# Je vérifie que j’ai compris :

* Nécessite un ordinateur physique pour fonctionner
* Nécessite un hôte sous linux
* Un conteneur est un processus

## Images et Conteneurs

### Images

Pour créer un conteneur, il faut une image.

Image = modèle qui sert de base de départ pour la création d’un conteneur.

Assez proche d’une décompression d’un fichier zip.

Une image sert à créer des conteneurs qui peuvent être aussi nombreux qu’on le souhaite.

“docker pull ‘image’“

“docker run ‘image’“

Q1)

Oui, on voit le processus tictac.

11859 : PID processus tictac.

165536 est l’utilisateur qui a lancé tictac.

Q2)

Non on ne peut pas.

L’expérience montre que le conteneur utilise son environnement d’exécution.

### Exploration rapide d’un conteneur

“docker container clock which tictac“

**Which** 🡪 localiser une commande

Q3)

Le conteneur a affiché “/bin/tictac“.

Non, on ne voit pas la commande tourner.

Non, le conteneur dans T2 ne fonctionne plus, on peut en déduire que le conteneur a été kill.

* Evaporation de conteneur ?

La commande **which tictac** à bien arrêtée le conteneur après exécution.

Quand un processus se termine, il n’apparait plus dans la liste d’un ps.

Un conteneur est un processus.

“docker container ps“ 🡪 liste les conteneurs actifs et non les processus actifs.

Q4) Cette commande a affichée le ps du docker donc il y a une ligne avec l’image, la commande, le nombre de minutes depuis le moment où il a été lancé.

Voici les lignes :

● CONTAINER ID : un identiﬁant unique du conteneur (on y revient plus loin).

● IMAGE : l’image qui a servi de modèle pour créer le conteneur.

● COMMAND : la commande exécutée dans le conteneur, c’est-à-dire soit celle conﬁgurée par défaut dans l’image, soit celle passée en remplacement sur la ligne du docker container run.

● CREATED : âge du conteneur.

● STATUS : état actuel du conteneur (Up signiﬁe qu’il est actif, en cours d’exécution).

On y revient plus loin.

● PORTS : vu plus tard.

● NAMES : un petit nom unique pour chaque conteneur. On peut le spéciﬁer à la création du conteneur, ou laisser le hasard faire les choses.

“docker container ps -a “

Q5)

La commande a affichée toute la liste des containers lancée dans la machine.

“which tictac “ apparait dans la commande.

### Arrêt d’un conteneur

Q6)

On ne peut pas arrêter le processus au clavier.

“docker container stop <id>“

Q7)

Le conteneur dans le terminal 1 s’est arrêtté.

### Redémarrage d’un conteneur

Q8)

Le conteneur a redémarré.

* Logs d’un conteneur :

“docker container logs -f <id>“

“docker container exec -ti <id> sh“ 🡪 permet de devenir root du conteneur.

Q10)

FAIT

Q11)

Tictac

Sh

Ps -a

Q12)

On voit 2 tictac.