

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the text '2023 - 2024'.

**2023 - 2024**

# **Projet virtualisation et conteneurisation**

TC3 – TD49

Several thin, curved lines in shades of blue and grey originate from the bottom left and sweep upwards and to the right.

**TUMAR Berfan | KOFFI Assiè**  
ESIEA

Lien vers notre dépôt GitHub : [https://github.com/Berasow/projet-docker-tumar\\_koffi](https://github.com/Berasow/projet-docker-tumar_koffi)

## I. Introduction

Dans le cadre de nos études d'ingénieur en 4<sup>ème</sup> année, nous avons réalisé un projet consistant à déployer une application via des conteneurs Docker sur une machine virtuelle hébergée sur ESXi. L'objectif principal de ce projet est de détailler le processus de configuration, de déploiement et de gestion des conteneurs Docker pour l'installation d'une application spécifique.

Dans ce rapport, nous allons détailler et montrer grâce à des captures d'écrans le déploiement de l'application avec les différentes commandes utilisées.

## II. Processus d'installation

Tout d'abord, nous avons effectué un fork du projet initial pour pouvoir travailler dessus en local depuis nos machines virtuelles. Une fois le projet récupéré, nous avons procédé aux étapes suivantes :

### 1. Création des images

Dans un premier, nous avons créé les images nécessaires au fonctionnement de l'application. Pour ce faire, nous avons créé un fichier Dockerfile pour chacun des services se situant dans le dossier de chaque service. Dans ces fichiers Dockerfile nous avons copié les données fournies dans le sujet du projet sur GitHub.

Ensuite nous avons créé un fichier docker-compose.build.yml dont voici sa composition :

```
version: '3'

services:
  worker:
    build:
      context: ./worker
      dockerfile: Dockerfile

  vote:
    build:
      context: ./vote
      dockerfile: Dockerfile

  seed-data:
    build:
      context: ./seed-data
      dockerfile: Dockerfile
```

```
result:
  build:
    context: ./result
    dockerfile: Dockerfile
```

Ce fichier nous a permis de créer les images en utilisant la commande suivante :

```
root@berfan-vm:/home/berfan/docker-project# docker compose -f docker-compose.build.yml build
[+] Building 1.8s (51/53) FINISHED
=> [seed-data internal] load build definition from Dockerfile
=> => transferring dockerfile: 346B
=> [seed-data internal] load .dockerignore
=> => transferring context: 2B
=> [result internal] load build definition from Dockerfile
=> => transferring dockerfile: 525B
=> [result internal] load .dockerignore
=> => transferring context: 50B
=> [seed-data internal] load metadata for docker.io/library/python:3.9-slim
=> [worker internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.04kB
=> [worker internal] load .dockerignore
=> => transferring context: 2B
=> [vote internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.09kB
=> [vote internal] load .dockerignore
=> => transferring context: 2B
docker:default 0.1s
0.1s
0.0s
0.0s
0.1s
0.1s
0.1s
1.6s
0.1s
0.0s
0.1s
0.0s
0.1s
0.0s
0.1s

root@berfan-vm:/home/berfan/docker-project# docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
docker-project-seed-data latest       b00576a939b6     6 hours ago     129MB
docker-project-vote  latest      e08c7b193df7     6 hours ago     154MB
docker-project-worker latest      7244c5d6835c     45 hours ago    194MB
docker-project-result latest      e110e6e0b751     45 hours ago    224MB
```

Ensuite nous avons créer un registre public (Docker Hub : <https://hub.docker.com/repositories/berasow>) et un registre privé :

```
root@berfan-vm:/home/berfan/docker-project# docker run -d -p 5000:5000 --restart=always --name registry registry:2
Unable to find image 'registry:2' locally
2: Pulling from library/registry
c926b61bad3b: Pull complete
5501dced60f8: Pull complete
e875fe5e6b9c: Pull complete
21f4bf2f86f9: Pull complete
98513cca25bb: Pull complete
Digest: sha256:0a182cb82c93939407967d6d71d6caf11dcef0e5689c6afe2d60518e3b34ab86
Status: Downloaded newer image for registry:2
98ebfee28808f4335e38c7b951f2dd6063db7f9757a9e394c017dfaca07d8ba1
root@berfan-vm:/home/berfan/docker-project# docker tag docker-project-worker:latest localhost:5000/docker-project-worker
root@berfan-vm:/home/berfan/docker-project# docker tag docker-project-vote:latest localhost:5000/docker-project-vote
root@berfan-vm:/home/berfan/docker-project# docker tag docker-project-result:latest localhost:5000/docker-project-result
root@berfan-vm:/home/berfan/docker-project# docker tag docker-project-seed-data:latest localhost:5000/docker-project-seed-data
root@berfan-vm:/home/berfan/docker-project# docker push localhost:5000/docker-project-worker:latest
The push refers to repository [localhost:5000/docker-project-worker]
b04d73d1057c: Pushed
f2c9fd0d2dac: Pushed
5ee537aacfb3: Pushed
7c73e219acdd: Pushed
d47be6633206: Pushed
e6cd39d4cea4: Pushed
latest: digest: sha256:925d735fc0b0d59558190719d5c2959a07d110bfc48b37d9303df4c6b8b585a size: 1577
```

```
root@berfan-vm:/home/berfan/docker-project# curl http://localhost:5000/v2/_catalog
{"repositories":["docker-project-result","docker-project-seed-data","docker-project-vote","docker-project-worker"]}
```

