

TP3 – Docker

6h

Objectifs :- Découvrir une solution open source de gestion de gestion de conteneurs

- Installation de Docker dans un environnement Linux
- Aborder la construction d'image et docker-compose



Installation de Docker

- Installer Docker en suivant les instructions du site Docker ; vous détaillerez dans votre rapport les étapes d'installation:
 - o <https://docs.docker.com/engine/install/debian/>
- A l'issue de l'installation, vérifier le service docker:

```
sudo systemctl status docker
```

```
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
root@doker:~# ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:dc:a3:cf brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 84914sec preferred_lft 84914sec
    inet6 fe80::a00:27ff:fedc:a3cf/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:5b:36:60 brd ff:ff:ff:ff:ff:ff
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:51:11:95:23 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
root@doker:~#
```

```
survivor@Tkhetet:~$ ssh m1wbe@10.42.0.89 -p 22
m1wbe@10.42.0.89's password:
Permission denied, please try again.
m1wbe@10.42.0.89's password:
Linux docker 4.19.0-12-amd64 #1 SMP Debian 4.19.152-1 (2020-10-18) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Oct 24 19:11:35 2020 from 10.42.0.1
m1wbe@doker:~$
```

- L'environnement de travail docker est désormais disponible dans le répertoire `var/lib/docker` son existence.

```
m1wbe@doker:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor p
   Active: active (running) since Sun 2020-10-25 01:22:13 CEST; 31min ag
     Docs: https://docs.docker.com
    Main PID: 598 (dockerd)
      Tasks: 41
     Memory: 141.4M
    CGroup: /system.slice/docker.service
            └─598 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/
lines 1-9/9 (END)
```

```
root@doker:/var/lib/docker# ls -l
total 48
drwx----- 2 root root 4096 oct. 16 01:04 builder
drwx--x--x 4 root root 4096 oct. 16 01:04 buildkit
drwx----- 2 root root 4096 oct. 16 01:04 containers
drwx----- 3 root root 4096 oct. 16 01:04 image
drwxr-x--- 3 root root 4096 oct. 16 01:04 network
drwx----- 3 root root 4096 oct. 22 19:00 overlay2
drwx----- 4 root root 4096 oct. 16 01:04 plugins
drwx----- 2 root root 4096 oct. 22 19:00 runtimes
drwx----- 2 root root 4096 oct. 16 01:04 swarm
drwx----- 2 root root 4096 oct. 22 19:00 tmp
drwx----- 2 root root 4096 oct. 16 01:04 trust
drwx----- 2 root root 4096 oct. 16 01:04 volumes
root@doker:/var/lib/docker#
```

```

m1wbe@docker:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
    default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast s
    tate UP group default qlen 1000
    link/ether 08:00:27:dc:a3:cf brd ff:ff:ff:ff:ff:ff
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast s
    tate UP group default qlen 1000
    link/ether 08:00:27:5b:36:60 brd ff:ff:ff:ff:ff:ff
    inet 10.42.0.89/24 brd 10.42.0.255 scope global dynamic noprefixrou
    t e enp0s8
        valid_lft 3439sec preferred_lft 3439sec
    inet6 fe80::a00:27ff:fe5b:3660/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue s
    tate DOWN group default
    link/ether 02:42:01:2d:10:22 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
m1wbe@docker:~$

```

- Je constate qu'il y'a u,e nouvelle interface docker0 qui vient de se créer
- **Gestion d'un premier conteneur « Hello-world »**
- **Conteneur de base:**
- Récupérer l'image « Hello World »:

```
sudo docker pull tutum/hello-world
```

- ```

root@docker:/home/m1wbe# sudo docker pull tutum/hello-world
Using default tag: latest
latest: Pulling from tutum/hello-world
Image docker.io/tutum/hello-world:latest uses outdated schema1 manifest
format. Please upgrade to a schema2 image for better future compatibilit
y. More information at https://docs.docker.com/registry/spec/deprecated-
schema-v1/
658bc4dc7069: Already exists
a3ed95caeb02: Already exists
af3cc4b92fa1: Already exists
d0034177ece9: Already exists
983d35417974: Already exists
Digest: sha256:0d57def8055178aafb4c7669cbc25ec17f0acdab97cc587f30150802d
a8f8d85
Status: Image is up to date for tutum/hello-world:latest
docker.io/tutum/hello-world:latest
root@docker:/home/m1wbe#

```

- Vérifier la présence sur votre serveur

```
sudo docker image ls
```

```

root@docker:/home/m1wbe# docker image ls
REPOSITORY TAG IMAGE ID CREATED
hello-world latest bf756fb1ae65 9 months ago
tutum/hello-world latest 31e17b0746e4 4 years ago

```

- 
- Créer le conteneur associé :

```
sudo docker run -d -p 80 tutum/hello-world
```

- 

```

root@docker:/home/m1wbe# sudo docker run -d -p 80 tutum/hello-world
8d0b732a26b9e06c767ff59b369e39aa2a63abda874477a98565f6c9da123211
root@docker:/home/m1wbe# sudo docker container ls
CONTAINER ID IMAGE COMMAND CREATED
8d0b732a26b9 tutum/hello-world "/bin/sh -c 'php-fpm..." 7 secon
ds ago Up 4 seconds 0.0.0.0:32768->80/tcp gifted_curran
root@docker:/home/m1wbe#

```

- Qu'est ce que renvoie cette commande ?
- Vérifier la présence du nouveau container:

```
sudo docker container ls
```

```

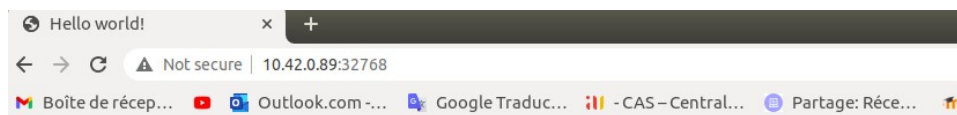
root@docker:/home/m1wbe# sudo docker container ls
CONTAINER ID IMAGE COMMAND CREATED
4450059ca2ca tutum/hello-world "/bin/sh -c 'php-fpm..." About a
minute ago Up About a minute 0.0.0.0:8080->80/tcp kind_knuth
8d0b732a26b9 tutum/hello-world "/bin/sh -c 'php-fpm..." 12 minu
tes ago Up 12 minutes 0.0.0.0:32768->80/tcp gifted_curran
root@docker:/home/m1wbe#

```

- 
- Quel est le port d'écoute utilisé par docker pour accéder au contenu de l'application Hello-world
- C'est le port 32768

```
sudo docker port <id container> 80
```

- Puis vérifier à quel service correspond ce port grâce à la commande netstat.
- tcp 0 0 docker:38922 ec2-34-211-222-43:https ESTABLISHED 1425/firefox-esr
- Depuis un navigateur du PC « hôte », saisir l'url permettant d'accéder au contenu web de l'application « Hello World » hébergé dans le conteneur Docker.
- <http://10.42.0.49:32768>



**Hello world!**

My hostname is 8d0b732a26b9

- 
- 



- Le conteneur dispose-t-il d'une carte réseau ?

