

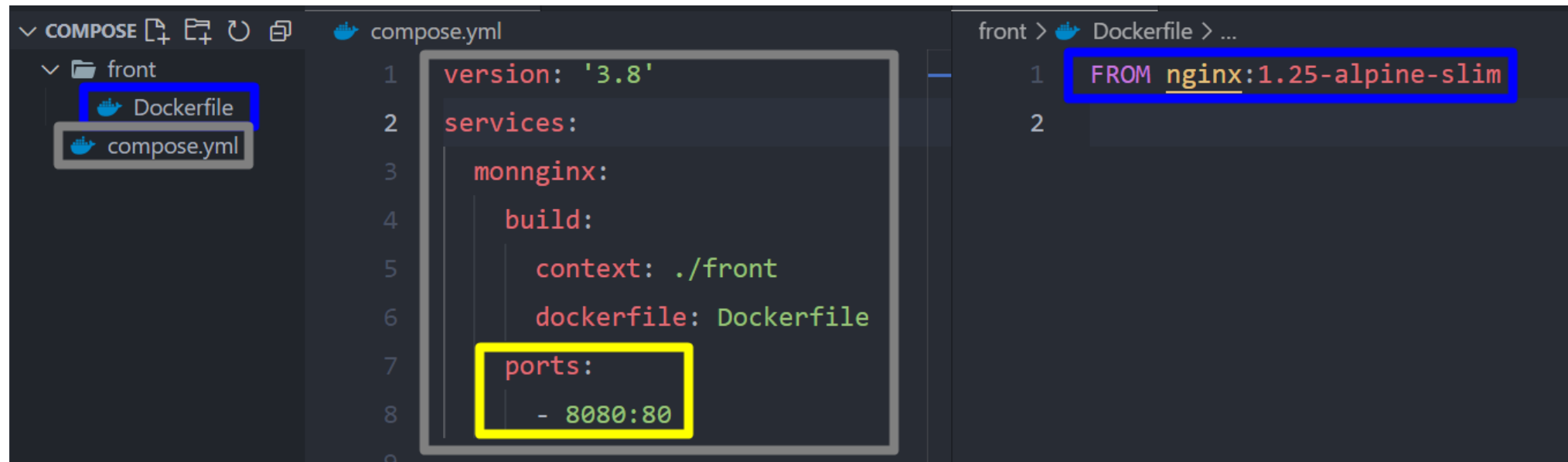
# **Docker-compose**

## **Partie 2**

# Les ports

## Bind ports :

- ✓ Pour rendre nos conteneurs accessibles il faut binder leur ports 80



The screenshot shows the Visual Studio Code interface with two files open. On the left, the 'front' directory contains 'Dockerfile' and 'compose.yml'. The 'compose.yml' file is open in the center editor, showing a configuration for a service named 'monnginx'. The 'ports' section is highlighted with a yellow box, showing the configuration '- 8080:80'. The 'Dockerfile' is open in the right editor, showing the first line 'FROM nginx:1.25-alpine-slim' highlighted with a blue box.

```
1 version: '3.8'
2 services:
3   monnginx:
4     build:
5       context: ./front
6       dockerfile: Dockerfile
7     ports:
8       - 8080:80
```

```
1 FROM nginx:1.25-alpine-slim
2
```

## Bind ports :

- ✓ Pour rendre nos conteneurs accessibles il faut binder leur ports 80

```
$ docker-compose up -d
[+] Building 0.6s (5/5) FINISHED
=> [monnginx internal] load build definition from Dockerfile
=> => transferring dockerfile: 66B
=> [monnginx internal] load .dockerignore
=> => transferring context: 2B
=> [monnginx internal] load metadata for docker.io/library/nginx:1.25-alpine-slim
=> CACHED [monnginx 1/1] FROM docker.io/library/nginx:1.25-alpine-slim@sha256:d50b0d87b5d34da5649e3acfc969011b2012ab9be
=> [monnginx] exporting to image
=> => exporting layers
=> => writing image sha256:0156fe97ad8ea76386bb87daf3cb1091cb22d4b57236b52873b92860b6cfb458
=> => naming to docker.io/library/compose-monnginx
[+] Running 2/2
✓ Network compose_default          Created
✓ Container compose-monnginx-1    Started
```

**Bind ports :**

The image shows a Docker Desktop interface with a sidebar on the left containing 'CONTAINERS', 'IMAGES', 'Volumes', and 'Networks'. The 'compose-monnginx' container is selected, and its details are shown in the main panel. A yellow box highlights the 'Ports' section, which lists '8080 → 80 tcp'. Below this, the 'Volumes' and 'Networks' sections are visible, both showing 'None'. In the foreground, a web browser window is open to 'localhost:8080', displaying a 'Welcome to nginx!' message. The browser's address bar shows 'localhost:8080' and the page content includes the text: 'Welcome to nginx! If you see this page, the nginx web server is successfully installed and working. Further configuration is required.'

**compose-monnginx-1 (a366743bd9bc)**

**Image**  
compose-monnginx (0156fe97ad8e)

**Ports**

- 8080 → 80 tcp

**Volumes**  
None

**Networks**

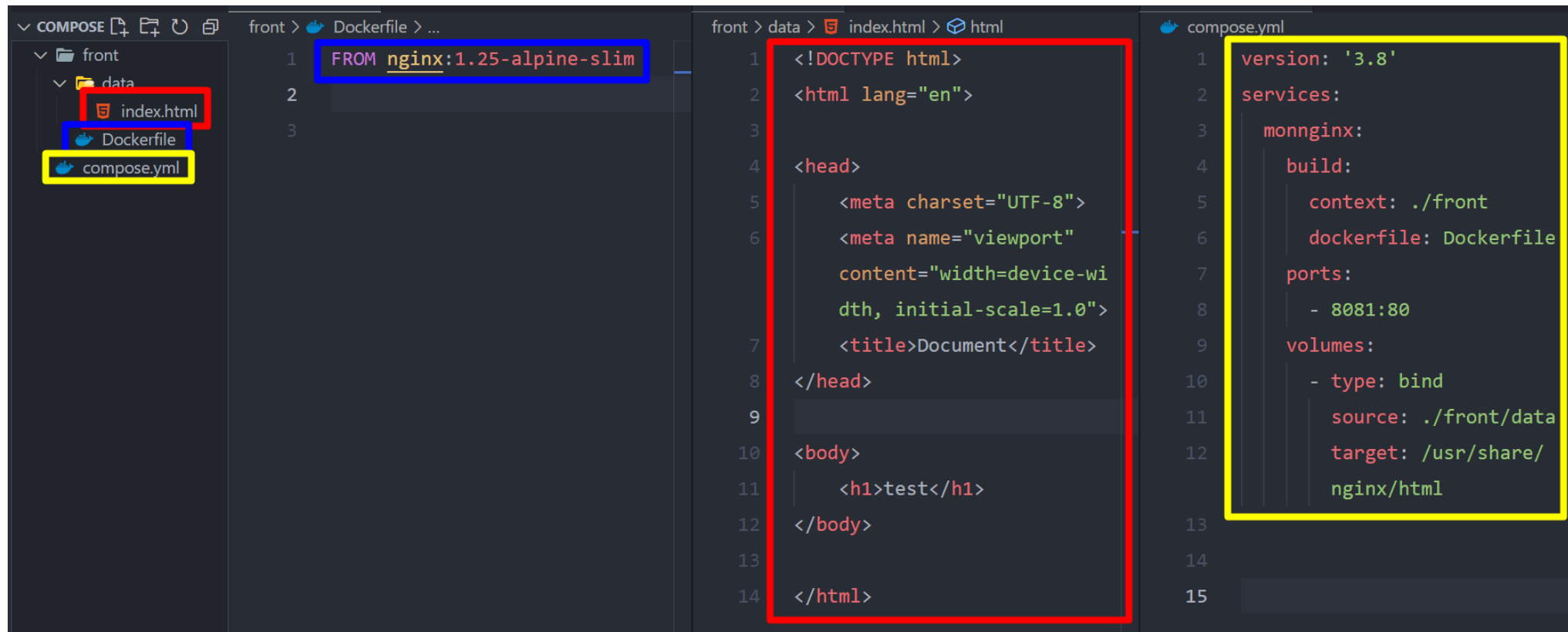
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

# **Les points de montage**

## Bind mounts :

✓ Point de montage local



The screenshot displays a code editor with three panels. The left panel shows a file explorer with the following structure:

- front
  - data
    - index.html (highlighted with a red box)
    - Dockerfile (highlighted with a blue box)
    - compose.yml (highlighted with a yellow box)

The middle panel shows the content of the Dockerfile:

```
1 FROM nginx:1.25-alpine-slim
2
3
```

The right panel shows the content of the compose.yml file:

```
1 version: '3.8'
2 services:
3   monnginx:
4     build:
5       context: ./front
6       dockerfile: Dockerfile
7     ports:
8       - 8081:80
9     volumes:
10      - type: bind
11        source: ./front/data
12        target: /usr/share/
13          nginx/html
14
15
```



## Bind mounts :

```
$ docker-compose up -d
[+] Building 2.0s (6/6) FINISHED                                docker:default
=> [monnginx internal] load .dockerignore                      0.0s
=> => transferring context: 2B                                  0.0s
=> [monnginx internal] load build definition from Dockerfile    0.0s
=> => transferring dockerfile: 97B                              0.0s
=> [monnginx internal] load metadata for docker.io/library/nginx:1.25-alpine-slim 1.1s
=> [monnginx 1/2] FROM docker.io/library/nginx:1.25-alpine-slim@sha256:d50b0d87b5d34da5649e3acfc969011b2012a 0.8s
=> => resolve docker.io/library/nginx:1.25-alpine-slim@sha256:d50b0d87b5d34da5649e3acfc969011b2012ab9be62b70 0.0s
=> => sha256:d50b0d87b5d34da5649e3acfc969011b2012ab9be62b70f8fb97c4d6467a317a 6.77kB / 6.77kB 0.0s
=> => sha256:a3612cfdeb39a0715f0a6100157f3edde0d2eb27226c2b8bd614d52538f5eb78 2.00kB / 2.00kB 0.0s
=> => sha256:c926b61bad3b94ae7351bafd0c184c159ebf0643b085f7ef1d47ecdc7316833c 3.40MB / 3.40MB 0.3s
=> => sha256:fed54a1dc458a7f591fa1c986669998655ad54d260d53691c8ef4841185883d4 1.90MB / 1.90MB 0.3s
=> => sha256:0a9dafac78ccc7a7a179404b0f68ac9b4d98975e695856def6a379714997e54d 7.42kB / 7.42kB 0.0s
=> => sha256:d4735778d47c0be8db66c446904aa2ba47f3e7509c0c9c3985ecb3b96bb7179f 629B / 629B 0.1s
=> => sha256:8695c106552e600555fefc1bc2b299b420c52583bbf537e6c0468bc7821a3f7b 955B / 955B 0.3s
-> [monnginx] exporting to image                                0.0s
=> => exporting layers                                          0.0s
=> => writing image sha256:79fcb20e0401308090fb68a4095ee6ce26aa8718f89cc40f5248f0b35ceabde8 0.0s
=> => naming to docker.io/library/compose-monnginx            0.0s
[+] Running 2/2
✓ Network compose_default Created                                0.0s
✓ Container compose-monnginx-1 Started                          0.0s
```

