

TP6 – Docker Compose

Brahim Hamdi

Générer des bitcoins

Le minage c'est le procédé par lequel les transactions Bitcoin sont sécurisées. A cette fin les mineurs effectuent avec leur matériel informatique des calculs mathématiques pour le réseau Bitcoin. Comme récompense pour leurs services, ils collectent les bitcoins nouvellement créés ainsi que les frais des transactions qu'ils confirment.

Les mineurs sont en concurrence et leurs revenus sont proportionnels à la puissance de calcul déployée.

— minage de bitcoins, <https://bitcoin.fr/minage/>

Dockercoins

Notre application exemple est Dockercoins, une application de minage de Dockercoins!

Toutes les ressources de cette application (fictive et complètement inutile) sont disponibles sur le dépôt <https://github.com/brahimhamdi/dockercoins>.

Elle est composée de 4 microservices dans le dossier dockercoins (+une base de données redis):

- rng = un service web(Python) générant en sortie des nombres aléatoires
- hasher = un service web(Ruby) générant en sortie un hash des données qui lui sont envoyées par HTTP POST
- worker = processus(Python) utilisant rng et hasher
- webui = web interface (JS)

Principe :

- worker demande à rng de lui fournir des données aléatoires
- worker injecte ces données dans hasher , hasher génère un hash, récupéré par worker,
- Pour chaque hash commençant par 0, worker génère un DockerCoin
- Le worker stocke les DockerCoins générés dans une base de données Redis,
- La webui affiche le taux d'hashage par seconde.

Vous remarquerez dans les codes précédents que chaque service est appelé par un nom DNS simple (hasher, rng, etc..). Ce sera notre rôle de démarrer ces différents services avec ce nom pour que le code soit valide.

Péparation

1. Installez la dernière version de Docker Compose :

```
ubuntu@formation1:~$ sudo apt install docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker-compose is already the newest version (1.29.2-2).
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
ubuntu@formation1:~$
ubuntu@formation1:~$
```

- Vérifiez ensuite que docker-compose est bien installé, et que vous avez la bonne version (>= 1.8):

```
ubuntu@formation1:~$ docker-compose version
docker-compose version 1.29.2, build unknown
docker-py version: 5.0.3
CPython version: 3.10.7
OpenSSL version: OpenSSL 3.0.5 5 Jul 2022
ubuntu@formation1:~$
```

2. Git clonez l'application Dockercoins sur votre hôte Docker:

```
ubuntu@formation1:~$ git clone https://github.com/brahimhamdi/dockercoins
Cloning into 'dockercoins'...
remote: Enumerating objects: 147, done.
remote: Counting objects: 100% (147/147), done.
remote: Compressing objects: 100% (105/105), done.
remote: Total 147 (delta 45), reused 136 (delta 34), pack-reused 0
Receiving objects: 100% (147/147), 146.78 KiB | 6.38 MiB/s, done.
Resolving deltas: 100% (45/45), done.
ubuntu@formation1:~$
```

- Sous le dossier dockercoins, créez le fichier *docker-compose.yml* et y ajoutez le contenu suivant :

```
version: "3"
services:
  worker:
    image: worker
    build: ./worker

  rng:
    image: rng
    build: ./rng

  hasher:
    image: hasher
```

```
ubuntu@formation1:~$ cd dockercoins/
ubuntu@formation1:~/dockercoins$ tree
.
├── hasher
│   ├── Dockerfile
│   └── hasher.rb
├── rng
│   ├── Dockerfile
│   └── rng.py
├── webui
│   ├── Dockerfile
│   ├── Dockerfile.old
│   └── files
│       ├── d3.min.js
│       ├── index.html
│       ├── jquery-1.11.3.min.js
│       ├── jquery.js
│       ├── rickshaw.min.css
│       └── rickshaw.min.js
└── worker
    ├── Dockerfile
    └── worker.py

5 directories, 15 files
ubuntu@formation1:~/dockercoins$ nano docker-compose.yml
```

build: ./hasher

webui:

image: brahimhamdi/webui

ports:

