
DOCKER

Sommaire

Partie Cours :	1
Principe	1
Origines	2
Fonctionnement	2
Exemples d'utilisation	2
Images et Conteneurs	2
Où trouver les images ?	3
Avantages	3
Inconvénients	3
Gestion des images	4
Partie TP :	4
Installation du service Docker	4
Installation d'un conteneur	6
Téléchargement de l'image	6
Création et lancement du conteneur Docker	6
Mise en place d'un conteneur Ubuntu	8
Lancement du conteneur Ubuntu	9
Afficher la page Ubuntu sur le navigateur de la machine hôte	9
Remplacer la page par défaut de Ubuntu par un site vitrine	10
Créer ou récupérer l'image d'une archive	12

Partie Cours :

Principe

Docker est un logiciel libre destiné aux développeurs et administrateurs systèmes, dont l'objectif est de faciliter le développement, la diffusion et le déploiement d'applications web autonomes.

Son principal intérêt est **d'assembler les briques d'une application en conteneurs pouvant être partagés, sous forme d'images et de les exécuter quels que soient la plateforme et l'environnement.**

Grâce à Docker, **les applications deviennent portables et exécutables n'importe où** : sur des ordinateurs portables Mac ou Windows, des machines virtuelles, de serveurs de production, etc.

Origines

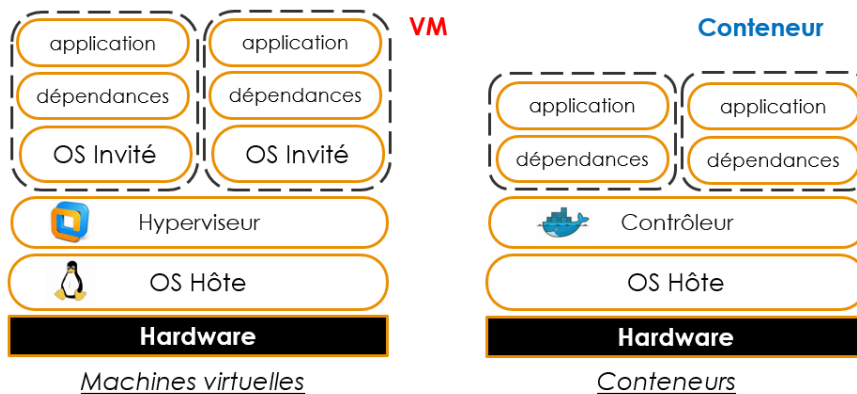
Docker a été initialement développé par Solomon Hykes pour un projet interne à la société de plateforme en tant que service (platform as a service - PaaS) qu'il a créée en 2008, dotCloud.

Docker a été distribué en tant que projet open source en 2013, projet qui a rapidement été très actif avec de nombreux contributeurs.

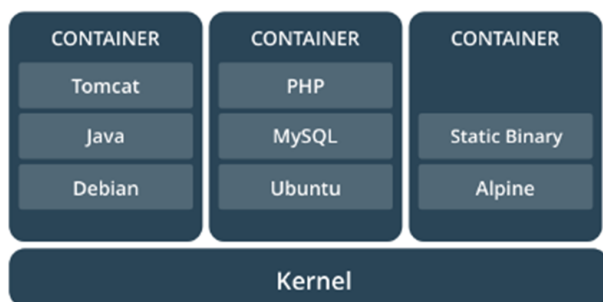
Fonctionnement

Docker utilise des fonctionnalités natives au noyau Linux mais offre les outils pour le faire de manière simplifiée pour permettre, entre autres :

- La duplication et la suppression des conteneurs ;
- L'accessibilité des conteneurs à travers la gestion des API et CLI ;
- La migration (à froid ou à chaud) de conteneurs.
- Docker est une technologie de **virtualisation par conteneurs**.



Exemples d'utilisation



Les conteneurs Docker offrent une grande flexibilité qui permet de les créer, déployer, mettre à jour, copier et déplacer d'un environnement à un autre.

Les conteneurs Docker sont actuellement pris en charge par les principaux acteurs du cloud. Ainsi, un conteneur Docker peut se déployer en de multiples instances sur toutes les principales distributions Linux, Microsoft Windows, et sur toutes les infrastructures.