## Bind mounts :

The image shows a web browser window at the top with the address bar displaying `localhost:8081`. Below it, the Docker Desktop interface is visible. On the left, the 'CONTAINERS' tab is active, showing a tree view of the container's file system. The 'compose-monnginx' container is selected, and its file system is expanded, showing a list of directories including `bin`, `dev`, `docker-e`, `etc`, `home`, `lib`, `media`, `mnt`, and `opt`. On the right, the details for the container `compose-monnginx-1 (70cc2839114b)` are shown. The 'Image' section indicates the container is using `compose-monnginx (79fcb20e0401)`. The 'Ports' section shows a mapping of `8081 → 80 tcp`. The 'Volumes' section, which is highlighted with a yellow box, shows a bind mount configuration: `D:\AFCI\Cours\8 - conteneurisation\cours\Compose\front\data → /usr/share/nginx/html (Bind mount, RW)`. The background of the Docker Desktop interface shows a terminal window with the command `FROM nginx:1.25-alpine-slim`.

test

compose-monnginx-1 (70cc2839114b)

Image

compose-monnginx (79fcb20e0401)

Ports

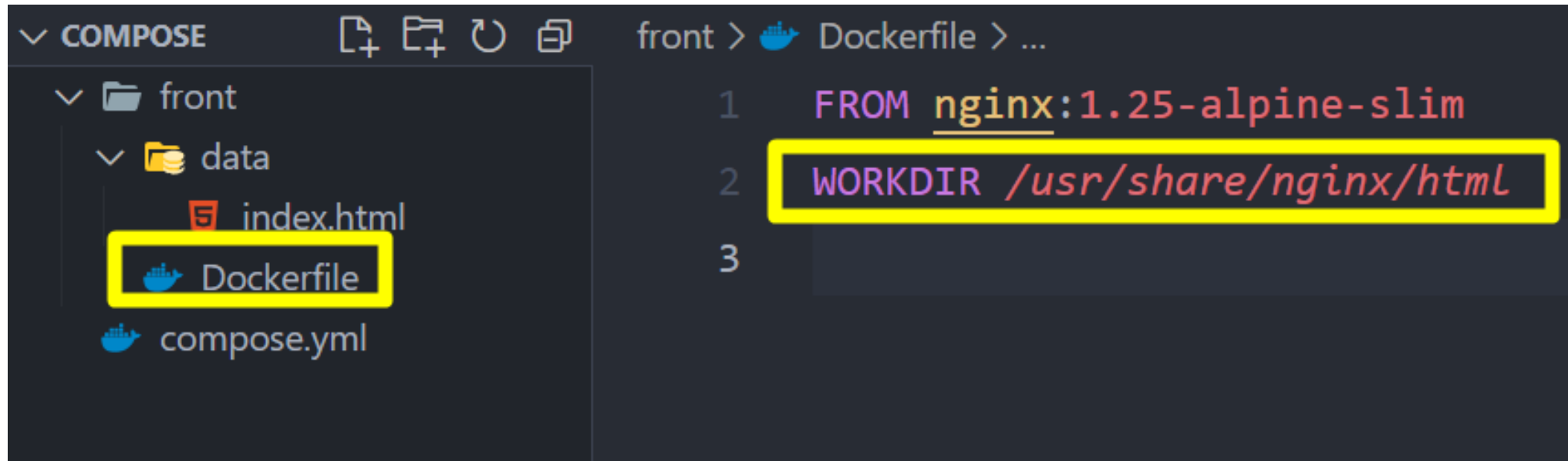
- 8081 → 80 tcp

Volumes

- D:\AFCI\Cours\8 - conteneurisation\cours\Compose\front\data → /usr/share/nginx/html (Bind mount, RW)

## Bind mounts :

- ✓ Test écriture à partir du conteneur



The screenshot shows a code editor with a dark theme. On the left, a file explorer sidebar is open, showing a project structure. The 'front' directory is expanded, revealing a 'data' subdirectory. Inside 'data', there is an 'index.html' file and a 'Dockerfile' file, which is highlighted with a yellow rectangular box. Below 'data', a 'compose.yml' file is also visible. On the right, the code editor displays the content of the 'Dockerfile'. The first line is 'FROM nginx:1.25-alpine-slim'. The second line, 'WORKDIR /usr/share/nginx/html', is highlighted with a yellow rectangular box. The third line is empty. The editor's status bar at the top indicates the current file is 'Dockerfile'.

```
front > Dockerfile > ...  
1 FROM nginx:1.25-alpine-slim  
2 WORKDIR /usr/share/nginx/html  
3
```

## Bind mounts :

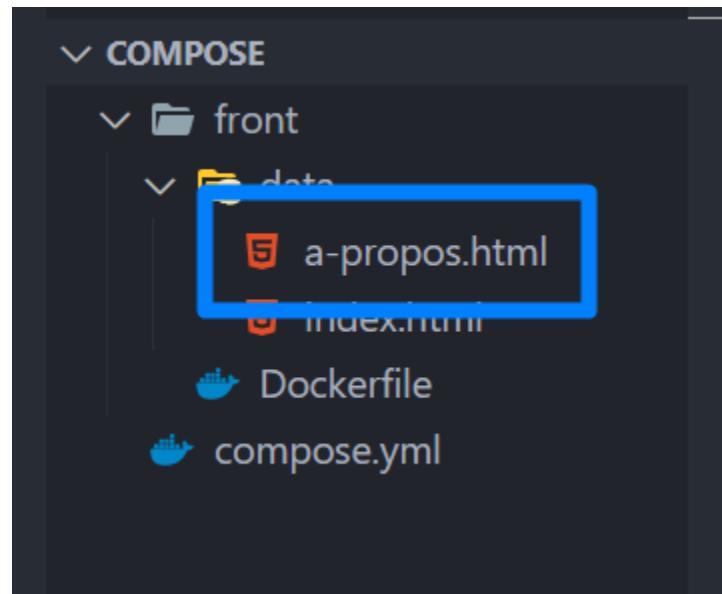
```
Christophe@ChrisBell-MINGW64 /d/APCI/cours/8 - conteneurisation/cours/compose
$ docker-compose up -d
[+] Building 2.4s (7/7) FINISHED
fault
=> [monnginx internal] load .dockerignore
0.0s
=> => transferring context: 2B
0.0s
0.0s
[+] Running 2/2
✓ Network compose_default      Created
0.0s
✓ Container compose-monnginx-1 Started
0.0s
```

## Bind mounts :

```
Christophe@ChrisDe11 MINGW64 /d/AFCI/Cours/8 - conteneurisation/cours/Compose
$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
03788dbe9dde   compose-monngx "/docker-entrypoint..." 4 minutes ago  Up 4 minutes  0.0.0.0:8081->80/tcp      compose-monng

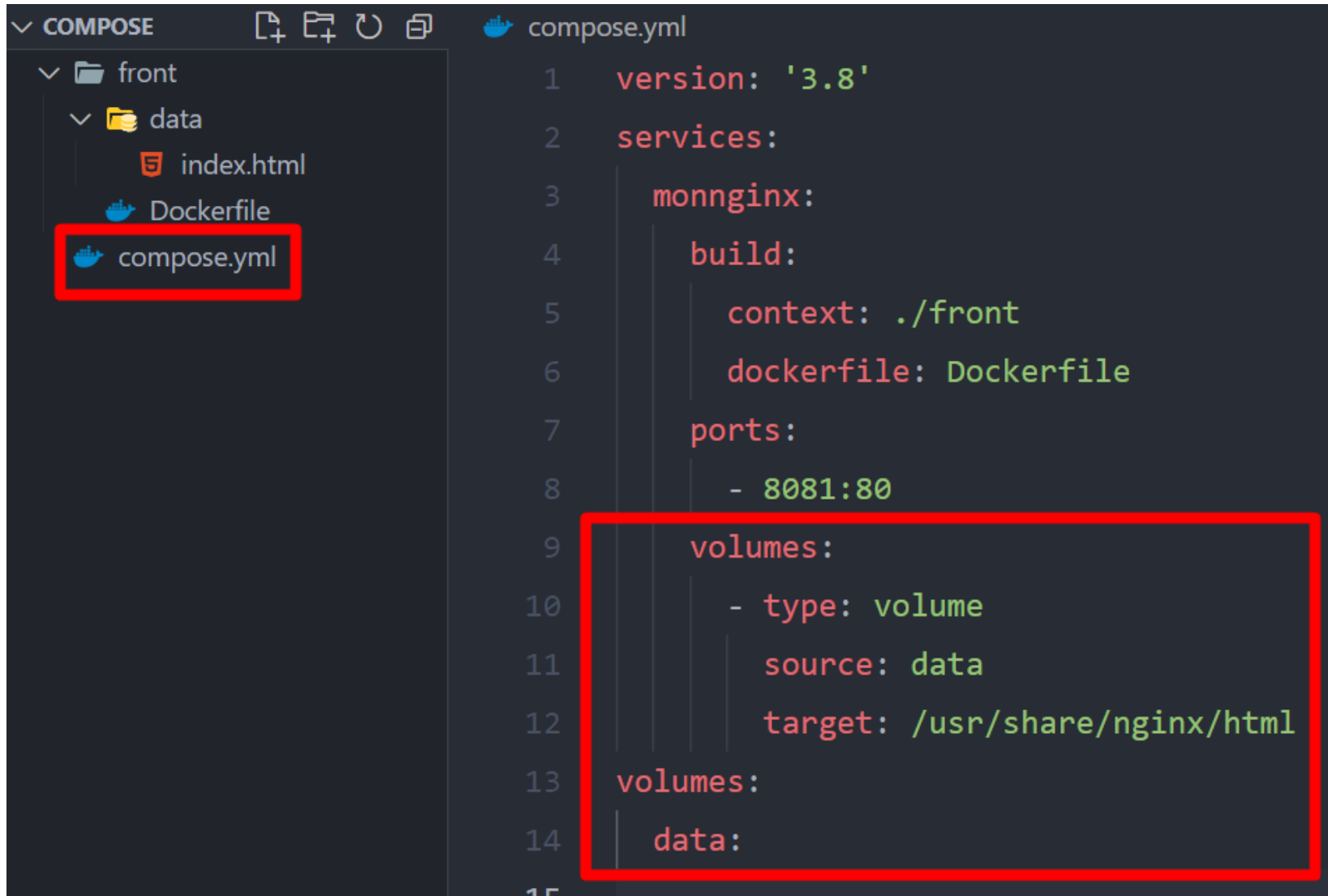
Christophe@ChrisDe11 MINGW64 /d/AFCI/Cours/8 - conteneurisation/cours/Compose
$ docker exec -ti 037 sh
/usr/share/nginx/html # touch a-propos.html
/usr/share/nginx/html #
```

