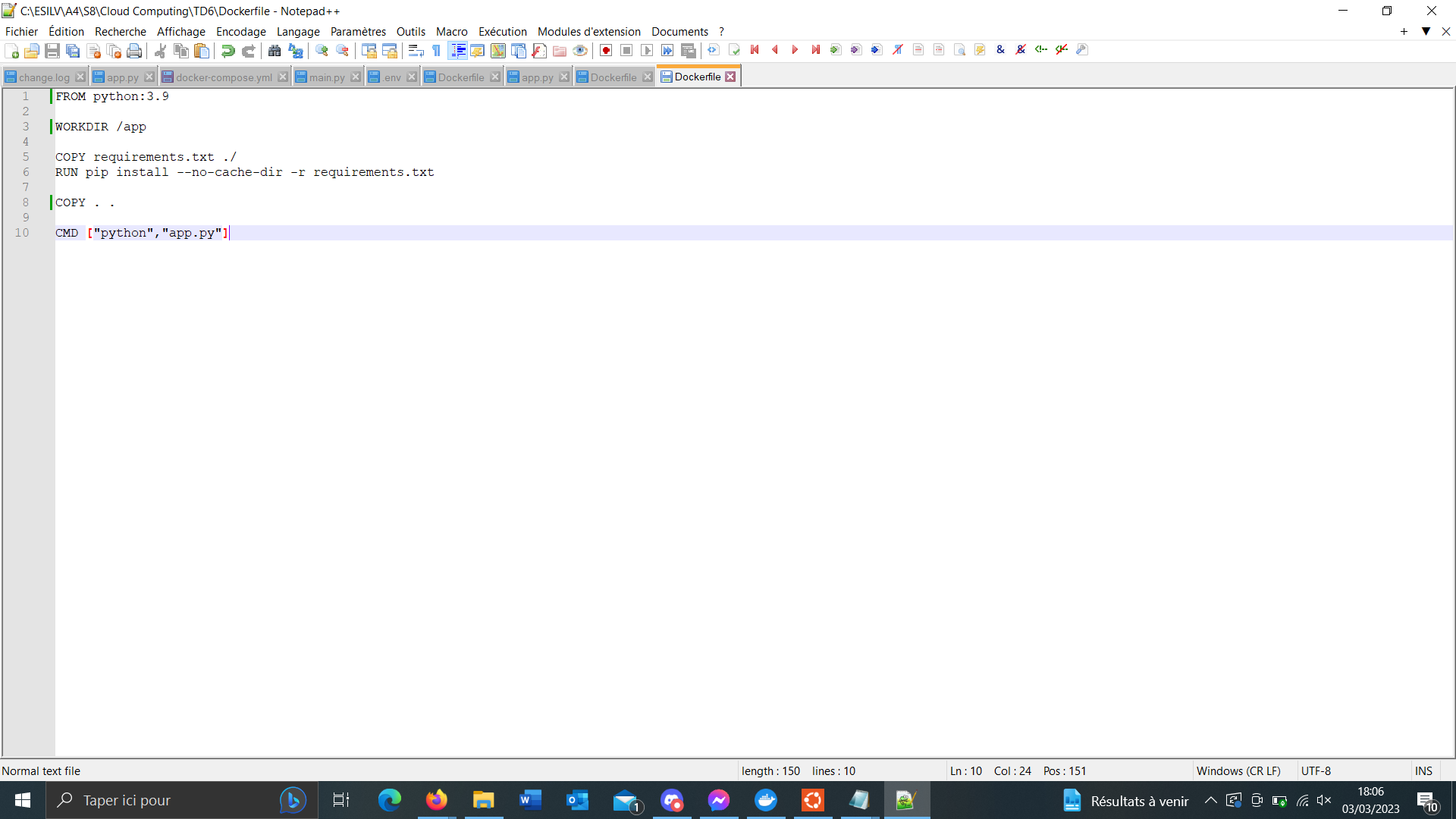
TD6

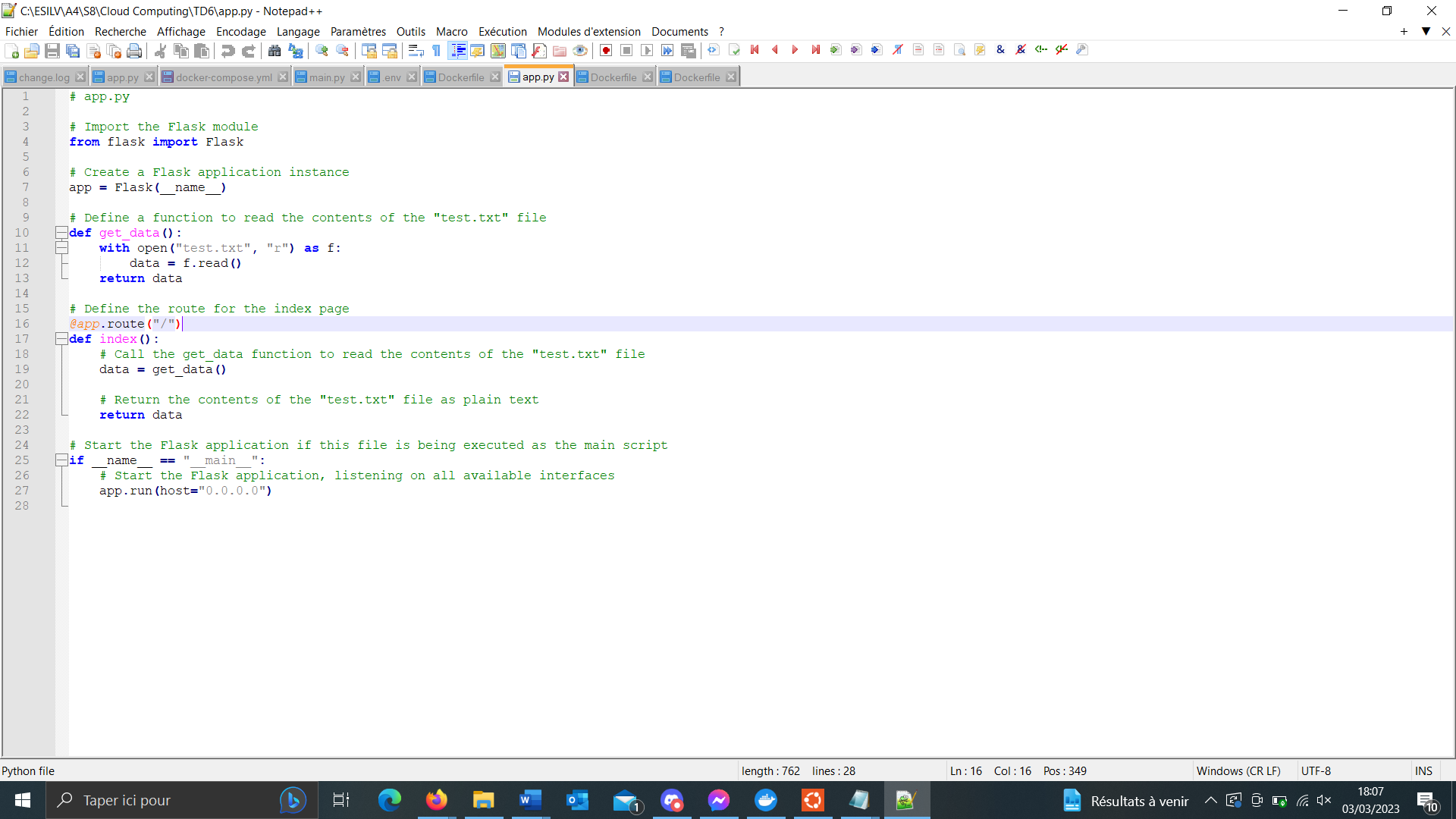
Step 1:

Add to your application a feature, which reads a **text file on your host** and shows the content on a web page. **Any changes in the content of this file** should be shown by refreshing the web page. (by using a bind mount)

For that we’ll create a Dockerfile to build the web image.



Here’s the app.py that will open “test.txt”



Then we go on Ubuntu, we build the image thanks to the Dockerfile.

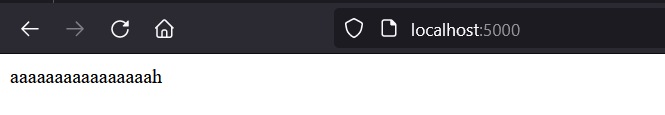


Then here’s the important part: we do our docker run to run our container but we add -v <path of my local test.txt: path to my text.txt in my container> to bind mount and be able to modify the text displayed.

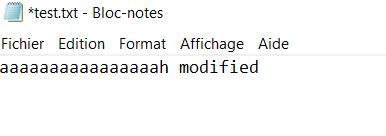
-d is for the detached part so that we can still use our terminal.