•

Oui docker0

- **Docker Proxy:**

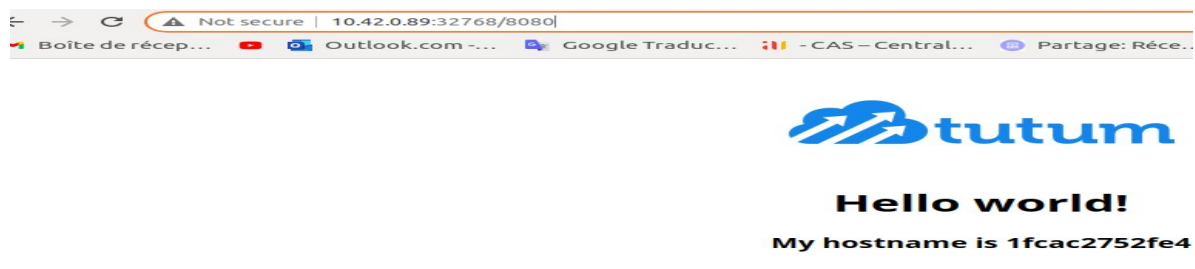
- Le service Docker Proxy permet notamment d'effectuer de la redirection de port pour protéger l'accès direct à des applications hébergées dans les containers.
- Créer un nouveau conteneur « Hello word » pour autoriser l'accès au service web sur le port 8080:

```
docker run -d -p <proxy_port>:<container_port> tutum/hello-world
```

•

```
root@docker:/home/m1wbe# sudo docker run -d -p 8080:80 tutum/hello-world
4450059ca2ca86b39af9da438112a3e815bd9752828ad82a4d751e816217f93c
root@docker:/home/m1wbe#
```

- Valider le bon fonctionnement depuis un navigateur du PC « hôte».



- Vérifier la présence des deux conteneurs sur votre serveur Docker.

Arrêt/Relance de conteneur:

```
root@docker:/home/m1wbe# sudo docker container ls
CONTAINER ID IMAGE PORTS COMMAND NAMES CREATED
4450059ca2ca tutum/hello-world 0.0.0.0:8080->80/tcp "/bin/sh -c 'php-fpm..." kind_knuth About a minute ago
Up About a minute
8d0b732a26b9 tutum/hello-world 0.0.0.0:32768->80/tcp "/bin/sh -c 'php-fpm..." gifted_curran 12 minutes ago
Up 12 minutes
root@docker:/home/m1wbe# docker kill 4450059ca2ca
4450059ca2ca
root@docker:/home/m1wbe# sudo docker container ls
CONTAINER ID IMAGE PORTS COMMAND NAMES CREATED
8d0b732a26b9 tutum/hello-world 0.0.0.0:32768->80/tcp "/bin/sh -c 'php-fpm..." gifted_curran 14 minutes ago
Up 14 minutes
root@docker:/home/m1wbe# docker start 4450059ca2ca
4450059ca2ca
root@docker:/home/m1wbe# sudo docker container ls
CONTAINER ID IMAGE PORTS COMMAND NAMES CREATED
4450059ca2ca tutum/hello-world 0.0.0.0:8080->80/tcp "/bin/sh -c 'php-fpm..." kind_knuth 4 minutes ago
Up 4 seconds
8d0b732a26b9 tutum/hello-world 0.0.0.0:32768->80/tcp "/bin/sh -c 'php-fpm..." gifted_curran 15 minutes ago
Up 15 minutes
root@docker:/home/m1wbe#
```

Arrêter le deuxième conteneur

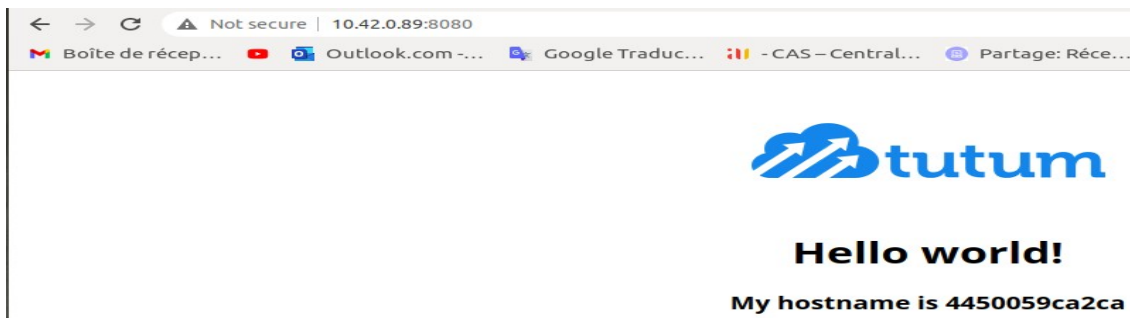
```
docker stop <id container>
```

- Vérifier que le service n'est plus opérationnel (sur la VM et le PC « hôte»)

Redémarrer le deuxième conteneur

```
docker start <id container>
```

- Vérifier que le service est opérationnel (sur la VM et le PC « hôte»)



## Modification de la page php du conteneur:

pour exécuter une commande dans un conteneur :