## Résultat :



# Les volumes

## Volumes :



```
1  version: '3.8'
2  services:
3    monnginx:
4      build:
5        context: ./front
6        dockerfile: Dockerfile
7      ports:
8        - 8081:80
9      volumes:
10       - type: volume
11         source: data
12         target: /usr/share/nginx/html
13  volumes:
14    data:
```

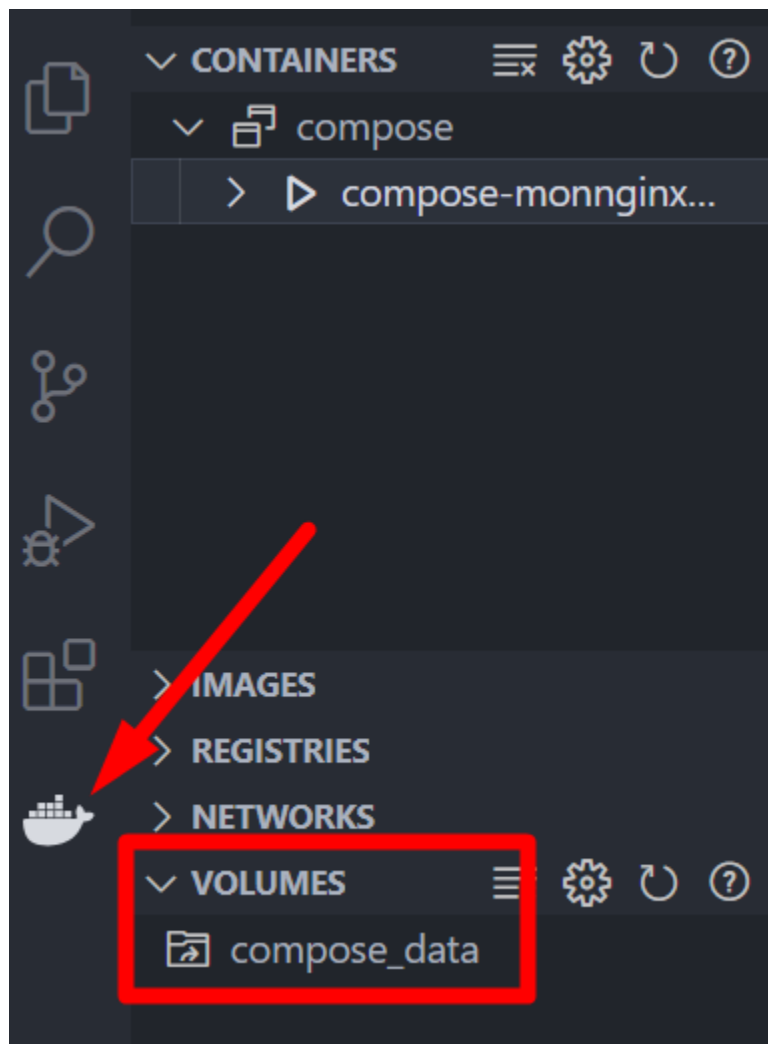
## Volumes :

```

$ docker-compose up -d
[+] Building 1.6s (7/7) FINISHED
=> [monnginx internal] load build definition from Dockerfile                                docker:default 0.0s
=> => transferring dockerfile: 97B                                                         0.0s
=> [monnginx internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                             0.0s
=> [monnginx internal] load metadata for docker.io/library/nginx:1.25-alpine-slim          1.5s
=> [monnginx auth] library/nginx:pull token for registry-1.docker.io                     0.0s
=> [monnginx 1/2] FROM docker.io/library/nginx:1.25-alpine-slim@sha256:d50b0d87b5d34da5649e3acfc969011b2012ab9be62b7 0.0s
=> CACHED [monnginx 2/2] WORKDIR /usr/share/nginx/html                                   0.0s
=> [monnginx] exporting to image                                                           0.0s
=> => exporting layers                                                                     0.0s
=> => writing image sha256:fe9a64130400ba93ab91b2c279501ccbf5bea4a4bb86bee28de0649dc2cbd6d7 0.0s
=> => naming to docker.io/library/compose-monnginx                                       0.0s
[+] Running 3/3
✓ Network compose_default                       Created                                0.0s
✓ Volume "compose_data"                        Created                                0.0s
✓ Container compose-monnginx-1                 Started                                0.1s
```



## Volumes :



## Volumes :

```

$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
7572bf44c8d6   compose-monginx-1  "/docker-entrypoint..." 31 seconds ago Up 30 seconds  0.0.0.0:8081->80/tcp      compose-mon
x-1

```

```

$ docker exec -ti 757 sh
/usr/share/nginx/html # ls
50x.html    index.html
/usr/share/nginx/html # touch contact.html
/usr/share/nginx/html # ls
50x.html    contact.html  index.html
/usr/share/nginx/html # exit

```

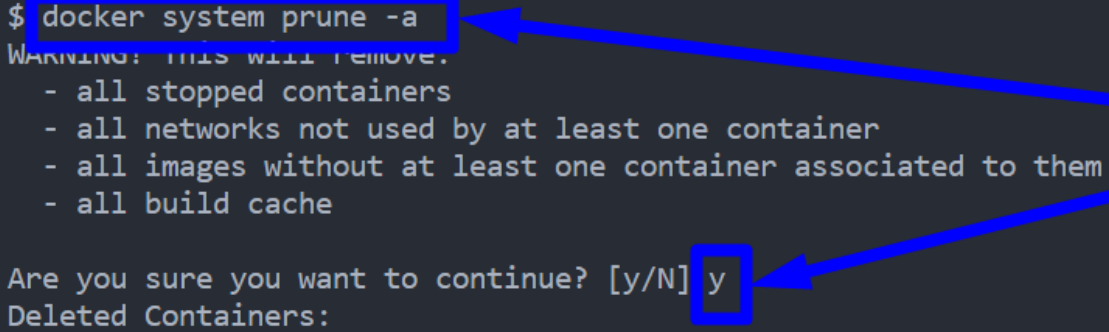
## Volumes :

```
C:\Users\mondel\MinGW64 /d/APC1/Cours/78 - conteneurisation/cours/compose
$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
7572bf44c8d6        compose-monnginx   "/docker-entrypoint..." 10 minutes ago     Up 10 minutes      0.0.0.0:8081->80/tcp  compose
x-1
```

```
C:\Users\mondel\MinGW64 /d/APC1/Cours/78 - conteneurisation/cours/compose
$ docker stop 757
757
```

```
$ docker system prune -a
WARNING: This will remove:
 - all stopped containers
 - all networks not used by at least one container
 - all images without at least one container associated to them
 - all build cache

Are you sure you want to continue? [y/N] y
Deleted Containers:
7572bf44c8d6248e681777cf3721c52cad180e6167cad4a203cd8d6a138faa62
```



## Volumes :

```

$ docker-compose up -d
[+] building 2/2 (///) FINISHED                                docker:default
=> [monnginx internal] load build definition from Dockerfile    0.0s
=> => transferring dockerfile: 97B                               0.0s
=> [monnginx internal] load .dockerignore                       0.0s
=> => transferring context: 2B                                    0.0s
=> [monnginx internal] load metadata for docker.io/library/nginx:1.25-alpine-slim 1.5s
=> [monnginx auth] library/nginx:pull token for registry-1.docker.io 0.0s
=> [monnginx 1/2] FROM docker.io/library/nginx:1.25-alpine-slim@sha256:d50b0d87b5d34da5649e3acfc969011b2012ab9be62b7 1.0s
=> => resolve docker.io/library/nginx:1.25-alpine-slim@sha256:d50b0d87b5d34da5649e3acfc969011b2012ab9be62b70f8fb97c4 0.0s
=> => sha256:d50b0d87b5d34da5649e3acfc969011b2012ab9be62b70f8fb97c4d6467a317a 6.77kB / 6.77kB 0.0s
=> => naming to docker.io/library/compose-monnginx             0.0s
[+] Running 2/2
✓ Network compose_default      Created      0.0s
✓ Container compose-monnginx-1 Started      0.0s
```

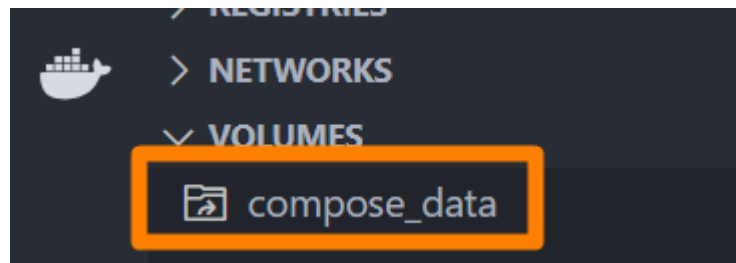
## Volumes :

```
$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
024336a02ef1	compose-monnginx	"/docker-entrypoint...."	40 seconds ago	Up 40 seconds	0.0.0.0:8081->80/tcp	compose-mon

