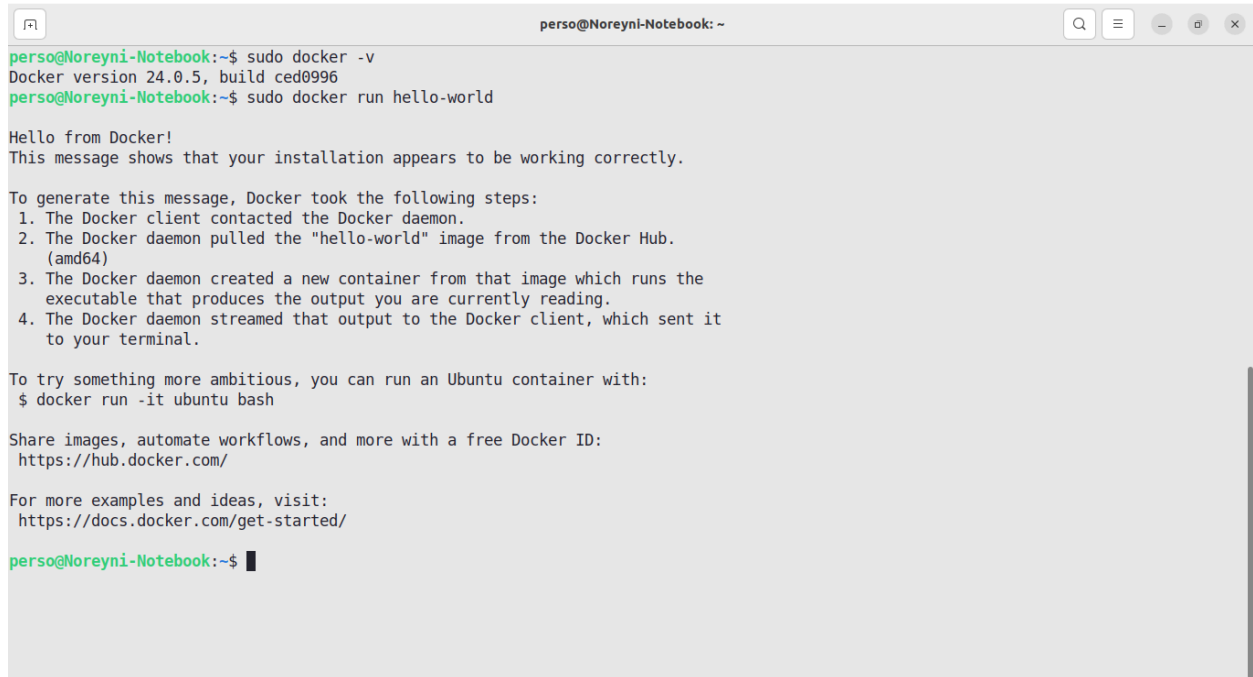


Classe: DUT1 INFO

Auteur: Ousmane SOW

Livrable 1: une capture d'écran sur la version de docker ainsi que le démarrage du docker hello-world