- 8000:80

redis:

image: redis

```
ubuntu@formation1:~/dockercoins$ cat docker-compose.yml
version: "3"
services:
  worker:
    image: worker
    build: ./worker
  rng:
    image: rng
    build: ./rng
  hasher:
    image: hasher
    build: ./hasher
  webui:
    image: brahimhamdi/webui
    ports:
      - 8000:80
  redis:
    image: redis
ubuntu@formation1:~/dockercoins$
```

Lancement de l'application

3. Lancez l'application en arrière plan. Docker Compose va lire et appliquer le contenu du fichier yaml.

```
ubuntu@formation1:~/dockercoins$ docker-compose up -d
Creating network "dockercoins_default" with the default driver
Building worker
Sending build context to Docker daemon 4.608kB
Step 1/6 : FROM python:alpine
alpine: Pulling from library/python
31e352740f53: Pull complete
cfcc276e4459: Pull complete
78d6a2cf8d41: Pull complete
129922027d23: Pull complete
c15df0cb361b: Pull complete
Digest: sha256:25df32b602118dab046b58f0fe920e3301da0727b5b07430c8bcd4b139627fdc
Status: Downloaded newer image for python:alpine
--> 9a2ccd0e4ef5
Step 2/6 : MAINTAINER brahim.hamdi.consult@gmail.com
--> Running in 756aa3ced4cc
Removing intermediate container 756aa3ced4cc
--> 02357d63cd78
Step 3/6 : RUN pip install redis
--> Running in e59076e5fa83
Collecting redis
  Downloading redis-4.6.0-py3-none-any.whl (241 kB)
    241.1/241.1 kB 14.9 MB/s eta 0:00:00
Installing collected packages: redis
Successfully installed redis-4.6.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the
recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.1.2 -> 23.2
[notice] To update, run: pip install --upgrade pip
Removing intermediate container e59076e5fa83
--> ea19431dd007
Step 4/6 : RUN pip install requests
```

4. Une fois le déploiement de l'application terminé, En utilisant le navigateur, affichez l'interface web de votre application.



- Qu'affiche l'interface de l'application Dockercoins ?

5. Vérifiez que tous les services sont à l'état up.

```
ubuntu@formation1:~/dockercoins$ docker-compose ps
-----
Name                                Command                                State          Ports
-----
dockercoins_hasher_1                ruby hasher.rb                        Up (healthy)   80/tcp
dockercoins_redis_1                 docker-entrypoint.sh redis ...       Up             6379/tcp
dockercoins_rng_1                    python rng.py                          Up             80/tcp
dockercoins_webut_1                  node webui.js                          Up             0.0.0.0:8000->80/tcp, :::8000->80/tcp
dockercoins_worker_1                 python worker.py                       Up

ubuntu@formation1:~/dockercoins$
ubuntu@formation1:~/dockercoins$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
31c610766f14   worker                              "python worker.py"      4 minutes ago Up 4 minutes
dockercoins_worker_1
b17477df119e   hasher                              "ruby hasher.rb"        4 minutes ago Up 4 minutes (healthy)   80/tcp
dockercoins_hasher_1
c1c7a38d0fb0   brahimhandi/webui                  "node webui.js"         4 minutes ago Up 4 minutes             0.0.0.0:8000->80/tcp, :::8000->80/tcp
dockercoins_webut_1
e2d7e6e1a0fc   redis                               "docker-entrypoint.s..." 4 minutes ago Up 4 minutes             6379/tcp
dockercoins_redis_1
23ad3ba4969a   rng                                 "python rng.py"         4 minutes ago Up 4 minutes             80/tcp
dockercoins_rng_1
ubuntu@formation1:~/dockercoins$
```

- Étudiez le résultat, et comparez avec la sortie de la commande docker ps.