```
$ docker exec -ti 024 sh
/usr/share/nginx/html # ls
50x.html      contact.html  index.html
/usr/share/nginx/html # exit
```

✓ Notre volume



Nous a permis de sauvegarder notre fichier « contact.html »

# **Les variables d'environnement**

## Variables d'environnement :

✓ Link en CLI

```
$ env | grep USER
FPS_BROWSER_USER_PROFILE_STRING=Default
USERDOMAIN_ROAMINGPROFILE=ChrisDell
USERDOMAIN=ChrisDell
USERPROFILE=C:\Users\conta
ALLUSERSPROFILE=C:\ProgramData
USERNAME=Christophe
```

Nous allons passer notre USERNAME au conteneur :


```
s/Compose
$ docker-compose run -e USERNAME monginx
[+] Building 0.0s (0/0)
                                     docker:default
[+] Building 0.0s (0/0)
                                     docker:default
```

## Variables d'environnement :

✓ Résultat :

```
chris@chris-laptop:~/Documents/containers/compose$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  STATUS              PORTS
8fc55e513720   compose-monginx                    "/docker-entrypoin...." 50 seconds ago     Up 49 seconds       80
cp             compose-monginx-run-e267ef710853
```

```
chris@chris-laptop:~/Documents/containers/compose$ docker exec -ti 8fc sh
/usr/share/nginx/html # env
HOSTNAME=8fc55e513720
SHLVL=1
HOME=/root
PKG_RELEASE=1
TERM=xterm
USERNAME=Christophe
NGINX_VERSION=1.25.3
```

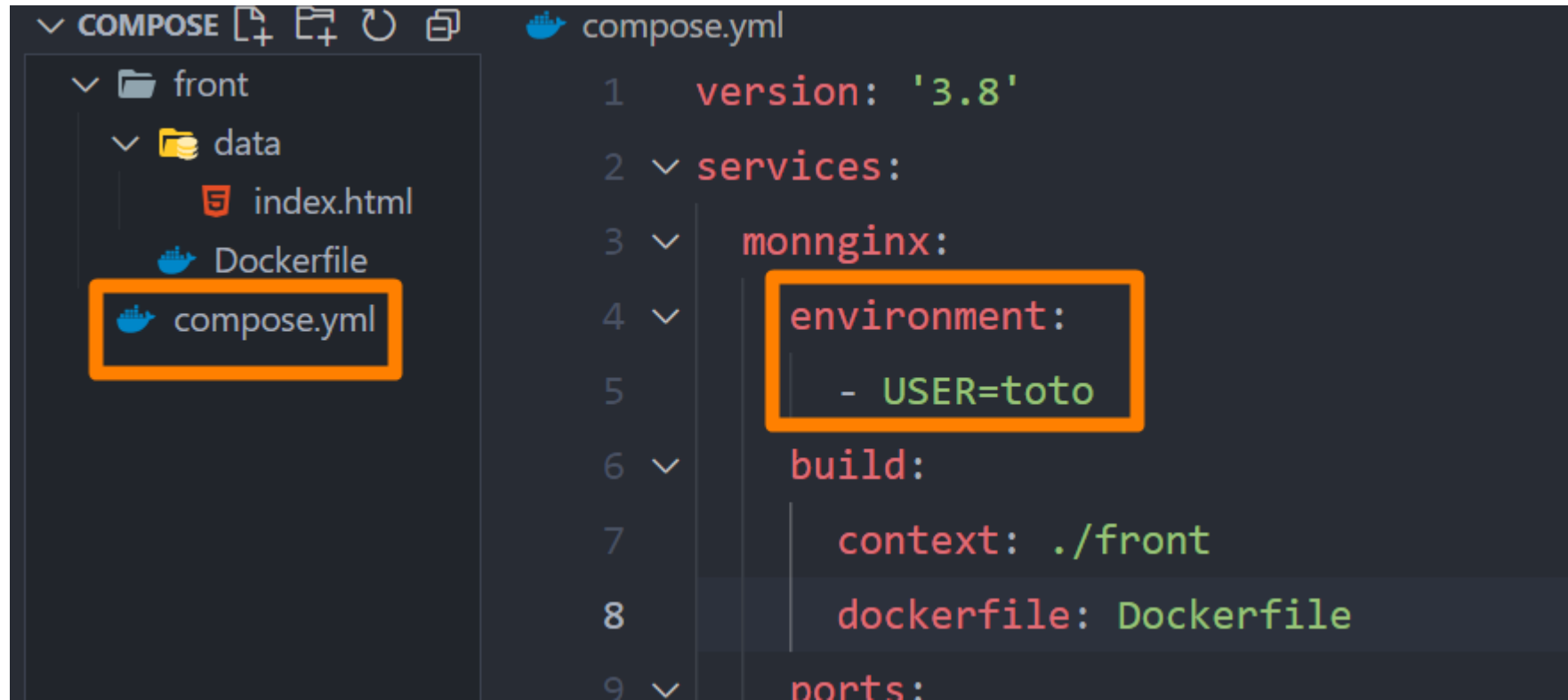


The diagram consists of two orange arrows originating from a single point on the right side of the terminal output. One arrow points to the command `docker exec -ti 8fc sh` in the first terminal block, and the other points to the `USERNAME=Christophe` line in the second terminal block, illustrating that environment variables are shared between the host and the container.



## Variables d'environnement :

✓ Link via « compose.yml » :



The screenshot shows a code editor with a dark theme. On the left, a file explorer shows a directory structure: 'COMPOSE' (expanded), 'front' (expanded), 'data' (expanded), 'index.html', 'Dockerfile', and 'compose.yml' (highlighted with an orange box). On the right, the 'compose.yml' file is open, showing the following content:

```
1  version: '3.8'
2  services:
3    monnginx:
4      environment:
5        - USER=toto
6      build:
7        context: ./front
8        dockerfile: Dockerfile
9      ports:
```

The 'environment' section for the 'monnginx' service is highlighted with an orange box, showing the variable 'USER=toto'.

## Variables d'environnement :

✓ Link via « compose.yml » :

```
Christophe@ChrisDe11 MINGW64 /d/AFCI/Cours/8 - conteneurisation/cours/Compose
$ docker-compose up -d
2023/12/27 10:30:20 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[+] Building 2.8s (7/7) FINISHED                                docker:default
=> [monnginx internal] load .dockerignore                      0.0s
=> => transferring context: 2B                                  0.0s
=> [monnginx internal] load build definition from Dockerfile    0.0s
=> => transferring dockerfile: 97B                              0.0s
=> => naming to docker.io/library/compose-monnginx             0.0s
[+] Running 3/3
✓ Network compose_default      Created                        0.0s
✓ Volume "compose_data"        Created                        0.0s
✓ Container compose-monnginx-1 Started                        0.1s
```

## Variables d'environnement :

✓ Link via « compose.yml » - résultat :

```
$ docker ps -a
```

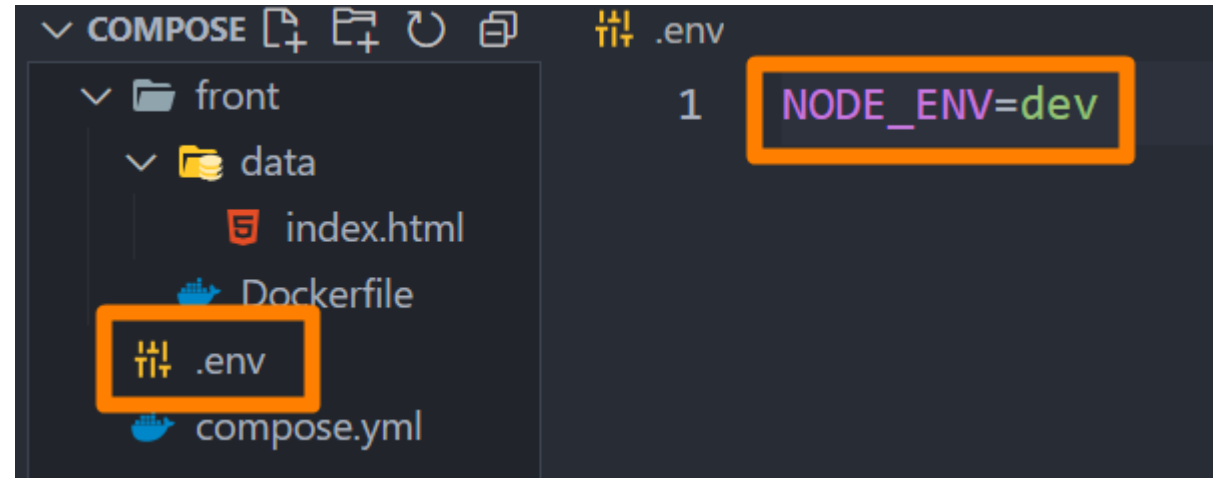
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ccc50ba23821	compose-monnginx	"/docker-entrypoint...."	8 seconds ago	Up 8 seconds	0.0.0.0:8081->80/tcp	compose-monnginx-1

