Département Informatique
RA-4.08 Cybersécurité – TP Docker
Responsable : X.Roirand + M. Le Lain
Durée : 180 mn (sur machine)

Introduction Docker (sur 2 séances)

Le but de ce TP va être d'utiliser une technique de containérisation appelée Docker et très répandu dans le cloud ou dans le déploiement d'applications de type cloud.

Attention : certaines commandes comme le démarrage d'un container peuvent prendre parfois jusqu'a 30 secondes, voir un peu plus, soyez patient :)

1) Connection sur la machine distante ou on doit travailler

Pour se connecter sur la machine distante, il faut utiliser une connexion ssh. C'est une connexion sécurisée qui permet de ne pas faire passer le mot de passe en clair sur le réseau.

Vous pouvez utiliser le programme « putty » sur windows ou « ssh » sur linux.

La connexion se fait sur une machine dont l'adresse IP est 149.202.85.73. Pour que chaque étudiant ait sa propre machine, les ports ssh de connexion diffèrent. Lors de la mise en place du TP, l'enseignant vous a donné un numéro. Ce numéro est important car il va être utilisé pour le port ssh de connexion et le mot de passe.

Pour résumer:

IP: 149.202.85.73
port: 122<votre numéro sur 2 digits>
login: student
mot de passe est: ?XXStudent_56

ou XX est votre numéro sur 2 digits (ou 1 digit si votre numéro est inférieur à 10)

Ex:

Si votre numéro est 86 alors le port sera 12286

Une fois sur votre machine distante, vous pouvez l'administrer totalement. Vous devez absolument changer le mot de passe par défaut le plus rapidement possible.

Pour pouvoir noter le TP, il faut que vous mettiez dans un fichier nommé /root/info.txt votre nom et prénom. Attention, pas de fichier = zéro !

Tout ce qui est noté sur fond bleu … (xxx réponse à fournir)…correspond à une question à laquelle vous devez répondre dans le document de rendu.

2) Qu'est ce que Docker ?

Créez un document qui va contenir les réponses que vous allez donner au fur et à mesure de

l'avancement dans le TP (ce document sera à rendre dans la zone de rendu sur Moodle).

A la fin de la 1ère séance, vous devez rendre ce document dans la zone de rendu « Rendu séance 1 », même si vous n'avez pas fini le TP, normal puisqu'il est sur 2 séances.

A ma fin de la 2ème séance, vous devez rendre ce document dans la zone de rendu « Rendu séance 2 ».

La première question à laquelle je vous demander de répondre est « Qu'est ce que Docker » ? Idéalement je voudrais une réponse qui montre que vous avez fait des recherches et qui m'indique aussi pourquoi c'est différent de VMWare, VirtualBox ou KVM par exemple ? Aussi sur quels OS ça tourne, les limitations et en terme de sécurité est-ce que c'est bien ou pas, etc…mes indications ici ne sont pas exhaustives.
Ce qui m'intéresse aussi c'est ce que vous en avez compris, je ferais une recherche sur les phrases que vous indiquerez dans votre réponse, si c'est un copié / collé ca sera considéré comme pas de réponse.
Le temps estimé de travail pour cette partie est d'environ 15mn. (1ère réponse à fournir)

Docker est une plateforme de conteneurisation dont l'objectif principal est de permettre l'exécution d'applications dans des environnements isolés, reproductibles et portables. Docker cherche à encapsuler une application avec l'ensemble de ses dépendances afin qu'elle puisse s'exécuter de manière identique quel que soit l'environnement. Un conteneur Docker correspond essentiellement à un ou plusieurs processus isolés, exécutés sur le système hôte.

La différence entre Docker et des solutions comme VMware, VirtualBox ou KVM réside dans le niveau d'abstraction utilisé. Les solutions de virtualisation classiques reposent sur la virtualisation matérielle : chaque machine virtuelle embarque son propre noyau, son système d'exploitation complet et ses services système, le tout étant géré par un hyperviseur. Cela offre une isolation très forte, mais au prix d'un coût élevé en ressources. Docker, à l'inverse, repose sur la virtualisation au niveau du système d'exploitation : tous les conteneurs partagent le même noyau que l'hôte, ce qui les rend beaucoup plus légers et rapides à démarrer, mais aussi moins hermétiques qu'une machine virtuelle complète.

