

TP FINAL Docker – 21/12/2023
DUMANOIR David & POIVET Antoine

- **Commande utilisée** : docker pull redis

- **Description** : On récupère l'image de la base de donnée Redis pour pouvoir l'utiliser par la suite.

- **Commande utilisée** : docker pull postgres:15-alpine

- **Description** : On récupère l'image de la base de données PostgreSQL version 15 avec une base Alpine pour une installation légère.

```
antoine@vmdocker:~$ docker pull redis
Using default tag: latest
latest: Pulling from library/redis
af107e978371: Pull complete
b031def5f2c4: Pull complete
bf7f0c8796d3: Pull complete
e3b2691a4104: Pull complete
190b4d7a237a: Pull complete
797591c7970a: Pull complete
4f4fb700ef54: Pull complete
45ce3854ac9a: Pull complete
Digest: sha256:7f26b254f1169010a2c0e160d166f0e12fb05ceab21a1efc342e43ad397a2674
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
antoine@vmdocker:~$ docker pull postgres:15-alpine
15-alpine: Pulling from library/postgres
661ff4d9561e: Already exists
e4a3f96ea8e5: Pull complete
0c1e2e159eal: Pull complete
26c071a8426e: Pull complete
e9a1ba05d22c: Pull complete
efc39a79d7dc: Pull complete
72124e665f9e: Pull complete
aa569f3e770e: Pull complete
86d5fe07cb37: Pull complete
Digest: sha256:e2a22801fcab638f9491039f8257e9f719ab02e8c78c6a6f2c0349505f92dc35
Status: Downloaded newer image for postgres:15-alpine
docker.io/library/postgres:15-alpine
```

- **Commande utilisée** : docker run -d -p 5000:5000 --restart always --name registry registry:2

- **Description** : On lance un conteneur Docker nommé "registry" en tant que registre local à l'écoute du port 5000, avec redémarrage automatique, permettant le stockage d'images Docker.

- **Commande utilisée** : docker tag redis:latest localhost:5000/redis:latest

- **Description** : On attribue une nouvelle étiquette à l'image Redis, la préparant ainsi pour le stockage dans le registre local à l'adresse localhost:5000.

- **Commande utilisée** : docker push localhost:5000/redis:latest

- **Description** : On pousse l'image Redis étiquetée vers le registre local à l'adresse localhost:5000, la rendant ainsi accessible à d'autres utilisateurs et systèmes.

- **Commande utilisée** : docker tag postgres:15-alpine localhost:5000/postgres:15-alpine

- **Description** : On attribue une nouvelle étiquette à l'image PostgreSQL, la préparant ainsi pour le stockage dans le registre local à l'adresse localhost:5000.

```
antoine@vmdocker:~$ docker tag postgres:15-alpine localhost:5000/postgres:15-alpine
antoine@vmdocker:~$ docker tag redis:latest localhost:5000/redis:latest
```

- **Commande utilisée** : `docker push localhost:5000/postgres:15-alpine`

- **Description** : On pousse l'image PostgreSQL étiquetée vers le registre local à l'adresse localhost:5000, la rendant ainsi accessible à d'autres utilisateurs et systèmes.

```
antoine@vmdocker:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ docker push localhost:5000/redis:latest
The push refers to repository [localhost:5000/redis]
3cb938f918a7: Pushed
5f70bf18a086: Pushed
97c159eabe6d: Pushed
987aefc1915a: Pushed
3dbd108cdf2f: Pushed
5b323dfbafb0: Pushed
a3688c42eece: Pushed
7292cf786aa8: Pushed
latest: digest: sha256:309f99718ff2424f4ae5ebf0e46f7f0ce03058bf47d9061d1d66e4af53b70ffc size: 1986
antoine@vmdocker:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ docker push localhost:5000/postgres:15-alpine
The push refers to repository [localhost:5000/postgres]
dc78fc630a07: Pushed
451ded513733: Pushed
67795a05c4c0: Pushed
470a5180ab3d: Pushed
3c45e8136f73: Pushed
5e62c73ab95f: Pushed
5df9ba38e251: Pushed
d7d3ab5a9bf2: Pushed
5af4f8f59b76: Pushed
15-alpine: digest: sha256:ae1d1f5e40efe311f7ac36037e6996305c97fc083a1918bd34935eab824507d3 size: 2192
```

- **Commande utilisée** : `git clone https://github.com/pascalito007/ynov-resources.git`

- **Description** : On clone le dépôt Git contenant les ressources Ynov depuis le lien spécifié.

```
antoine@vmdocker:~$ git clone https://github.com/AntoinePvt19/ynov-resources.git
Cloning into 'ynov-resources'...
remote: Enumerating objects: 402, done.
remote: Counting objects: 100% (26/26), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 402 (delta 4), reused 20 (delta 2), pack-reused 376
Receiving objects: 100% (402/402), 11.55 MiB | 7.51 MiB/s, done.
Resolving deltas: 100% (59/59), done.
antoine@vmdocker:~$ ls
composetest  Tp  Tp2  ynov-resources
```

- **Commande utilisée** : `cd /ynov-resources/2023/m2/dataeng/humans-best-friend`