```
christophe@host1: /d/AFCI/Cours/8 - conteneurisation/cours/compose  
$ docker exec -ti ccc sh  
/usr/share/nginx/html # env  
USER=toto  
HOSTNAME=ccc50ba23821  
SHLVL=1  
HOME=/root
```



## Variables d'environnement :

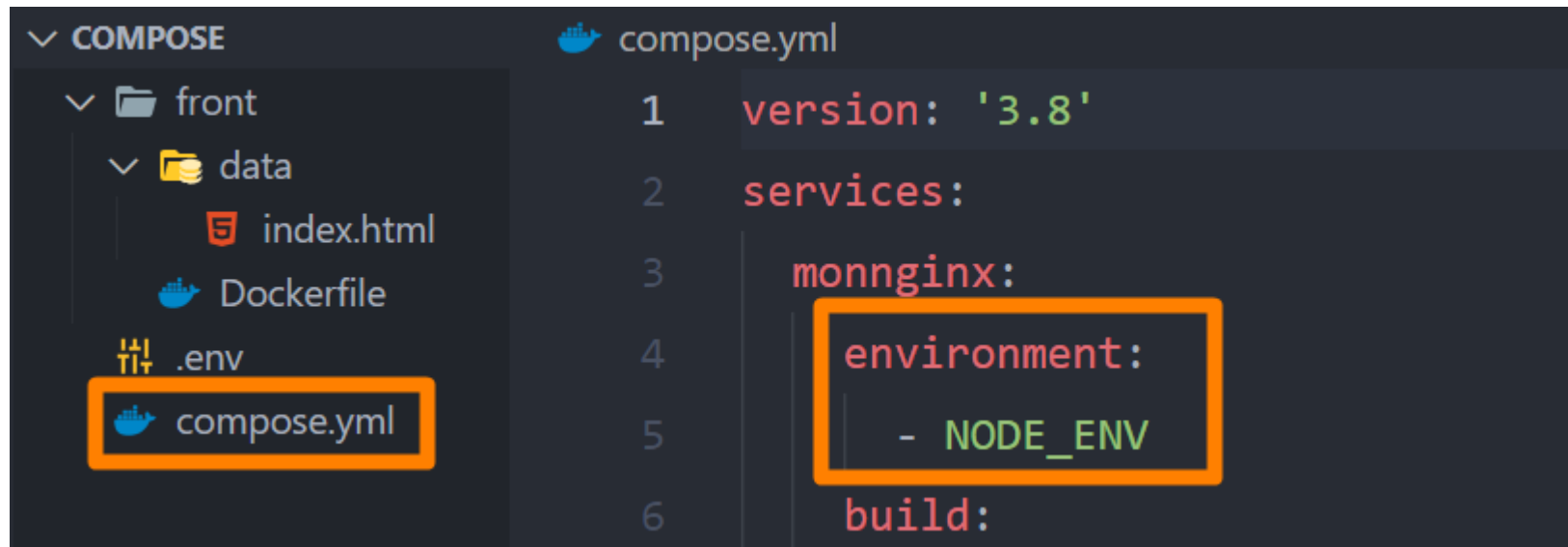
✓ Link via « .env » :



This screenshot shows a file explorer on the left and a code editor on the right. The file explorer is expanded to show the 'front' directory, which contains a 'data' subdirectory with 'index.html', 'Dockerfile', and '.env' files. The '.env' file is highlighted with an orange box. The code editor on the right shows the content of the '.env' file, which is 'NODE\_ENV=dev'. This line is also highlighted with an orange box.

```
COMPOSE [icons] .env
  front
    data
      index.html
      Dockerfile
      .env
      compose.yml
```

```
1 NODE_ENV=dev
```



This screenshot shows a file explorer on the left and a code editor on the right. The file explorer is expanded to show the 'front' directory, which contains a 'data' subdirectory with 'index.html', 'Dockerfile', and '.env' files. The 'compose.yml' file is highlighted with an orange box. The code editor on the right shows the content of the 'compose.yml' file. The 'environment' section under the 'monnginx' service is highlighted with an orange box, showing '- NODE\_ENV'.

```
COMPOSE compose.yml
  front
    data
      index.html
      Dockerfile
      .env
      compose.yml
```

```
1 version: '3.8'
2 services:
3   monnginx:
4     environment:
5       - NODE_ENV
6   build:
```

## Variables d'environnement :

✓ Link via « .env » :

```
Christophe@CHRISDELL MINGW64 /d/AFC1/Cours/8 - conteneurisation/cours/compose
$ docker-compose up -d
[+] Building 0.0s (0/0)
[+] Running 1/1
✓ Container compose-monginx-1 Started
```

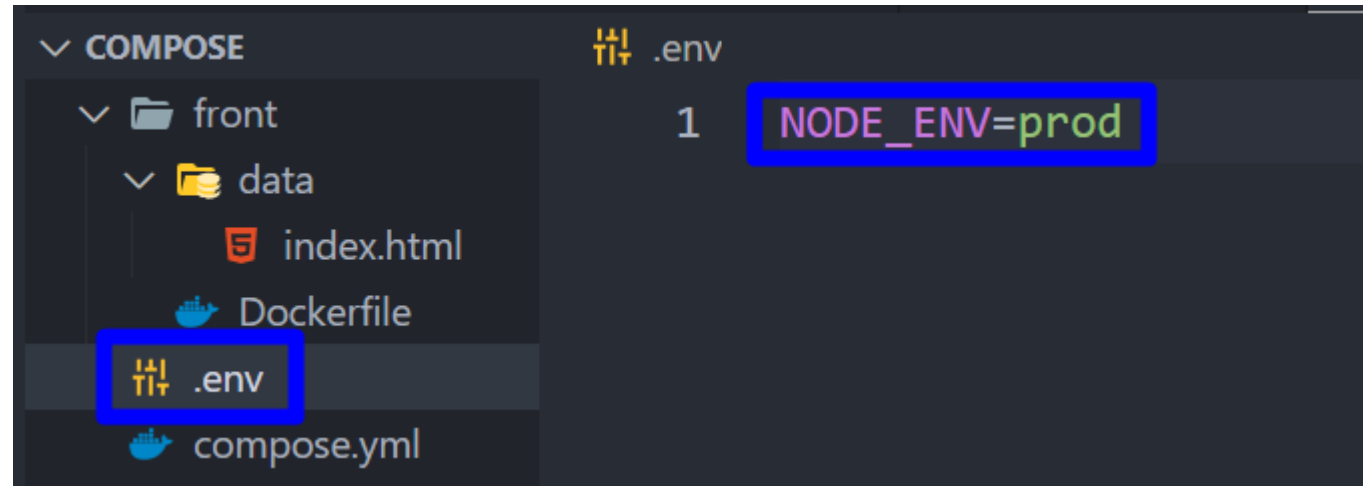
```
Christophe@CHRISDELL MINGW64 /d/AFC1/Cours/8 - conteneurisation/cours/compose
$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
47bfb8918ded	compose-monginx	"/docker-entrypoint..."	7 seconds ago	Up 5 seconds	0.0.0.0:8081->80/tcp	compose-mong

```
Christophe@CHRISDELL MINGW64 /d/AFC1/Cours/8 - conteneurisation/cours/Compose
$ docker exec -ti 47b sh
/usr/share/nginx/html # env
HOSTNAME=47bfb8918ded
SHLVL=1
HOME=/root
PKG_RELEASE=1
TERM=xterm
NGINX_VERSION=1.25.3
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/usr/share/nginx/html
NODE_ENV=dev
/usr/share/nginx/html #
```

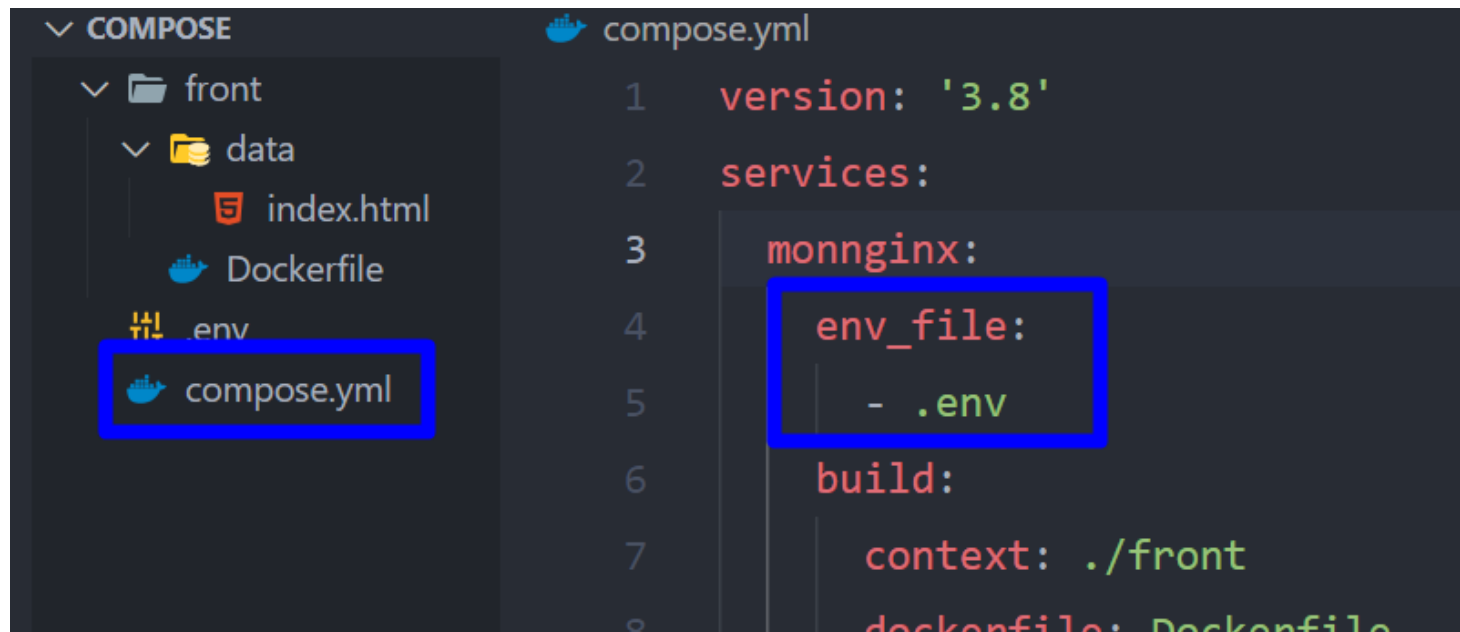
## Variables d'environnement :

✓ Link via « .env » :



This screenshot shows the VS Code interface with the file explorer on the left and the editor on the right. The file explorer shows a project structure with a 'COMPOSE' folder containing 'front', 'data', 'Dockerfile', '.env', and 'compose.yml'. The '.env' file is selected and highlighted with a blue box. The editor displays the content of the '.env' file, which is a single line: `NODE_ENV=prod`. This line is also highlighted with a blue box.

```
COMPOSE
  front
    data
      index.html
      Dockerfile
      .env
      compose.yml
.env
1 NODE_ENV=prod
```



