download complete
edf85873f64e:
Downloading  7.713MB/47.31MB
edf85873f64e:
Downloading   14.9MB/47.31MB

```

0d74d296605b:
Downloading 64.87kB/6.173MB

edf85873f64e:
Downloading 22.58MB/47.31MB

297d04cfe470:
Downloading 2.607kB/2.607kB

297d04cfe470: Verifying Checksum

297d04cfe470: Download complete

0d74d296605b:
Downloading 5.046MB/6.173MB

edf85873f64e:
Downloading 28.33MB/47.31MB

0d74d296605b: Verifying Checksum

0d74d296605b: Download complete

edf85873f64e:
Downloading 35.52MB/47.31MB

edf85873f64e:
Downloading 42.23MB/47.31MB

4c8a3e0d4e4b:
Downloading 335B/335B

4c8a3e0d4e4b: Verifying Checksum

4c8a3e0d4e4b: Download complete

edf85873f64e: Verifying Checksum

edf85873f64e: Download complete

edf85873f64e:
Extracting 491.5kB/47.31MB

a63160a5eda1:
Downloading 507kB/49.93MB

edf85873f64e:
Extracting 983kB/47.31MB

a63160a5eda1:
Downloading 7.095MB/49.93MB

7534d1db9f8d:
Downloading 316B/316B

7534d1db9f8d: Verifying Checksum

7534d1db9f8d: Download complete

edf85873f64e:
Extracting 1.966MB/47.31MB

a63160a5eda1:
Downloading 12.67MB/49.93MB

49ec2dab01d9:
Downloading 539.4kB/129.4MB

edf85873f64e:
Extracting 3.932MB/47.31MB

a63160a5eda1:
Downloading 17.24MB/49.93MB

49ec2dab01d9:
Downloading 5.353MB/129.4MB

edf85873f64e:
Extracting 5.898MB/47.31MB

a63160a5eda1:
Downloading 21.8MB/49.93MB

49ec2dab01d9:
Downloading 9.616MB/129.4MB

ab24264a27e9:
Downloading 3.478kB/5.327kB

ab24264a27e9:
Downloading 5.327kB/5.327kB

ab24264a27e9: Verifying Checksum

ab24264a27e9: Download complete

a63160a5eda1:
Downloading 26.88MB/49.93MB

49ec2dab01d9:
Downloading 18.17MB/129.4MB

a63160a5eda1:
Downloading 32.98MB/49.93MB

49ec2dab01d9:
Downloading 21.91MB/129.4MB

a63160a5eda1:
Downloading 37.56MB/49.93MB

edf85873f64e:
Extracting 7.373MB/47.31MB

49ec2dab01d9:
Downloading 26.7MB/129.4MB

96d30d9fbee8:
Downloading 120B/120B

96d30d9fbee8: Download complete

a63160a5eda1:
Downloading 42.63MB/49.93MB

49ec2dab01d9:
Downloading 33.07MB/129.4MB

edf85873f64e:

Extracting 9.339MB/47.31MB

a63160a5eda1:
Downloading 48.74MB/49.93MB

a63160a5eda1: Verifying Checksum

a63160a5eda1: Download complete

49ec2dab01d9:
Downloading 41.1MB/129.4MB

edf85873f64e:
Extracting 12.78MB/47.31MB

49ec2dab01d9:
Downloading 42.16MB/129.4MB

edf85873f64e:
Extracting 15.73MB/47.31MB

49ec2dab01d9:
Downloading 49.65MB/129.4MB

edf85873f64e:
Extracting 18.19MB/47.31MB

49ec2dab01d9:
Downloading 54.99MB/129.4MB

49ec2dab01d9:
Downloading 61.38MB/129.4MB

edf85873f64e:
Extracting 21.14MB/47.31MB

49ec2dab01d9:
Downloading 67.75MB/129.4MB

edf85873f64e:
Extracting 23.59MB/47.31MB

49ec2dab01d9:

Downloading 69.34MB/129.4MB

edf85873f64e:

Extracting 24.58MB/47.31MB

49ec2dab01d9:

Downloading 75.73MB/129.4MB

edf85873f64e:

Extracting 27.53MB/47.31MB

49ec2dab01d9:

Downloading 82.13MB/129.4MB

edf85873f64e:

Extracting 30.47MB/47.31MB

49ec2dab01d9:

Downloading 91.71MB/129.4MB

edf85873f64e:

Extracting 31.95MB/47.31MB

49ec2dab01d9:

Downloading 101.3MB/129.4MB

edf85873f64e:

Extracting 33.91MB/47.31MB