**docker exec <id container> <commande unix>**

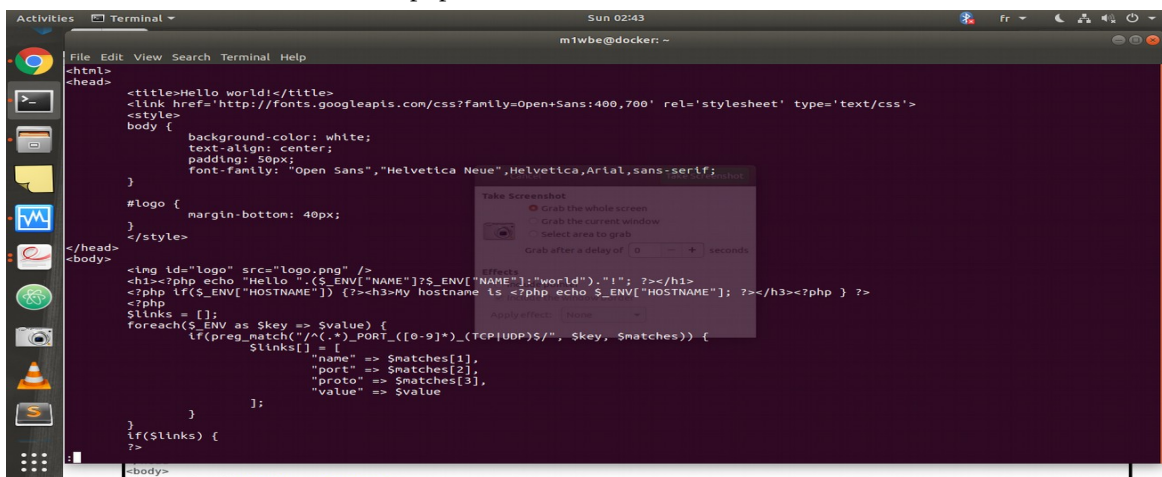
- Afficher le contenu du répertoire www dans le premier conteneur.

```
root@docker:~# docker exec 4450059ca2ca ls /www
index.php
logo.png
root@docker:~#
```

- Copier le fichier index.php du conteneur dans le répertoire /tmp du serveur Docker:

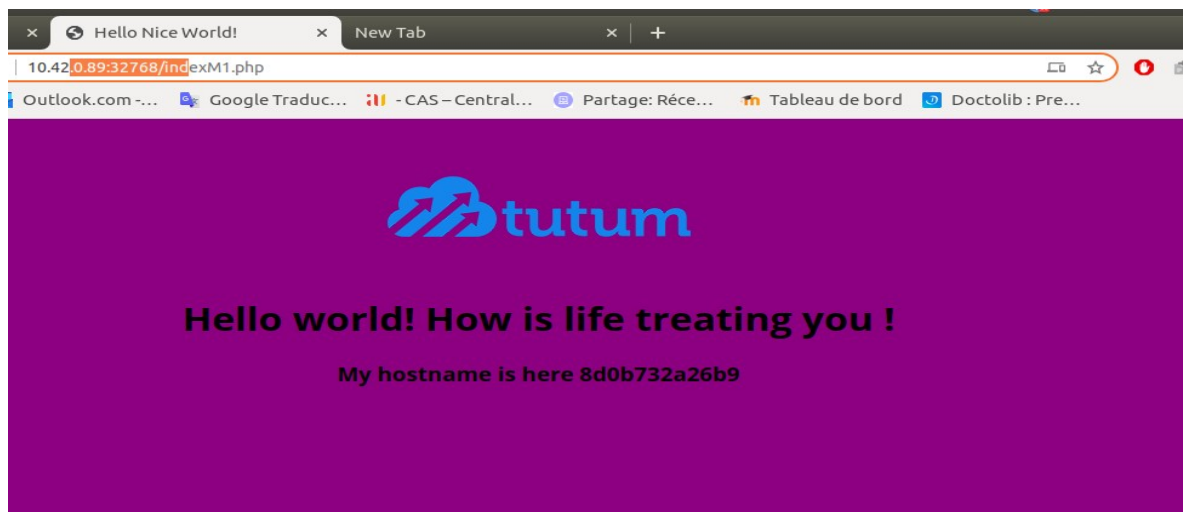
**docker cp <containerId>:/file/path/within/container /host/path/target**

- Modifier le contenu du fichier /tmp/index.php en modifiant la couleur de fond par exemple et modifier nom du fichier en indexM1.php



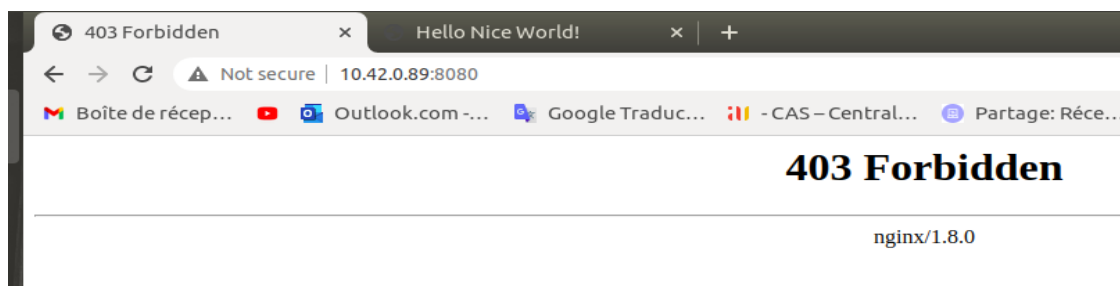
- Copier le fichier indexM1.php dans le répertoire www du conteneur puis valider le bon fonctionnement à travers le navigateur du PC «hôte».





## Modification d'un conteneur:

- Afficher le contenu du répertoire www dans le deuxième conteneur.
- Supprimer le fichier index.php. Que se passe-t-il le serveur Docker et le PC « hôte »?



Quand le browser envoie la requête pour récupérer la page index.php il ne la retrouve pas donc il renvoie

**docker kill <id container>**

Suppri

mer vos deux conteneurs :

```
root@docker:~# docker kill 4450059ca2ca
4450059ca2ca
root@docker:~# docker kill 8d0b732a26b9
8d0b732a26b9
root@docker:~# ps -l
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
4 S 0 1882 1861 0 80 0 - 2443 - pts/0 00:00:00 su
4 S 0 1883 1882 0 80 0 - 1793 - pts/0 00:00:00 bash
0 T 0 2365 1883 0 80 0 - 1400 - pts/0 00:00:00 less
0 T 0 2680 1883 0 80 0 - 1400 - pts/0 00:00:00 less
0 T 0 2736 1883 0 80 0 - 1400 - pts/0 00:00:00 less
0 R 0 2932 1883 0 80 0 - 2630 - pts/0 00:00:00 ps
root@docker:~# docker ps -l
CONTAINER ID IMAGE COMMAND
NAMES
4450059ca2ca tutum/hello-world "/bin/sh -c 'php-fpm..."
About a minute ago kind_knuth 2 hours ago Exited (137)
```

Les conteneurs sont-ils vraiment supprimés ? Vérifier dans l'arborescence de docker.