Images et Conteneurs

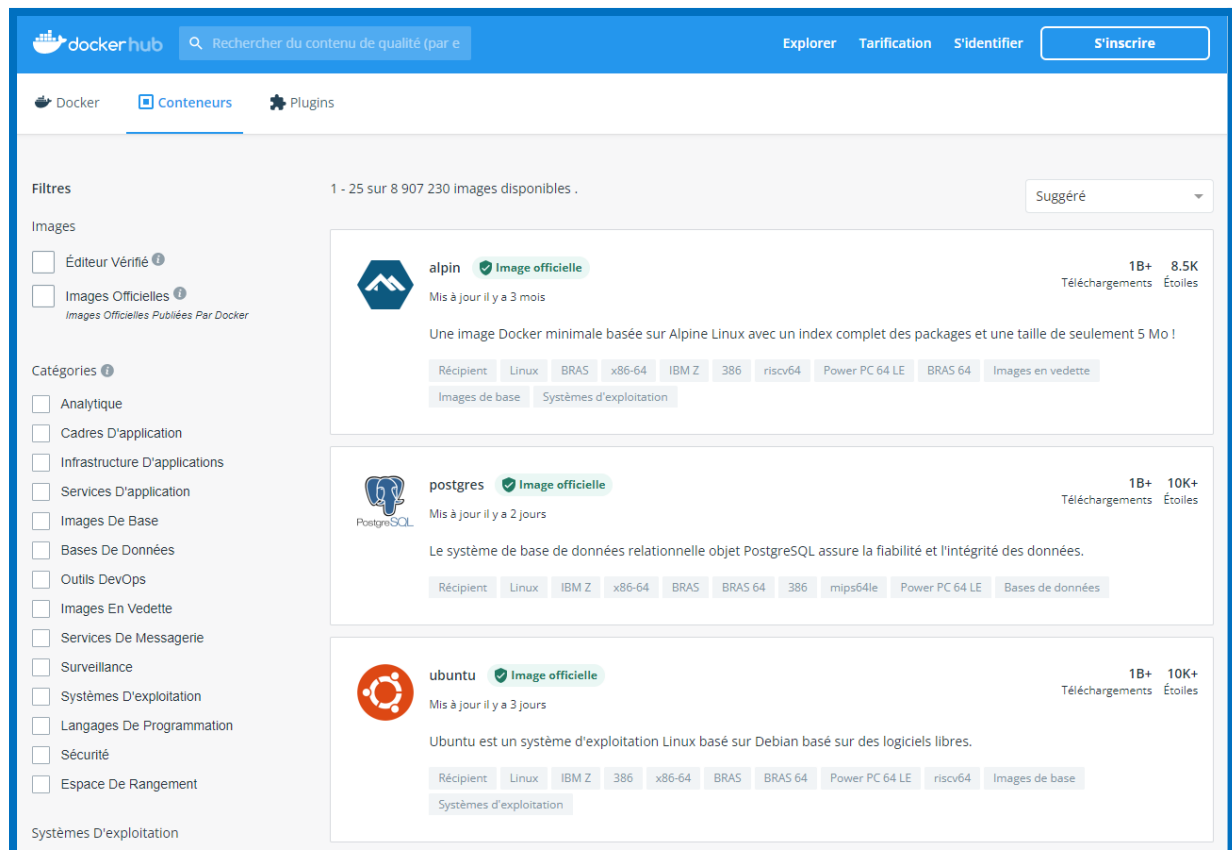
Un conteneur est une image en cours d'exécution : lorsqu'un conteneur est lancé à partir d'une image, Docker va monter le système de fichiers de l'image en lecture seule et une couche accessible en lecture/écriture.

Un conteneur peut être utilisé pour créer une nouvelle image.

Où trouver les images ?

De nombreuses images allant de la simple application comme Nextcloud à un système d'exploitation complet comme Debian sont disponibles sur :

- Le registre officiel (appelé hub) : <https://registry.hub.docker.com> ;
- De nombreux dépôts initiés par de « simples » utilisateurs.
- Il est bien sûr possible de proposer des images, d'en modifier d'autres et de déposer la modification sur le dépôt officiel.



Avantages

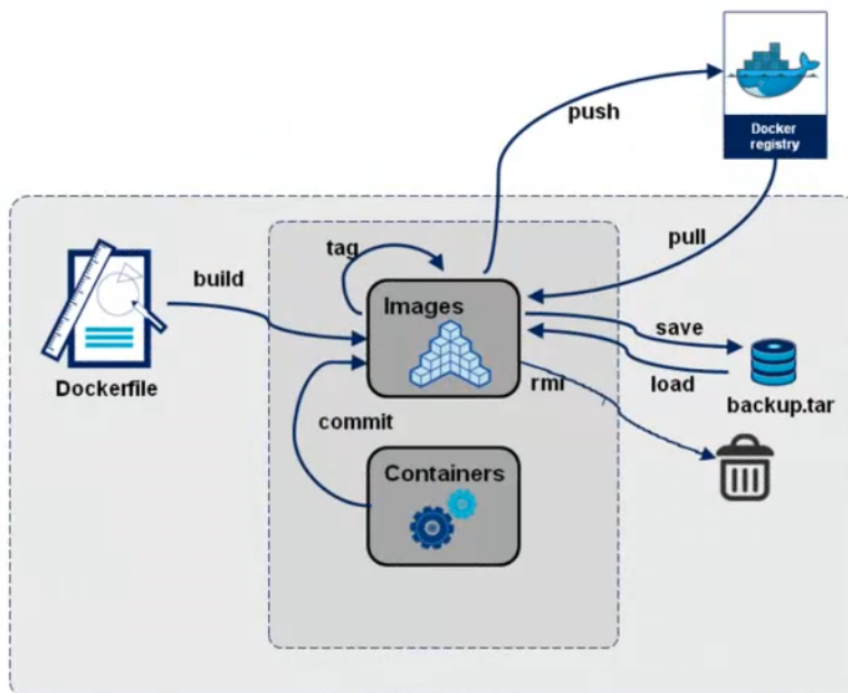
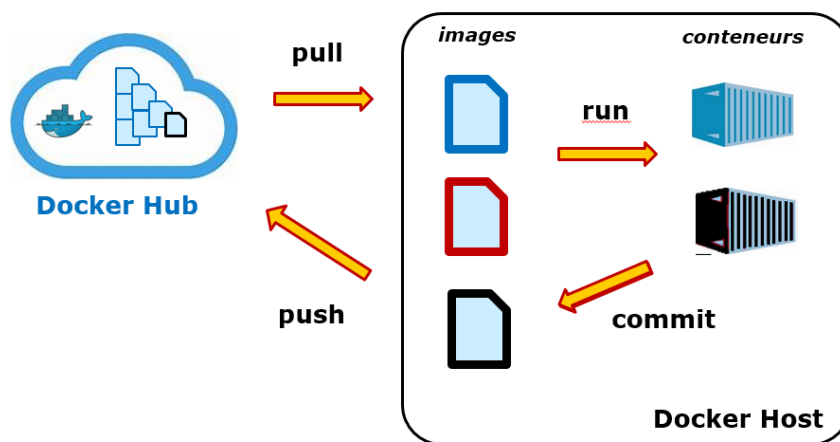
- Exécuter plusieurs machines virtuelles sur un même serveur demande des ressources suffisamment performantes pour assumer plusieurs machines virtuelles
- Les conteneurs Docker démarrent très rapidement et utilisent peu de ressources (processeur, mémoire vive, etc.)