49ec2dab01d9:

Downloading 108.8MB/129.4MB

edf85873f64e:

Extracting 40.8MB/47.31MB

49ec2dab01d9:

Downloading 115.2MB/129.4MB

edf85873f64e:

Extracting 41.78MB/47.31MB

edf85873f64e:

Extracting 42.27MB/47.31MB

49ec2dab01d9:

Downloading 121.6MB/129.4MB

49ec2dab01d9:

Downloading 129.1MB/129.4MB

49ec2dab01d9: Verifying Checksum

49ec2dab01d9: Download complete

edf85873f64e:

Extracting 42.76MB/47.31MB

edf85873f64e:

Extracting 44.24MB/47.31MB

edf85873f64e:

Extracting 46.2MB/47.31MB

edf85873f64e:

Extracting 47.31MB/47.31MB

edf85873f64e: Pull complete

6ef6c7b50a93: Extracting 884B/884B

6ef6c7b50a93: Extracting 884B/884B

6ef6c7b50a93: Pull complete

e3e5d1ac74c1:

Extracting 32.77kB/783.6kB

e3e5d1ac74c1:

Extracting 783.6kB/783.6kB

e3e5d1ac74c1:

Extracting 783.6kB/783.6kB

e3e5d1ac74c1: Pull complete

0d74d296605b:

Extracting 65.54kB/6.173MB

0d74d296605b:

Extracting 1.18MB/6.173MB

0d74d296605b:

Extracting 2.425MB/6.173MB

0d74d296605b:

Extracting 5.833MB/6.173MB

0d74d296605b:

Extracting 6.173MB/6.173MB

0d74d296605b: Pull complete

297d04cfe470:

Extracting 2.607kB/2.607kB

297d04cfe470:

Extracting 2.607kB/2.607kB

297d04cfe470: Pull complete

4c8a3e0d4e4b: Extracting 335B/335B

4c8a3e0d4e4b: Extracting 335B/335B

4c8a3e0d4e4b: Pull complete

a63160a5eda1:

Extracting 524.3kB/49.93MB

a63160a5eda1:

Extracting 3.146MB/49.93MB

a63160a5eda1:

Extracting 6.291MB/49.93MB

a63160a5eda1:

Extracting 8.913MB/49.93MB

a63160a5eda1:

Extracting 11.01MB/49.93MB

a63160a5eda1:

Extracting 13.63MB/49.93MB

a63160a5eda1:

Extracting 15.73MB/49.93MB

a63160a5eda1:

Extracting 17.83MB/49.93MB

a63160a5eda1:

Extracting 21.5MB/49.93MB

a63160a5eda1:

Extracting 23.59MB/49.93MB

a63160a5eda1:

Extracting 26.21MB/49.93MB

a63160a5eda1:

Extracting 29.88MB/49.93MB

a63160a5eda1:

Extracting 34.08MB/49.93MB

a63160a5eda1:

Extracting 37.22MB/49.93MB

a63160a5eda1:

Extracting 38.27MB/49.93MB

a63160a5eda1:

Extracting 40.89MB/49.93MB

a63160a5eda1:

Extracting 43.52MB/49.93MB

a63160a5eda1:

Extracting 46.14MB/49.93MB

a63160a5eda1:

Extracting 48.76MB/49.93MB

a63160a5eda1:
Extracting 49.93MB/49.93MB

a63160a5eda1: Pull complete

7534d1db9f8d: Extracting 316B/316B

7534d1db9f8d: Extracting 316B/316B

7534d1db9f8d: Pull complete

49ec2dab01d9:
Extracting 557.1kB/129.4MB

49ec2dab01d9:
Extracting 3.342MB/129.4MB

49ec2dab01d9:
Extracting 6.685MB/129.4MB

49ec2dab01d9:
Extracting 9.47MB/129.4MB

49ec2dab01d9:
Extracting 12.81MB/129.4MB

49ec2dab01d9:
Extracting 15.04MB/129.4MB

49ec2dab01d9:
Extracting 17.83MB/129.4MB

49ec2dab01d9:
Extracting 19.5MB/129.4MB

49ec2dab01d9:
Extracting 21.73MB/129.4MB

49ec2dab01d9:
Extracting 23.95MB/129.4MB

49ec2dab01d9:

Extracting 25.07MB/129.4MB

49ec2dab01d9:
Extracting 27.3MB/129.4MB

49ec2dab01d9:
Extracting 28.97MB/129.4MB

49ec2dab01d9:
Extracting 31.2MB/129.4MB

49ec2dab01d9:
Extracting 32.31MB/129.4MB

49ec2dab01d9:
Extracting 32.87MB/129.4MB

49ec2dab01d9:
Extracting 33.42MB/129.4MB

49ec2dab01d9:
Extracting 33.98MB/129.4MB

49ec2dab01d9:
Extracting 34.54MB/129.4MB

49ec2dab01d9:
Extracting 35.09MB/129.4MB

49ec2dab01d9:
Extracting 35.65MB/129.4MB

49ec2dab01d9:
Extracting 36.21MB/129.4MB

49ec2dab01d9:
Extracting 36.77MB/129.4MB

49ec2dab01d9:
Extracting 37.32MB/129.4MB

49ec2dab01d9:
Extracting 37.88MB/129.4MB

49ec2dab01d9:
Extracting 38.44MB/129.4MB

49ec2dab01d9:
Extracting 38.99MB/129.4MB

49ec2dab01d9:
Extracting 39.55MB/129.4MB

49ec2dab01d9:
Extracting 40.11MB/129.4MB

49ec2dab01d9:
Extracting 40.67MB/129.4MB

49ec2dab01d9:
Extracting 41.22MB/129.4MB

49ec2dab01d9:
Extracting 41.78MB/129.4MB

49ec2dab01d9:
Extracting 42.34MB/129.4MB

49ec2dab01d9:
Extracting 42.89MB/129.4MB

49ec2dab01d9:
Extracting 43.45MB/129.4MB

49ec2dab01d9:
Extracting 44.01MB/129.4MB

49ec2dab01d9:
Extracting 44.56MB/129.4MB

49ec2dab01d9:
Extracting 45.12MB/129.4MB

49ec2dab01d9:
Extracting 45.68MB/129.4MB

49ec2dab01d9:
Extracting 46.24MB/129.4MB

49ec2dab01d9:
Extracting 46.79MB/129.4MB

49ec2dab01d9:
Extracting 47.91MB/129.4MB

49ec2dab01d9:
Extracting 49.02MB/129.4MB

49ec2dab01d9:
Extracting 49.58MB/129.4MB

49ec2dab01d9:
Extracting 50.14MB/129.4MB

49ec2dab01d9:
Extracting 51.25MB/129.4MB

49ec2dab01d9:
Extracting 53.48MB/129.4MB

49ec2dab01d9:
Extracting 55.71MB/129.4MB

49ec2dab01d9:
Extracting 58.49MB/129.4MB

49ec2dab01d9:
Extracting 61.28MB/129.4MB

49ec2dab01d9:
Extracting 64.62MB/129.4MB

49ec2dab01d9:
Extracting 67.4MB/129.4MB

49ec2dab01d9:
Extracting 70.19MB/129.4MB

49ec2dab01d9:

Extracting 73.53MB/129.4MB

49ec2dab01d9:

Extracting 76.87MB/129.4MB

49ec2dab01d9:

Extracting 79.66MB/129.4MB

49ec2dab01d9:

Extracting 82.44MB/129.4MB

49ec2dab01d9:

Extracting 85.23MB/129.4MB

49ec2dab01d9:

Extracting 88.01MB/129.4MB

49ec2dab01d9:

Extracting 91.36MB/129.4MB

49ec2dab01d9:

Extracting 93.03MB/129.4MB

49ec2dab01d9:

Extracting 95.26MB/129.4MB

49ec2dab01d9:

Extracting 98.6MB/129.4MB

49ec2dab01d9:

Extracting 103.6MB/129.4MB

49ec2dab01d9:

Extracting 107.5MB/129.4MB

49ec2dab01d9:

Extracting 112.5MB/129.4MB

49ec2dab01d9:

Extracting 117.5MB/129.4MB

49ec2dab01d9:

Extracting 122.6MB/129.4MB

49ec2dab01d9:

Extracting 124.8MB/129.4MB

49ec2dab01d9:

Extracting 127.6MB/129.4MB

49ec2dab01d9:

Extracting 129.4MB/129.4MB

49ec2dab01d9: Pull complete

ab24264a27e9:

Extracting 5.327kB/5.327kB

ab24264a27e9:

Extracting 5.327kB/5.327kB

ab24264a27e9: Pull complete

96d30d9fbee8: Extracting 120B/120B

96d30d9fbee8: Extracting 120B/120B

96d30d9fbee8: Pull complete

Digest: sha256:7dcddc01f13bab2f15cde676d44d01f61fc9f99f

Status: Downloaded newer image for

mysql:8.0

51350b8fa93ab2d0c4f86fc57c94e271fcf817711300e48fdd95295

.....

mysql-db

mysql-db

db16b2ae5c48b13c00701348c02e2c6cf478915b56a7f8483e1c29e

..

2026-10-14T06:25:59.736163Z 0

[Warning] [MY-011068] [Server]

The syntax '--skip-host-cache' is

deprecated and will be removed in a

future release. Please use SET GLOBAL

host_cache_size=0 instead.

2026-10-14 06:25:58+00:00 [Note]

[Entrypoint]: Entrypoint script for

MySQL Server 8.0.46-1.el9 started.