Sur le plan technique, Docker s'appuie principalement sur des mécanismes natifs du noyau Linux, notamment les namespaces pour l'isolation (processus, réseau, système de fichiers, utilisateurs) et les cgroups pour la limitation des ressources (CPU, mémoire, I/O). C'est pour cette raison que Docker est nativement conçu pour Linux. Sur macOS et Windows, Docker fonctionne indirectement via une machine virtuelle Linux légère (HyperKit, Hyper-V ou WSL2), ce qui signifie que, même sur ces systèmes, les conteneurs exécutés sont majoritairement des conteneurs Linux.

Docker présente néanmoins des limitations structurelles. Le partage du noyau impose que tous les conteneurs utilisent le même kernel, ce qui empêche, d'exécuter des applications nécessitant des versions de noyau différentes. L'accès au matériel spécifique peut également être plus complexe que dans une machine virtuelle classique.

En matière de sécurité, le partage du noyau implique qu'une vulnérabilité critique au niveau du kernel peut potentiellement affecter l'ensemble des conteneurs d'un hôte. En revanche, lorsqu'il est correctement configuré, Docker peut offrir un niveau de sécurité tout à fait satisfaisant, parfois supérieur à celui de systèmes traditionnels mal cloisonnés. La sécurité avec Docker relève donc davantage de la discipline opérationnelle que de l'outil lui-même.

3) Installation de docker :

Vous avez une machine ubuntu 18.04 qui vous est assignée et vous allez devoir installer docker dessus et vérifier qu'ensuite docker fonctionne bien, puis indiquez ce que vous avez fait pour que tout fonctionne dans le document de rendu (2ème réponse à fournir) et comment vous avez fait pour vérifier que docker fonctionne bien sur la machine (3ème réponse à fournir).

1. Vérifier la présence de docker

```
student@test:~$ docker --version

Command 'docker' not found, but can be installed with:

sudo apt install docker.io
```

Puis, d'après : https://docs.docker.com/engine/install/ubuntu/

```
student@test:~$ sudo apt remove $(dpkg --get-selections docker.io
docker-compose docker-compose-v2 docker-doc podman-docker containerd
runc | cut -f1)
dpkg: no packages found matching docker.io
dpkg: no packages found matching docker-compose
dpkg: no packages found matching docker-compose-v2
dpkg: no packages found matching docker-doc
dpkg: no packages found matching podman-docker
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no
longer required:
  aufs-tools cgroupfs-mount containerd.io libltdl7 pigz
Use 'sudo apt autoremove' to remove them.

0 upgraded, 0 newly installed, 0 to remove and 263 not
upgraded.
```

```
# Add Docker's official GPG key:

sudo apt update

sudo apt install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc
```

```
# Add the repository to Apt sources:

sudo tee /etc/apt/sources.list.d/docker.sources <<EOF

Types: deb

URIs: https://download.docker.com/linux/ubuntu

Suites: $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}")

Components: stable

Signed-By: /etc/apt/keyrings/docker.asc

EOF

sudo apt update
```

Install the Docker packages.

```
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

The Docker service starts automatically after installation. To verify that Docker is running, use:

```
student@test:~$ sudo systemctl status docker
* docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled;
vendor preset: enabled)
   Active: active (running) since Fri 2026-01-23 07:43:04 UTC; 1min
33s ago
     Docs: https://docs.docker.com
 Main PID: 7896 (dockerd)
    Tasks: 9
   CGroup: /system.slice/docker.service
           `-7896 /usr/bin/dockerd -H fd:// --
containerd=/run/containerd/containerd.sock