6. A quel réseau les conteneurs sont-ils connectés ?

```
ubuntu@formation1:~/dockercoins$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
15c26d892c9b   bridge    bridge    local
fd5a725c0ae0   dockercoins_default bridge    local
2a55a3d75042   host      host      local
9d7cf89781de   none      null      local

ubuntu@formation1:~/dockercoins$ docker network inspect dockercoins_default
[
  {
    "Name": "dockercoins_default",
    "Id": "fd5a725c0ae01c83fe5def6b05e7f67caa1576ebf729422473e8f2bf74b94d0c",
    "Created": "2023-07-16T12:04:02.077580342Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "ConfigOnly": false,
    "Containers": {
      "23ad3ba4969a2e3696f7a9404b5c761c34e2e1442ecf540f52b482d23d63baa8": {
        "Name": "dockercoins_rng_1",
        "EndpointID": "398e7ae7e68b39ecae9c6e62cbb31659387cf017ac6917b80e7a3a420e004ea",
        "MacAddress": "02:42:ac:12:00:02",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
      },
      "31c610766f1436fc4871246f94e808b1c1b0d6e34c3fb32b6f5c241db852f31c": {
        "Name": "dockercoins_worker_1",
        "EndpointID": "55c1ec2a8f983555221b361d2193363dfbf968614f7b7b80a5b28c274f62d606",
        "MacAddress": "02:42:ac:12:00:06",
        "IPv4Address": "172.18.0.6/16",
        "IPv6Address": ""
      },
      "b17477df119e8163107049cbb15eec764c42d7e51506ff4efbd5912db6d4479a": {
        "Name": "dockercoins_hasher_1",
        "EndpointID": "cb005ec762c0752f5a4c60f88e415fc8c50b6826ce19529055c3ebb7d1df0ff01a",
        "MacAddress": "02:42:ac:12:00:04",
        "IPv4Address": "172.18.0.4/16",
        "IPv6Address": ""
      },
      "c1c7a38d0fb00208e83244bb48cc7b9a9b9ed82541a1c66fac9d6875a575e150": {
        "Name": "dockercoins_webut_1",
        "EndpointID": "60b83ca217572ee99b1c614f568c983498a6dd4c2bc7e7b8e330d4d7ecea7f",
        "MacAddress": "02:42:ac:12:00:05",
        "IPv4Address": "172.18.0.5/16",
        "IPv6Address": ""
      },
      "e2d7e6e1a0fc04d1fa865081da141ccce7f0b4bccdd50a7e9db0a60af833b1f1": {
        "Name": "dockercoins_redis_1",
        "EndpointID": "9cab4f302a40c759b3b3e59ca9d18bf8e080850af2348f3b8ed0b7ac75ab93d5",
        "MacAddress": "02:42:ac:12:00:03",
        "IPv4Address": "172.18.0.3/16",
        "IPv6Address": ""
      }
    }
  }
]
```

7. Vérifiez que les nouvelles images sont bien créées.

```
ubuntu@formation1:~/dockercoins$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hasher	latest	aa73eda4cd1e	11 minutes ago	324MB
rng	latest	806c78e39770	11 minutes ago	68.3MB
worker	latest	1e43eba78d6e	12 minutes ago	68.7MB
brahimhamdi/demoapp	latest	d8424b0449af	22 hours ago	102MB
demoapp	2.0	d8424b0449af	22 hours ago	102MB
demoapp	1.0	aa6e1f1a82ac	22 hours ago	113MB
redis	latest	7e89539dd8bd	5 days ago	130MB
mariadb	latest	011343cf3ec3	11 days ago	403MB
nginx	latest	021283c8eb95	11 days ago	187MB
ruby	alpine	49414a6e0413	4 weeks ago	74.2MB
python	alpine	9a2ccd0e4ef5	5 weeks ago	52MB
brahimhamdi/webui	latest	e174d2bf3a61	23 months ago	218MB
openjdk	8-jre-alpine	f7a292bbb70c	4 years ago	84.9MB

```
ubuntu@formation1:~/dockercoins$
```

- Créez un compte sur Dockerhub, taggez les nouvelles images et poussez-les vers cette registry.