- **Description** : On se déplace dans le répertoire spécifié où se trouve le projet "humans-best-friend".

```
antoine@vmdocker:~$ cd ynov-resources/
antoine@vmdocker:~/ynov-resources$ ls
2023  README.md
antoine@vmdocker:~/ynov-resources$ cd 2023
antoine@vmdocker:~/ynov-resources/2023$ ls
m2
antoine@vmdocker:~/ynov-resources/2023$ cd m2
antoine@vmdocker:~/ynov-resources/2023/m2$ ls
dataeng  webdev
antoine@vmdocker:~/ynov-resources/2023/m2$ cd dataeng/
antoine@vmdocker:~/ynov-resources/2023/m2/dataeng$ ls
humans-best-friend
antoine@vmdocker:~/ynov-resources/2023/m2/dataeng$ cd humans-best-friend/
antoine@vmdocker:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ ls
architecture.png  healthchecks  README.md  result  seed-data  vote  worker
```

- **Commande utilisée** : vi docker-compose.build.yml

- **Description** : On ouvre l'éditeur de texte Vi pour créer ou éditer le fichier de configuration Docker Compose "docker-compose.build.yml". Vous trouverez le fichier dans mon Git. Copiez le dans l'éditeur Vi.

```
antoine@vmdocker:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ vi docker-compose.build.yml
```

- **Commande utilisée** : vi compose.yml

- **Description** : On ouvre l'éditeur de texte Vi pour créer ou éditer le fichier de configuration Docker Compose "compose.yml". Vous trouverez le fichier dans mon Git. Copiez le dans l'éditeur Vi.

```
antoine@vmdocker:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ vi compose.yml
```

- **Commande utilisée** : docker compose -f docker-compose.build.yml up -d

- **Description** : On lance les services spécifiés dans le fichier Docker Compose "docker-compose.build.yml" en mode détaché.

- **Commande utilisée** : docker compose up -d

- **Description** : On lance les services spécifiés dans le fichier Docker Compose "compose.yml" en mode détaché.

```
antoine@vmdocker:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ docker compose -f docker-compose.build.yml up -d
[+] Running 6/6
  ✓ Network humans-best-friend_humansbestfriend-network Created
  ✗ Container humans-best-friend-db-1 Error
  ✗ Container humans-best-friend-redis-1 Error
  ✓ Container humans-best-friend-result-1 Created
  ✓ Container humans-best-friend-worker-1 Created
  ✓ Container humans-best-friend-vote-1 Created
dependency failed to start: container humans-best-friend-db-1 is unhealthy
antoine@vmdocker:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ docker compose up -d
[+] Running 5/5
  ✗ Container humans-best-friend-db-1 Error
  ✗ Container humans-best-friend-redis-1 Error
  ✓ Container humans-best-friend-vote-1 Recreated
  ✓ Container humans-best-friend-worker-1 Recreated
  ✓ Container humans-best-friend-result-1 Recreated
dependency failed to start: container humans-best-friend-db-1 is unhealthy
```

-**Erreur rencontrée** : Ca nous indique l'erreur ci-dessus (le conteneur est en 'mauvaise santé'). Pour résoudre, nous avons essayé d'indiquer les bons accès à la base de données (voir images ci-dessous) car nous avons vu dans le postre.sh cela :

Avant :

```
args=(
    # force postgres to not use the local unix socket (test "external" connectivity)
    --host "$host"
    --username "$user"
    --dbname "$db"
    --quiet --no-align --tuples-only
)
```

Donc nous avons mis cela dans le build :

```
db:
  image: localhost:5000/postgres:15-alpine
  environment:
    POSTGRES_HOST: "db"
    POSTGRES_USER: "postgres"
    POSTGRES_PASSWORD: "postgres"
    POSTGRES_DBNAME: "db_tp"
```

Mais nous n'avons pas réussi à résoudre le problème. Nous avons essayé de démarrer manuellement les conteneurs mais toujours pas. Nous n'avons pas réussi à démarrer le conteneur 'vote' et les conteneurs db et Redis car ils sont 'unhealthy'.

- **Commande utilisée** : docker start humans-best-friend-vote-1

- **Description** : On démarre le conteneur Docker nommé "humans-best-friend-vote-1".

```
antoine@vmdocker:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ docker start humans-best-friend-vote-1
```

- **Commande utilisée** : docker start humans-best-friend-worker-1

- **Description** : On démarre le conteneur Docker nommé "humans-best-friend-worker-1".

```
antoine@vmdocker:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ docker start humans-best-friend-worker-1
```

- **Commande utilisée** : docker start humans-best-friend-result-1

- **Description** : On démarre le conteneur Docker nommé "humans-best-friend-result-1".

```
antoine@vmdocker:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ docker start humans-best-friend-result-1
```

- **Commande utilisée** : docker ps

- **Description** : On affiche la liste des conteneurs Docker en cours d'exécution.

```
antoine@vmdocker:~/ynov-resources/2023/m2/dataeng/humans-best-friend$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6219cb293a63	humans-best-friend-worker	"dotnet Worker.dll"	12 minutes ago	Up 6 seconds		humans-best-friend-worker-1
9a11aa9f56a	humans-best-friend-result	"/usr/bin/tini -- no."	12 minutes ago	Up 11 seconds		humans-best-friend-result-1
f7fbc2b12959	localhost:5000/postgres:15-alpine	"docker-entrypoint.s."	12 minutes ago	Up 12 minutes (unhealthy)	5432/tcp	humans-best-friend-db-1
85a15fa20155	localhost:5000/redis:latest	"docker-entrypoint.s."	12 minutes ago	Up 12 minutes (unhealthy)	6379/tcp	humans-best-friend-redis-1
75b973ec1616	registry:2	"/entrypoint.sh /etc."	7 days ago	Up 5 hours	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp	registry