This screenshot shows the VS Code interface with the file explorer on the left and the editor on the right. The file explorer shows the same project structure as the previous screenshot, but now the 'compose.yml' file is selected and highlighted with a blue box. The editor displays the content of the 'compose.yml' file, which is a YAML configuration for a Docker service. The configuration includes a 'version' of '3.8', a 'services' section with a 'monnginx' service, and a 'build' section. The 'env\_file' property of the 'monnginx' service is set to '.env', and this is highlighted with a blue box. The 'build' section is also highlighted with a blue box.

```
COMPOSE
  front
    data
      index.html
      Dockerfile
      .env
      compose.yml
compose.yml
1 version: '3.8'
2 services:
3   monnginx:
4     env_file:
5     - .env
6   build:
7     context: ./front
8     dockerfile: Dockerfile
```

## Variables d'environnement :

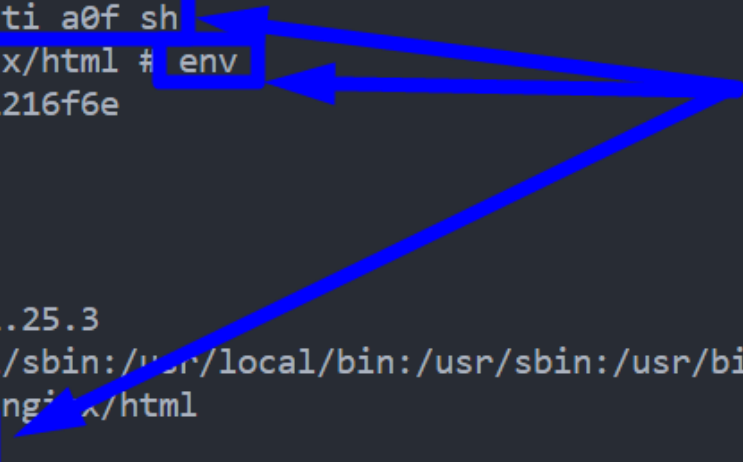
✓ Link via « .env » :

```
CHRISTOPHER@CHRISTOPHER-MINGW64 /d/AFC1/Cours/8 - conteneurisation/cours/Compose
$ docker-compose up -d
[+] Building 0.0s (0/0)
[+] Running 1/1
✓ Container compose-monginx-1 Started
```

```
CHRISTOPHER@CHRISTOPHER-MINGW64 /d/AFC1/Cours/8 - conteneurisation/cours/Compose
$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a0f541216f6e	compose-monginx	"/docker-entrypoint..."	5 seconds ago	Up 3 seconds	0.0.0.0:8081->80/tcp	compose

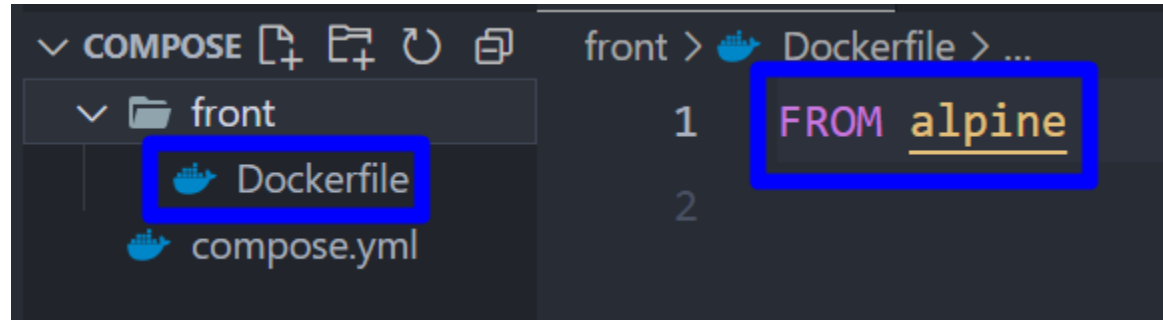
```
CHRISTOPHER@CHRISTOPHER-MINGW64 /d/AFC1/Cours/8 - conteneurisation/cours/Compose
$ docker exec -ti a0f sh
/usr/share/nginx/html # env
HOSTNAME=a0f541216f6e
SHLVL=1
HOME=/root
PKG_RELEASE=1
TERM=xterm
NGINX_VERSION=1.25.3
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/usr/share/nginx/html
NODE_ENV=prod
/usr/share/nginx/html #
```



# **Gestion des réseaux**



Le réseau :



This screenshot shows a code editor with a sidebar on the left displaying a file tree. The 'front' directory is expanded, showing 'Dockerfile' and 'compose.yml'. The 'Dockerfile' file is selected and highlighted with a blue box. The main editor area shows the content of the Dockerfile, with the first line 'FROM alpine' highlighted by a blue box. The breadcrumb at the top indicates the path 'front > Dockerfile > ...'.

```
front > Dockerfile > ...  
1 FROM alpine  
2
```



This screenshot shows a code editor with a sidebar on the left displaying a file tree. The 'front' directory is expanded, showing 'Dockerfile' and 'compose.yml'. The 'compose.yml' file is selected and highlighted with a blue box. The main editor area shows the content of the compose.yml file, with the entire content highlighted by a blue box. The breadcrumb at the top indicates the path 'compose.yml'.

```
compose.yml  
1 version: '3.8'  
2 services:  
3   a:  
4     build:  
5       context: ./front  
6       dockerfile: Dockerfile  
7     command: ["ping", "google.fr"]  
8   b:  
9     image: alpine  
10    command: ["ping", "google.fr"]
```

## Le réseau :

```
Christophe@ChrisDell MINGW64 /d/AFCI/Cours/8 - conteneurisation/cours/Compose
$ docker-compose up
[+] Running 2/2
 ✓ b 1 layers [::]      0B/0B      Pulled
 ✓ 661ff4d9561e Pull complete
2023/12/27 12:19:10 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[+] Building 0.1s (5/5) FINISHED
=> [a internal] load build definition from Dockerfile
=> => transferring dockerfile: 50B
=> [a internal] load .dockerignore
=> => transferring context: 2B
.....
=> => naming to docker.io/library/compose-a
[+] Running 3/0
 ✓ Network compose_default Created
 ✓ Container compose-b-1 Created
 ✓ Container compose-a-1 Created
Attaching to compose-a-1, compose-b-1
compose-a-1 | PING google.fr (216.58.214.163): 56 data bytes
compose-a-1 | 64 bytes from 216.58.214.163: seq=0 ttl=63 time=13.135 ms
compose-b-1 | PING google.fr (216.58.214.163): 56 data bytes
compose-b-1 | 64 bytes from 216.58.214.163: seq=0 ttl=63 time=9.599 ms
```

Nos PING

## Le réseau :

```
compose-b-1 | 64 bytes from 216.58.214.163: seq=14 ttl=63 time=15.121 ms
Gracefully stopping... (press Ctrl+C again to force)
Aborting on container exit...
[+] Stopping 2/2
✓ Container compose-a-1 Stopped
✓ Container compose-b-1 Stopped
canceled
```

- Ctrl+C pour quitter les processus

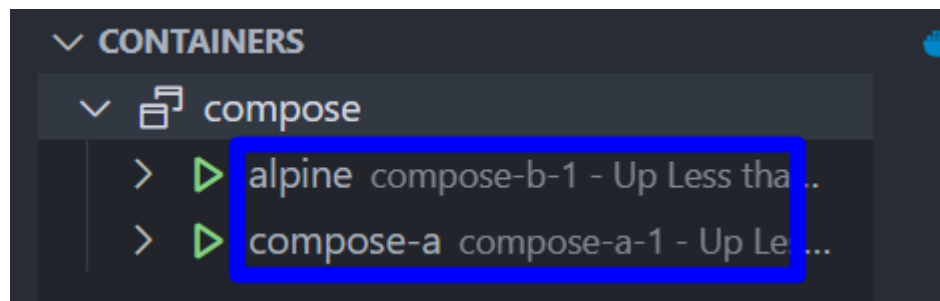
```
CHETST@BNAME:CHETSDAT1 MINGW64 /d/AFC1/Cours/8 - conteneurisation/cours/compose
$ docker-compose down
[+] Running 3/3
✓ Container compose-b-1 Removed
✓ Container compose-a-1 Removed
✓ Network compose_default Removed
```

- Nettoyage

## Le réseau :

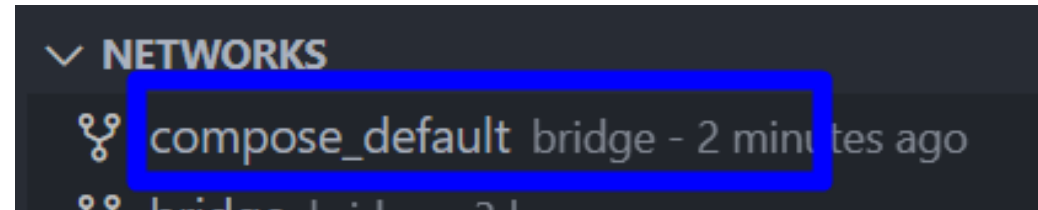
- ✓ Essai en mode « --detach »

```
$ docker-compose up -d
[+] Building 0.0s (0/0)
docker:default
[+] Running 3/3
✓ Network compose_default Created
  0.0s
✓ Container compose-a-1 Started
  0.0s
✓ Container compose-b-1 Started
  0.0s
```



## Le réseau :

- ✓ Test réseau : ping entre les 2 conteneurs



```
$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a3df8ee1001d	alpine	"ping google.fr"	5 minutes ago	Up 5 minutes		compose-b-1
0efcb235a02a	compose-a	"ping google.fr"	5 minutes ago	Up 5 minutes		compose-a-1