```
ubuntu@formation1:~/dockercoins$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
```

Login Succeeded

```
ubuntu@formation1:~/dockercoins$
ubuntu@formation1:~/dockercoins$ docker tag worker brahimhamdi/worker
ubuntu@formation1:~/dockercoins$ docker tag rng brahimhamdi/rng
ubuntu@formation1:~/dockercoins$ docker tag hasher brahimhamdi/hasher
```

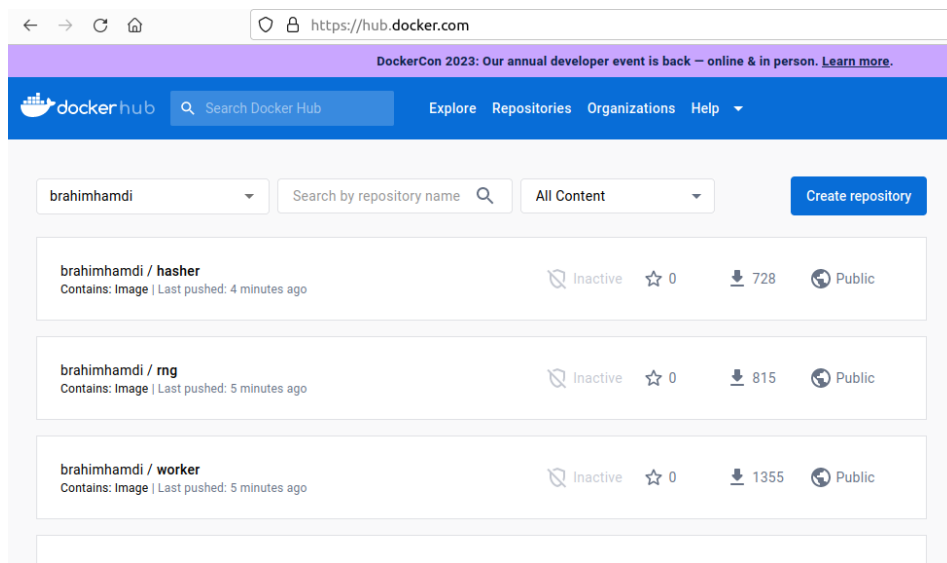
```
ubuntu@formation1:~/dockercoins$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
brahimhamdi/hasher	latest	aa73eda4cd1e	14 minutes ago	324MB
hasher	latest	aa73eda4cd1e	14 minutes ago	324MB
rng	latest	806c78e39770	15 minutes ago	68.3MB
brahimhamdi/rng	latest	806c78e39770	15 minutes ago	68.3MB
brahimhamdi/worker	latest	1e43eba78d6e	15 minutes ago	68.7MB
worker	latest	1e43eba78d6e	15 minutes ago	68.7MB
brahimhamdi/demoapp	latest	d8424b0449af	22 hours ago	102MB
demoapp	2.0	d8424b0449af	22 hours ago	102MB
demoapp	1.0	aa6e1f1a82ac	22 hours ago	113MB
redis	latest	7e89539dd8bd	5 days ago	130MB
mariadb	latest	011343cf3ec3	11 days ago	403MB
nginx	latest	021283c8eb95	11 days ago	187MB
ruby	alpine	49414a6e0413	4 weeks ago	74.2MB
python	alpine	9a2ccd0e4ef5	5 weeks ago	52MB
brahimhamdi/webui	latest	e174d2bf3a61	23 months ago	218MB
openjdk	8-jre-alpine	f7a292bbb70c	4 years ago	84.9MB

```
ubuntu@formation1:~/dockercoins$
```

```
ubuntu@formation1:~/dockercoins$ docker push brahimhamdi/worker
Using default tag: latest
The push refers to repository [docker.io/brahimhamdi/worker]
2841c90a8d86: Pushed
1465d77f32ce: Pushed
1ed71f1687eb: Pushed
776df4582527: Layer already exists
42568fdeffd1: Layer already exists
9ec06db39a30: Layer already exists
cd18acbc1cce: Layer already exists
78a822fe2a2d: Layer already exists
latest: digest: sha256:977b982b12500b3553bce1f994af56749c659b0901d134f2ce49f46476c21a4b size: 1997
ubuntu@formation1:~/dockercoins$ docker push brahimhamdi/rng
Using default tag: latest
The push refers to repository [docker.io/brahimhamdi/rng]
b2f5d0ec8054: Pushed
a6bb9d621208: Pushed
776df4582527: Mounted from brahimhamdi/worker
42568fdeffd1: Mounted from brahimhamdi/worker
9ec06db39a30: Mounted from brahimhamdi/worker
cd18acbc1cce: Mounted from brahimhamdi/worker
78a822fe2a2d: Mounted from brahimhamdi/worker
latest: digest: sha256:df6581d4735b6aaa3818ac9bcf51379258f293e6f1481a3d7ddc0f75921f7a0 size: 1786
ubuntu@formation1:~/dockercoins$
ubuntu@formation1:~/dockercoins$ docker push brahimhamdi/hasher
Using default tag: latest
The push refers to repository [docker.io/brahimhamdi/hasher]
025121d5a8ce: Pushed
986ed0eb803d: Pushed
1398410b199f: Pushed
5b9fd8bad923: Pushed
656256d87668: Mounted from library/ruby
25e2c51db494: Mounted from library/ruby
a71a2f625bbd: Mounted from library/ruby
345bf535117: Mounted from library/ruby
78a822fe2a2d: Mounted from brahimhamdi/rng
latest: digest: sha256:daf0b356be41705c01bb1dac5afb0fd6a2e8dee5acbd1987b9f52f6b101424 size: 2205
ubuntu@formation1:~/dockercoins$
```