Jan 23 07:42:53 test systemd[1]: Starting Docker Application
Container Engine...
Jan 23 07:42:57 test dockerd[7896]: time="2026-01-
23T07:42:57.755945144Z" level=info msg="Starting up"
Jan 23 07:43:01 test dockerd[7896]: time="2026-01-
23T07:43:01.534657783Z" level=info msg="[graphdriver] using prior
storage driver: overlay2"
Jan 23 07:43:01 test dockerd[7896]: time="2026-01-
23T07:43:01.622466927Z" level=info msg="Loading containers: start."
Jan 23 07:43:03 test dockerd[7896]: time="2026-01-
23T07:43:03.428995169Z" level=info msg="Default bridge (docker0) is
assigned with an IP address 172.17.0.0/16. Daemon option --bip can be
used to set a prefe
Jan 23 07:43:03 test dockerd[7896]: time="2026-01-
23T07:43:03.519240431Z" level=info msg="Loading containers: done."
```

```
Jan 23 07:43:04 test dockerd[7896]: time="2026-01-
23T07:43:04.019941880Z" level=info msg="Docker daemon" commit=659604f
graphdriver=overlay2 version=24.0.2
Jan 23 07:43:04 test dockerd[7896]: time="2026-01-
23T07:43:04.044006035Z" level=info msg="Daemon has completed
initialization"
Jan 23 07:43:04 test dockerd[7896]: time="2026-01-
23T07:43:04.479183600Z" level=info msg="API listen on
/run/docker.sock"
Jan 23 07:43:04 test systemd[1]: Started Docker Application Container
Engine.
```

Verify that the installation is successful by running the **`hello-world`** image :

```
student@test:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest:
sha256:05813aedc15fb7b4d732e1be879d3252c1c9c25d885824f6295cab4538cb85
cd
Status: Downloaded newer image for hello-world:latest


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/
```

Pour cela cherchez sur internet mais attention, tous les packages docker ne fonctionnent pas forcément sur cette distribution là, il faut choisir le bon, donc peut-être parfois désinstaller le mauvais et réinstaller le bon.

4) Mise en place des droits pour votre utilisateur :

Pour pouvoir utiliser docker sur la machine qui vous est assignée, il faut qu'il y ait un daemon docker qui tourne, afin de…afin de quoi ? Il sert à quoi ce daemon ? (4ème réponse à fournir).

Pour utiliser Docker, il faut qu'un daemon Docker (dockerd) soit en cours d'exécution, car c'est lui qui fait réellement le travail. Les commandes Docker que l'on lance depuis le terminal ne sont que des requêtes envoyées à ce daemon ; elles n'agissent jamais directement sur le système.
Le daemon est responsable de la création, du démarrage et de l'arrêt des conteneurs. Il interagit avec le noyau Linux pour mettre en place l'isolation (namespaces), la limitation des ressources (cgroups), le réseau et le stockage. Il gère également les images Docker : téléchargement, stockage et construction à partir des Dockerfiles.
En résumé, le daemon Docker sert de moteur d'exécution : sans lui, aucune opération Docker n'est possible, car il n'y a aucun composant pour transformer les commandes de l'utilisateur en actions concrètes sur la machine.

Ce daemon docker tourne toujours en tant que "root". Donc si vous voulez pouvoir faire des commandes plus évoluées, vous serez vite confronté à un problème de droit car vos commandes demanderont une élévation de privilège (sudo). Pour éviter cela, et demander en permanence une élévation de privilège faite……Trouvez la bonne commande qui résoudra ce problème (5ème réponse à fournir).

La solution consiste à autoriser un utilisateur non-root à accéder au socket Docker en l'ajoutant au groupe docker. Une fois membre de ce groupe, l'utilisateur peut exécuter les commandes Docker sans sudo

```
student@test:~$ sudo usermod –aG docker $USER
```

Après l'exécution de cette commande, il est nécessaire de se déconnecter

5) Vérification du bon fonctionnement du daemon docker:

Trouvez la commande systeme qui permet d'afficher l'état du daemon docker et copiez/coller la commande et la sortie écran correspondante (6ème réponse à fournir).