- Les conteneurs sont légers : cela réduit l'utilisation du disque, ils se migrent plus facilement d'une machine physique à une autre et les téléchargements d'images sont beaucoup plus rapides

Inconvénients

- Les conteneurs peuvent être vulnérables, car ils partagent un noyau et des composants systèmes et leur fonctionnement exige généralement l'accès root : si toute l'architecture est basée sur Docker et si le système hôte est attaqué tous les services seront « accessibles » et exposés plus facilement

Gestion des images



Partie TP :

Installation du service Docker

Afin d'installer Docker, nous avons besoin de faire la commande wget.

```
apt update
wget https://get.docker.com/
bash index.html
```

Nous retrouvons ensuite un fichier index.html qui correspond au fichier d'installation de toutes les ressources pour Docker.

```
#!/bin/sh
set -e
# Docker CE for Linux installation script
#
# See https://docs.docker.com/engine/install/ for the installation steps.
#
# This script is meant for quick & easy install via:
#   $ curl -fsSL https://get.docker.com -o get-docker.sh
#   $ sh get-docker.sh
#
# For test builds (ie. release candidates):
#   $ curl -fsSL https://test.docker.com -o test-docker.sh
#   $ sh test-docker.sh
#
# NOTE: Make sure to verify the contents of the script
#       you downloaded matches the contents of install.sh
#       located at https://github.com/docker/docker-install
#       before executing.
#
# Git commit from https://github.com/docker/docker-install when
# the script was uploaded (Should only be modified by upload job):
```

Nous avons désormais Docker !

```
root@buster:~# systemctl status docker
• docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2023-05-05 11:23:39 CEST; 23h ago
    Docs: https://docs.docker.com
  Main PID: 7159 (dockerd)
    Tasks: 8
   Memory: 33.2M
    CGroup: /system.slice/docker.service
            └─7159 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Nous allons pouvoir tester différentes commandes avant de mettre en place des images et des conteneurs.

```
## List Docker CLI commands
docker
docker container --help

## Display Docker version and info
docker --version
docker version
docker info

## Execute Docker image
docker run hello-world

## List Docker images
docker image ls

## List Docker containers (running, all, all in quiet mode)
docker container ls
docker container ls --all
docker container ls -a -q
```

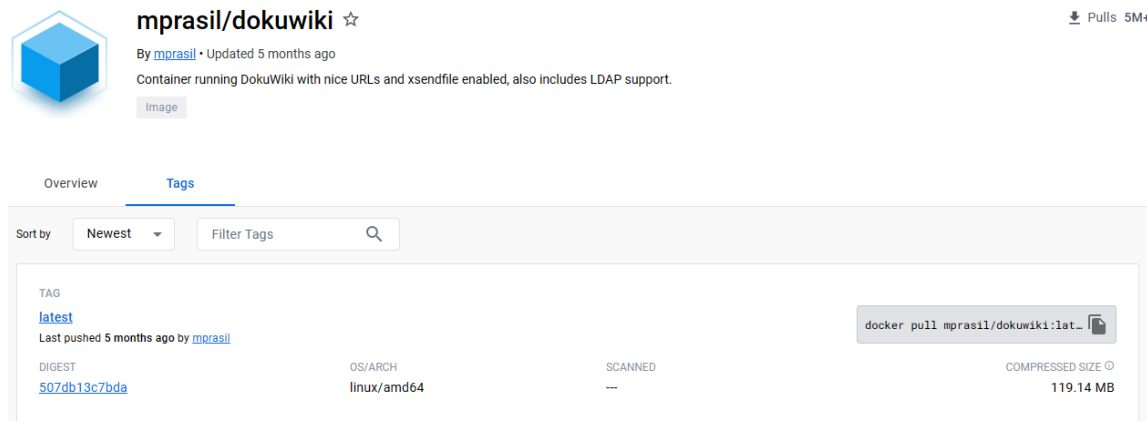
Dans cette liste de commande nous allons surtout utiliser les commandes :

- ***docker image ls*** : afficher la liste des images.
- ***docker ps*** : afficher la liste des conteneurs actifs.
- ***docker ps -a*** : afficher la liste des conteneurs.
- ***docker rm*** : supprimer un conteneur
- ***docker start*** : démarrer un docker

Installation d'un conteneur

Téléchargement de l'image

Dans cette partie nous allons installer un conteneur DokuWiki. Pour cela, nous allons tout d'abord aller sur dockerhub afin de récupérer la commande pour télécharger l'image.



mprasil/dokuwiki ☆ Pulls 5M+

By [mprasil](#) • Updated 5 months ago

Container running DokuWiki with nice URLs and xsendfile enabled, also includes LDAP support.

Image

Overview **Tags**

Sort by **Newest** Filter Tags

TAG

[latest](#)

Last pushed 5 months ago by [mprasil](#)

DIGEST [507db13c7bda](#) OS/ARCH [linux/amd64](#) SCANNED --- COMPRESSED SIZE [119.14 MB](#)

`docker pull mprasil/dokuwiki:latest`

Nous pouvons vérifier que celui-ci est bien installé.