- Vérifiez sur Dockerhub que ces nouvelles images ont été bien poussées.



Passage à l'échelle

8. Nous voulons augmenter le nombre de *Dockercoins* générés par l'application, pour cela on vous demande d'augmenter à 2 le nombre de conteneurs (réplicas) exécutant le service worker.

```
ubuntu@formation1:~/dockercoins$ docker-compose scale worker=2
WARNING: The scale command is deprecated. Use the up command with the --scale flag instead.
Creating dockercoins_worker_2 ... done
ubuntu@formation1:~/dockercoins$ docker-compose ps
```

Name	Command	State	Ports
dockercoins_hasher_1	ruby hasher.rb	Up (healthy)	80/tcp
dockercoins_redis_1	docker-entrypoint.sh redis ...	Up	6379/tcp
dockercoins_rng_1	python rng.py	Up	80/tcp
dockercoins_webui_1	node webui.js	Up	0.0.0.0:8000->80/tcp, :::8000->80/tcp
dockercoins_worker_1	python worker.py	Up	
dockercoins_worker_2	python worker.py	Up	

```
ubuntu@formation1:~/dockercoins$
```

- Regardez maintenant la courbe sur l'interface de l'application : Quel est le taux de génération de dockercoins ?



9. Augmentez une autre fois le nombre de worker à 10.

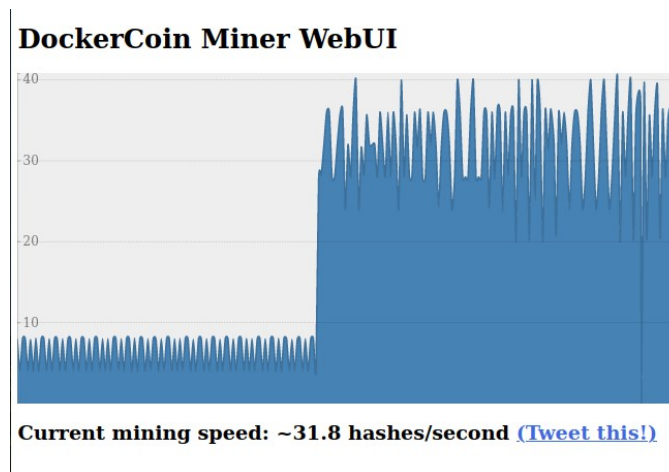
```
ubuntu@formation1:~/dockercoins$ docker-compose scale worker=10
WARNING: The scale command is deprecated. Use the up command with the --scale flag instead.
Creating dockercoins_worker_3 ... done
Creating dockercoins_worker_4 ... done
Creating dockercoins_worker_5 ... done
Creating dockercoins_worker_6 ... done
Creating dockercoins_worker_7 ... done
Creating dockercoins_worker_8 ... done
Creating dockercoins_worker_9 ... done
Creating dockercoins_worker_10 ... done
ubuntu@formation1:~/dockercoins$
ubuntu@formation1:~/dockercoins$ docker-compose ps worker

```

Name	Command	State	Ports
dockercoins_worker_1	python worker.py	Up	
dockercoins_worker_10	python worker.py	Up	
dockercoins_worker_2	python worker.py	Up	
dockercoins_worker_3	python worker.py	Up	
dockercoins_worker_4	python worker.py	Up	
dockercoins_worker_5	python worker.py	Up	
dockercoins_worker_6	python worker.py	Up	
dockercoins_worker_7	python worker.py	Up	