```
student@test:~$ systemctl status docker
* docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled;
vendor preset: enabled)
   Active: active (running) since Fri 2026-01-23 07:43:04 UTC; 15min
ago
     Docs: https://docs.docker.com
 Main PID: 7896 (dockerd)
    Tasks: 10
   CGroup: /system.slice/docker.service

           `–7896     /usr/bin/dockerd     –H     fd://     --
containerd=/run/containerd/containerd.sock
```

Trouvez la commande système qui permet de stopper le daemon docker et copiez/coller la commande et la sortie écran correspondante (7ème réponse à fournir).

```
student@test:~$ sudo systemctl stop docker
[sudo] password for student:
Warning: Stopping docker.service, but it can still be activated by:
  docker.socket
student@test:~$ systemctl status docker
* docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled;
vendor preset: enabled)
   Active: inactive (dead) since Fri 2026-01-23 08:00:51 UTC; 10s
ago
     Docs: https://docs.docker.com
  Process: 7896 ExecStart=/usr/bin/dockerd -H fd:// --
containerd=/run/containerd/containerd.sock (code=exited,
status=0/SUCCESS)
 Main PID: 7896 (code=exited, status=0/SUCCESS)
student@test:~$ systemctl status docker.socket
* docker.socket - Docker Socket for the API
   Loaded: loaded (/lib/systemd/system/docker.socket; enabled;
vendor preset: enabled)
   Active: active (listening) since Fri 2026-01-23 07:42:53 UTC;
18min ago
   Listen: /run/docker.sock (Stream)
    Tasks: 0 (limit: 76956)
   CGroup: /system.slice/docker.socket
```

Trouvez la commande système qui permet de redémarrer le daemon docker et copiez/coller la commande et la sortie écran correspondante (8ème réponse à fournir).

```
student@test:~$ systemctl status docker
* docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled;
vendor preset: enabled)
   Active: active (running) since Fri 2026-01-23 08:03:09 UTC; 4s ago
     Docs: https://docs.docker.com
 Main PID: 9449 (dockerd)
    Tasks: 9
   CGroup: /system.slice/docker.service
           `-9449 /usr/bin/dockerd -H fd:// --
containerd=/run/containerd/containerd.sock
```

6) Recherche de container/images :

Maintenant que tout est bien installé, vous allez cherchez les différents images Docker des distributions suivantes :

Fedora
Alpine
OpenSuse

On vous demande juste de chercher, pas d'installer.

Indiquez dans le document de rendu la commande utilisée et l'output obtenu (9ème réponse à fournir).