Les containers ne sont pas vraiment supprimés mais ça reste leurs configurations initiales

```
root@docker:~# mv /tmp/index.php /tmp/indexM1.php
root@docker:~# docker cp /tmp/indexM1.php 8d0b732a26b9:/www/
root@docker:~# docker exec 8d0b732a26b9 ls /www
index.php
indexM1.php
logo.png
root@docker:~# less /www/indexM1.php
```

## Docker et ses images

## Recherche d'une image dans un Docker Hub Registry:

Il existe une bibliothèque géante regroupant des images de conteneurs officielles ou non officielles. Il s'agit du HUB Docker disponible sur Internet : <https://hub.docker.com/>

Il n'est pas nécessaire de créer un compte utilisateur, il suffit de cliquer « Explore ». Il suffit ensuite de saisir dans la barre de recherche une application et vérifier l'existence d'une image associée.

## Exploration du répertoire de stockage des images

Le répertoire par défaut de stockage des images est :

```
/var/lib/docker/overlay2
```

Lister ce répertoire, et expliquez pourquoi il y a autant de contenu dedans ?

```
oot@docker:~# ls /var/lib/docker/overlay2/
b1591b8a023f9d82672d94286ce90060b518bc6f8a44c1041107a545415a9a7
b1591b8a023f9d82672d94286ce90060b518bc6f8a44c1041107a545415a9a7-init
ce731de1cebcfd084c669a3b6f8a2dd7c8ab974c74e4a460cf3cb08a9d038a9
ce731de1cebcfd084c669a3b6f8a2dd7c8ab974c74e4a460cf3cb08a9d038a9-init
dbfb0336104670681bc3b58f5c250846137e0f06bdc070af05c84928cc80818
dbfb0336104670681bc3b58f5c250846137e0f06bdc070af05c84928cc80818-init
21677a610e74d81b934819cabc9c7db9fae525ca9d095c92eb9a1bd6e7aa85e
21677a610e74d81b934819cabc9c7db9fae525ca9d095c92eb9a1bd6e7aa85e-init
3c10f3454eda1941bfff6157b279b345e6fa470ccd940c651e057d30bac2dab3
3c10f3454eda1941bfff6157b279b345e6fa470ccd940c651e057d30bac2dab3-init
6b067d5ff4a5f867ad91e7f66607c233ed6282b82f405f24096345c74f28684
6b067d5ff4a5f867ad91e7f66607c233ed6282b82f405f24096345c74f28684-init
b1dbf42b9623b714cce9f19514f304646e1b9d4857273be87954fa128e4f5a2
d0293835920f2c266713f96466a601338fddbf36fbabf8cdfd8d73b2e46300d
e33df355a4a84eb5a83e0ffdbbe4868520f177b4bab41226bccb8f7c2e274b06
02161b5fe44a47324d024f12e75afedb07315daa65fd5de316089d6d80131b5
02161b5fe44a47324d024f12e75afedb07315daa65fd5de316089d6d80131b5-init
37be9881b209a702a40215834e9f399fceed29fe80f24b2172b6ad25b0df37f
37be9881b209a702a40215834e9f399fceed29fe80f24b2172b6ad25b0df37f-init
96be06d369426011604f53a2114017f0920083199560bdef1895ab31a5bcs5a5
9d4f8e0fa7a4c5adef31e378916d3cb8f27c7a91be16312101871021a6f5c46
9d4f8e0fa7a4c5adef31e378916d3cb8f27c7a91be16312101871021a6f5c46-init
bc8731594be7d958eeff09d8ea60d4722fa826abd67353fe553914b25163859
bc8731594be7d958eeff09d8ea60d4722fa826abd67353fe553914b25163859-init
2f9a4915c9bed7762e90326f37e270a30a71ab7b871fcabdef12bf95b8bdf12
2f9a4915c9bed7762e90326f37e270a30a71ab7b871fcabdef12bf95b8bdf12-init
441056bbd3cf3f6ec14618c64bee17f037fa54310823f0d82813d47fcd8a35
86659199e893026a8c8feae0739a4b52f439407b1620006074c020fed6bdf45
86659199e893026a8c8feae0739a4b52f439407b1620006074c020fed6bdf45-init
ec87d3c643913dd8960eb898ae3526e13f9c9155d57859063c484b279fe91ff
63a605b65901d38295001720ca8aa5d6a31204296c70302424a1e73903000a6
868a1638aa5bcbdfaf3b297902e99a688cb7b37ae4fb7aa7d90975043c0f6e42d-init
8d3903bfbfd08d70a966dddcf73ff0469800a083d156520686d13dd083de7310e
90573186540be164391b3b821d4ad17b26eeb9561d8e60914ff621ce190254b8
90573186540be164391b3b821d4ad17b26eeb9561d8e60914ff621ce190254b8-init
991bbc78d09ec72a654b92146dc1866b88deabddacd383d1c3f9a7852a9c355d
9feff52608b483666d4286f0cbee1038f2dadda8f8c30522f2f470a59183c4bf7
a22c884c9e52048a2b62914cea82be1a807cb151295d6e5644f923797aeaff86
a22c884c9e52048a2b62914cea82be1a807cb151295d6e5644f923797aeaff86-init
a666cd84fbf6697d37e8d9aa8210963f8c0fb8407530c0cd13b3012037439290
a7776cfad03b4ba75aa1f2e0926b5fd64f204d0938fa56104a5ef2b3cd37dd84
aaa8743ba74cb9bde1c7614ffc883adbe4b2c2f77c70ecabd96ca9ab4ed609c7
aaa8743ba74cb9bde1c7614ffc883adbe4b2c2f77c70ecabd96ca9ab4ed609c7-init
b596c1e1219665ae7304bb5d88fe22e45ae4eac09afe4c883206cbb1044b46a6
b596c1e1219665ae7304bb5d88fe22e45ae4eac09afe4c883206cbb1044b46a6-init
c259bc93b87cdf54055a767b7740486a9567e960bb4754bc493ef6ae3ce3d7ac
c259bc93b87cdf54055a767b7740486a9567e960bb4754bc493ef6ae3ce3d7ac-init
c6614b04dcf00cf0928ec6638b7941d04d581867439445b8ab06d56ceaf5a7b1
c6614b04dcf00cf0928ec6638b7941d04d581867439445b8ab06d56ceaf5a7b1-init
cb013d82c999212c3ad0965d0b9f35fe359404c3d6ee586e54d4ccfc6a530f7e
cb013d82c999212c3ad0965d0b9f35fe359404c3d6ee586e54d4ccfc6a530f7e-init
d6c65117896e712d7497ce70c7a6e232eb19ec1b1ed2d86b989150da94211740
d6c65117896e712d7497ce70c7a6e232eb19ec1b1ed2d86b989150da94211740-init
e037011b0e59a0aac6138ad8373411ba5bd9cf1ffd8decba7328b1d52ac00383
ec6658803ddb25d4c062fcbdb036cafc98ff4498dfe43d46c50db0206de120d94
ec6658803ddb25d4c062fcbdb036cafc98ff4498dfe43d46c50db0206de120d94-init
ede19cf0e61c68cbc45378aaf6c675390e4700d535ad2ae9ac06c85372ef6d56
f2d7ced7d3fb52b1d9828fe2a3085b9f3c99c8d1ed9f879f566f85138d46ee92
f2d7ced7d3fb52b1d9828fe2a3085b9f3c99c8d1ed9f879f566f85138d46ee92-init
fd885f41beee7bbecd21132a77d5b14c160b0dad405af513196d72df47c2c511
fd885f41beee7bbecd21132a77d5b14c160b0dad405af513196d72df47c2c511-init
fe53559ae6454e2b8b553f640c039e30aebdfa08f94d59372a34b096ee60e2ef
fe53559ae6454e2b8b553f640c039e30aebdfa08f94d59372a34b096ee60e2ef-init
l
```



```

root@docker:~# docker rmi 31e17b0746e4
Error response from daemon: conflict: unable to delete 31e17b0746e4 (
ust be forced) - image is being used by stopped container 711127fdd42
root@docker:~# docker rmi -f 31e17b0746e4
Untagged: tutum/hello-world:latest
Untagged: tutum/hello-world@sha256:0d57def8055178aafb4c7669cbc25ec17f
acdab97cc587f30150802da8f8d85
Deleted: sha256:31e17b0746e48958b27f1d3dd4fe179fbbba7e8efe14ad7a51e964
81a92847a6
root@docker:~# █

root@docker:~# 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] █

Deleted Images:
untagged: hello-world:latest
untagged: hello-world@sha256:8c5aeeb6a5f3ba4883347d3747a7249f491766ca1caa47e5da5dfcf6b9b717c0
deleted: sha256:bf756fb1ae65adf866bd8c456593cd24beb6a0a061dedf42b26a993176745f6b
deleted: sha256:9c27e219663c25e0f28493790cc0b88bc973ba3b1686355f221c38a36978ac63

Total reclaimed space: 67.58kB
root@docker:~# █