```
root@buster:~# docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
mprasil/dokuwiki    latest      78c138e0b03c  5 months ago  319MB
root@buster:~#
```

Création et lancement du conteneur Docker

Afin de créer un conteneur d'une image, nous allons utiliser la commande : **docker run**.

```
root@Matteo:~# docker run -d -p 8001:80 --name wiki01 mprasil/dokuwiki
```

Dans cette commande on retrouve plusieurs paramètres :

- L'option **-d** permet de lancer en arrière plan.
- L'option **-p 8001:80** permet de rediriger le port 8001 de la machine physique vers le port 80 du conteneur.
- L'option **--name wiki01** permet d'affecter un nom au conteneur, en l'occurrence ici wiki01.
- L'option **mprasil/dokuwiki** correspond à l'image que nous avons choisie.

```
root@buster:~# docker run -d -p 8001:80 --name wiki01 mprasil/dokuwiki
f64df4d10c0270c23f3285ea38b12697b69684ccfdeb94767a087c0fcb5f691b
root@buster:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
f64df4d10c02   mprasil/dokuwiki  "/startup.sh run"       5 seconds ago  Up 4 seconds  0.0.0.0:8001->80/tcp, :::8001->80/tcp
wiki01
```

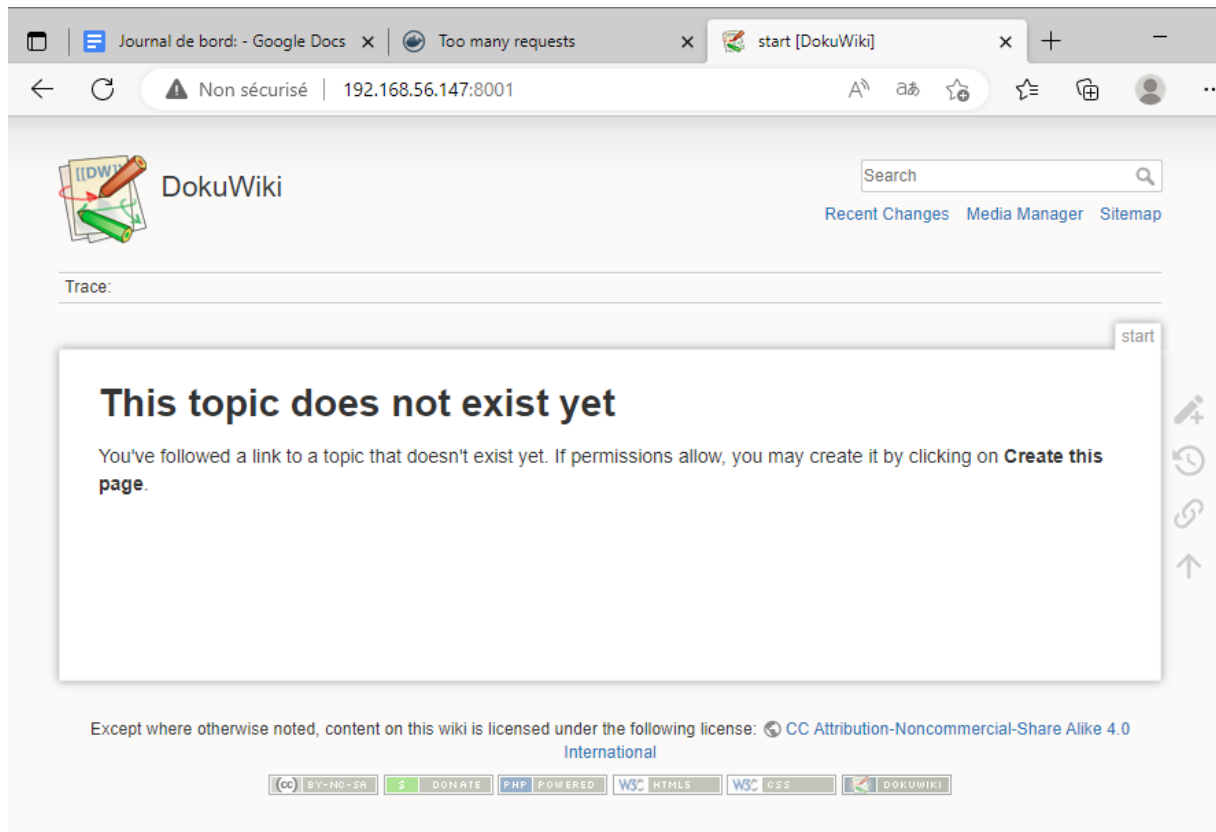
De plus, nous pouvons créer plusieurs conteneurs de la même image.