```
student@test:~$ docker search fedora
NAME                     DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
fedora                   Official Docker builds of Fedora                1264      [OK]
srcml/fedora             Build, package, and test srcml on Fedora        0
fedora/apache                                                            34                   [OK]
fedora/nginx                                                             19                   [OK]
fedora/ssh                                                               22                   [OK]
fedora/tools             Docker image that has systems administration…   3                    [OK]
fedora/mariadb                                                           23                   [OK]
fedora/cockpitws                                                         0                    [OK]
fedora/memcached                                                         19                   [OK]
fedora/couchdb                                                           33                   [OK]
fedora/firefox                                                           24                   [OK]
fedora/rabbitmq                                                          18                   [OK]
fedora/systemd-systemd                                                   25                   [OK]
fedora/qpid                                                              20                   [OK]
smartentry/fedora        fedora with smartentry                          0                    [OK]
ppc64le/fedora           Official Docker builds of Fedora                1
amd64/fedora             Official Docker builds of Fedora                0
teachable/fedora                                                         0
arm64v8/fedora           Official Docker builds of Fedora                2
s390x/fedora             Official Docker builds of Fedora                0
fedora/postgresql        postgresql                                      3                    [OK]
fedora/python                                                            20                   [OK]
arm32v7/fedora           Official Docker builds of Fedora                8
fedora/earthquake                                                        17                   [OK]
fedora/redis                                                             19                   [OK]
student@test:~$ docker search opensuse
NAME                     DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
opensuse                 DEPRECATED; for current images by the openSU…   332       [OK]
srcml/opensuse           Development environment for srcml on OpenSUS…   0
opensuse/leap            Official openSUSE Leap Images                   105
opensuse/tumbleweed      Official openSUSE Tumbleweed images             84
opensuse/portus          Authorization service and frontend for Docke…   97                   [OK]
opensuse/clair           CoreOS Clair image based on openSUSE. Has th…   0                    [OK]
opensuse/python          openSUSE base image with python                 0                    [OK]
opensuse/archive         Archive of the old opensuse images from the …   3
opensuse/amd64           x86_64 releases of openSUSE Docker images       8
gmao/opensuse                                                            0
opensuse/registry        Docker registry based on openSUSE.              1                    [OK]
opensuse/ruby            Ruby Docker images based on openSUSE Leap 42…   3                    [OK]
opensuse/nailed          Collect and visualize product related data f…   2                    [OK]
opensuse/etcd            etcd image                                      0                    [OK]
opensuse/s390x           Experimental s390x images of openSUSE           0
opensuse/salt-master     salt-master image                               2                    [OK]
opensuse/salt-minion     salt-minion image                               0                    [OK]
opensuse/salt-api        salt-api image                                  0                    [OK]
fispact/opensuse         Automated build of openSUSE images with the …   0                    [OK]
raccos/opensuse                                                          0
robertdebock/opensuse    Container to test Ansible roles in, includin…   0
```

```
amd64/opensuse          DEPRECATED; for current images by the openSU…   0
buluma/opensuse                                                         1
rkrahl/opensuse         Slightly modified version of the openSUSE im…   1                    [OK]
hachque/opensuse        A base openSUSE 42.1 image, built on SUSE St…   4
student@test:~$ docker search alpine
NAME                DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
alpine              A minimal Docker image based on Alpine Linux…   11447     [OK]
alpine/git          A  simple git container running in alpine li…   250                  [OK]
alpine/socat        Run socat command in alpine container           115                  [OK]
alpine/curl                                                         12
alpine/helm         Auto-trigger docker build for kubernetes hel…   70
alpine/k8s          Kubernetes toolbox for EKS (kubectl, helm, i…   65
alpine/httpie       Auto-trigger docker build for `httpie` when …   21                   [OK]
alpine/bombardier   Auto-trigger docker build for bombardier whe…   28
alpine/terragrunt   Auto-trigger docker build for terragrunt whe…   18
alpine/openssl      openssl                                         7
alpine/kubectl      Kubernetes command-line tool for managing cl…   1
alpine/flake8       Auto-trigger docker build for fake8 via ci c…   2
alpine/psql         psql - The PostgreSQL Command-Line Client       4
alpine/ansible      run ansible and ansible-playbook in docker      35
alpine/jmeter       run jmeter in Docker                            9
alpine/semver       Docker tool for semantic versioning             4                    [OK]
alpine/java         Repo containing the build scripts to produce…   5
alpine/mongosh      mongosh - MongoDB Command Line Database Tools   2
alpine/xml          several xml tools to work on xml file as jq …   2
alpine/mysql        mysql client                                    7
alpine/bundle       This repository has been archived by the own…   1
alpine/gcloud       Auto-trigger docker build for gcloud (google…   0
alpine/cfn-nag      Auto-trigger docker build for cfn-nag when n…   0
alpine/crane                                                        0
alpine/dfimage      reverse Docker images into Dockerfiles          70
```

Trouvez à quel endroit ces images sont stockées sur votre système de fichier avec la commande locate (elle n'est peut-être pas installée…). Utilisez des mots clés tels que docker ou container dans la commande, de plus, il faut voir comment fonctionne la commande locate car elle a besoin d'une 1ère phase avant d'être exploitable.

ATTENTION : les répertoires dans lesquels sont stockés les images étant accessibles uniquement à certains utilisateurs, il faut faire la commande préfixé de « sudo ». D'ailleurs à quoi sert cette commande (10ème réponse à fournir) ? Indiquez dans le document de rendu les commandes utilisées et l'output obtenu (11ème réponse à fournir) pour trouver l'endroit ou les images sont stockées.

La commande locate permet de rechercher rapidement des fichiers ou répertoires sur le système à partir d'une base de données, et non par un parcours en temps réel du système de fichiers. Cette base n'est pas mise à jour automatiquement ; elle doit être générée ou rafraîchie avant la première utilisation. Sans cette étape, locate ne retourne soit rien, soit des résultats obsolètes.