A screenshot of a terminal window titled 'perso@Noreyni-Notebook: ~'. The terminal shows the following commands and output:

```
perso@Noreyni-Notebook:~$ sudo docker -v
Docker version 24.0.5, build ced0996
perso@Noreyni-Notebook:~$ sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

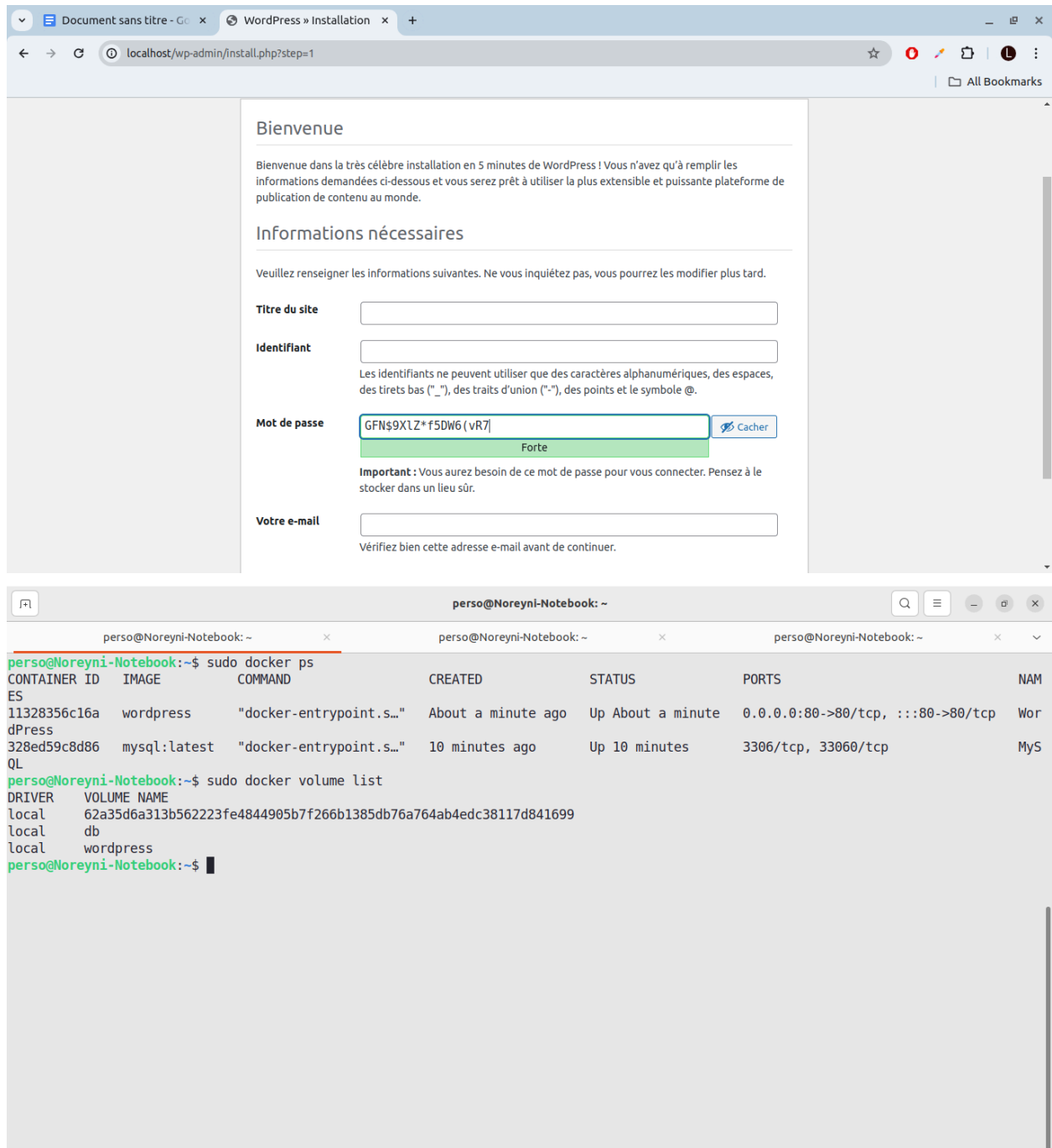
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

perso@Noreyni-Notebook:~$
```

livrable 2:

- soumettre une capture d'écran du CMS WordPress au niveau du tableau de bord principal accessible via l'URL <http://localhost>.
- soumettre une capture d'écran de la liste des conteneurs Docker en cours d'exécution,
- le mappage des ports pour le conteneur WordPress, et
- les volumes Docker sur le système hôte.



commande

sudo docker ps: lister les containers en cours d'exécution

sudo docker volume list: pour lister les volumes existants

Livrable 3:

- Soumettre une capture d'écran de docker container ls listant les deux conteneurs en cours d'exécution qui viennent d'être créés par la commande docker-compose up.

- Soumettez une capture d'écran de docker volume ls listant tous les volumes de stockage. Vous devriez au moins voir les deux volumes originaux que vous avez créés manuellement (db et wordpress), ainsi que les volumes que Docker Compose a créés dans le cadre de votre pile d'applications (ubuntu_db et ubuntu_wordpress).

