



**POLYTECHNIQUE  
MONTRÉAL**

UNIVERSITÉ  
D'INGÉNIERIE

**Polytechnique Montréal**

**Département de génie informatique et génie logiciel**

**Cours LOG3000  
Processus du génie logiciel**

**A2022 - Travail Pratique 4 - INFRASTRUCTURE EN TANT QUE  
CODE - Groupe 01**

**Rendu par :  
Aghilès Gasselin 2013772  
Maximiliano Falicoff 2013658  
Mohammed Fenjiro 1901744**

**Soumis à  
Varun Shiri**

**Date:  
Pour le 16 novembre 2022**

# Questions

## Questions d'analyse sur la section 3.1

1. Docker pull va puller (télécharger) l'image spécifiée depuis le registry configuré, ici c'est docker hub. Dans notre cas, on veut télécharger l'image la plus récente de l'alpine. Alpine étant un distro léger linux.

```
miameme@DESKTOP-88K4TLP:~$ docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
213ec9aee27d: Pull complete
Digest: sha256:bc41182d7ef5ffc53a40b044e725193bc10142a1243f395ee852a8d9730fc2ad
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
```

2. Lorsque l'on roule une commande docker run {commande}. Docker va lancer le container et rouler la commande spécifiée puis il s'arrête par la suite.

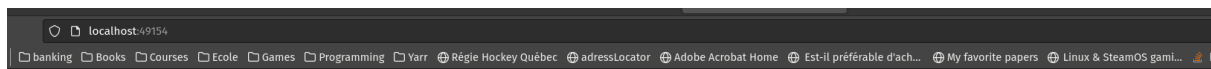
```
mazilious@mazil in ~ as 🐙 took 247ms
λ sudo docker run alpine ls -l
total 0
drwxr-xr-x  1 root    root      862 Aug  9 08:47 bin
drwxr-xr-x  5 root    root     340 Nov  3 13:49 dev
drwxr-xr-x  1 root    root     528 Nov  3 13:49 etc
drwxr-xr-x  1 root    root        0 Aug  9 08:47 home
drwxr-xr-x  1 root    root     290 Aug  9 08:47 lib
drwxr-xr-x  1 root    root      28 Aug  9 08:47 media
drwxr-xr-x  1 root    root        0 Aug  9 08:47 mnt
drwxr-xr-x  1 root    root        0 Aug  9 08:47 opt
dr-xr-xr-x 547 root    root        0 Nov  3 13:49 proc
drwx----- 1 root    root        0 Aug  9 08:47 root
drwxr-xr-x  1 root    root        0 Aug  9 08:47 run
drwxr-xr-x  1 root    root     800 Aug  9 08:47 sbin
drwxr-xr-x  1 root    root        0 Aug  9 08:47 srv
dr-xr-xr-x 13 root    root        0 Nov  3 13:49 sys
drwxrwxrwt  1 root    root        0 Aug  9 08:47 tmp
drwxr-xr-x  1 root    root      40 Aug  9 08:47 usr
drwxr-xr-x  1 root    root      86 Aug  9 08:47 var
```

```
miameme@DESKTOP-88K4TLP:~$ docker run alpine echo "hello from
alpine"
hello from
alpine
```

3. Il s'agit d'une sortie de Linux Alpine. Docker lance le container Alpine, exécute la commande echo "hello from alpine" et stop le container.
4. Une image est un 'snapshot' du système ou de l'application alors que le conteneur est le système ou logiciel qui roule concrètement.
5. Ce sont des containers très légers, qui permettent à tout le monde de créer des containers facilement et permettent d'exécuter des commandes très rapidement (avec docker run {commande}) comparé à une machine virtuelle.

## Questions d'analyse sur la section 3.2

6.



```
mazilious@mazil in ~ as 🐙 took 8ms
[❌] x sudo docker run --name static-site -e AUTHOR="mazil" -d -P dockersamples/static-site
947e0fe2a277a0ccb955007c2d4aaed68e2ad6f9565b9ebc441ee9dd2d6b3b64

mazilious@mazil in ~ as 🐙 took 175ms
λ docker run --name static-site -e
^C

mazilious@mazil in ~ as 🐙 took 175ms
λ sudo docker port static-site
443/tcp -> 0.0.0.0:49153
443/tcp -> :::49153
80/tcp -> 0.0.0.0:49154
80/tcp -> :::49154

mazilious@mazil in ~ as 🐙 took 16ms
λ _
```

On veut rouler instancier un conteneur avec le nom static-site. Le -e va passer une variable d'environnement AUTHOR="<YOUR\_NAME>" qui va ensuite être utilisé par l'application pour afficher cette variable. le -d initialise le conteneur en mode détaché, donc qui roule détaché du terminal et en background. Le -P va exposer tous les ports du conteneur à notre machine puis le dockersamples/static-site est l'image que l'on veut instancier.