```
student@test:~$ sudo apt install locate
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  locate
0 upgraded, 1 newly installed, 0 to remove and 256 not upgraded.
Need to get 52.3 kB of archives.
After this operation, 181 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/universe amd64 locate amd64 4.6.0+git+20170828-2 [52.3
kB]
Fetched 52.3 kB in 0s (139 kB/s)
Selecting previously unselected package locate.
(Reading database ... 25908 files and directories currently installed.)
Preparing to unpack .../locate_4.6.0+git+20170828-2_amd64.deb ...
Unpacking locate (4.6.0+git+20170828-2) ...
Setting up locate (4.6.0+git+20170828-2) ...
Processing triggers for man-db (2.8.3-2) ...
student@test:~$ sudo updatedb
student@test:~$ sudo locate docker
… très long
student@test:~$ sudo locate /var/lib/docker/vfs
/var/lib/docker/vfs
/var/lib/docker/vfs/backingFsBlockDev
/var/lib/docker/vfs/dir
/var/lib/docker/vfs/dir/02fd6db4c3d2b3fc54f548c42816fe09ba9085ebf93845e986ede4c9a68edb17
/var/lib/docker/vfs/dir/5a53df8400a7658fe4ced4845022f7c9af3b963d8ec43a96dba83d57de9557a6
/var/lib/docker/vfs/dir/6e1970cdfa16fe0f088faa28ff5f73dafac53773513c62d9de6438f0f6cb8b38
/var/lib/docker/vfs/dir/828e28f108199a59365584abc3228a6d7d173d660c209a97ecc9d66ca4ddd87b
/var/lib/docker/vfs/dir/9b33161efc7dd5ec838ea11ed1633f926edb206163a7e417e9886f5f69f7de46
/var/lib/docker/vfs/dir/9b33161efc7dd5ec838ea11ed1633f926edb206163a7e417e9886f5f69f7de46-init
…
/var/lib/docker/vfs/dir/aff00b66817f89d091c37768a4dbe0ef64235aa42de850dd67347a09efdc98a2-
init/var/log/wtmp
/var/lib/docker/vfs/dir/aff00b66817f89d091c37768a4dbe0ef64235aa42de850dd67347a09efdc98a2-
```

```
init/var/log/apt/eipp.log.xz
/var/lib/docker/vfs/dir/aff00b66817f89d091c37768a4dbe0ef64235aa42de850dd67347a09efdc98a2-
init/var/log/apt/history.log
/var/lib/docker/vfs/dir/aff00b66817f89d091c37768a4dbe0ef64235aa42de850dd67347a09efdc98a2-
init/var/spool/mail
student@test:~$ sudo du -h --max-depth=1 /var/lib/docker/vfs
483M  /var/lib/docker/vfs/dir
483M    /var/lib/docker/vfs
```

7) **Manipulation de container:**

Au départ aucune image n'existe en local sur votre machine, il faut donc en télécharger une et l'exécuter. Prenez la distribution « alpine » et trouvez la bonne commande pour lancer l'instance docker correspondante, et constatez que lorsque vous lancez le container, sans paramètre supplémentaire, il vous rend la main rapidement. Mettez cette commande et la sortie écran produite lorsque vous avez rentré cette commande dans le document de rendu (12ème réponse à fournir).