```

2026-10-14T06:25:59.738511Z      0
[System] [MY-010116] [Server] /usr/
sbin/mysqld (mysqld 8.0.46) starting
as process 1
2026-10-14 06:25:59+00:00 [Note]
[Entrypoint]: Switching to dedicated
user 'mysql'
2026-10-14 06:25:59+00:00 [Note]
[Entrypoint]: Entrypoint script for
MySQL Server 8.0.46-1.el9 started.
'/var/lib/mysql/mysql.sock' -> '/
var/run/mysqld/mysqld.sock'
2026-10-14T06:25:59.745937Z      1
[System] [MY-013576] [InnoDB] InnoDB
initialization has started.
2026-10-14T06:26:00.712713Z      1
[System] [MY-013577] [InnoDB] InnoDB
initialization has ended.
2026-10-14T06:26:01.383455Z      0
[Warning] [MY-010068] [Server] CA
certificate ca.pem is self signed.
2026-10-14T06:26:01.383548Z      0
[System] [MY-013602] [Server] Channel
mysql_main configured to support
TLS. Encrypted connections are now
supported for this channel.
2026-10-14T06:26:01.399271Z      0
[Warning] [MY-011810] [Server]
Insecure configuration for --pid-
file: Location '/var/run/mysqld' in
the path is accessible to all OS
users. Consider choosing a different
directory.
2026-10-14T06:26:01.428267Z      0
[System] [MY-011323] [Server] X
Plugin ready for connections. Bind-
address: '::' port: 33060, socket: /
var/run/mysqld/mysqlx.sock
2026-10-14T06:26:01.428367Z      0
[System] [MY-010931] [Server] /usr/
sbin/mysqld: ready for connections.
Version: '8.0.46' socket: '/var/

```