```

## Création d'une

### image à partir d'undockerfile

Le dockerfile contient toutes les instructions docker nécessaires à la compilation (build) d'une image.

Pour créer une image :

- Créer un répertoire **m1-container** à la racine.
- Dans ce répertoire, créer un fichier vide, nommé Dockerfile (la casse est importante).

```

root@docker:~/m1-container# touch Dockerfile
root@docker:~/m1-container# nano Dockerfile

```

- Copier dans ce fichier le contenu de ce fichier image (il s'agit d'un container node.js):

```

from base image node (node.js)
ARG NODE_VERSION=8.11-slim
FROM node:$NODE_VERSION
LABEL "about"="This file is just am example to demonstarte the LABEL" ENV
workdirectory /home/m1xxx
RUN mkdir /dockerexample
VOLUME /dockerexample
COPY package.json .
RUN ls -ll &&\
 npm install
RUN useradd m1xxx &&\
 mkdir -p $workdirectory &&\
 chown m1xxx $workdirectory
USER m1xxx
WORKDIR $workdirectory
ADD index.js .
RUN ls -l
EXPOSE 3070
command executable and version
ENTRYPOINT ["node", "index.js"]

```

- Copier dans le répertoire M1-Container les fichiers index.js package.json
- Expliquer chaque prédicat à l'aide d'un tableau du type:

| Nom du prédicat | Description | Obligatoire / facultatif | Exemple |
|-----------------|-------------|--------------------------|---------|
|-----------------|-------------|--------------------------|---------|

|                      |                                                                                                                                                                                         |                    |                                                                                                                                        |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <i>Exemple : RUN</i> | <i>Définit une instruction à exécuter lors de la construction du container ; rajoute une couche à l'image</i>                                                                           | <i>Facultatif</i>  | <i>ls -l</i>                                                                                                                           |
| <i>FROM</i>          | <i>PREMIÈRE FONCTION Du DOCKFILE</i>                                                                                                                                                    | <i>OBLIGATOIRE</i> | <i>from folder</i>                                                                                                                     |
| <i>ARG</i>           | <i>Définir une variable à passer au constructeur au moment de la construction avec la commande de construction du docker en utilisant le flag --build-arg &lt;var&gt;=&lt;value&gt;</i> | <i>Facultatif</i>  | <i>ARG python_ver=3.0</i>                                                                                                              |
| <i>LABEL</i>         | <i>Ajouter des infos supp sur à une image</i>                                                                                                                                           | <i>Facultatif</i>  | <i>LABEL "desc"="describe image"</i>                                                                                                   |
| <i>VOLUME</i>        | <i>Crée un point de montage avec le nom spécifié et le marque comme contenant des volumes montés extérieurement à partir d'un hôte natif ou d'autres conteneurs.</i>                    | <i>Facultatif</i>  | <i>VOLUME /home/m1wbe/volume</i>                                                                                                       |
| <i>ENTRYPOINT</i>    | <i>Permet de configurer un conteneur qui fonctionnera comme un exécutable.</i>                                                                                                          | <i>Facultative</i> | <i>ENTRYPOINT ["&lt;executable&gt;", "&lt;parametre 1&gt;", "&lt;parametre 2&gt;"]<br/>Ex: ENTRYPOINT ['python', 'hello-world.py']</i> |
| <i>COPY</i>          | <i>Copier les nouveaux fichiers ou répertoires de &lt;src&gt; et les ajouter au système de fichiers</i>                                                                                 | <i>Facultative</i> | <i>COPY /home/m1wbe/web_app /var/www/html/</i>                                                                                         |
| <i>USER</i>          | <i>définir le nom d'utilisateur à utiliser lors de l'exécution de l'image et pour toutes les instructions qui le suivent RUN, CMD et ENTRYPOINT le Dockerfile.</i>                      | <i>Facultative</i> | <i>USER m1wbe</i>                                                                                                                      |
| <i>WORKDIR</i>       | <i>Définir le répertoire de travail pour toutes les instructions RUN, CMD, ENTRYPOINT, COPY et ADD qui le suivent.</i>                                                                  | <i>Facultative</i> | <i>WORKDIR /home/m1wbe</i>                                                                                                             |
| <i>EXPOSE</i>        | <i>INFORMER Docker que le conteneur écoute sur le(s) port(s) réseau spécifié(s) au moment de l'exécution.</i>                                                                           | <i>Facultative</i> | <i>Port number 8080</i>                                                                                                                |
| <i>ADD</i>           | <i>Copier les nouveaux fichiers, répertoires ou URL de fichiers distants à partir de &lt;src&gt; et les ajoute au système de fichiers de l'image au chemin &lt;dest&gt;.</i>            | <i>Facultative</i> | <i>ADD index.js style.css package.json</i>                                                                                             |