## Questions d'analyse sur la section 3.3

7.

```
mazilious@mazil in ~ took 2s
[❌] x sudo docker images
[sudo] password for mazilious:
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
alpine               latest     9c6f07244728  2 months ago  5.54MB
dockersamples/static-site latest     f589ccde7957  6 years ago   190MB

mazilious@mazil in ~ as 🐙 took 1s
λ _
```

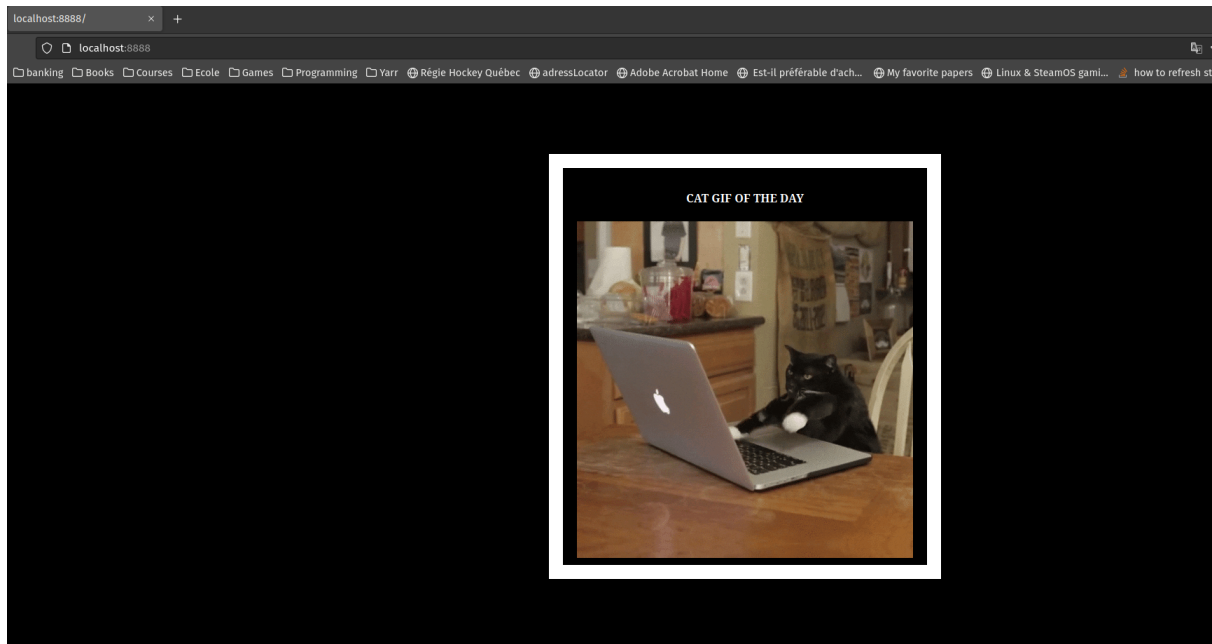
Si on veut une image autre que latest, il faut spécifier un tag spécifique, si on veut par exemple alpine avec une version de par exemple 2.3, on roule la commande docker pull alpine:2.3