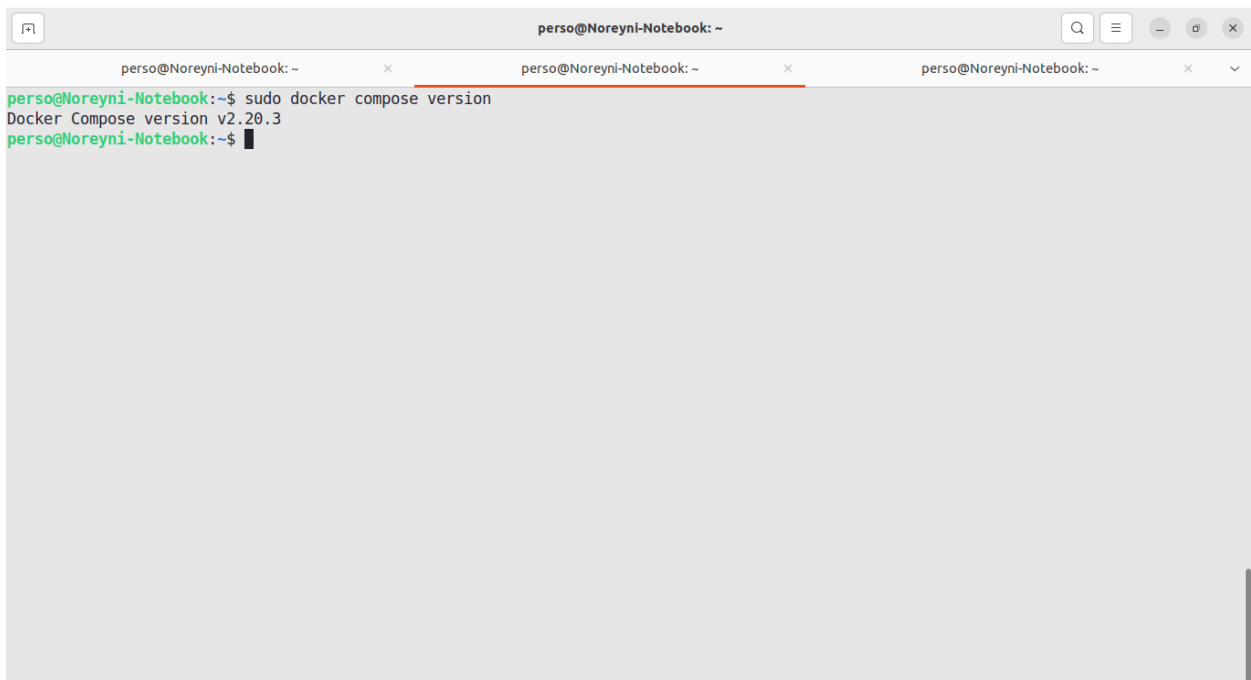
installation de docker compose(en utilisant le repository):

commande

sudo apt-get update

sudo apt-get install docker-compose-plugin

NB: j'avais installé docker avec snap donc c'est fonctionnel(c'était pas intentionnel c'est la première fois que j'utilisais docker avant le cours)



```
perso@Noreyni-Notebook: ~  
perso@Noreyni-Notebook:~$ sudo docker compose version  
Docker Compose version v2.20.3  
perso@Noreyni-Notebook:~$
```

configuration du fichier docker-compose(la casse est prise en compte par ce fichier)

commande entre : sudo vim docker-compose.yml ou sudo nano docker-compose.yml

```
Version: '3.8'
services:
  db:
    image: mysql:latest
    restart: always
    environment:
      MYSQL_DATABASE: wordpress
      MYSQL_USER: user1
      MYSQL_PASSWORD: passer123
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql
  wordpress:
    image: wordpress
    depends_on:
      - db
    restart: always
    ports:
      - "80:80"
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: user1
      WORDPRESS_DB_PASSWORD: passer123
      WORDPRESS_DB_NAME: wordpress
    volumes:
      - wordpress:/var/www/html
volumes:
  wordpress: {}
```

