Job2
sudo docker run hello-world

Voici quelques exemples de commande utilisable avec docker
https://docs.docker.com/go/guides/
sudo usermod -aG docker ${USER}
docker ps - a (permet de lister container)
docker rm num_id (supprimer container)
docker start num_id (demarrer container)
docker stop num_id (arreter container)
docker run nom_image (créer et démarre container)
docker images (lister images)
docker rmi nom_repo (supprimer image)
docker pull nom_image (permet d'installer via Docker Hub)
docker build nom_image (permet d'installer via DockerFile)

Job3
Voici le contenu du fichier dockerfile qui permet de recréer l'image helloworld
FROM debian:buster0

CMD ["echo", "helloworld"]

Lancer la commande "docker build -t dockerfile . " afin de build l'image dockerfile. Puis la commande "docker run dockerfile"

Job4
Voici le contenu du fichier dockerfile qui permet de recréer l'image ssh
RUN apt-get update && apt-get install -y openssh-server
RUN mkdir /var/run/sshd
Set root password for SSH access (change 'your_password' to your desired password)
RUN echo 'root:your_password' | chpasswd
RUN sed -i 's/#Port 22/Port 8000/' /etc/ssh/sshd_config
RUN sed -i 's/#PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config
RUN sed 's@session\s*required\s*pam_loginuid.so@session optional pam_loginuid.so@g' -i /etc/pam.d/sshd
EXPOSE 8000
CMD ["/usr/sbin/sshd", "-D"]

Lancer la commande "docker build -t dockerfile . " afin de build l'image dockerfile. Puis la commande "docker run -d -p 8000:8000 --name dockerfile-ssh dockerfile-ssh"

Job5
Voici les alias que j'ai rajouté dans mon fichier .bashrc
alias dbu="docker build -t"
alias dim="docker images"
alias dps="docker ps"
alias dpsa="docker ps -a"
alias drm="docker rm"
alias drmi="docker rmi"
alias drun="docker run"

Job6

Le système de fichiers Docker
Une image Docker est constituée de plusieurs couches, chacune d'elles en lecture seule. Pour toute image lancée depuis un conteneur, Docker ajoute une nouvelle couche inscriptible au système. Dans Docker, ce service est connu sous le nom d'« Union File System ».

Chaque fois qu'un fichier est modifié, Docker crée une copie de celui-ci à partir des couches en lecture seule, qu'il ajoute à la couche inscriptible supérieure. Le fichier original (en lecture seule) n'est donc pas modifié.

En supprimant un conteneur, vous perdrez également sa couche inscriptible supérieure. Cela signifie que toute modification effectuée après le lancement du conteneur est alors supprimée.

Un volume de conteneur est hébergé par l'ordinateur hôte, en dehors du véritable conteneur. Pour le conteneur, le volume joue le même rôle qu'un dossier ; vous pouvez donc y stocker ou en extraire des données. Ce service est assuré à l'aide d'un « point de montage » sur l'un des répertoires de l'hôte.

Vous pouvez créer et gérer des volumes Docker de différentes manières. Chacune d'elles comporte des avantages et des inconvénients.

gestion volume docker:
volume creeate
volume inspect
volume rm

Job 7
Voici le contenu du fichier docker-compose.yml

```
services:
  nginx:
    image: nginx:latest
    network_mode: host
    volumes:
      -serveurs:/usr/share/nginx/html
    restart: always

  ftp:
    image: delfer/alpine-ftp-server:latest
    network_mode: host
    volumes:
      - serveurs:/ftp/jerome
    environment:
      - USERS=jerome|123456789
    restart: always

volumes:
  serveurs: {}
```

Pour l'exécuter, il faut lancer la commande “docker compose up -d”

Job 8
Voici le contenu du fichier dockerfile qui permet de recréer l'image nginx

```
FROM debian:bookworm
```

RUN apt-get update && apt-get upgrade -y && apt-get install nginx -y

RUN sed -i 's/listen 80 default_server;/listen 3200 default_server;/' /etc/nginx/sites-enabled/default
RUN sed -i 's/listen [::]:80_default_server;/[::]:3200_default_server;/' /etc/nginx/sites-enabled/default

EXPOSE 3200

CMD ["nginx", "-g", "daemon off;"]

Lancer la commande “docker build -t dockerfile . “ afin de build l’image dockerfile. Puis la commande “docker run -d -p 3200:3200 --name dockerfile-ssh dockerfile-ssh”

Job 10
Voici le script pour désinstaller complètement docker
#!/bin/bash
sudo apt-get purge docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin docker-ce-rootless-extras

sudo rm -rf /var/lib/docker
sudo rm -rf /var/lib/containerd

Voici un script pour installer docker de manière automatique
sudo apt-get update
sudo apt-get install ca-certificates curl -y
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y