```
student@test:~$ docker run alpine
student@test:~$ docker ps -a
CONTAINER ID   IMAGE         COMMAND     CREATED          STATUS                      PORTS     NAMES
3033e449eb3f   alpine        "/bin/sh"   9 seconds ago    Exited (0) 7 seconds ago              quirky_hoover
```

En effet, une image Docker est juste un environnement dans lequel il va exécuter la ou les commandes que vous lui donnez, si vous n'en donnez aucune, alors il lance l'image, puis va finir l'exécution, puisqu'il n'y a rien à exécuter. En général les images ont un point d'entrée (de démarrage) mais pour les images simples, basées sur des distributions, ça n'est pas le cas.

~~Lancer une image docker CentOS sans rien exécuter dans le container docker. Mettez cette commande et la sortie écran produite lorsque vous avez rentré cette commande dans le document de rendu (13ème réponse à fournir).~~

On va faire la même chose mais la commande à exécuter va être un shell interactif, ce qui permettra, d'être en live dans l'image docker et de faire des choses. Pour lancer un docker avec un shell de manière interactive, trouvez une commande unique (pas avec 2 commandes, et il faut mettre des options) qui permet de le faire. L'image à exécuter est une image de la distribution Alpine (c'est moi qui veut ça).

Mettez cette commande et la sortie écran produite lorsque vous avez rentré cette commande dans le document de rendu (14ème réponse à fournir).

```
student@test:~$ docker run -it alpine /bin/sh

/ #
/ # ls
bin    dev    etc    home   lib    media  mnt    opt    proc   root   run    sbin   srv    sys    tmp    usr    var
/ #
```

On va tester que la partie interactive fonctionne bien et qu'on arrive bien à utiliser des fonctionnalités qui ne sont pas sur la machine qu'on vous met à disposition mais que les dockers peuvent bien apporter cette fonctionnalité.

Pour chaque élement ci-dessous, vérifiez que votre machine en fournit pas la fonctionnalité/language mais qu'un docker peut le faire. Affichez l'output de la vérification/commande sur votre machine et pareil dans le docker. (15ème réponse à fournir).

Ex avec Java :

Sur la machine : java -help

Command 'java' not found, but can be installed with:

apt install default-jre

apt install openjdk-11-jre-headless

apt install openjdk-8-jre-headless

```
student@test:~$ java -version

Command 'java' not found, but can be installed with:

sudo apt install default-jre
sudo apt install openjdk-11-jre-headless
sudo apt install openjdk-8-jre-headless
sudo apt install openjdk-9-jre-headless
```

Avec le bon docker en interactif :

root@05e34a9c317d:/# java -h
Usage: java [-options] class [args...]
(to execute a class)
or java [-options] -jar jarfile [args...]
(to execute a jar file)
where options include:
-cp <class search path of directories and zip/jar files>
[…]

```
student@test:~$ docker run -it eclipse-temurin:17 /bin/bash
Unable to find image 'eclipse-temurin:17' locally
17: Pulling from library/eclipse-temurin
a3629ac5b9f4: Pull complete
f72f0f1ec9ea: Pull complete
3404c492510b: Pull complete
682d1a3197c6: Pull complete
c940cc503ed4: Pull complete
Digest: sha256:710bbe5d41a4c48ecd1e8d5be5f05d49132a102ab70961006d0675ed8b387d86
Status: Downloaded newer image for eclipse-temurin:17
root@79cbf5f46792:/# java -version
openjdk version "17.0.17" 2025-10-21
OpenJDK Runtime Environment Temurin-17.0.17+10 (build 17.0.17+10)
OpenJDK 64-Bit Server VM Temurin-17.0.17+10 (build 17.0.17+10, mixed mode, sharing)
root@79cbf5f46792:/# exit
exit
student@test:~$
```

Voici les éléments :

- Python3 (permet d'avoir une version différente très facilement)

