# Containers

1. Launch a postgres database with docker

docker run --name some-postgres -e POSTGRES\_PASSWORD=mysecretpassword -d postgres



1. Log in to the database and create a table with the following SQL statement

docker exec -it some-postgres psql -U postgres

CREATE TABLE users (firstname VARCHAR(50) NOT NULL UNIQUE, lastname VARCHAR(50) NOT NULL UNIQUE);

1. Exit and stops the postgres container.

docker stop some-postgres

1. In your opinion, is the data lost?

non

1. Display all information about the postgres container, where is the data stored?

docker inspect some-postgres

1. Restart the container and check the data.

la data est la

1. Delete the postgres container. In your opinion, is the data lost?

docker rm some-postgres

the data is lost

1. Restart the postgres container with the following constraints:
   * the database must be accessible on the host on port 2345
   * the database must be named devops
   * the connection identifiers must be admin / foo123
   * the container has a memory limit of 50 MB
   * the container cannot consume more than one CPU;

docker run --name devops -e POSTGRES\_PASSWORD=foo123 -e POSTGRES\_USER=admin -d -p 2345:5432 --memory=50m --cpus=1 postgres

# Image

1. Launch an alpine container in its latest version with an interactive shell.

docker run -it alpine:latest sh

1. Install the curl command using apk and test it with curl [www.google.com](http://www.google.com)

apk update && apk add curl

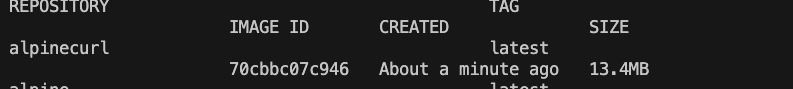
curl google.com

1. Create an image using the instructions you used in the alpine image to install curl.
2. FROM alpine:latest
3. RUN apk update && apk add curl
4. CMD ["/bin/sh"]

docker build -t alpinecurl .

1. List your images and check that it is present

docker image ls



1. Test the installation of your image in a single command line.

docker run -it alpinecurl curl google.com

# Volumes

In this exercise, we will learn how to manipulate volumes.

1. As the exercise 1, launch a postgres database. The postgres database should have:

* persistent data
* the database must be named devops

1. Create a table and add some data
2. Delete the container and recreate it.
3. In your opinion, is the data lost?

NOOOOO

For information about the postgres database container, see the dockerhub documentation page.

# Docker Compose

The objective of this exercise is to start up a postgres database and an administration interface with adminer.

1. Create a docker-compose file with :
   * two services (postgres, admirer)
   * connect it in the same network
   * use a volume to store postgres data
2. Create a new database and a new table with random data.
3. Connect with bash command and display this data