## 2. Déploiement des conteneurs

La seconde étape du projet consistait à déployer les conteneurs à partir des images que nous avons créées précédemment. Nous avons donc créé un autre fichier `compose.yml` pour parvenir à cela, ce fichier comporte toutes les configurations comme le numéro de port, le network, les dépendences, etc. nécessaires au bon déploiement des conteneurs. Voici le contenu de ce fichier :

```
version: '3'

services:
  worker:
    image: localhost:5000/docker-project-worker:latest
    depends_on:
      - redis
      - db
    networks:
      - back-tier

  vote:
    image: localhost:5000/docker-project-vote:latest
    volumes:
      - ./vote:/usr/local/app
    ports:
      - "5002:80"
    networks:
      - front-tier
      - back-tier
    healthcheck:
      test: ["CMD", "curl", "-f", "http://localhost"]
      interval: 15s
      timeout: 5s
      retries: 3
      start_period: 10s

  seed-data:
    image: localhost:5000/docker-project-seed-data:latest
    depends_on:
      - vote
    restart: "no"
    networks:
      - front-tier

  result:
    image: localhost:5000/docker-project-result:latest
    depends_on:
      - db
    volumes:
```

```
- ./result:/usr/local/app
ports:
- "5001:80"
- "127.0.0.1:9229:9229"

db:
  image: postgres:15-alpine
  volumes:
    - "db-data:/var/lib/postgresql/data"
    - "./healthchecks:/healthchecks"
  healthcheck:
    test: /healthchecks/postgres.sh
    interval: "5s"
  environment:
    - POSTGRES_PASSWORD=admin
    - POSTGRES_USER=admin
  networks:
    - back-tier

redis:
  image: redis:latest
  volumes:
    - "./healthchecks:/healthchecks"
  healthcheck:
    test: /healthchecks/redis.sh
    interval: "5s"
  networks:
    - back-tier

volumes:
  db-data:

networks:
  front-tier:
    driver: bridge
  back-tier:
    driver: bridge
```

Ce fichier de configuration va également permettre de créer les volumes et les networks qui sont demandés dans le sujet du projet.

```

root@berfan-vm:/home/berfan/docker-project# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
4ab22f6c48da        bridge              bridge               local
00467b701c4f        docker-project_back-tier  bridge              local
6dcec97d5857        docker-project_default  bridge              local
9257efa16646        docker-project_front-tier bridge              local
6cee4b2fb2a3        host                host                 local
ff033a2f9bda        none                 null                 local
root@berfan-vm:/home/berfan/docker-project# docker volume ls
DRIVER              VOLUME NAME
local               3d66478dcaaad96585c3e91057c57bb6cbbcb6b115004eb06ff6b119c3f2cd36
local               838f1fdb7340da3a7ede01fc3fd16a1a7a829c9ae73e787e3d4342822f5db4e2
local               bb0425612f062e17f07ff849e35aab953f861f8152e65e98ff8f9b0d235cf64f
local               c549376a0f3967d80639403a00f6d9c1ed2c5ec4129489204372432d735a8a23
local               cf5ecfc000b284f62551d805f05826600ca6ced6c08e9043ac35468e1349f78c
local               d6f703b5bbcd927be3c25ac1e6f86d06f1d4f02d5c6035c75e53ae28aff48b42
local               docker-project_db-data

```

Lorsque les fichiers docker-compose.build.yml, compose.yml et les Dockerfile sont tous dans leur répertoire respectif. Nous pouvons démarrer tous les conteneurs avec la commande :

```
docker compose up
```

```

root@berfan-vm:/home/berfan/docker-project# docker compose up
[*] Running 6/0
✓ Container docker-project-vote-1      Running      0.0s
✓ Container docker-project-seed-data-1 Running      0.0s
✓ Container docker-project-db-1        Running      0.0s
✓ Container docker-project-result-1    Running      0.0s
✓ Container docker-project-redis-1     Running      0.0s
✓ Container docker-project-worker-1    Running      0.0s
Attaching to docker-project-db-1, docker-project-redis-1, docker-project-result-1, docker-project-seed-data-1, docker-project-vote-1, docker-project-worker-1
docker-project-vote-1 | [2024-01-06 19:21:44,234] INFO in app: Received vote for b
docker-project-vote-1 | [2024-01-06 19:21:44 +0000] [9] [INFO] Received vote for b
docker-project-vote-1 | 172.20.0.3 - - [06/Jan/2024:19:21:44 +0000] "POST / HTTP/1.0" 200 1688 "-" "ApacheBench/2.3"
docker-project-vote-1 | 172.20.0.3 - - [06/Jan/2024:19:21:44 +0000] "POST / HTTP/1.0" 200 1688 "-" "ApacheBench/2.3"
docker-project-vote-1 | 172.20.0.3 - - [06/Jan/2024:19:21:44 +0000] "POST / HTTP/1.0" 200 1688 "-" "ApacheBench/2.3"
docker-project-vote-1 | 172.20.0.3 - - [06/Jan/2024:19:21:44 +0000] "POST / HTTP/1.0" 200 1688 "-" "ApacheBench/2.3"
docker-project-vote-1 | [2024-01-06 19:21:44,253] INFO in app: Received vote for b
docker-project-vote-1 | [2024-01-06 19:21:44,255] INFO in app: Received vote for b
docker-project-vote-1 | [2024-01-06 19:21:44 +0000] [7] [INFO] Received vote for b
docker-project-vote-1 | [2024-01-06 19:21:44 +0000] [9] [INFO] Received vote for b
docker-project-vote-1 | [2024-01-06 19:21:44,257] INFO in app: Received vote for b
docker-project-vote-1 | [2024-01-06 19:21:44 +0000] [8] [INFO] Received vote for b

```

On constate ici que tous les conteneurs démarrent et que l'application tourne correctement, on voit les votes qui sont reçus pour a ou b.