1,1 Top

exécution du fichier docker-compose.yml

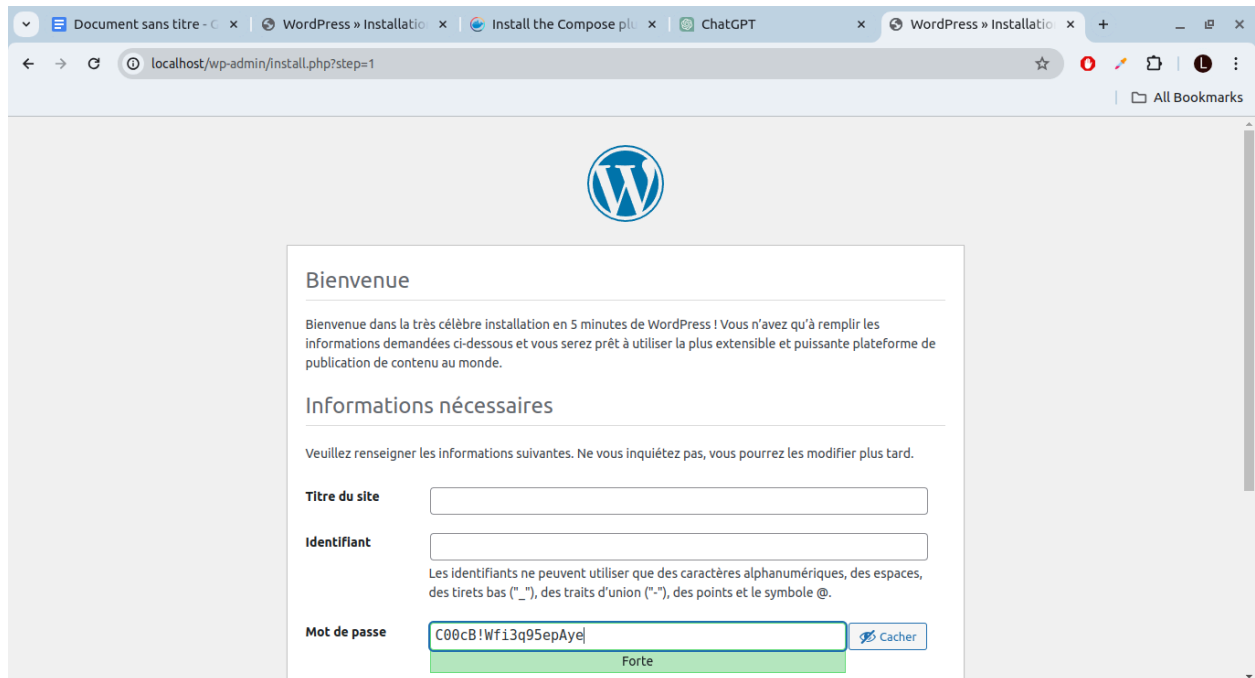
commande:

`sudo docker-compose up -d`

Note: si le docker du port 80 est toujours en exécution(le docker wordpress dans notre situation) alors le fichier ne fonctionnera pas donc on peut en déduire que le port 80 est utilisé par un seul docker

```
perso@Noreyni-Notebook:~$ sudo docker-compose up -d
[+] Running 2/2
 ✓ Container perso-db-1      Running      0.0s
 ✓ Container perso-wordpress-1 Started      6.3s
perso@Noreyni-Notebook:~$
```

