

Dockerfile –partie 3

Introduction

Sources :

<https://docs.docker.com/engine/reference/builder/#dockerfile-reference>

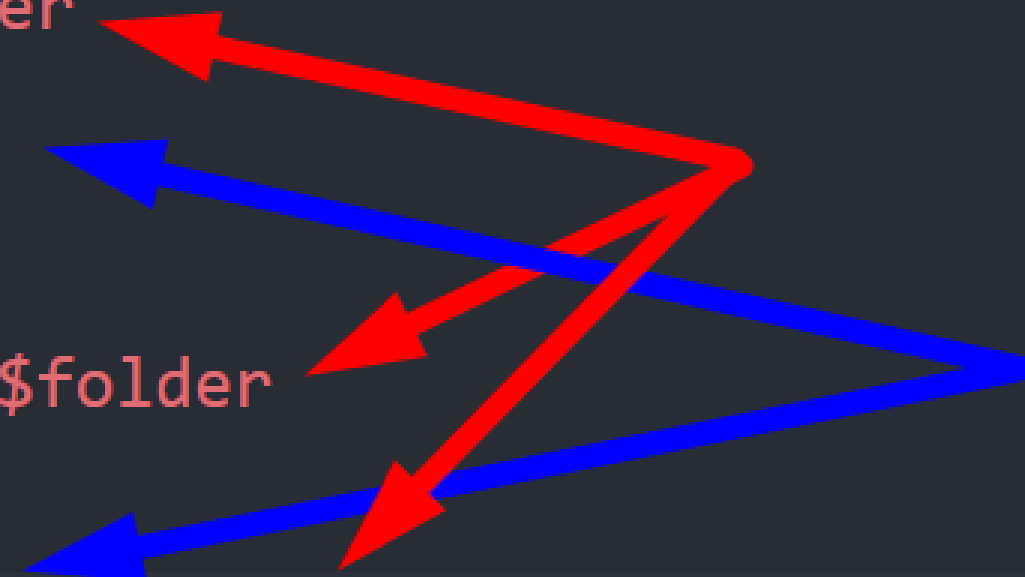
<https://docs.docker.com/build/building/packaging/>

Les arguments

L'instruction ARG :

- Permet de paramétrer des arguments dans le dockerfile et les passer au moment du build.

```
11 ARG folder
12 ARG file
13
14 WORKDIR $folder
15
16 COPY $file $folder
```



The diagram illustrates the resolution of variables in the Dockerfile instructions. Red arrows indicate the resolution of the 'folder' argument: one arrow points from 'folder' in line 11 to '\$folder' in line 14, and another points from 'folder' in line 16 to '\$folder' in line 14. Blue arrows indicate the resolution of the 'file' argument: one arrow points from 'file' in line 12 to '\$file' in line 16, and another points from 'file' in line 12 to '\$folder' in line 14. This shows that both '\$file' and '\$folder' are resolved to the same source at the time of the COPY instruction.

L'image :

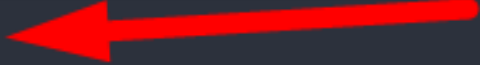
```
Christophe@ChrisDell MINGW64 /d/AFCI/Cours/8 - conteneurisation/demos/Dockerfile
$ docker build -t monimage:dev --build-arg folder=/var/www/ --build-arg file=hello-world.js .
[+] Building 1.4s (10/10) FINISHED
```

Résultat :

```
Christophe@ChrisDell MINGW64 /d/AFCI/Cours/8 - conteneurisation/demos/Dockerfile
$ docker run -ti monimage:dev bash
root@bc7f324d9e61:/var/www# ls
var
root@bc7f324d9e61:/var/www# cd var/
root@bc7f324d9e61:/var/www/var# ls
www
root@bc7f324d9e61:/var/www/var# cd www/
root@bc7f324d9e61:/var/www/var/www# ls
hello-world.js
root@bc7f324d9e61:/var/www/var/www#
```

L'instruction ARG :

- Valeur par défaut.

```
10 |  
11 ARG folder=/var/www/   
12 ARG file
```

L'image :

```
Christophe@ChrisDell MINGW64 /d/AFCI/Cours/8 - conteneurisation/demo:  
$ docker build -t monimage:dev --build-arg file=hello-world.js .  
[+] Building 0.7s (9/9) FINISHED
```


Résultat :

```
Christophe@ChrisDell MINGW64 /d/AFCI/Cours/8 - cd
$ docker run -ti monimage:dev bash
root@aaedceb85612:/var/www# ls
hello-world.js
root@aaedceb85612:/var/www#
```

Point particulier :

```
1  ARG version
2  FROM php:$version
```

Si placé avant le FORM :

- UNIQUEMENT visible par ce dernier.
- ✓ Pratique pour choisir une version de PHP au moment du build par exemple.