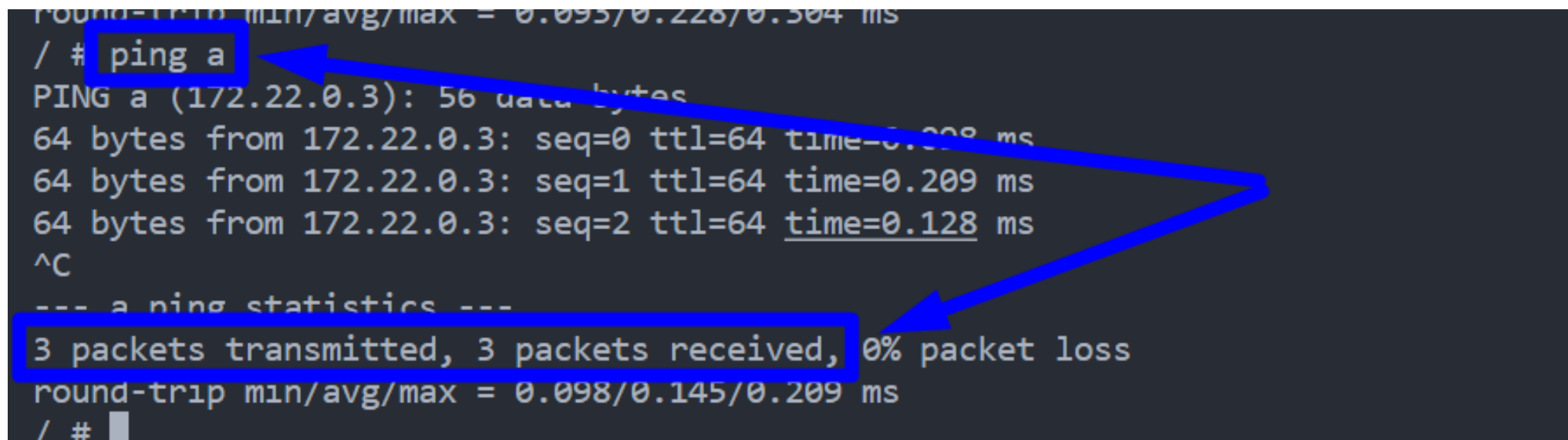
```
$ docker exec -it compose-b-1 sh
/ # ping compose-a-1
PING compose-a-1 (172.22.0.3): 56 data bytes
64 bytes from 172.22.0.3: seq=0 ttl=64 time=0.093 ms
64 bytes from 172.22.0.3: seq=1 ttl=64 time=0.304 ms
64 bytes from 172.22.0.3: seq=2 ttl=64 time=0.287 ms
^C
--- compose-a-1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.093/0.228/0.304 ms
/ #
```

A diagram with three blue arrows pointing from the ping statistics section of the terminal output to the three ping lines above it, indicating that the statistics apply to the entire sequence of pings.

## Le réseau :

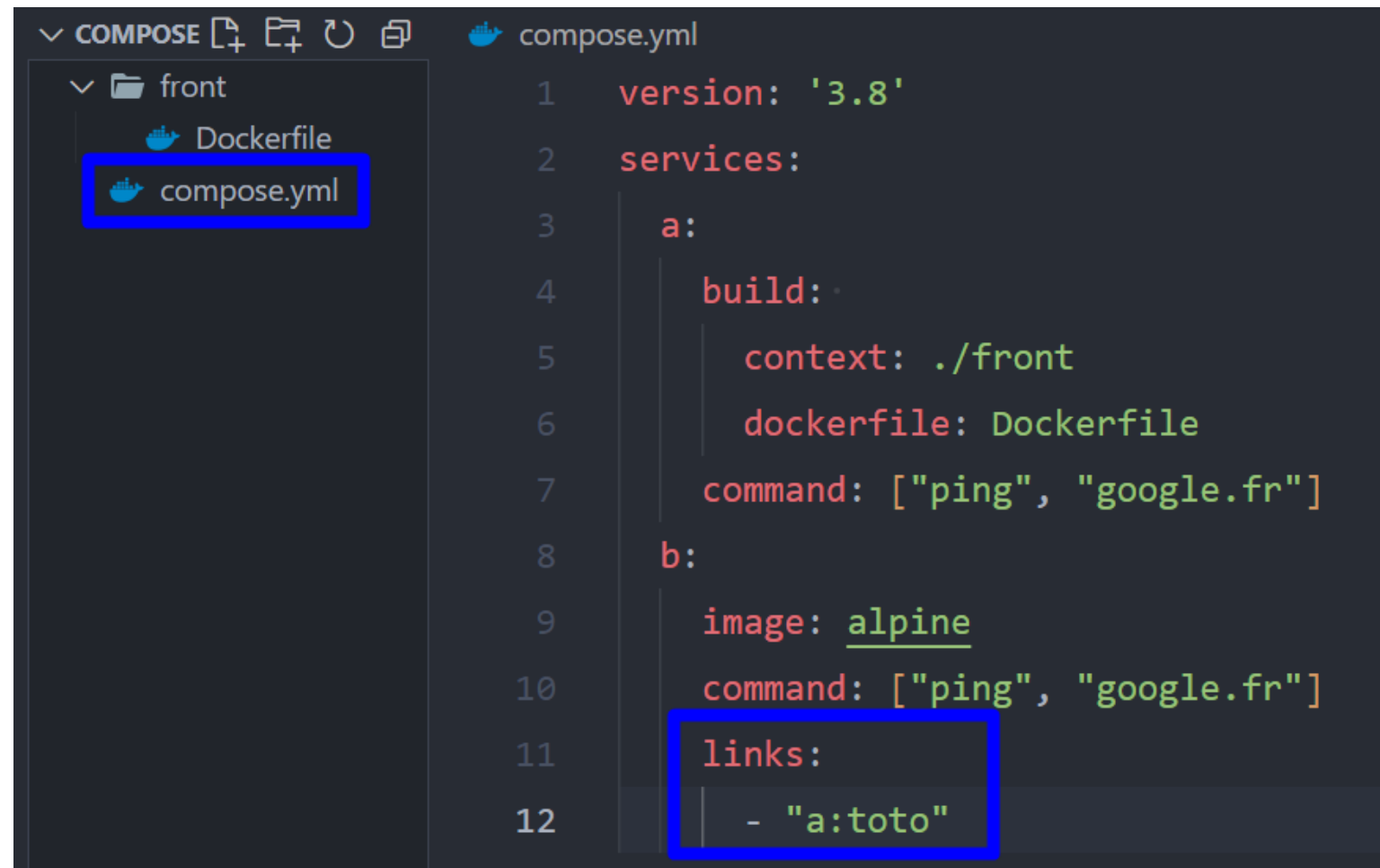
- ✓ On peut aussi utiliser le nom du service

```
Round-trip min/avg/max = 0.095/0.228/0.304 ms
/ # ping a
PING a (172.22.0.3): 56 data bytes
64 bytes from 172.22.0.3: seq=0 ttl=64 time=0.208 ms
64 bytes from 172.22.0.3: seq=1 ttl=64 time=0.209 ms
64 bytes from 172.22.0.3: seq=2 ttl=64 time=0.128 ms
^C
--- a ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.098/0.145/0.209 ms
/ #
```



## Le réseau :

- ✓ Les links (alias)



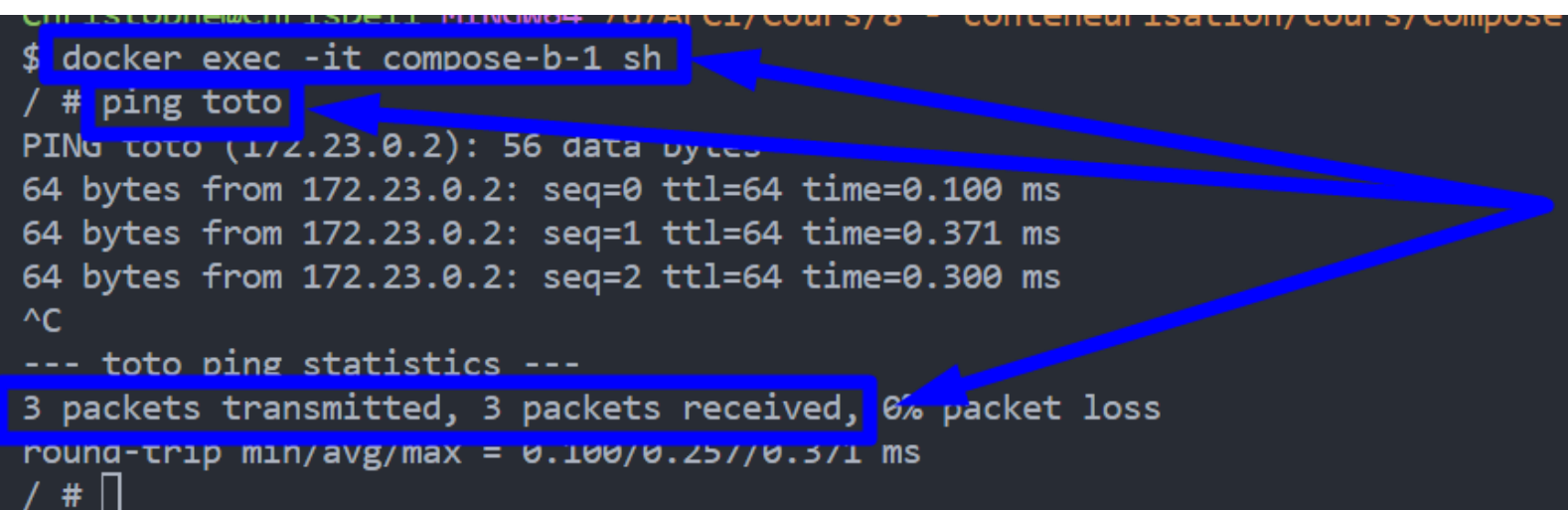
```
1  version: '3.8'
2  services:
3      a:
4          build:
5              context: ./front
6              dockerfile: Dockerfile
7          command: ["ping", "google.fr"]
8      b:
9          image: alpine
10         command: ["ping", "google.fr"]
11         links:
12             - "a:toto"
```

## Le réseau :

### ✓ Les links (alias)

```
Christophe@Christofell-MINGW64 /d/AFCI/Cours/8 - conteneurisation/cours/Compose
$ docker-compose up -d
[+] Building 0.0s (0/0)
[+] Running 3/3
✓ Network compose_default Created 0.0s
✓ Container compose-a-1 Started 0.0s
✓ Container compose-b-1 Started 0.0s
```

```
Christophe@Christofell-MINGW64 /d/AFCI/Cours/8 - conteneurisation/cours/Compose
$ docker exec -it compose-b-1 sh
/ # ping toto
PING toto (172.23.0.2): 56 data bytes
64 bytes from 172.23.0.2: seq=0 ttl=64 time=0.100 ms
64 bytes from 172.23.0.2: seq=1 ttl=64 time=0.371 ms
64 bytes from 172.23.0.2: seq=2 ttl=64 time=0.300 ms
^C
--- toto ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.100/0.257/0.371 ms
/ #
```



