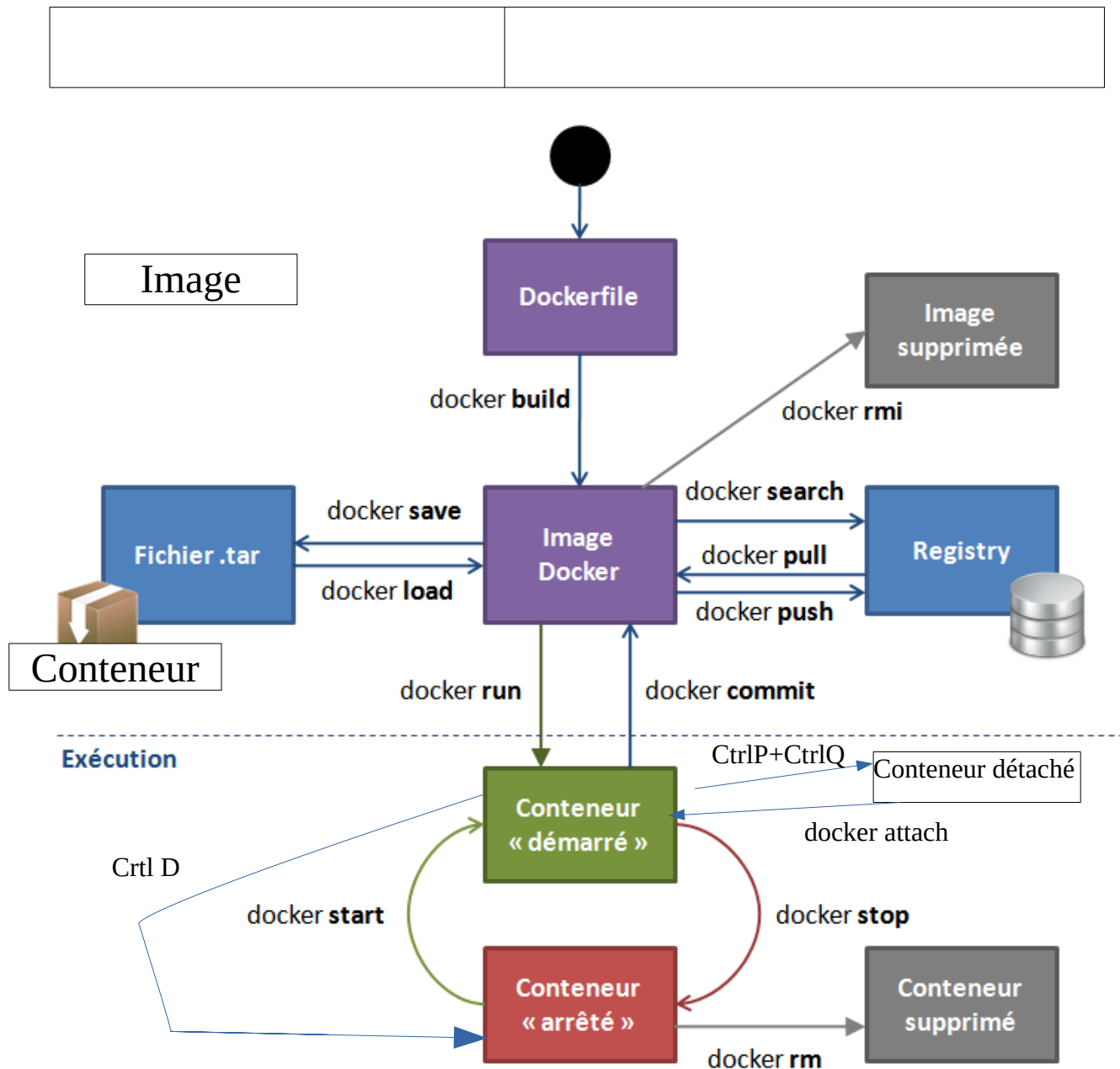


## Docker aide mémoire

docker <b>images</b>	liste des images du cache machine (-a pour toutes les images mêmes les images non actives)
docker <b>search</b> --stars=5 samba	Recherche des images "samba" avec 5 étoiles
docker <b>pull</b> ubuntu:latest	Récupération d'une image dans sa dernière version
docker run ubuntu	lancement d'un <b>nouveau</b> conteneur
docker start/stop ubuntu	<b>Après un "docker run..."</b>
docker ps	Liste des conteneurs actifs (-a avec les inactifs)
Ctrl D	Quitte et arrête le conteneur==> docker start conteneur+ <b>docker attach conteneur</b> (pour revenir dans le conteneur)
Ctrl P+ Ctrl Q	Quitte sans arrêter le conteneur ==> docker attach conteneur (pour revenir dans le conteneur)
docker <b>commit</b> nom_conteneur nouveau_nom_de_l'image	sauvegarder les changements d'un conteneur arrêté ou en cours de fonctionnement dans une nouvelle image
docker <b>cp</b> scripts.tar.gz nomconteneur:/rep_conteneur	<i>Copie de l'hôte locale vers un conteneur</i>
docker run -d --name mysql --expose 3306 imagemysql	<i>1/ ...on expose le port 3306</i>
docker run -d --name Apache2 -P --link mysql:mysql imageApache conteneur-mysql:nom-libre pour le nommage des variables	<i>2/...Apache2 se lie au conteneur mysql</i>
docker build -t nom/image :1.0 .	<i>compilation...ne pas oublier le point à la fin.</i>
docker rm \$(docker ps -a -q)	<i>supprime tous conteneurs</i>
<b>docker exec -it conteneur commande</b>	<i>Exécute une commande à partir de l'hôte sur le conteneur</i>
docker inspect....conteneur	<i>paramètres du conteneur (nom, ip, service...)</i>
docker run ...-p port1:port2...image	<i>port1 : machine physique port2 :conteneur</i>
docker save -o image_save.tar image_a_sauvegarder	Sauvegarde d'une image
docker load -i image_save.tar	Restauration d'une image



## Définition des variables d'environnement

### Exemple#

```
$ docker run -e "ENV_VAR=foo" ubuntu /bin/bash
```

Les deux `--env -e` et `--env` peuvent être utilisées pour définir des variables d'environnement à l'intérieur d'un conteneur. Il est possible de fournir de nombreuses variables d'environnement en utilisant un fichier texte:

```
$ docker run --env-file ./env.list ubuntu /bin/bash
```

Exemple de fichier de variable d'environnement:

```
# This is a comment
TEST_HOST=10.10.0.127
```