- Construire l'image (se mettre dans le répertoire m1-container):

```
sudo docker -t m1 build .
```

Vérifier la création de l'image:

```

root@docker:~/m1-container# sudo docker build -t m1 .
Sending build context to Docker daemon 4.608kB
Step 1/15 : ARG NODE_VERSION=8.11-slim
Step 2/15 : FROM node:$NODE_VERSION
----> 1aa312dac871
Step 3/15 : LABEL "about"="This file is just an example to demonstarte the LABEL"
----> Using cache
----> 5dc3599240b9
Step 4/15 : ENV workdirectory /home/m1wbe
----> Using cache
----> 84644d94b1b2
Step 5/15 : RUN mkdir /dockerexample
----> Using cache
----> 0e4a94c35d2b
Step 6/15 : VOLUME /dockerexample
----> Using cache
----> 8a58c2148dc1
Step 7/15 : COPY package.json .
----> c17b768a7d8c
Step 8/15 : RUN ls -ll &&npm install
----> Running in 55daf08f8f4c
----> Running in 55daf08f8f4c
total 72
drwxr-xr-x 2 root root 4096 Aug 31 2018 bin
drwxr-xr-x 2 root root 4096 Jun 14 2018 boot
drwxr-xr-x 5 root root 340 Oct 25 10:34 dev
drwxr-xr-x 2 root root 4096 Oct 25 10:13 dockerexample
drwxr-xr-x 1 root root 4096 Oct 25 10:34 etc
drwxr-xr-x 1 root root 4096 Sep 4 2018 home
drwxr-xr-x 1 root root 4096 Nov 27 2014 lib
drwxr-xr-x 2 root root 4096 Aug 31 2018 lib64
drwxr-xr-x 2 root root 4096 Aug 31 2018 media
drwxr-xr-x 2 root root 4096 Aug 31 2018 mnt
drwxr-xr-x 1 root root 4096 Sep 4 2018 opt
-rw-r--r-- 1 root root 595 Oct 25 10:34 package.json
dr-xr-xr-x 159 root root 0 Oct 25 10:34 proc
drwx----- 1 root root 4096 Sep 4 2018 root
drwxr-xr-x 3 root root 4096 Aug 31 2018 run
drwxr-xr-x 2 root root 4096 Aug 31 2018/sbin
drwxr-xr-x 2 root root 4096 Aug 31 2018/srv
dr-xr-xr-x 13 root root 0 Oct 25 10:34 sys
drwxrwxrwt 1 root root 4096 Sep 4 2018 tmp
drwxr-xr-x 1 root root 4096 Aug 31 2018/usr
drwxr-xr-x 1 root root 4096 Aug 31 2018/var
npm notice created a lockfile as package-lock.json. You should commit this file.
added 50 packages in 4.935s
Removing intermediate container 55daf08f8f4c
----> 69647f57bd19
Step 9/15 : RUN useradd m1wbe &&mkdir -p $workdirectory &&chown m1wbe $workdirectory
----> Running in 338aab5d85a1
Removing intermediate container 338aab5d85a1
----> 92202c2658bf
Step 10/15 : USER m1wbe
----> Running in 84b4742e0a50
Removing intermediate container 84b4742e0a50
----> cc49389c8db3
Step 11/15 : WORKDIR $workdirectory
----> Running in 33e8d34f94a5
Removing intermediate container 33e8d34f94a5

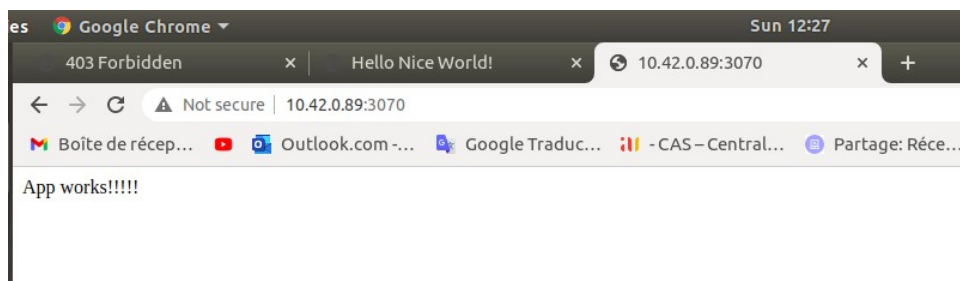
```

```
sudo docker image ls
```

- Exécuter l'image m1:

```
sudo docker container run -it -p 3070:3070 m1
```

Cette dernière instruction doit afficher l'invit de node.js dans laquelle on peut exécuter des commandes javascript. Vérifier que le port exposé est bien ouvert sur le host.



## Docker-compose

Docker-compose est un outil qui permet de gérer des ensemble de conteneurs de façon à pouvoir modéliser une application.