8.

```
Collecting MarkupSafe>=0.23 (from Jinja2>=2.4->Flask==0.10.1->-r /usr/src/app/requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/b9/2e/64db92e53b86efccfaea71321f597fa2e1b2bd3853d8ce6585
68f7a13094/MarkupSafe-1.1.1.tar.gz
Installing collected packages: Werkzeug, MarkupSafe, Jinja2, itsdangerous, Flask
  Running setup.py install for MarkupSafe: started
    Running setup.py install for MarkupSafe: finished with status 'done'
  Running setup.py install for Flask: started
    Running setup.py install for Flask: finished with status 'done'
Successfully installed Flask-0.10.1 Jinja2-2.11.3 MarkupSafe-1.1.1 Werkzeug-1.0.1 itsdangerous-1.1.0
You are using pip version 9.0.0, however version 22.3 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Removing intermediate container a5af6915b61a
----> 85c96491f64c
Step 5/8 : COPY app.py /usr/src/app/
----> d31a63890cd0
Step 6/8 : COPY templates/index.html /usr/src/app/templates/
----> c35e28dfc0ff
Step 7/8 : EXPOSE 5000
----> Running in afe22f91e14c
Removing intermediate container afe22f91e14c
----> 2bc0ac88dbbf
Step 8/8 : CMD ["python", "/usr/src/app/app.py"]
----> Running in e1926251afec
Removing intermediate container e1926251afec
----> d625e1065fcb
Successfully built d625e1065fcb
Successfully tagged mazil007/myfirstapp:latest
```

```
mazilious@mazil in ~/Downloads/flask-app via v3.10.8 as took 14s
```

```
mazilious@mazil in ~/Downloads/flask-app via v3.10.8 as took 8ms
❯ x sudo docker run -p 8888:5000 --name flask-app mazil007/myfirstapp
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
172.17.0.1 - - [03/Nov/2022 14:22:46] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [03/Nov/2022 14:22:46] "GET /favicon.ico HTTP/1.1" 404 -
```



Une image de base est l'image sur laquelle une child image se base. Dans notre cas par exemple, notre application flask est la child image qui utilise alpine comme base image.

9. Une image officielle est une image sur le dockerhub qui sont soigneusement sélectionnés par Docker, Inc. Une image user est une image que n'importe qui avec un compte sur dockerhub peut téléverser, comme nous on pourrait téléverser notre image de notre application flask.
10. Le Dockerfile est le fichier de configuration permettant de partir d'une certaine image (commande FROM dans ce fichier) pour ensuite configurer notre nouvelle image et pour finir l'exécuter (commande CMD dans ce fichier).
11. Chaque ligne à partir de la ligne 2 va ajouter un layer sur l'image de base.
  1. On choisi l'image de base sur lequel on veut rouler notre conteneur
  2. On veut que notre image contienne pip donc on installe pip sur cette image
  3. On copie le fichier requirements.txt localement à un répertoire spécifié dans l'image.
  4. On roule la commande pip install pour avoir nos dépendances installé sur l'image.
  5. On copie notre fichier python à l'image
  6. On copie le fichier html à l'image
  7. On dit quel port notre image doit exposer lorsqu' elle roule.
  8. On écrit quelle commande doit être exécuté lorsque lors fait appelle a docker run {notre image}

12. Lorsque l'on exécute cette commande, docker va prendre notre fichier Dockerfile et exécuter chacune des lignes à l'intérieur de celui-ci afin de créer une nouvelle image docker qui sera notre application customisée.

