Cloud Computing and Visualization: TD6

# Part 1:

Since we need to use our previous app, we create the same Dockerfile into a new directory.

Graphical user interface, text, application

Description automatically generated

Screenshot 1: Dockerfile content

Then we modify the app.py, in order to get the content of the file benefits.txt.

Graphical user interface, text, application, email

Description automatically generated

Screenshot 2: app.py

We remove all the database calls from the previous exercise and define the function get\_benefits() and call in into index after the route.

NB: I forget remove the unused libraries !!!

Now, we still have to build the image TD6 using the following command: docker build –t td6bis .

To run the image and bind mount a local file we need to us the option -v and we use -p to specify the port : docker run -v Desktop/ESILV/Cloud\ Computing\ and\ Visualization/TDs/TD6 -p 5001:5001 td6bis

Graphical user interface, text, application

Description automatically generated

Screenshot 3: Connection to the dev sever

Graphical user interface, application, Word

Description automatically generated

# Part 2:

Here the first command is : docker volume create mydatabase. It creates a Docker volume called mydatabase. We can then use this volume to persist the data in our database container. Then we run: docker run -v mydatabase:/data/db mongo. It will tell Docker to create a container running the mongo image, and to mount the mydatabase volume to the /data/db directory inside the container. Any data written to the /data/db directory will be stored in the mydatabase volume.

Text

Description automatically generated

Screenshot 4: Volume creation

# Part 3:

For this part, we need to create a docker-compose.yaml file.

Graphical user interface, text, letter

Description automatically generated

Screenshot 5:docker-compose.yaml file content

* The app service builds the image from the current directory, sets the port to 5001, mounts the benefits.txt file as a read-only volume, sets the DATABASE\_URI environment variable to mongodb://db:27017/mydatabase, and depends on the db service.
* The db service uses the official mongo image and creates a named volume called mongodb to store the database data.

Graphical user interface, text, application, email

Description automatically generated

Screenshot 6: Dockerfile

We keep the same app.py in the part 1.

Then, we run: docker-compose up -d

This command starts the services defined in the **docker-compose.yaml** file with the background option **-d** .

Graphical user interface, application, Word

Description automatically generated