Les variables d'environnement

Variables d'environnement :

```
11  WORKDIR /var/www/  
12  
13  COPY ./hello-world.js .
```

```
Christophe@ChrisDell MINGW64 /d/AFCI/Cours/8 - conteneurisation/demos/Dockerfile  
$ docker build -t monimage:dev .  
[+] Building 1.1s (10/10) FINISHED  
=> [internal] load .dockerignore
```

Variables d'environnement :

```
Christophe@ChrisDell MINGW64 /d/AFCI/Cours/8 - conteneurisation/demos/Dockerfile
```

```
$ docker run -ti monimage:dev bash
```

```
root@4637b06f2ca4:/var/www# env
```

```
HOSTNAME=4637b06f2ca4
```

```
PWD=/var/www
```

```
HOME=/root
```

```
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;42:ow=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.xgz=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.xspf=00;36:
```

```
TERM=xterm
```

```
SHLVL=1
```

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

```
_=/usr/bin/env
```

```
root@4637b06f2ca4:/var/www#
```

Variables d'environnement :

```
6      curl -fsSL https://deb.nodesource.com/gpgkey/nodesour
7
8
9
10
11  ENV
12
```

ENV **key** value **Clé valeur**

The name of the environment variable.

Set an environment variable to the specified value. The value will be in the environment of any descendent

1/3 Dockerfiles

Test :

```
9 apt-get install nodejs -y
```

```
10
```

```
11 ENV environnement=production
```

```
12
```

```
13 WORKDIR /var/www/
```

Résultat :

```
Christophe@ChrisDell MINGW64 /d/AFCI/Cours/8 - conteneurisation/demos/Dockerfile
$ docker run -ti monimage:dev bash
root@64c5fd710987:/var/www# env
HOSTNAME=64c5fd710987
PWD=/var/www
HOME=/root
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.a=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.win:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:1;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=01;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:px=00;36:*.xspf=00;36:
TERM=xterm
SHLVL=1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
environnement=production
_/usr/bin/env
root@64c5fd710987:/var/www#
```


Exemple :

```
11 ARG environment
```

```
13 ENV environnement=$environment
```

```
15 WORKDIR /var/www/
```

L'image :

```
Christophe@ChrisDell MINGW64 /d/AFCI/Cours/8 - conteneurisation/demos/Dockerfile
$ docker build -t monimage:dev --build-arg environment=dev .
[+] Building 0.7s (9/9) FINISHED
=> [internal] load .dockerignore
```

Résultat :

```
*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.
.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=01;35:*.
.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=
1;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35
*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wa
px=00;36:*.xspf=00;36:
TERM=xterm
SHLVL=1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
environnement=dev
_=/usr/bin/env
root@98ceaf98651b:/var/www#
```

Les Metadata

Les labels :

```
9      apt-get install nodejs -y
10
11  LABEL author="Christophe"
12  LABEL email="contact@christophe-le-goff.com"
13
14  WORKDIR /var/www/
15
```

L'image :

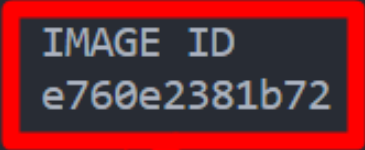
```
Christophe@ChrisDell MINGW64 /d/AFCI/Cours/8 - conteneurisation/demos/Dockerfile
$ docker build -t monimage:dev .
[+] Building 1.3s (10/10) FINISHED
```

Inspection :

```
Christophe@ChrisDell MINGW64 /d/AFCI/Cours/8 - conteneurisation/demos/Dockerfile
$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
monimage	dev	e760e2381b72	38 minutes ago	325MB