```
root@buster:~# docker run -d -p 8002:80 --name wiki02 mprasil/dokuwiki
dfed11dafd3953543c5a4131b23532976ba3d211d445e28744152747d0b8ea7a
root@buster:~# docker run -d -p 8003:80 --name wiki03 mprasil/dokuwiki
4042a5504f14a8ebfcc6bdd3aa9e87c80a5e3e007e15c5f7bbe573a3d3d063
root@buster:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
4042a5504f14   mprasil/dokuwiki  "/startup.sh run"       5 seconds ago  Up 4 seconds  0.0.0.0:8003->80/tcp, :::8003->80/tcp
wiki03
dfed11dafd39   mprasil/dokuwiki  "/startup.sh run"       17 seconds ago  Up 15 seconds  0.0.0.0:8002->80/tcp, :::8002->80/tcp
wiki02
f64df4d10c02   mprasil/dokuwiki  "/startup.sh run"       10 minutes ago  Up 10 minutes  0.0.0.0:8001->80/tcp, :::8001->80/tcp
wiki01
root@buster:~#
```

Il est également possible d'avoir accès au mode console d'un conteneur à partir d'une image.

```
root@Matteo:~# docker exec -it wiki01 bash
root@0f02d2117b57:/# ls
bin    dev      dokuwiki.tgz  home  lib32  libx32  mnt  proc  run  srv      sys  usr
boot  dokuwiki  etc          lib   lib64  media  opt  root  sbin  startup.sh  tmp  var
root@0f02d2117b57:/# _
```

Dans le navigateur de la machine hôte :



Nous retrouvons le contenu du conteneur wiki01.

Rappel :

- Un conteneur est la version exécutée d'une image docker. Il possède la copie du système de fichiers de l'image, ainsi que la capacité de lancer des processus.
- Lorsqu'un conteneur est lancé à partir d'une image, docker va monter le système de fichiers de l'image en lecture seule et monter une couche accessible en lecture/écriture
- Dans ce conteneur, il sera donc possible d'interagir avec les applications installées dans l'image, d'exécuter des scripts, de modifier des fichiers, d'installer d'autres applications

Mise en place d'un conteneur Ubuntu

Nous allons télécharger l'image Ubuntu que nous retrouvons dans dockerhub.

```
root@buster:~# docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
ubuntu              latest      3b418d7b466a  10 days ago    77.8MB
mprasil/dokuwiki    latest      78c138e0b03c  5 months ago   319MB
root@buster:~#
```

Remarque : Pour récupérer une autre version de l'image, il faut associer le nom du tag au nom de l'image. Dans l'exemple précédent, pour récupérer la version xenial, il faut utiliser la commande suivante : **docker pull ubuntu:xenial**.

Nous pouvons lancer un conteneur en lui faisant exécuter une commande spécifique (et non seulement celle prévue par défaut) :


```
root@Matteo:~# docker run ubuntu cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=22.04
DISTRIB_CODENAME=jammy
DISTRIB_DESCRIPTION="Ubuntu 22.04.2 LTS"
root@Matteo:~# _
```

```
root@buster:~# docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
PORTS         NAMES
70542666306c   ubuntu    "cat /etc/lsb-release"  7 seconds ago Exited (0) 7 seconds ago
vigilant_newton
4042a5504f14   mprasil/dokuwiki "/startup.sh run"      15 minutes ago Up 15 minutes
0.0.0.0:8003->80/tcp, :::8003->80/tcp wiki03
dfed11dafd39   mprasil/dokuwiki "/startup.sh run"      15 minutes ago Up 15 minutes
0.0.0.0:8002->80/tcp, :::8002->80/tcp wiki02
f64df4d10c02   mprasil/dokuwiki "/startup.sh run"      25 minutes ago Up 25 minutes
0.0.0.0:8001->80/tcp, :::8001->80/tcp wiki01
root@buster:~# _
```

Lancement du conteneur Ubuntu

Pour lancer un container avec l'image Ubuntu et pouvoir l'utiliser, je vais utiliser le mode interactif via les options **-ti** (-t va créer un terminal dans le conteneur et -i va permettre de se connecter à celui-ci).

