

Job 1

Exécution du conteneur « Hello World »

Pour exécuter le conteneur « Hello World », utilisez la commande suivante :

```
docker run hello-world
```

```
user@SOUFIANE:~$ 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/
```

Job 2

Voici comment recréer le conteneur helloworld en utilisant un Dockerfile basé sur une image minimale Ubuntu.

1. Structure du projet

Créez un répertoire pour votre projet :

```
mkdir helloworld-ubuntu  
cd helloworld-ubuntu
```

Voici comment recréer le conteneur helloworld en utilisant un Dockerfile basé sur une image minimale Ubuntu.

1. Structure du projet

Créez un répertoire pour votre projet :

```
bash  
Copier le code  
mkdir helloworld-ubuntu  
cd helloworld-ubuntu
```

2. Créer le Dockerfile

Créez un fichier nommé Dockerfile dans ce répertoire :

```
touch Dockerfile  
  
# Utiliser une image Ubuntu comme base  
FROM ubuntu:22.04  
  
# Mettre à jour les paquets et nettoyer  
RUN apt-get update && apt-get install -y \  
    ca-certificates \  
    && apt-get clean \  
    && rm -rf /var/lib/apt/lists/*
```

```
&& rm -rf /var/lib/apt/lists/*
```

Créer un script avec un shebang

```
RUN echo '#!/bin/sh' > /usr/local/bin/helloworld && \  
    echo 'echo "Hello from Docker Container using Ubuntu!"' >>  
/usr/local/bin/helloworld && \  
    chmod +x /usr/local/bin/helloworld
```

Commande par défaut à exécuter

```
CMD ["/usr/local/bin/helloworld"]
```

3. Construire l'image

À partir du répertoire contenant le `Dockerfile`, utilisez la commande suivante pour construire l'image Docker :

```
docker build -t helloworld:ubuntu .
```

4. Exécuter le conteneur

Une fois l'image construite, lancez un conteneur basé sur cette image :

```
docker run helloworld:ubuntu
```

```

user@SOUFIANE:~/helloworld-docker$ docker build -t helloworld:ubuntu .
[+] Building 1.5s (7/7) FINISHED          docker:default
=> [internal] load build definition from Dockerfile      0.0s
=> => transferring dockerfile: 556B                      0.0s
=> [internal] load metadata for docker.io/library/ubuntu 0.9s
=> [internal] load .dockerignore                        0.0s
=> => transferring context: 2B                            0.0s
=> [1/3] FROM docker.io/library/ubuntu:22.04@sha256:0e5e 0.0s
=> => resolve docker.io/library/ubuntu:22.04@sha256:0e5e 0.0s
=> CACHED [2/3] RUN apt-get update && apt-get install -y 0.0s
=> [3/3] RUN echo '#!/bin/sh' > /usr/local/bin/helloworl 0.3s
=> exporting to image                                  0.2s
=> => exporting layers                                    0.1s
=> => exporting manifest sha256:05cfb8a088530d4c275191c6 0.0s
=> => exporting config sha256:68df6ed54af004d8842b16920a 0.0s
=> => exporting attestation manifest sha256:1e61c415fb6d 0.0s
=> => exporting manifest list sha256:5c17a697b0dce32ee49 0.0s
=> => naming to docker.io/library/helloworld:ubuntu      0.0s
=> => unpacking to docker.io/library/helloworld:ubuntu    0.0s
user@SOUFIANE:~/helloworld-docker$ docker run helloworld:ubuntu
Hello from Docker Container using Ubuntu!
user@SOUFIANE:~/helloworld-docker$ |

```

JOB 4

```

user@SOUFIANE:~$ cd ssh-docker/
user@SOUFIANE:~/ssh-docker$ ls
Dockerfile  ssh-servertest
user@SOUFIANE:~/ssh-docker$ cd ssh-servertest/
user@SOUFIANE:~/ssh-docker/ssh-servertest$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED
STATUS        PORTS      NAMES
4c007cd85697   busybox    "sh"                    41 minutes
ago          Up 41 minutes      reader
1e5e970d8a30   busybox    "sh"                    41 minutes
ago          Up 41 minutes      writer
cd9361def1b0   ssh-servertest "/usr/sbin/sshd -D"     5 hours ag
o            Up 5 hours      0.0.0.0:3333->2222/tcp  ssh-container_te
st
user@SOUFIANE:~/ssh-docker/ssh-servertest$ ssh root@localhost -p
3333
root@localhost's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 5.15.167.4-microsoft-st
andard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

This system has been minimized by removing packages and content
that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Mon Dec  9 21:12:44 2024 from 172.17.0.1
root@cd9361def1b0:~# |

```

Job 5

```
GNU nano 7.2 /home/user/.bashrc
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

# Docker command aliases
alias dps="docker ps"
alias dpsa="docker ps -a"
alias drm="docker rm"
alias drma="docker rm $(docker ps -aq)"
alias dimg="docker images"
alias drmi="docker rmi"
alias drmia="docker rmi $(docker images -q)"
alias dexec="docker exec -it"
alias dstart="docker start"
alias dstop="docker stop"
alias dbuild="docker build -t"
alias drun="docker run -it"
alias drund="docker run -d"
alias dlogs="docker logs"
alias dlogsf="docker logs -f"
alias dinsp="docker inspect"
alias dstopall="docker stop $(docker ps -q)"
alias dclean="docker system prune -f"

# Liste les cont>
# Liste tous les>
# Supprime un co>
# Supprime tous >
# Liste toutes l>
# Supprime une i>
# Supprime toute>
# Exécute une co>
# Démarre un con>
# Arrête un cont>
# Construit une >
# Lance un conte>
# Lance un conte>
# Affiche les lo>
# Suivi en temps>
# Affiche les dé>
# Arrête tous le>
# Nettoie Docker>

^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste    ^J Justify
```

Job 6

```
user@SOUFIANE:~$ docker volume ls
DRIVER      VOLUME NAME
local       partage
user@SOUFIANE:~$ docker volume inspect partage
[
  {
    "CreatedAt": "2024-12-09T21:19:00Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/partage/_data",
    "Name": "partage",
    "Options": null,
    "Scope": "local"
  }
]
user@SOUFIANE:~$ |
```