```
Christophe@ChrisDell MINGW64 /d/AFCI/Cours/8 - conteneurisation/demos/Dockerfile
$ docker inspect e760e2381b72
```



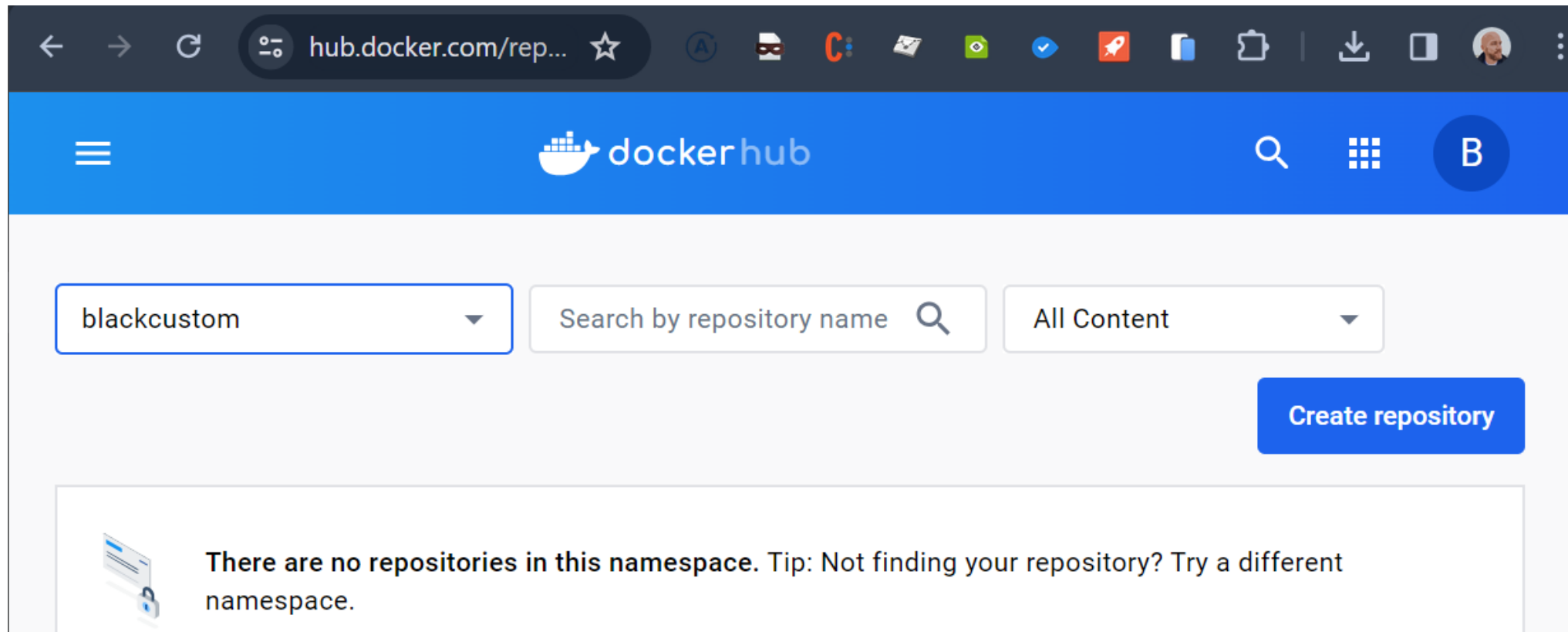
Résultat :