```
root@buster:~# docker run -ti ubuntu /bin/bash
root@e71093d8678d:/# apt update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [925 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1083 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1010 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [41.2 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1137 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1356 kB]
88% [9 Packages store 0 B] [14 Packages 0 B/1356 kB 0%]
```

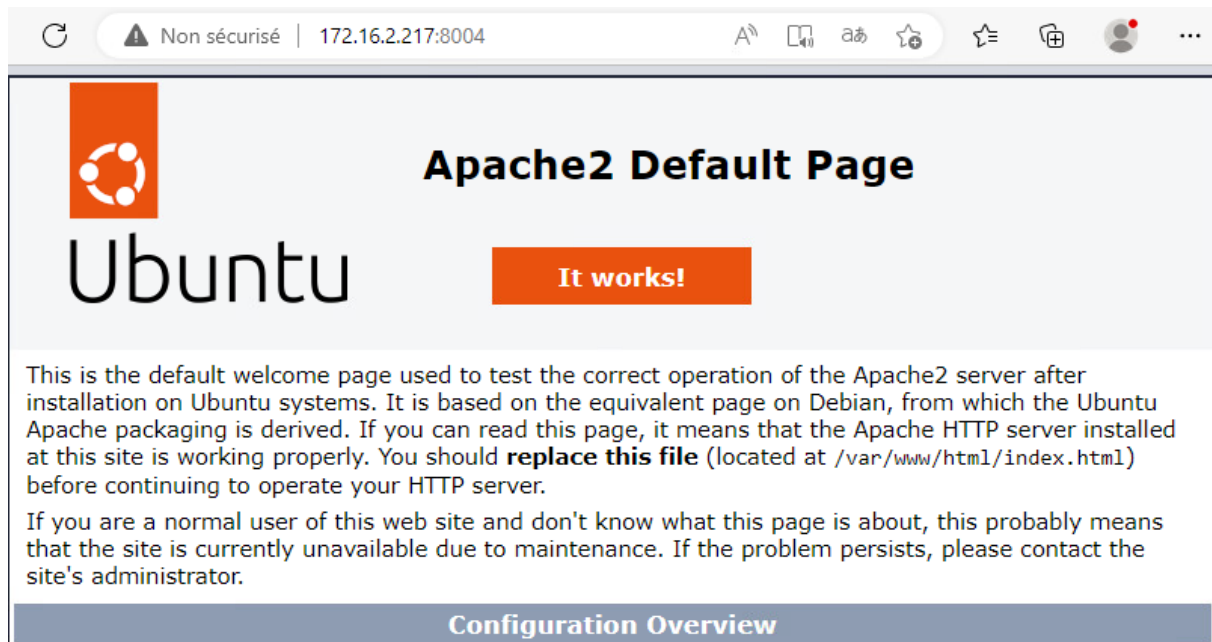
Exercice à faire :

1. Afficher la page Ubuntu sur la machine hôte
2. Remplacer la page Ubuntu par un site vitrine
3. Afficher le site vitrine

Afficher la page Ubuntu sur le navigateur de la machine hôte

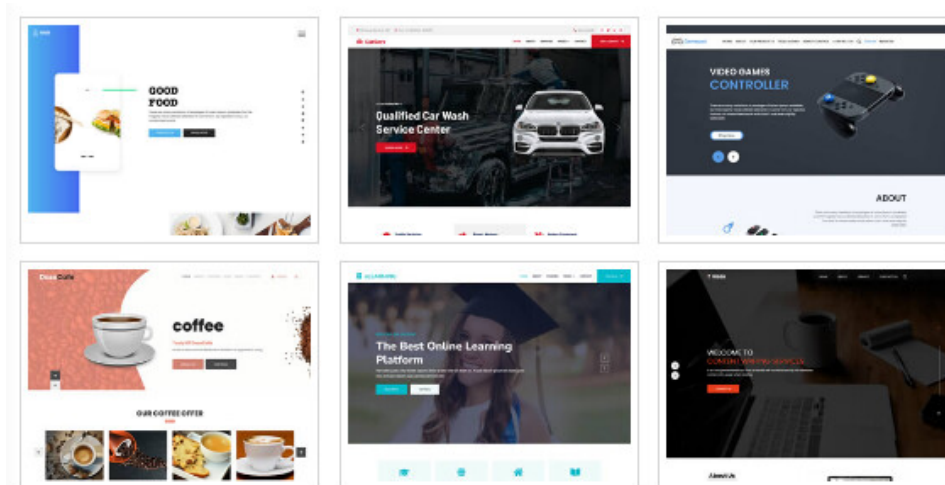
Pour cela, nous allons devoir installer Apache sur la machine et démarrer le service afin de pouvoir l'afficher sur la machine hôte :

```
root@Matteo:~# docker run --name ubuntuapache01 -dit -p 8005:80 ubuntu_
root@08f5d738c2a9:/# service apache2 start_
```



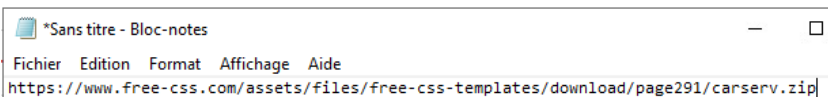
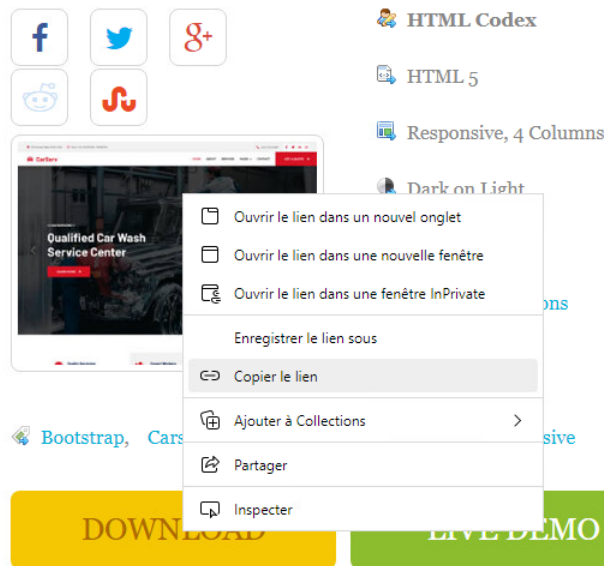
Remplacer la page par défaut de Ubuntu par un site vitrine

Ensuite on va aller chercher un template sur Internet afin de prendre un site vitrine. Free CSS est un site avec de bon template donc nous allons en choisir un dessus.



Nous allons copier le lien du download et le coller sur un bloc-notes.

CARSERV FREE CSS TEMPLATE



Cette étape va nous permettre de stocker le lien et de pouvoir utiliser la commande wget dans notre conteneur afin de télécharger le site vitrine.

Nous allons remplacer index.html par notre site donc nous pouvons le supprimer ou le déplacer peu importe.