le fichier est en marche donc sur le navigateur saisir : <http://localhost>

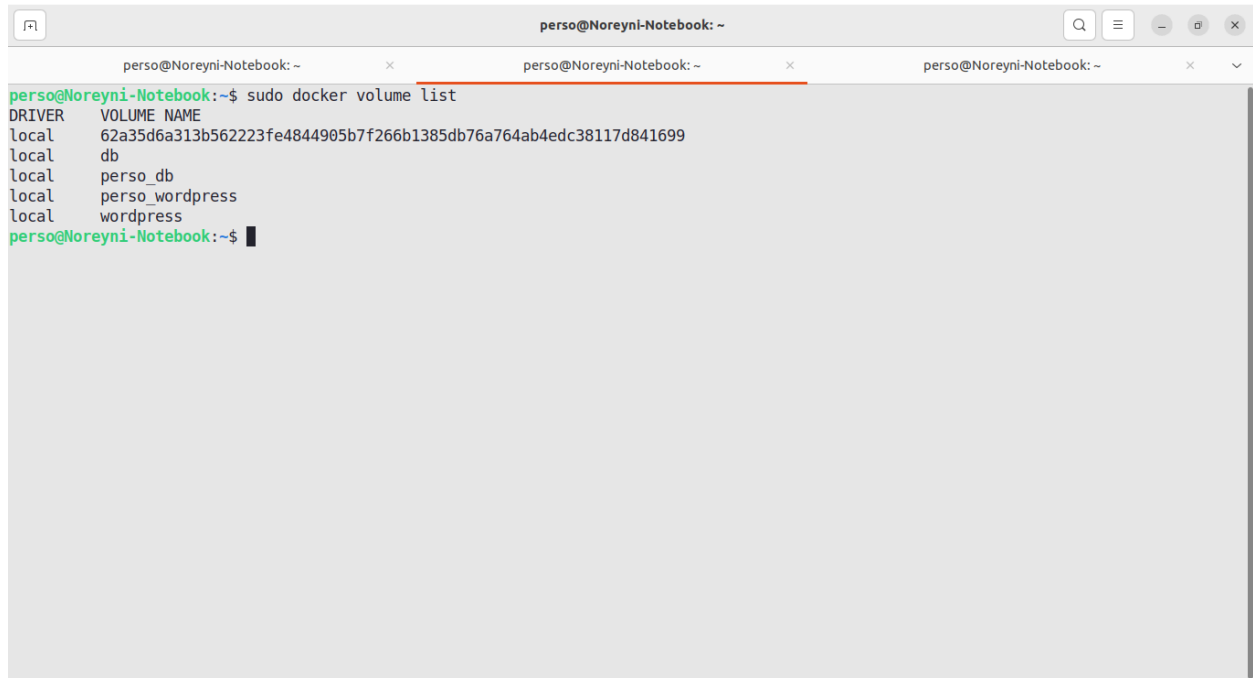


capture pour vérifier les docker en marche:

```
perso@Noreyni-Notebook: ~  
perso@Noreyni-Notebook:~$ sudo docker ps  
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES  
88f6de680073   wordpress  "docker-entrypoint.s..." 6 minutes ago  Up 5 minutes  0.0.0.0:80->80/tcp, :::80->80/tcp  perso-wordp  
ress-1  
c99e4364e979   mysql:late "docker-entrypoint.s..." 6 minutes ago  Up 6 minutes  3306/tcp, 33060/tcp               perso-db-1  
328ed59c8d86   mysql:late "docker-entrypoint.s..." 53 minutes ago  Up 53 minutes  3306/tcp, 33060/tcp               MySQL  
perso@Noreyni-Notebook:~$
```

Note: on voit plus le docker WordPress initialement mise en place car il y aurait conflit de port donc c'est pour cette raison qu'il a été éteint avec la commande `sudo docker stop WordPress`

Pour les nouveaux volumes on voit l'apparition de `perso_db` et `perso_wordpress`:



```
perso@Noreyni-Notebook: ~  
perso@Noreyni-Notebook:~$ sudo docker volume list  
DRIVER      VOLUME NAME  
local       62a35d6a313b562223fe4844905b7f266b1385db76a764ab4edc38117d841699  
local       db  
local       perso_db  
local       perso_wordpress  
local       wordpress  
perso@Noreyni-Notebook:~$
```

livrable 4:

- Soumettre votre fichier docker-compose.yml pour le site Plone.
- Soumettre une capture d'écran de docker container ls montrant vos trois conteneurs en cours d'exécution (1 serveur ZEO + 2 clients ZEO).
- Soumettre une capture d'écran de votre nouveau site Plone accessible depuis <http://localhost:8082>.