```

run/mysqld/mysqld.sock' port: 3306
MySQL Community Server - GPL.

```

9 Gestion des ressources (Avancé)

- Docker permet de limiter les ressources utilisées par un conteneur.
- Cela inclut la mémoire, le CPU, et le nombre de PIDs.

Limiter CPU / mémoire / PIDs

```

#| label: lst-limit-cpu-mem

docker rm -f cpu-test >/dev/null 2>&1
|| true
docker run --label ebpro-render=true
--detach --name cpu-test \
--memory="512m" \
--cpus="0.5" \
python:3.11-slim \
python -c "while True: pass"

```

```

Unable to find image 'python:3.11-
slim' locally
3.11-slim: Pulling from library/
python

```

```
ecc510cle359: Pulling fs layer
```

```
0f752a1982d6: Pulling fs layer
```

```
759a7f30c318: Pulling fs layer
```

```
496c1a902b3a: Pulling fs layer
```

```
496c1a902b3a: Waiting
```

```
0f752a1982d6:
Downloading 15.79kB/1.294MB

```

759a7f30c318:
Downloading 146.9kB/14.46MB

ecc510cle359:
Downloading 310.7kB/29.84MB

0f752a1982d6: Verifying Checksum

0f752a1982d6: Download complete

759a7f30c318:
Downloading 3.243MB/14.46MB

ecc510cle359:
Downloading 4.046MB/29.84MB

759a7f30c318:
Downloading 8.552MB/14.46MB

ecc510cle359:
Downloading 9.027MB/29.84MB

759a7f30c318:
Downloading 13.86MB/14.46MB

ecc510cle359:
Downloading 14.32MB/29.84MB

759a7f30c318: Verifying Checksum

759a7f30c318: Download complete

ecc510cle359:
Downloading 20.54MB/29.84MB

ecc510cle359:
Downloading 26.15MB/29.84MB

496c1a902b3a: Download complete

ecc510cle359: Verifying Checksum

ecc510cle359: Download complete

ecc510cle359:
Extracting 327.7kB/29.84MB

ecc510cle359:
Extracting 1.966MB/29.84MB

ecc510cle359:
Extracting 3.604MB/29.84MB

ecc510cle359:
Extracting 5.243MB/29.84MB

ecc510cle359:
Extracting 6.881MB/29.84MB

ecc510cle359:
Extracting 8.847MB/29.84MB

ecc510cle359:
Extracting 10.16MB/29.84MB

ecc510cle359:
Extracting 11.8MB/29.84MB

ecc510cle359:
Extracting 14.75MB/29.84MB

ecc510cle359:
Extracting 18.35MB/29.84MB

ecc510cle359:
Extracting 21.63MB/29.84MB

ecc510cle359:
Extracting 24.9MB/29.84MB

ecc510cle359:
Extracting 26.21MB/29.84MB

ecc510cle359:
Extracting 28.51MB/29.84MB

ecc510cle359:
Extracting 29.16MB/29.84MB

ecc510cle359:
Extracting 29.82MB/29.84MB

ecc510cle359:
Extracting 29.84MB/29.84MB

ecc510cle359: Pull complete

0f752a1982d6:
Extracting 32.77kB/1.294MB

0f752a1982d6:
Extracting 229.4kB/1.294MB

0f752a1982d6:
Extracting 852kB/1.294MB

0f752a1982d6:
Extracting 1.294MB/1.294MB

0f752a1982d6:
Extracting 1.294MB/1.294MB

0f752a1982d6: Pull complete

759a7f30c318:
Extracting 163.8kB/14.46MB

759a7f30c318:
Extracting 983kB/14.46MB

759a7f30c318:
Extracting 2.949MB/14.46MB

759a7f30c318:
Extracting 3.768MB/14.46MB

759a7f30c318:
Extracting 7.373MB/14.46MB

759a7f30c318:
Extracting 7.537MB/14.46MB

759a7f30c318:
Extracting 9.339MB/14.46MB

759a7f30c318:
Extracting 10.32MB/14.46MB

759a7f30c318:
Extracting 10.65MB/14.46MB

759a7f30c318:
Extracting 11.47MB/14.46MB

759a7f30c318:
Extracting 12.12MB/14.46MB

759a7f30c318:
Extracting 12.78MB/14.46MB

759a7f30c318:
Extracting 13.27MB/14.46MB

759a7f30c318:
Extracting 14.09MB/14.46MB

759a7f30c318:
Extracting 14.46MB/14.46MB

759a7f30c318: Pull complete

496c1a902b3a: Extracting 250B/250B

496c1a902b3a: Extracting 250B/250B

496c1a902b3a: Pull complete

Digest: sha256:e88e9763f943ec1834f992a4b51e0f2450048680

Status: Downloaded newer image for
python:3.11-slim

2c1410f68dfc7d81f57dce0f5383a22f753d4be462db54ee9cc9648