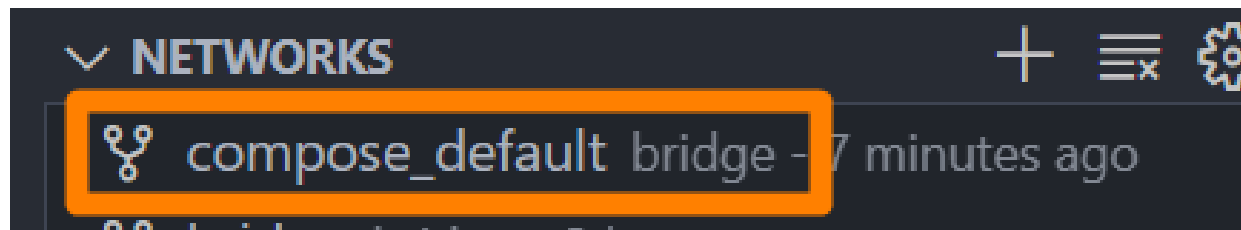


## Le réseau :

- ✓ Pour l'instant docker créer un réseau par défaut pour nous

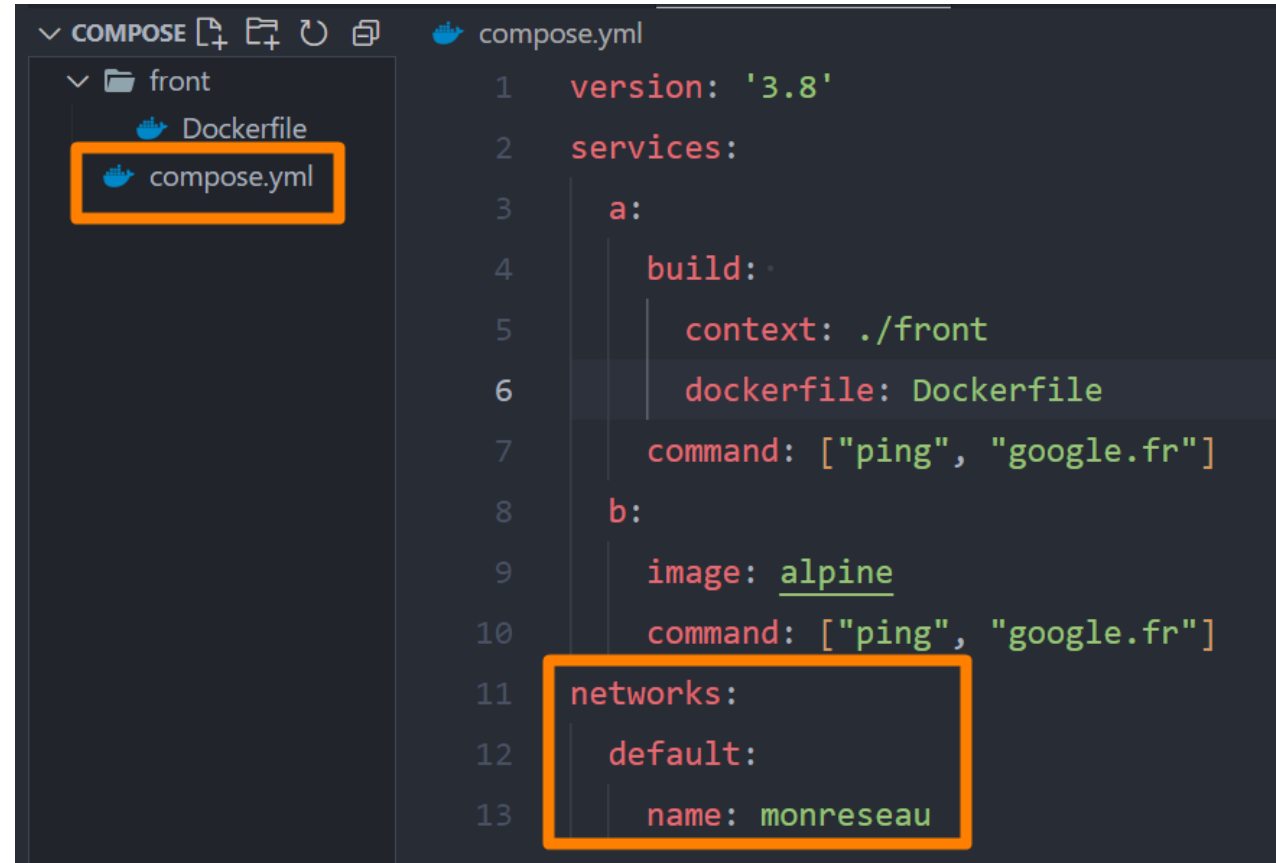
```
Christophe@ChristopheLL: MINGW64 /d/AFCI/Cours/8 - conteneurisation/cours/Compose
$ docker-compose up -d
[+] Building 0.0s (0/0)
[+] Running 3/3
✓ Network compose_default Created
✓ Container compose-a-1 Started
✓ Container compose-b-1 Started
```

Créé par Docker



## Le réseau :

- ✓ Créons le notre pour encapsuler nos conteneurs



```
COMPOSE [Icons] compose.yml
  front
    Dockerfile
    compose.yml
1  version: '3.8'
2  services:
3    a:
4      build:
5        context: ./front
6        dockerfile: Dockerfile
7      command: ["ping", "google.fr"]
8    b:
9      image: alpine
10     command: ["ping", "google.fr"]
11   networks:
12     default:
13       name: monreseau
```

## Le réseau :

### ✓ Nettoyage :

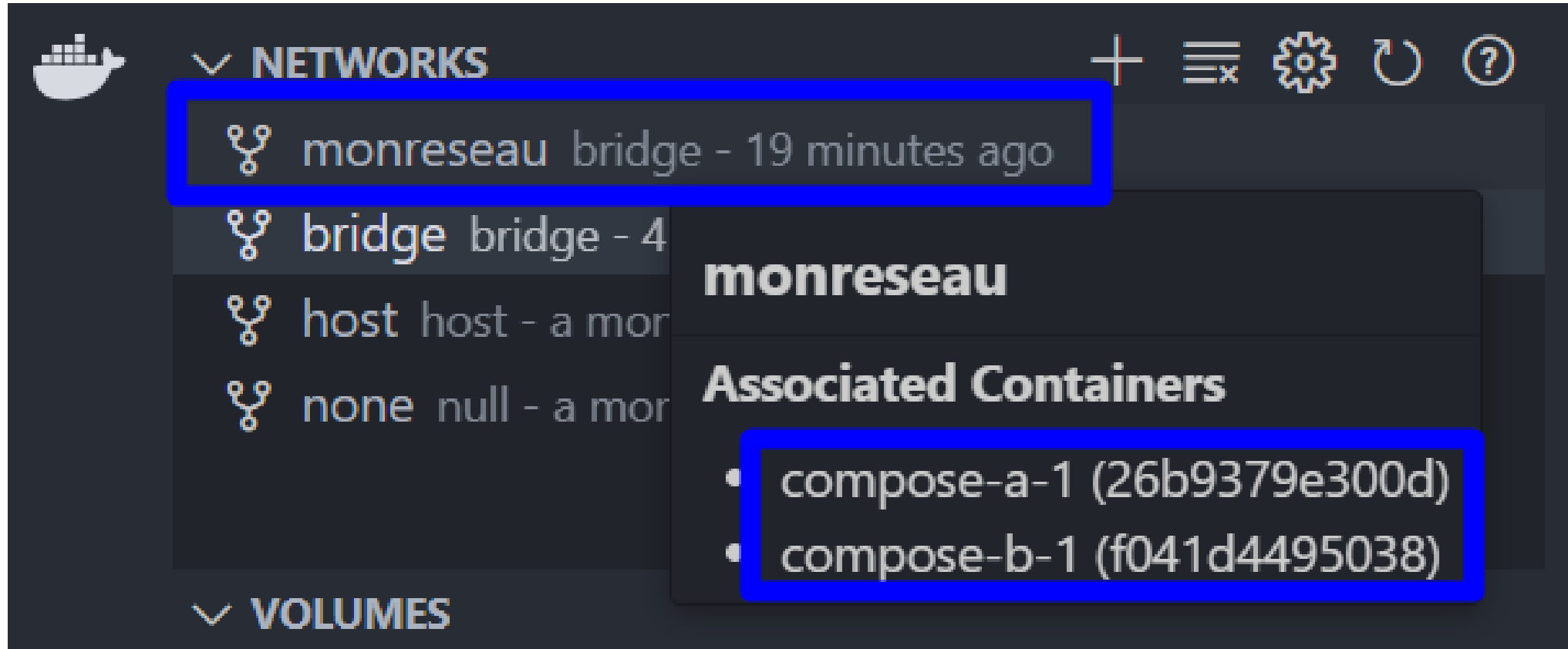
```
$ docker-compose down
[+] running 3/3
✓ Container compose-a-1   Removed
✓ Container compose-b-1   Removed
✓ Network monreseau       Removed
```

### ✓ Création de l'environnement :

```
$ docker-compose up -d
[+] Building 0.0s (0/0)
[+] Running 3/3
✓ Network monreseau       Created
✓ Container compose-a-1   Started
✓ Container compose-b-1   Started
```

Le réseau :

✓ Résultat :



The screenshot shows the Docker Desktop interface. On the left, the 'NETWORKS' section is expanded, displaying a list of networks. The 'monreseau' network is highlighted with a blue box. To the right, a detailed view of the 'monreseau' network is shown, including its type ('bridge') and a list of 'Associated Containers'. Two containers are listed: 'compose-a-1 (26b9379e300d)' and 'compose-b-1 (f041d4495038)', with the latter highlighted by a blue box.

**NETWORKS**

- monreseau bridge - 19 minutes ago
- bridge bridge - 4
- host host - a mor
- none null - a mor

**monreseau**

**Associated Containers**

- compose-a-1 (26b9379e300d)
- compose-b-1 (f041d4495038)

**VOLUMES**

**Questions ??**

Prochain chapitre: Cas concret  
MEP serveur LAMP.