```
root@0f6351d7221c:/var/www/html# ls
index.html
root@0f6351d7221c:/var/www/html# rm index.html
```

Nous devons également installer les paquets : wget et zip.

```
root@0f6351d7221c:/var/www/html# wget https://www.free-css.com/assets/files/free-css-templates/download/page291/carserv.zip
--2023-05-06 10:14:14-- https://www.free-css.com/assets/files/free-css-templates/download/page291/carserv.zip
Resolving www.free-css.com (www.free-css.com)... 217.160.0.242, 2001:8d8:100f:f000::28f
Connecting to www.free-css.com (www.free-css.com)|217.160.0.242|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1354335 (1.3M) [application/zip]
Saving to: 'carserv.zip'

carserv.zip      100%[=====] 1.29M --.-KB/s  in 0.1s

2023-05-06 10:14:14 (8.65 MB/s) - 'carserv.zip' saved [1354335/1354335]



root@0f6351d7221c:/var/www/html#
```

```
root@0f6351d7221c:/var/www/html# ls
car-repair-html-template carserv.zip
root@0f6351d7221c:/var/www/html#
```

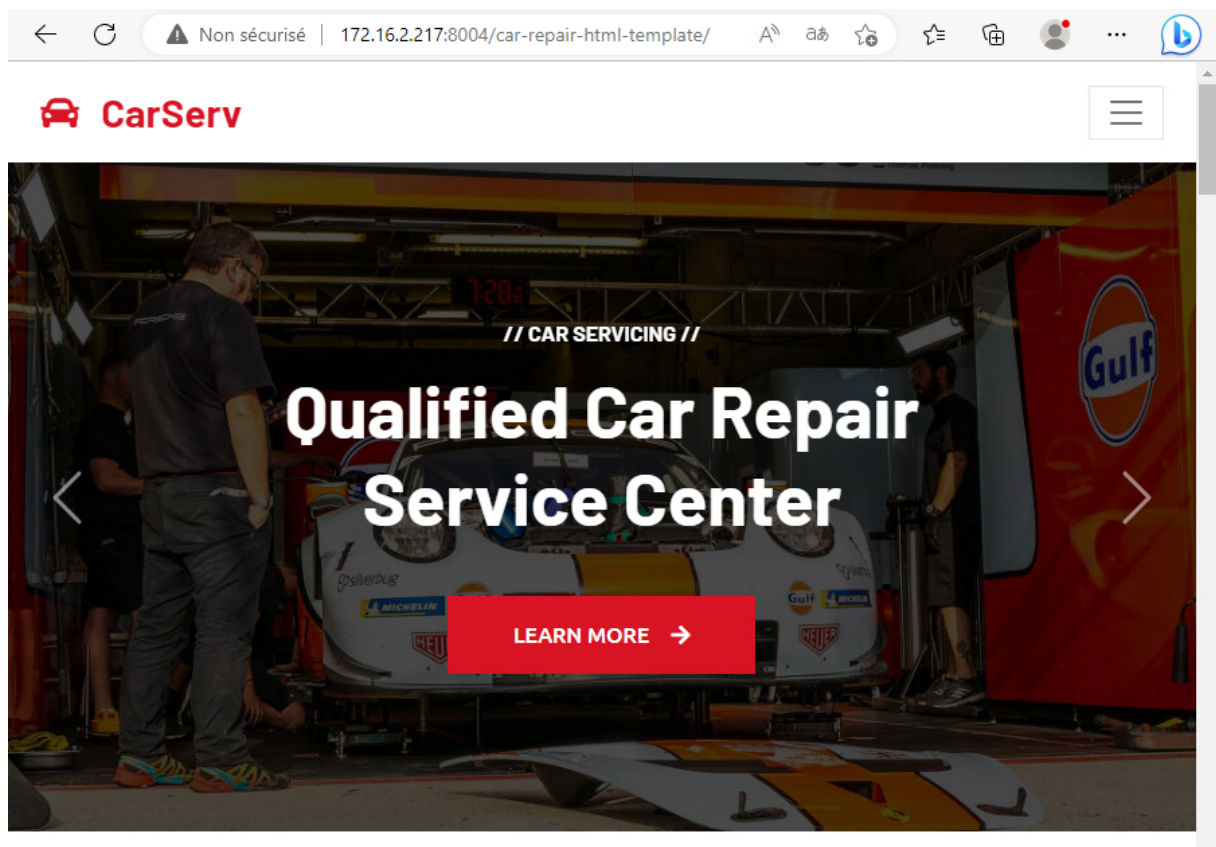
Nous pouvons maintenant retourner sur le navigateur de la machine hôte et mettre l'adresse IP de la machine Debian ainsi que le port du conteneur.



Index of /

Name	Last modified	Size	Description
 car-repair-html-template/	2021-11-17 17:39	-	
 carserv.zip	2022-01-07 13:53	1.3M	

Apache/2.4.52 (Ubuntu) Server at 172.16.2.217 Port 8004



Nous avons donc remplacé le site par défaut d'Ubuntu par un site vitrine téléchargé sur Internet !

Créer ou récupérer l'image d'une archive

En cas de problème de connexion Internet pour le téléchargement, une sauvegarde locale pour réutilisation est possible.

Nous devons commencer par compresser l'image au format .tar :