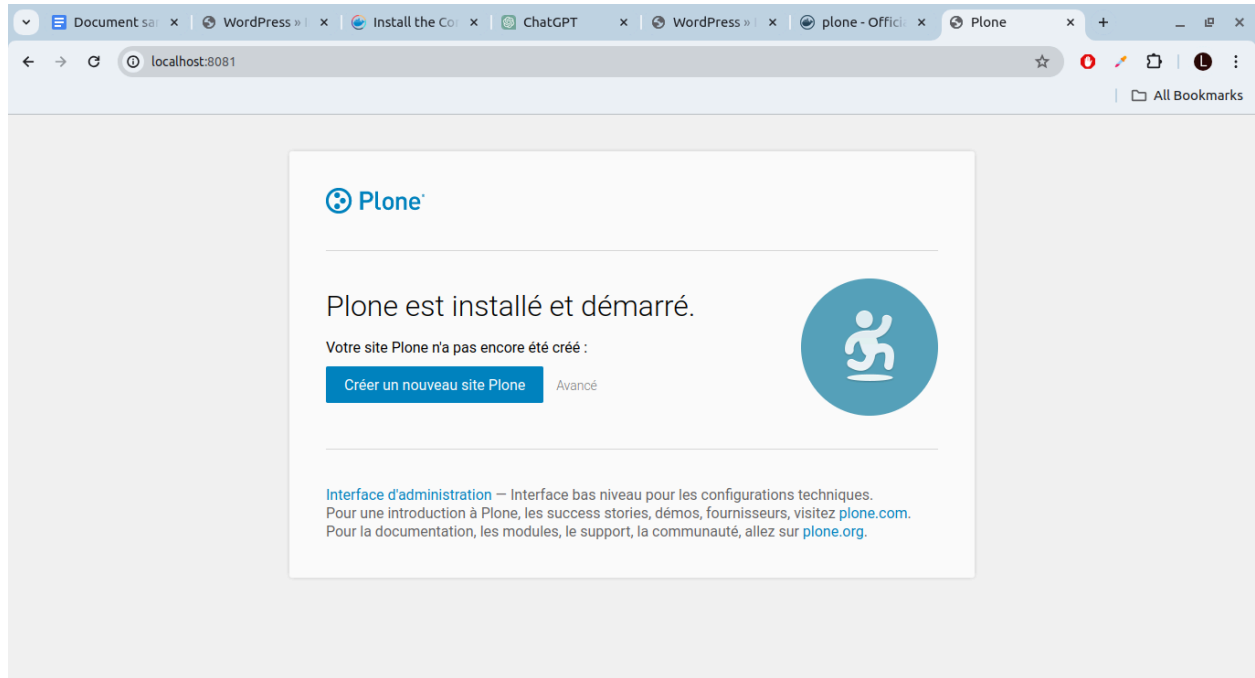
Document docker-compose.yml

```
perso@Noreyni-Notebook: ~/plone
version: '3'
services:
  zeo:
    image: plone
    command: zeo
    volumes:
      - zeo-data:/data
  instance1:
    image: plone
    depends_on:
      - zeo
    ports:
      - 8081:8080
    environment:
      ZEO_ADDRESS: "zeo:8080"
  instance2:
    image: plone
    depends_on:
      - zeo
    ports:
      - 8082:8080
    environment:
      ZEO_ADDRESS: "zeo:8080"
volumes:
  zeo-data:
~
~
"docker-compose.yml" 26L, 394B
1,1 All
```