dockercoins_worker_8	python worker.py	Up	
dockercoins_worker_9	python worker.py	Up	

```
ubuntu@formation1:~/dockercoins$
```

- Le taux de génération a-t-il multiplié par 10 ?



Conclusion :

Lorsque les ressources (RAM, CPU, ...) de l'hôte Docker sont limitées, nous ne pouvons pas "scaler" cette application, car en l'état, le service qui fait office de goulot d'étranglement nécessite d'être déployé sur plusieurs machines, et Compose travaille actuellement sur un mono-serveur Docker. Dans le TP suivant, nous allons déployer un swarm Docker (plusieurs hôtes Docker) pour remédier à ça.

Arrêt de l'application

10. Arrêtez et supprimez tous les services de l'application en une seule commande (y compris la dizaine de conteneurs worker lancés).

```

ubuntu@formation1:~/dockercoins$ docker-compose down
Stopping dockercoins_worker_4 ... done
Stopping dockercoins_worker_3 ... done
Stopping dockercoins_worker_5 ... done
Stopping dockercoins_worker_9 ... done
Stopping dockercoins_worker_10 ... done
Stopping dockercoins_worker_8 ... done
Stopping dockercoins_worker_6 ... done
Stopping dockercoins_worker_7 ... done
Stopping dockercoins_worker_2 ... done
Stopping dockercoins_worker_1 ... done
Stopping dockercoins_hasher_1 ... done
Stopping dockercoins_webui_1 ... done
Stopping dockercoins_redis_1 ... done
Stopping dockercoins_rng_1 ... done
Removing dockercoins_worker_4 ... done
Removing dockercoins_worker_3 ... done
Removing dockercoins_worker_5 ... done
Removing dockercoins_worker_9 ... done
Removing dockercoins_worker_10 ... done
Removing dockercoins_worker_8 ... done
Removing dockercoins_worker_6 ... done
Removing dockercoins_worker_7 ... done
Removing dockercoins_worker_2 ... done
Removing dockercoins_worker_1 ... done
Removing dockercoins_hasher_1 ... done
Removing dockercoins_webui_1 ... done
Removing dockercoins_redis_1 ... done
Removing dockercoins_rng_1 ... done
Removing network dockercoins_default
ubuntu@formation1:~/dockercoins$ docker-compose ps
Name      Command      State      Ports
-----
ubuntu@formation1:~/dockercoins$ 

```