```
#| label: lst-monitor-cpu-mem

docker stats --no-stream cpu-test
```

CONTAINER ID	NAME	CPU %	MEM
2c1410f68dfc	cpu-test	50.90%	
3.926MiB / 512MiB		0.77%	266B /
84B 0B / 81.9kB	1		

Pensez à arrêter et supprimer le conteneur après le test :

```
#| label: lst-cleanup-cpu-test

docker stop cpu-test
docker rm cpu-test
```

```
cpu-test
cpu-test
```

9.1 Modifier les limites d'un conteneur existant

- Il est possible de modifier les limites d'un conteneur en cours d'exécution avec `docker update`.

```
docker update --memory="1g" --cpus="1" cpu-test
```

10 Bonnes pratiques d'usage

- **Nettoyage régulier**

```
docker container prune
docker image prune
docker volume prune
docker network prune
docker system prune --all
```

```
# NETTOYAGE COMPLET (ATTENTION : SUPPRIME TOUT)
# Y COMPRIS LES DONNÉES DANS DES VOLUMES !
docker system prune --all --volumes --force # RISQUE
```

- **Éviter `latest` en production**

- Utiliser des tags de version stables.

- **Ne pas stocker de secrets dans l'environnement**

- Préférer `--secret` (ou mécanismes externes).

- **Logs sur `stdout/stderr`**

- Les conteneurs doivent produire des logs vers la sortie standard.

11 Atelier final

- Déployer une application simple composée de deux conteneurs :

- une application Java accédant à une base de données via JDBC
- une base de données PostgreSQL

- Contraintes :

- chaque composant doit être dans son propre conteneur
- les deux conteneurs doivent communiquer via un réseau Docker personnalisé
- les données de la base doivent être persistées dans un volume Docker
- l'application Java doit se connecter à la base par le nom du conteneur, pas par localhost

- Objectif :

- vérifier la communication entre conteneurs, l'usage des réseaux et la persistance des données.

12 Conclusion

! A Retenir

- Un conteneur est une instance d'une image, exécutant un processus isolé.
- Le cycle de vie d'un conteneur inclut les états : démarré, arrêté, supprimé.
- Les conteneurs peuvent être connectés via des réseaux custom pour la communication.
- Les volumes permettent de persister les données indépendamment du cycle de vie des conteneurs.
- Il est possible de limiter les ressources (CPU, mémoire) utilisées par un conteneur.
- Appliquer les bonnes pratiques d'usage pour une gestion efficace des conteneurs.
- Gérer les conteneurs manuellement n'est pas viable en production : il faut automatiser
 - Outils d'Orchestration : docker compose, Kubernetes, ...