Commencer par faire le ménage sur votre vm :

- Stopper lesconteneurs
- Supprimer conteneurs etimages.
- 

```
root@docker:/home/miwbe# docker images ls
REPOSITORY TAG IMAGE ID CREATED SIZE
root@docker:/home/miwbe# docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PO
RTS NAMES
781ac704e97b m1 "node index.js" 5 hours ago Up 5 hours 0.
0.0.0:3070->3070/tcp cranky_kowalevski
root@docker:/home/miwbe# docker rm -f cranky_kowalevski
cranky_kowalevski
root@docker:/home/miwbe# docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PO
RTS NAMES
root@docker:/home/miwbe#
```

### Installation de dockercompose

- Exécuter les commandes :

```
sudo curl -L
"https://github.com/docker/compose/releases/download/1.27.3/docker-compose-
$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
root@docker:/home/miwbe# curl -L "https://github.com/docker/compose/releases/download/1.27.3/docker-co
mpose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total % Received % Xferd Average Speed Time Time Time Current
 Dload Upload Total Spent Left Speed
100 651 100 651 0 0 2552 0 --:--:-- --:--:-- --:--:-- 2562
100 11.6M 100 11.6M 0 0 670k 0 0:00:17 0:00:17 --:--:-- 261k
root@docker:/home/miwbe#
```

Puis :

```
sudo chmod +x /usr/local/bin/docker-compose
```

```
root@docker:/home/miwbe# sudo chmod +x /usr/local/bin/docker-compose
root@docker:/home/miwbe# docker-compose --version
docker-compose version 1.27.3, build 4092ae5d
root@docker:/home/miwbe#
```

- 
- 
- Vérifier la bonne installation de docker-compose:

```
docker-compose --version
```

La sortir attendue doit être :

```
docker-compose version 1.27.3, build 4092ae5d
```

### 5.1) Création d'une applicationcomplète

Nous allons définir une application Wordpress répartie sur 2 conteneurs :

- un conteneurmysql,
- un conteneur wordpress (serveur web + php) qui pointe vers la base mysql du précédentconteneur.
  - Créer un répertoire**m1-compose**à la racine, et y créer à l'intérieur un fichier**docker-compose.yml**

```
root@docker:~/m1-compose# nano docker-compose.yml
root@docker:~/m1-compose# docker-compose up -d
Creating network "m1-compose_default" with the default driver
Creating volume "m1-compose_db_data" with default driver
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
bb79b6b2107f: Pull complete
49e22f6fb9f7: Pull complete
842b1255668c: Pull complete
9f48d1f43000: Pull complete
c693f0615bce: Pull complete
8a621b9dbed2: Pull complete
0807d32aef13: Pull complete
f15d42f48bd9: Pull complete
098ceecc0c8d: Pull complete
b6fead9737bc: Pull complete
351d223d3d76: Pull complete
Digest: sha256:4d2b34e99c14edb99cdd95ddad4d9aa7ea3f2c4405ff0c3509a29dc40bcb10ef
Status: Downloaded newer image for mysql:5.7
Pulling wordpress (wordpress:latest)...
latest: Pulling from library/wordpress
bb79b6b2107f: Already exists
80f7a64e4b25: Pull complete
da391f3e81f0: Extracting [=====>
```

- Copier dans ce fichier les instructions suivantes ; vous décrirez dans votre rapport la signification des prédicats utilisés dans ce fichier:

```
version: '3.3'

services:
 db:
 image: mysql:5.7
 volumes:
 - db_data:/var/lib/
 mysqlrestart: always
 environment:
 MYSQL_ROOT_PASSWORD: somewordpress
 MYSQL_DATABASE: wordpress
 MYSQL_USER: wordpress
 MYSQL_PASSWORD: wordpress

 wordpress:dep
 ends_on:
 - db
 image: wordpress:latest
 ports:
 - "8000:80"
 restart: always
 environment:
 WORDPRESS_DB_HOST: db:3306
 WORDPRESS_DB_USER: wordpress
 WORDPRESS_DB_PASSWORD: wordpress
 WORDPRESS_DB_NAME: wordpress
volumes:
 db_data: {}
```

- Lancer l'application (commande à lancer dans le répertoire contenant le fichier**docker-compose.yml**):

```
docker-compose up -d
```



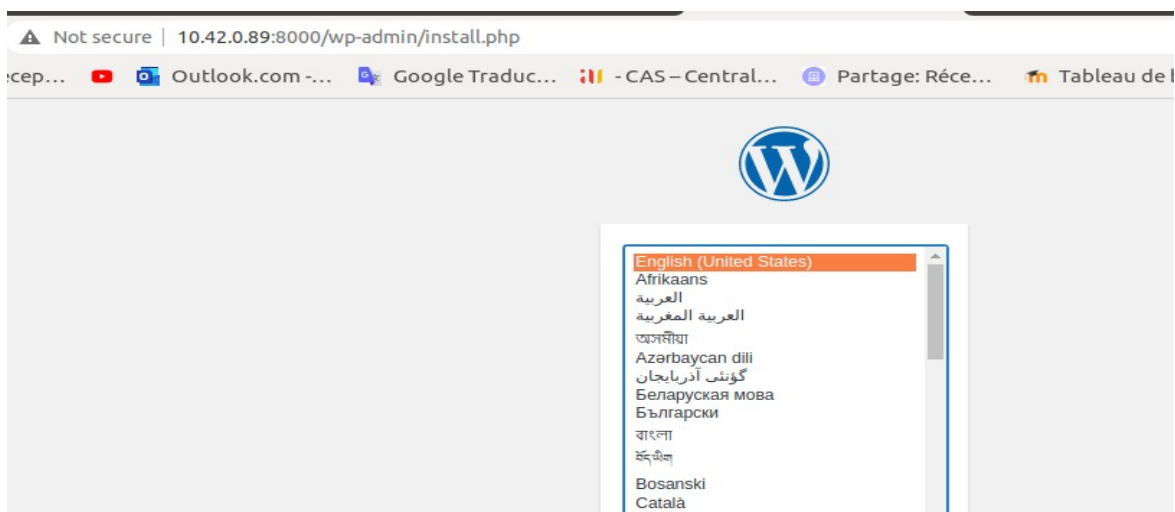
```
root@docker:~/m1-compose# nano docker-compose.yml
root@docker:~/m1-compose# docker-compose up -d
Creating network "m1-compose_default" with the default driver
Creating volume "m1-compose_db_data" with default driver
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
bb79b6b2107f: Pull complete
49e22f6fb9f7: Pull complete
842b1255668c: Pull complete
9f48d1f43000: Pull complete
c693f0615bce: Pull complete
8a621b9dbed2: Pull complete
0807d32aef13: Pull complete
f15d42f48bd9: Pull complete
098ceecc0c8d: Pull complete
b6fead9737bc: Pull complete
351d223d3d76: Pull complete
Digest: sha256:4d2b34e99c14edb99cdd95ddad4d9aa7ea3f2c4405ff0c3509a29dc40bcb10ef
Status: Downloaded newer image for mysql:5.7
Pulling wordpress (wordpress:latest)...
latest: Pulling from library/wordpress
bb79b6b2107f: Already exists
80f7a64e4b25: Pull complete
da391f3e81f0: Extracting [=====>
bc1e61f6eb18: Pull complete
033f43da5ae7: Pull complete
277905382399: Pull complete
a1b0f1f9235d: Pull complete
Digest: sha256:11ac02124f5e1d7dedd5ea1277f2a5ec2f7ec808c9969449635eddf72740a4b
Status: Downloaded newer image for wordpress:latest
Creating m1-compose_db_1 ... done
Creating m1-compose_wordpress_1 ... done
root@docker:~/m1-compose#
root@docker:~/m1-compose# docker-compose ps
```