```
"Config": {
  "Hostname": "",
  "Domainname": "",
  "User": "",
  "AttachStdin": false,
  "AttachStdout": false,
  "AttachStderr": false,
  "Tty": false,
  "OpenStdin": false,
  "StdinOnce": false,
  "Env": [
    "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
  ],
  "Cmd": [
    "/bin/bash"
  ],
  "Image": "",
  "Volumes": null,
  "WorkingDir": "/var/www/",
  "Entrypoint": null,
  "OnBuild": null,
  "Labels": {
    "author": "Christophe",
    "email": "contact@christophe-le-goff.com",
    "org.opencontainers.image.ref.name": "ubuntu",
    "org.opencontainers.image.version": "22.04"
  }
},
"Architecture": "amd64",
```

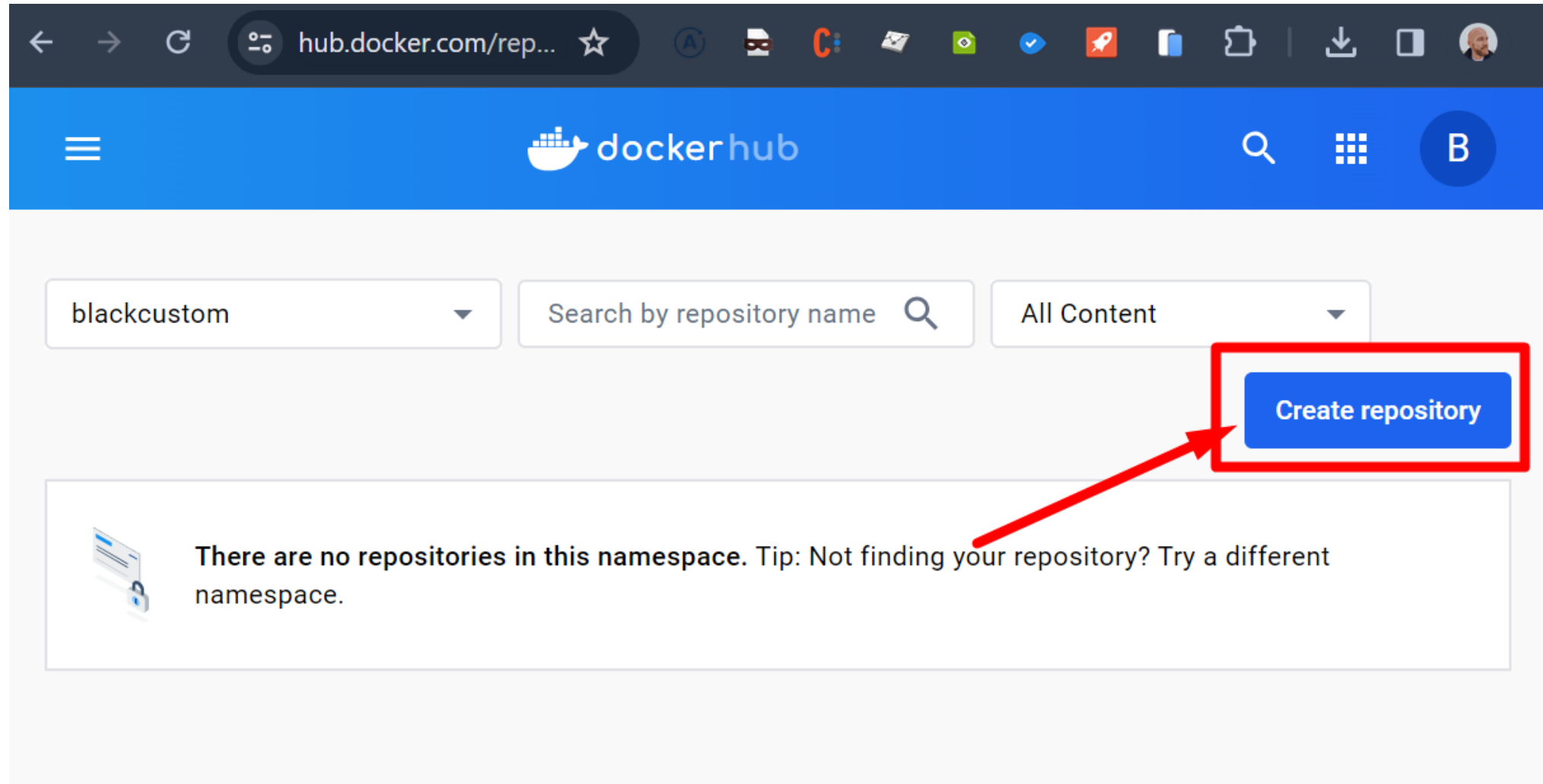
Import – Export image

Comme pour le versioning (GIT), il est possible de partager vos images sur le registry officiel de Docker :

<https://hub.docker.com/repositories/blackcustom>



Création du Repository :




Préparation du Repository :

Create repository

Namespace
blackcustom

Repository Name *
serveur-php-nginx





Short description
Un petit serveur à base de PHP et Nginx

A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.

Visibility

Using 0 of 1 private repositories. [Get more](#)


☒ Public 
Appears in Docker Hub search results


☐ Private 
Only visible to you


Cancel

Create

Résultat :

 **blackcustom / serveur-php-nginx**

Description
Un petit serveur à base de PHP et Nginx 

 Last pushed: a few seconds ago

Docker commands
To push a new tag to this repository:

`docker push blackcustom/serveur-php-nginx:tagname`

Public View

La commande pour pusher est indiquée à la création du repo.

Push de notre image :

➤ Avant de pouvoir pusher notre image il faut :

1. Vérifier la connexion au Docker Hub;
2. Reconstruire l'image.

1. Vérifier la connexion au Docker Hub;

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker login
Authenticating with existing credentials...
Login Succeeded
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> |
```