## Questions d'analyse sur la section 3.4

13. Le Dockerfile est disponible en pièce jointe.

```
miameme@miameme in repo: CoursPoly/LOG3000/tp4/nginx-app on  master [?] via  v14.15.0 took 23ms
[⊗] * sudo docker run -it --rm -d -p 8080:80 --name web webserver
30b8d7d8e1f9e13b4e0446869e99bcd1ca5a14da0baf8a1303ae1f8ed5b5a03d

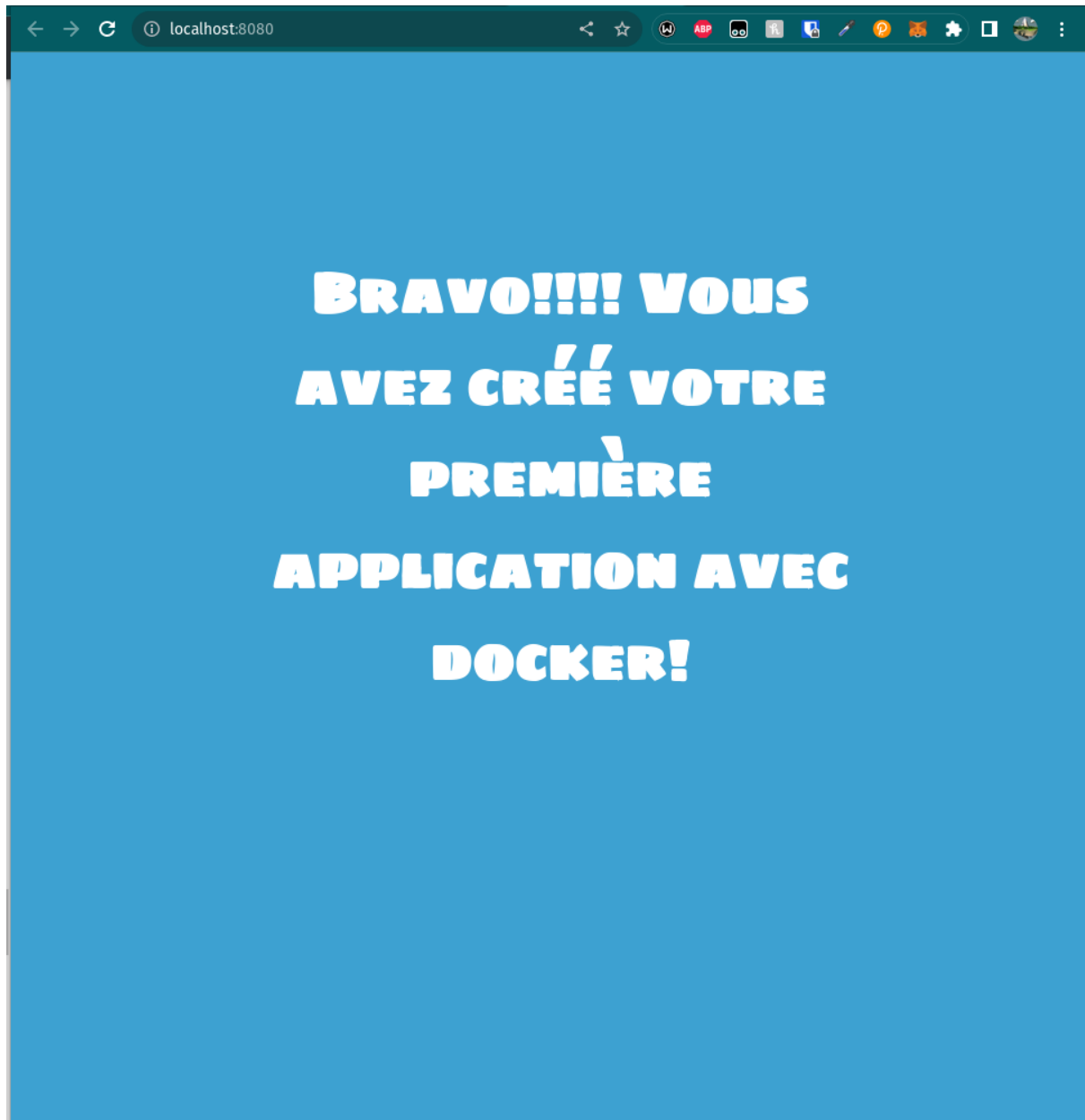
[miameme] password for miameme:
Sending build context to Docker daemon  6.656kB
Step 1/3 : FROM nginx:latest

miameme@miameme in repo: CoursPoly/LOG3000/tp4/nginx-app on  master [?] via  v14.15.0 took 3s
λ bat Dockerfile
```

	File: Dockerfile
1	FROM nginx:latest
2	COPY index.html /usr/share/nginx/html/index.html
3	COPY animation.js /usr/share/nginx/html/animation.js

```
miameme@miameme in repo: CoursPoly/LOG3000/tp4/nginx-app on  master [?] via  v14.15.0 took 21ms
λ ls
-rw-r--r-- 1.5k miameme  3 Nov 10:39 * animation.js
-rw-r--r-- 120 miameme  3 Nov 10:48 - Dockerfile
-rw-r--r-- 1.6k miameme  3 Nov 10:39  index.html
```

FROM nginx:latest : On choisi l'image de base sur lequel on veut rouler notre conteneur (en l'occurrence nginx)  
COPY index.html /usr/share/nginx/html/index.html : On copie le fichier html à l'image  
COPY animation.js /usr/share/nginx/html/animation.js : On copie le fichier javascript à l'image



### Question de rétroaction

1. Ce laboratoire nous a pris environ 4h par personnes à faire. Ce laboratoire présente une bonne charge de travail est l'effort est à un bon niveau,
2. Au début, le binôme qui était sur windows a rencontré des difficultés techniques mais lorsqu'il est passé à linux, tout a fonctionné normalement. A part cela nous n'avons pas rencontré d'autres difficultés techniques.