```
student@test:~$ python3 --version

Python 3.6.5
student@test:~$
student@test:~$ docker run -it python:3 /bin/bash
Unable to find image 'python:3' locally
3: Pulling from library/python
2ca1bfae7ba8: Pull complete
82e18c5e1c15: Pull complete
be442a7e0d6f: Pull complete
26d823e3848f: Pull complete
0a23b95910e1: Pull complete
3096782d2500: Pull complete
a53307d2fabc: Pull complete
Digest: sha256:17bc9f1d032a760546802cc4e406401eb5fe99dbcb4602c91628e73672fa749c
Status: Downloaded newer image for python:3
root@c9319588a4f7:/# python3 --version
Python 3.14.2
root@c9319588a4f7:/# exit
exit
student@test:~$
```

- Nodejs

```
student@test:~$ node --version

Command 'node' not found, but can be installed with:

sudo apt install nodejs

student@test:~$ docker run -it node /bin/bash
Unable to find image 'node:latest' locally
latest: Pulling from library/node
c1be109a62df: Pull complete
64538a062a61: Pull complete
fd1872fa12cc: Pull complete
4925cf9d8be8: Pull complete
a338e2f0574c: Pull complete
35ab4dc6aef4: Pull complete
2aca9f293459: Pull complete
880ae2fb2e5c: Pull complete
Digest: sha256:e6b32434aba48dcb8730d56de2df3d137de213f1f527a922a6bf7b2853a24e86
Status: Downloaded newer image for node:latest
root@213541065b62:/# node --version
v25.5.0
root@213541065b62:/# exit
exit
student@test:~$
```

- Tcl (voir indicateur %)

```
student@test:~$ tclsh

Command 'tclsh' not found, but can be installed with:

sudo apt install tcl

student@test:~$ docker run -it alpine /bin/sh
/ # apk add --no-cache tcl
(1/2) Installing tzdata (2025c-r0)
(2/2) Installing tcl (8.6.17-r0)
Executing busybox-1.37.0-r30.trigger
OK: 12.6 MiB in 18 packages
/ # tclsh
%
%
% exit
/ # exit
student@test:~$
```

- PHP

```
student@test:~$ php -v

Command 'php' not found, but can be installed with:

sudo apt install php7.2-cli
sudo apt install hhvm

student@test:~$ docker run -it php /bin/bash
Unable to find image 'php:latest' locally
latest: Pulling from library/php
119d43eec815: Pull complete
8dcc69194de6: Pull complete
93b931ac6d0e: Pull complete
fc0ff79a09c6: Pull complete
56feb94ffbb6: Pull complete
ad54ff4d476b: Pull complete
63482a9a6690: Pull complete
c0aa4f6243d4: Pull complete
603bfbff4470: Pull complete
Digest: sha256:40bb9e411d6f8b852365c56ce3bb665a4e58d37f3165db0a6d7139b8523affe2
Status: Downloaded newer image for php:latest
root@913a015613fe:/# php -v
PHP 8.5.2 (cli) (built: Jan 16 2026 23:13:28) (NTS)
Copyright (c) The PHP Group
Built by https://github.com/docker-library/php
Zend Engine v4.5.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.5.2, Copyright (c), by Zend Technologies
root@913a015613fe:/# exit
exit
student@test:~$
```

- Lua

```
student@test:~$ lua -v

Command 'lua' not found, but can be installed with:

sudo apt install lua5.1
sudo apt install lua5.2
sudo apt install lua50

student@test:~$ docker run -it nickblah/lua /bin/sh
Unable to find image 'nickblah/lua:latest' locally
latest: Pulling from nickblah/lua
2ca1bfae7ba8: Already exists
4666e3fe4d97: Pull complete
Digest: sha256:14ce3fc9f228b1e8dca8532d5d28fa840c0026a25e851f7629413b447b9faedd
Status: Downloaded newer image for nickblah/lua:latest
# lua -v
Lua 5.5.0  Copyright (C) 1994-2025 Lua.org, PUC-Rio
# exit
student@test:~$
```