2. Reconstruire l'image (prenez le nom mentionné lors de la création du repo).

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker build -t blackcustom/serveur-php-nginx:dev .
2023/12/13 10:30:45 http2: server: error reading preface from client //./pipe/docker_engine. File has already been closed
[+] Building 3.5s (4/13)
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.26kB
=> [internal] load metadata for docker.io/library/php:7.4-fpm
=> [internal] load build context
=> => transferring context: 980B
=> [1/9] FROM docker.io/library/php:7.4-fpm@sha256:3ac7c8c74b2b047c7cb273469d74fc0d59b857aa44043e6ea6a0084372811d5b
=> => resolve docker.io/library/php:7.4-fpm@sha256:3ac7c8c74b2b047c7cb273469d74fc0d59b857aa44043e6ea6a0084372811d5b
=> => sha256:7c6a2cba718f37c4b3ca3486476c7d3dc68dc3ea9b12bde9c57bd4b8d9479fc0 2.41kB / 2.41kB
=> => sha256:38f2b691dcb8c6d4630caa2999173e35be341f2f1264164ae045d9bfb3906c28 11.38kB / 11.38kB
=> => sha256:a603fa5e3b4127f210503aaa6189abf6286ee5a73deeaab460f8f33ebc6b64e2 27.26MB / 31.41MB
=> => sha256:c428f1a494230852524a2a5957cc5199c36c8b403305e0e877d580bd0ec9e763 226B / 226B
=> => sha256:156740b07ef8a632f9f7bea4e57e4ee5541ade376adf9169351a1265382e39de 24.12MB / 91.63MB
=> => sha256:3ac7c8c74b2b047c7cb273469d74fc0d59b857aa44043e6ea6a0084372811d5b 1.86kB / 1.86kB
=> => sha256:fb5a1c8a682f00720b74127e47bda7f76c0a2e2b8a0a421750bc0025aface08d8 270B / 270B
```

Le PUSH :

```
docker push blackcustom/serveur-php-nginx:tagname
```

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker image ls
REPOSITORY          TAG          IMAGE ID      CREATED      SIZE
blackcustom/serveur-php-nginx dev          a96063ee4e4d  56 seconds ago  1192MB
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker push blackcustom/serveur-php-nginx:dev
The push refers to repository [docker.io/blackcustom/serveur-php-nginx]
4f485b90bae9: Pushed
aa12d07887f1: Pushed
55fb49dc538e: Pushed
5e4e8a5ddb16: Pushed
f4269c1f417c: Pushed
66933ec951c9: Pushed
39fd11ccdade: Pushed
8d65b457856f: Pushing [=====>]  47.45MB
5e65a6c61859: Mounted from library/php
d78098596d78: Mounted from library/php
7c314756ee72: Mounted from library/php
```

Le PUSH - repository :

Tags

This repository contains 1 tag(s).


Tag	OS	Type	Pulled	Pushed
 dev		Image	---	5 minutes ago

[See all](#)

Le PULL :

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 dev		Image	---	5 minutes ago

[See all](#)

Le PULL :

TAG

[dev](#)

Last pushed 7 minutes ago by [blackcustom](#)

```
docker pull blackcustom/serveur-...
```

DIGEST

[e680f090b610](#)

OS/ARCH

linux/amd64

LAST PULL

COMPRESSED SIZE ⓘ

177.67 MB

Pour récupérer son image en local, il suffit d'exécuter la commande proposée.