```
root@buster:~# docker save -o ubuntu.tar ubuntu_
```

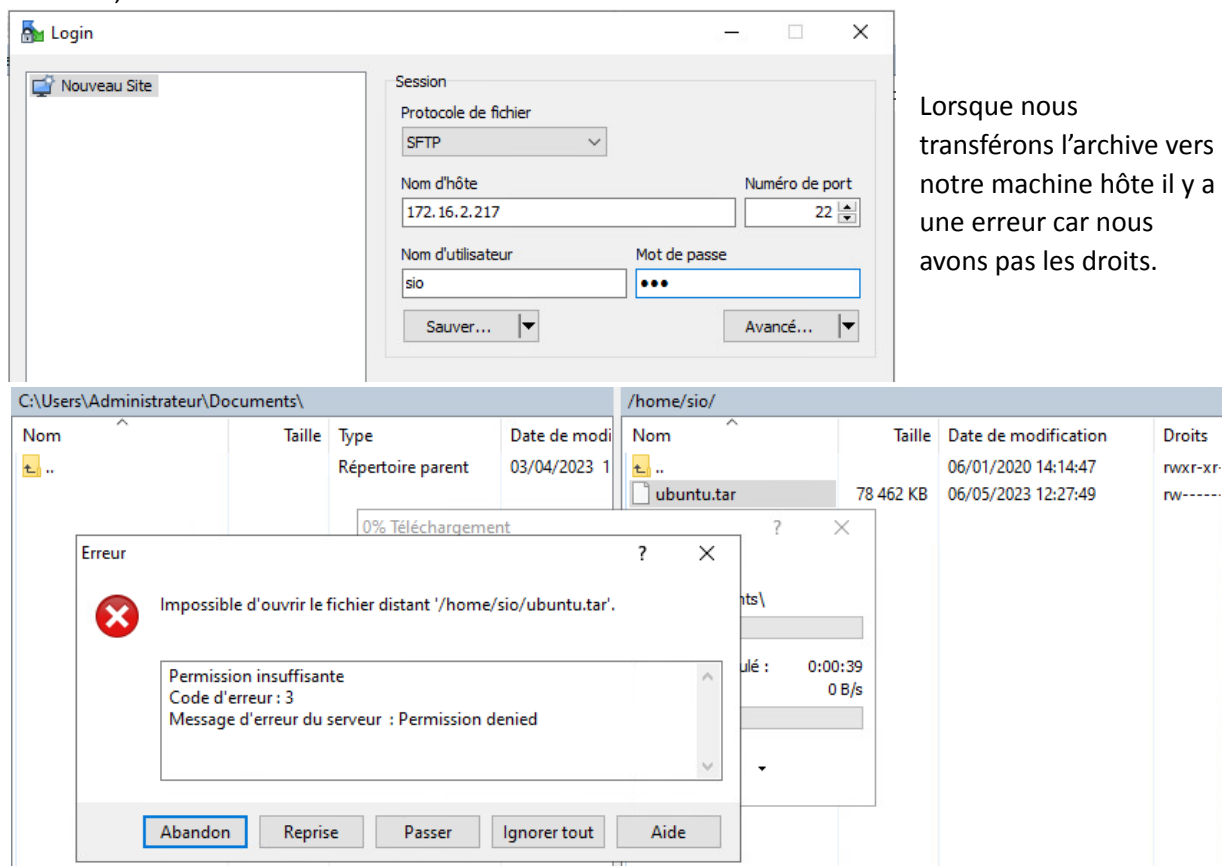
Ensuite nous allons déplacer l'archive dans /home/sio.

```
root@buster:~# cp ubuntu.tar /home/sio
root@buster:~# cd /home/sio
root@buster:/home/sio# ls
ubuntu.tar
root@buster:/home/sio#
```

Nous allons transférer l'archive .tar vers la machine hôte, nous pouvons utiliser Samba ou WinSCP mais dans notre cas nous allons faire avec WinSCP.

Remarque : installer le paquet SSH sur la machine Debian puis installer WinSCP sur la machine hôte.

Pour finir, nous allons établir la connexion entre les deux machines.



Lorsque nous transférons l'archive vers notre machine hôte il y a une erreur car nous avons pas les droits.

Nous allons donc nous donner les droits de le faire avec la commande chmod.

```
root@buster:/home/sio# chmod 777 ubuntu.tar
root@buster:/home/sio#
```

Désormais nous pouvons faire le transfert.

C:\Users\Administrateur\Documents\				/home/sio/			
Nom	Taille	Type	Date de modification	Nom	Taille	Date de modification	Droits
..		Répertoire parent	06/05/2023 1	..		06/01/2020 14:14:47	rw-r--r--
ubuntu.tar	78 462 KB	Fichier TAR	06/05/2023 1	ubuntu.tar	78 462 KB	06/05/2023 12:27:49	rw-r--r--