| Name                   | Command                           | State | Ports                |
|------------------------|-----------------------------------|-------|----------------------|
| m1-compose_db_1        | docker-entrypoint.sh<br>mysqld    | Up    | 3306/tcp, 33060/tcp  |
| m1-compose_wordpress_1 | docker-entrypoint.sh<br>apach ... | Up    | 0.0.0.0:8000->80/tcp |

Vérifier la présence des deux conteneurs.

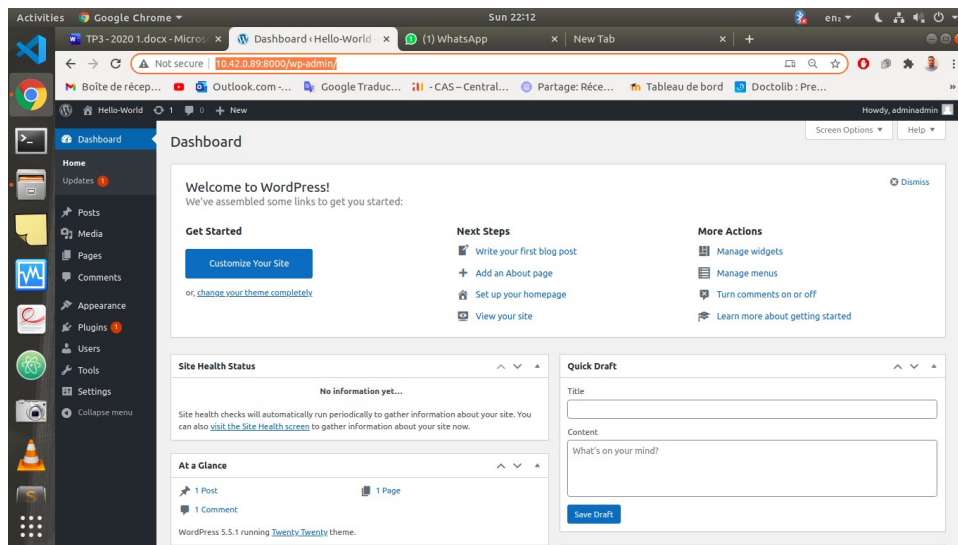
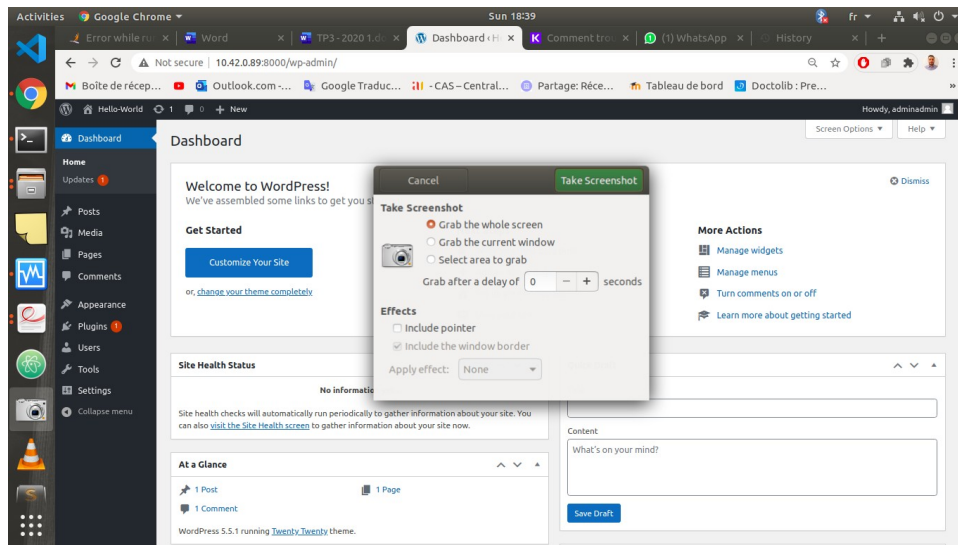
Quelle est l'url que vous devez utiliser dans un navigateur pour accéder à l'application Wordpress ?

- URI : ip adress +port 8000 . <http://10.42.0.89:8000/wp-admin/install.php>



- Connectez-vous à l'application Wordpress et finalisez l'installation.

083




chrome

file.ru | x Word | x TP3 - 2020 1.d | x WordPress | x Comment troi | x WhatsApp | x History

Not secure | 10.42.0.89:8000/wp-admin/install.php?step=1

récep... | Outlook.com -... | Google Traduc... | CAS - Central... | Partage: Réce... | Tableau de bord | Doctolib : Pre.



## Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

### Information needed

Please provide the following information. Don't worry, you can always change these settings later.

**Site Title**

**Username**   
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

**Password**    
Strong  
**Important:** You will need this password to log in. Please store it in a secure location.

**Your Email**   
Double-check your email address before continuing.

**Search engine visibility** ☐ Discourage search engines from indexing this site  
It is up to search engines to honor this request.