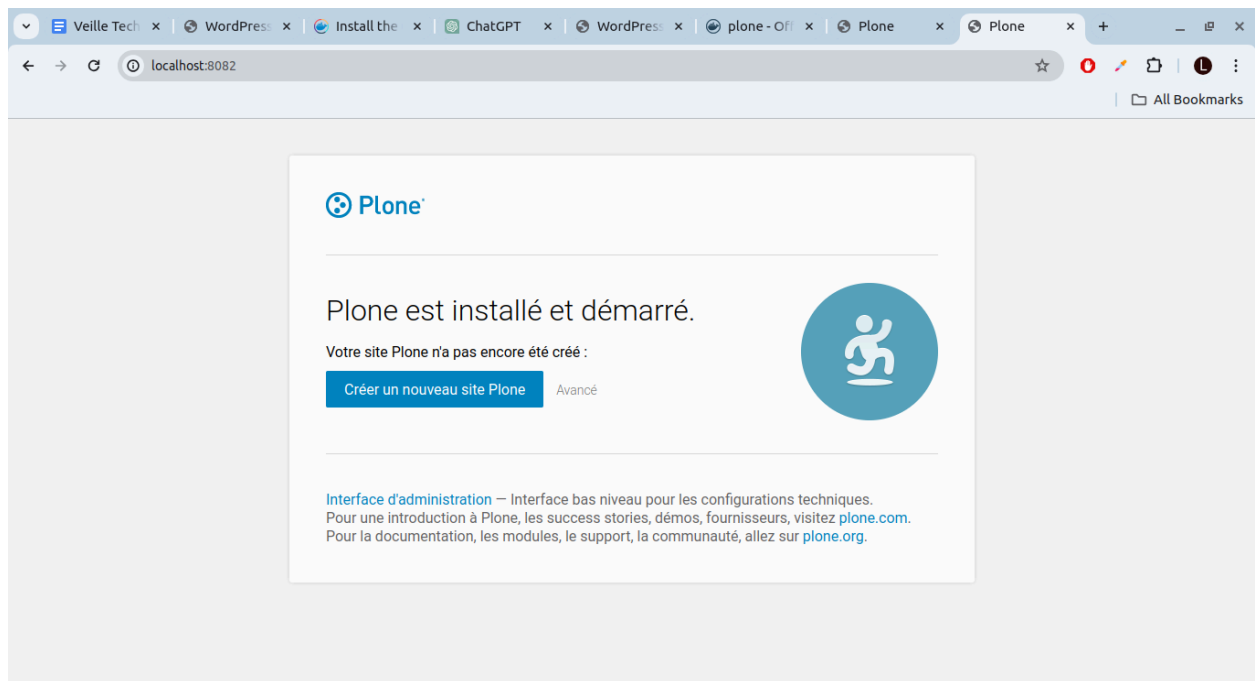
exécution du fichier:

```
perso@Noreyni-Notebook: ~/plone
perso@Noreyni-Notebook:~/plone$ sudo vim docker-compose.yml
perso@Noreyni-Notebook:~/plone$ sudo docker compose up -d
[+] Running 12/12
✓ instance2 5 layers [#####] 0B/0B Pulled 211.4s
✓ da9bdc27ef49 Pull complete 2.4s
✓ 748907db26d5 Pull complete 29.6s
✓ fcd635e41347 Pull complete 5.2s
✓ 6115e4a26909 Pull complete 29.2s
✓ d25590cc79cc Pull complete 31.1s
✓ zeo Pulled 211.5s
✓ instance1 4 layers [#####] 0B/0B Pulled 211.4s
✓ 6177a7f9989f Pull complete 25.4s
✓ dfe7d7a4d47e Pull complete 27.2s
✓ 655dd247e29c Pull complete 28.6s
✓ 9952d2948e28 Pull complete 149.9s
[+] Running 5/5
✓ Network plone_default Created 7.8s
✓ Volume "plone_zeo-data" Created 0.9s
✓ Container plone-zeo-1 Started 9.4s
✓ Container plone-instance2-1 Started 1.2s
✓ Container plone-instance1-1 Started 1.2s
perso@Noreyni-Notebook:~/plone$
```

<http://localhost:8081>



<http://localhost:8082>



Vérification des docker en cours d'exécution avec la commande
sudo docker ps


```
perso@Noreyni-Notebook: ~/plone
perso@Noreyni-Notebook: ~/plone$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
71a5b6706f90	plone	"/docker-entrypoint..."	8 minutes ago	Up 8 minutes (healthy)	0.0.0.0:8081->8080/tcp, ::
:8081->8080/tcp	plone-instance1-1				
5519d997db7c	plone	"/docker-entrypoint..."	8 minutes ago	Up 4 seconds (health: starting)	0.0.0.0:8082->8080/tcp, ::
:8082->8080/tcp	plone-instance2-1				
f49ab3d2bae6	plone	"/docker-entrypoint..."	9 minutes ago	Up 8 minutes (healthy)	8080/tcp
	plone-zeo-1				
88f6de680073	wordpress	"docker-entrypoint.s..."	37 minutes ago	Up 36 minutes	0.0.0.0:80->80/tcp, :::80-
>80/tcp	perso-wordpress-1				
c99e4364e979	mysql:latest	"docker-entrypoint.s..."	37 minutes ago	Up 37 minutes	3306/tcp, 33060/tcp
	perso-db-1				
328ed59c8d86	mysql:latest	"docker-entrypoint.s..."	About an hour ago	Up About an hour	3306/tcp, 33060/tcp
	MySQL				

```
perso@Noreyni-Notebook: ~/plone$
```

Fin du rapport.