

# Exercices corrections

## Accéder le shell dans un conteneur

```
docker run --interactive --tty alpine
```

## Utiliser des commandes dans un conteneur

```
docker run --interactive --tty --rm alpine:3.19
docker run -it --rm alpine:3.19
find / | wc -l
```

## Utiliser l'image officielle de PHP et installer des modules complémentaires

Tout est expliqué dans la [documentation](#) de l'image.

Pour récupérer le git, on peut :

- utiliser Docker Desktop en explorant les fichiers du conteneur
- utiliser la variante apache de l'image PHP
- monter l'export HTTP dans un répertoire local à MS Windows via bind mount

## Une base de données

### Charger les données

```
docker run `
  --detach `
  --volume ./data:/docker-entrypoint-initdb.d:ro `
  --env MARIADB_ALLOW_EMPTY_ROOT_PASSWORD=Y `
  --publish 3306:3306 `
  mariadb
```

### Rendre les données persistentes

```
docker run `
  --detach `
  --volume refuge:/var/lib/mysql:rw `
```

```
--volume ./data:/docker-entrypoint-initdb.d:ro `
--env MARIADB_ALLOW_EMPTY_ROOT_PASSWORD=Y `
--publish 3306:3306 `
    mariadb
```

## Faire communiquer les conteneurs

```
$Env:db_cid = docker run `
    --detach `
    --volume refuge:/var/lib/mysql:rw `
    --volume ./data:/docker-entrypoint-initdb.d:ro `
    --env MARIADB_ALLOW_EMPTY_ROOT_PASSWORD=Y `
    --publish 3306:3306 `
    mariadb
$Env:db_ip = docker exec -it $Env:db_cid hostname -i
docker exec -it $Env:db_cid mariadb -e "create user 'lord'@'%'"
identified by ''"
docker exec -it $Env:db_cid mariadb -e "grant all privileges on
refuge.* to 'lord'@'%'"
$Env:web_cid = docker run `
    --detach `
    --env DB_HOST=$Env:db_ip `
    --env DB_NAME=refuge `
    --env DB_USER=lord `
    --env DB_PASS= `
    --publish 80:80 `
    --volume ./public:/var/www/html:ro `
    php:8.2-apache
docker exec -it $Env:web_cid bash
```

```
docker-php-ext-configure mysqli &&
docker-php-ext-install -j$(nproc) mysqli &&
kill -HUP 1
```