Le PULL :

```
docker system prune -a
```

```
See 'docker --help'
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker image ls
REPOSITORY      TAG          IMAGE ID      CREATED      SIZE
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> |
```

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker pull blackcustom/serveur-php-nginx:dev
dev: Pulling from blackcustom/serveur-php-nginx
a603fa5e3b41: Pull complete
c428f1a49423: Pull complete
156740b07ef8: Extracting [=====>] 17.83MB/91.63MB
fb5a4c8af82f: Download complete
972155ae644b: Download complete
a8e3b94fe6c1: Download complete
93346a3f46bc: Download complete
b922b67ca46b: Download complete
6137f893bda6: Download complete
79b1a1b78461: Verifying Checksum
0d741653eeee: Downloading [=====>] 13.29MB/26.31MB
6cb891cddbdc: Download complete
```

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker image ls
REPOSITORY      TAG          IMAGE ID      CREATED      SIZE
blackcustom/serveur-php-nginx  dev          a96063ee4e4d  13 minutes ago  492MB
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> |
```



Test conteneur :

```
REPOSITORY          TAG          IMAGE ID      CREATED        SIZE
blackcustom/serveur-php-nginx  b92983251bcc7804742548a7cd006762c03c46f4298dad353be656ce03e83ca9  15 minutes ago  492MB
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker run -d -p 8080:80 --name monsite blackcustom/serveur-php-nginx:dev
b92983251bcc7804742548a7cd006762c03c46f4298dad353be656ce03e83ca9
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> |
```

Résultat :

Mon site en Php

localhost:8080

PHP Version 7.4.33

System	Linux b92983251bcc 5.15.133.1-microsoft-standard-WSL2 #1 SMP Thu Oct 5 21:02:42 UTC
Build Date	Nov 15 2022 06:05:32

Partage offline

Il est possible de partager vos images et/ou conteneur sans passer par le Docker Hub :

- ✓ Exporter une image et/ou un conteneur
- ✓ Importer une image et/ou un conteneur

Exporter une image :

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker build -t monimage:dev .
[+] Building 0.6s (14/14) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.26kB
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/php:7.4-fpm
=> [1/9] FROM docker.io/library/php:7.4-fpm@sha256:3ac7c8c74b2b047c7cb273469d74fc0d59b857aa44043e6ea6a0084372811d5b
=> [internal] load build context
```

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker image ls
REPOSITORY          TAG       IMAGE ID       CREATED        SIZE
monimage             dev       0c0eda85fa1b   About a minute ago  492MB
blackcustom/serveur-php-nginx dev       a96063ee4e4d   34 minutes ago  492MB
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker image save monimage:dev -o monimage.tar
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> ls
```

Répertoire : D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile

Mode	LastWriteTime	Length	Name
d----	12/12/2023 21:57		app
d----	12/12/2023 21:57		config
-a----	12/12/2023 22:50	1218	Dockerfile
-a----	13/12/2023 11:06	501675520	monimage.tar

Importer une image :

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> ls
```

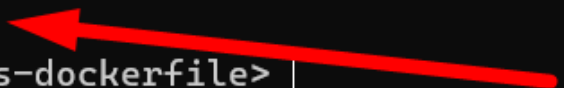
Répertoire : D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile

Mode	LastWriteTime	Length	Name
d-----	12/12/2023 21:57		app
d-----	12/12/2023 21:57		config
-a----	12/12/2023 22:50	1218	Dockerfile
-a----	13/12/2023 11:06	501675520	monimage.tar

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker image load -i .\monimage.tar
ec4a38999118: Loading layer [=====] 0.0MB/0.0MB
f60117696410: Loading layer [=====] 3.584kB/3.584kB
797a7c0590e0: Loading layer [=====] 273.5MB/273.5MB
44148371c697: Loading layer [=====] 6.144kB/6.144kB
08cc615b0242: Loading layer [=====] 11.67MB/11.67MB
91fd2792fa74: Loading layer [=====] 4.096kB/4.096kB
000000000000: Loading layer [=====] 0.0MB/0.0MB
```

Résultat :

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
monimage      dev       57aac837a976   3 minutes ago  494MB
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> |
```



Test :

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker ps -a
CONTAINER ID   IMAGE          COMMAND         CREATED        STATUS        PORTS          NAMES
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker run -d --name test monimage:dev
6f853202e238f9431f7b39b472169ba07f560d5bd84d8fcafff274251a69631ab
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker ps -a
CONTAINER ID   IMAGE          COMMAND         CREATED        STATUS        PORTS          NAMES
6f853202e238   monimage:dev   "/etc/entrypoint.sh"   3 seconds ago   Up 3 seconds   9000/tcp       test
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker exec -ti 6f8 bash
root@6f853202e238:/var/www/monsite# ls
index.php
root@6f853202e238:/var/www/monsite# |
```


Test :

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-  
root@6f853202e238:/var/www/monsite# nano index.php
```

