

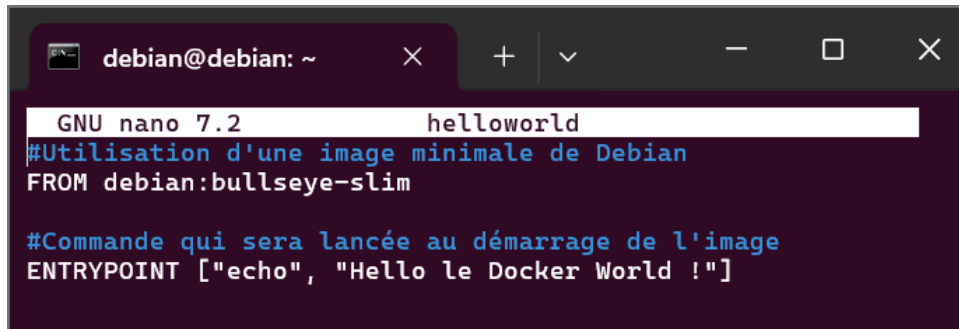
# Job3

## Job 03

---

Utilisation de « Dockerfile » pour recréer le conteneur « helloworld » depuis une image Debian minimum.

1. Création d'un dockerfile "helloworld" avec image minimale de Debian et un message à afficher



```
debian@debian: ~  
GNU nano 7.2 helloworld  
#Utilisation d'une image minimale de Debian  
FROM debian:bullseye-slim  
  
#Commande qui sera lancée au démarrage de l'image  
ENTRYPOINT ["echo", "Hello le Docker World !"]
```

#Utilisation d'une image minimale de Debian

FROM debian:bullseye-slim

#Commande qui sera lancée au démarrage de l'image

ENTRYPOINT ["echo", "Hello le Docker World !"]

PS: on j'aurais pu choisir alpine comme image légère de Linux

2. On build l'image avec : `sudo docker build -t helloworld -f helloworld` .

```
debian@debian:~$ sudo docker build -t helloworld -f helloworld .
[+] Building 0.5s (5/5) FINISHED                                docker:default
=> [internal] load build definition from helloworld            0.0s
=> => transferring dockerfile: 209B                             0.0s
=> [internal] load metadata for docker.io/library/debian:bullseye-slim 0.5s
=> [internal] load .dockerignore                               0.0s
=> => transferring context: 2B                                    0.0s
=> CACHED [1/1] FROM docker.io/library/debian:bullseye-slim@sha256:0e75 0.0s
=> exporting to image                                          0.0s
=> => exporting layers                                           0.0s
=> => writing image sha256:02c4e8c02d7050e56a9432fbf339300276232c2b1e79 0.0s
=> => naming to docker.io/library/helloworld                    0.0s
```

3. On lance l'image avec : `sudo docker run helloworld`

```
debian@debian:~$ sudo docker run helloworld
Hello le Docker World !
debian@debian:~$ |
```