```
GNU nano 5.4 index.php  
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Mon site en Php</title>  
</head>  
  
<body>  
  <?php echo phpinfo() ?>  
</body>  
  
</html>
```

Exporter un conteneur :

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
6f853202e238   monimage:dev   "/etc/entrypoint.sh"    20 minutes ago Up 20 minutes  9000/tcp       test
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker container export -o moncontainer.tar 6f8
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> ls
```

Répertoire : D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile

Mode		LastWriteTime	Length	Name
d----		12/12/2023 21:57		app
d----		12/12/2023 21:57		config
-a----		13/12/2023 11:25	1223	Dockerfile
-a----		13/12/2023 12:22	494003200	moncontainer.tar
-a----		13/12/2023 11:56	502922752	monimage.tar

Importer un conteneur :

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> ls
```

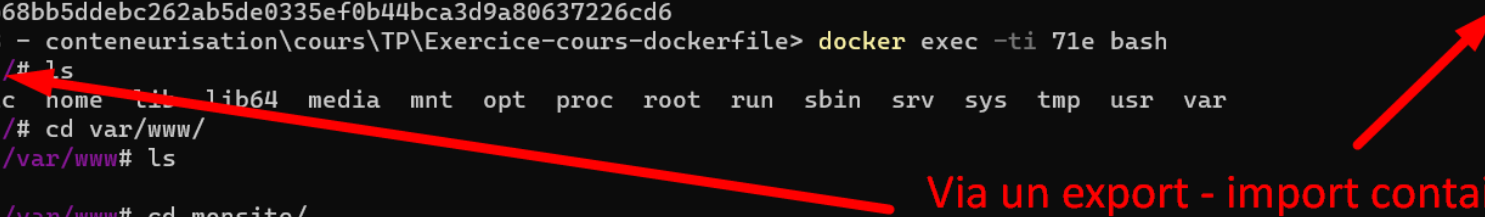
Répertoire : D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile

Mode		LastWriteTime		Length	Name
----		-----		-----	----
d-----		12/12/2023	21:57		app
d-----		12/12/2023	21:57		config
-a----		13/12/2023	11:25	1223	Dockerfile
-a----		13/12/2023	12:22	494003200	moncontainer.tar
-a----		13/12/2023	11:56	502922752	monimage.tar

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker image import .\moncontainer.tar moncontainer:dev
sha256:f2fdbad5b7c7b94ec10b9e0d92e09b1bcbef3e8f240cfa98b8801b506e0b12cf
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
moncontainer dev f2fdbad5b7c7 10 seconds ago 485MB
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile>
```

Test :

```
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker image ls
REPOSITORY      TAG          IMAGE ID       CREATED        SIZE
moncontainer    dev         0d1bb271fcf1   49 seconds ago 485MB
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker run -d --name test moncontainer:dev "etc/entrypoint.sh"
71e8555457b0db158b68bb5ddebc262ab5de0335ef0b44bca3d9a80637226cd6
PS D:\AFCI\Cours\8 - conteneurisation\cours\TP\Exercice-cours-dockerfile> docker exec -ti 71e bash
root@71e8555457b0:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@71e8555457b0:/# cd var/www/
root@71e8555457b0:/var/www# ls
html monsite
root@71e8555457b0:/var/www# cd monsite/
root@71e8555457b0:/var/www/monsite# ls
index.php
root@71e8555457b0:/var/www/monsite# |
```



Via un export - import container les
MetaData ne sont pas sauvegardées!!

Exercise

Petit exercice pour la route :

1. Créer un projet avec un fichier style.css pour : ex => mettre le 'body' en rouge
2. Créer un dockerfile à partir de l'image nginx (officielle)
3. Copier le fichier css et installer nano via le dockerfile
4. Builder l'image et la pusher sur le hub => tag 'dev'
5. M'envoyer le lien du repo pour test de mon côté
6. Récupérer l'image d'un camarade
7. lancer un container à partir de l'image et linker le css directement via le container
8. Vérifier dans un navigateur (body en rouge)

Correction

Liens :

- ✓ Image : <https://hub.docker.com/r/blackcustom/serveur-php-nginx>
- ✓ Github : <https://github.com/chrisalexlegoff/serveur-nginx-php-docker>

Questions ??

Prochain chapitre: Docker-compose – partie 1.