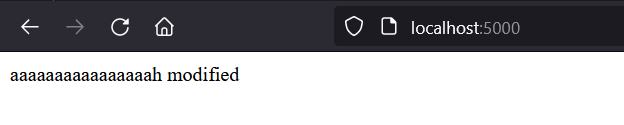
-p is for the port

Once we go on localhost:5000 (which was the port we defined), we can see the text displayed from test.txt



Once we modify it and refresh the webpage, the changes are applied.





Step 2:

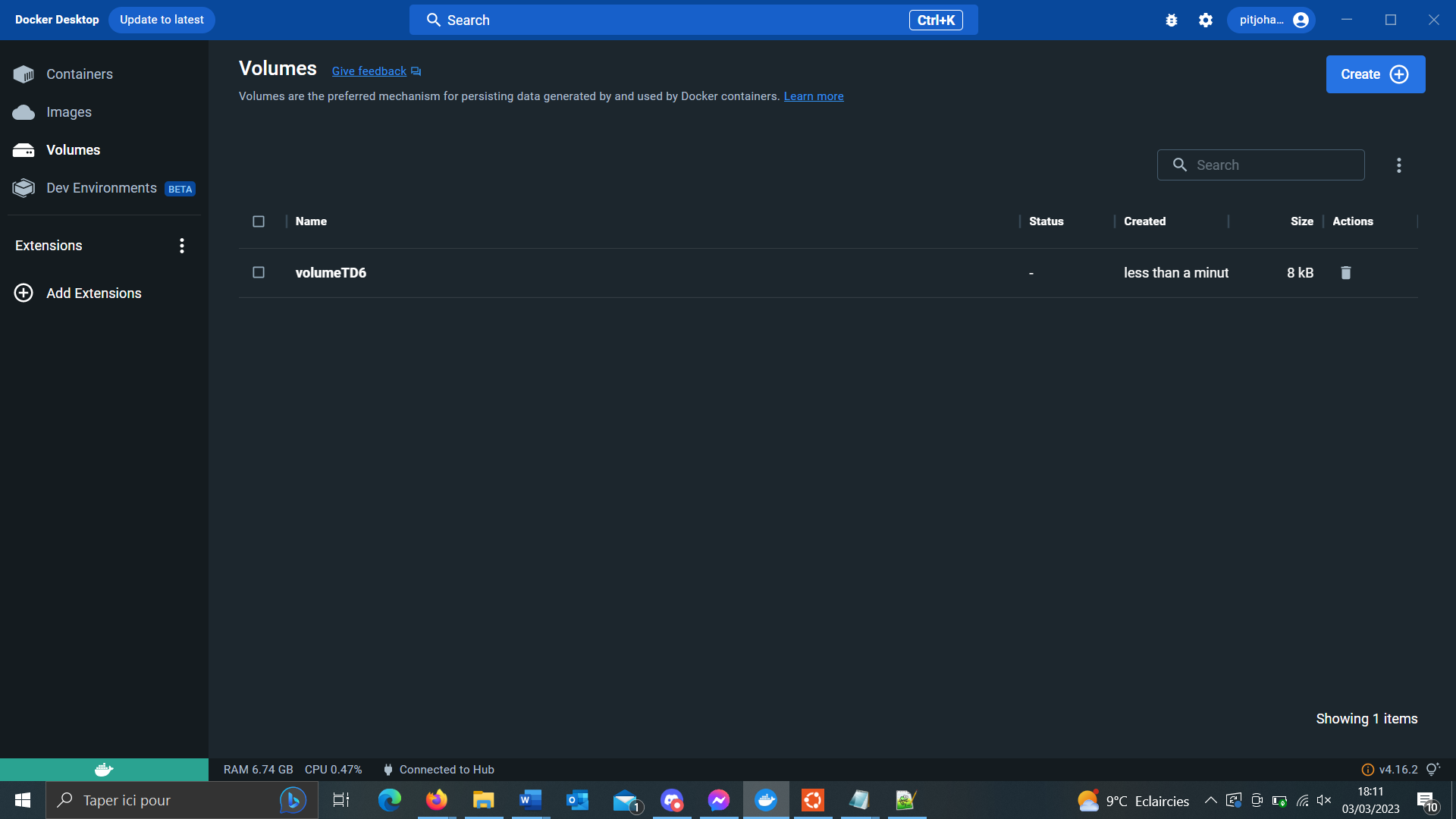
Make your database persistent. (by using Volume)

Explain how you can migrate it! How to share it with another instance of the same database engine.

On crée un volume



On peut le voir sur Docker Desktop



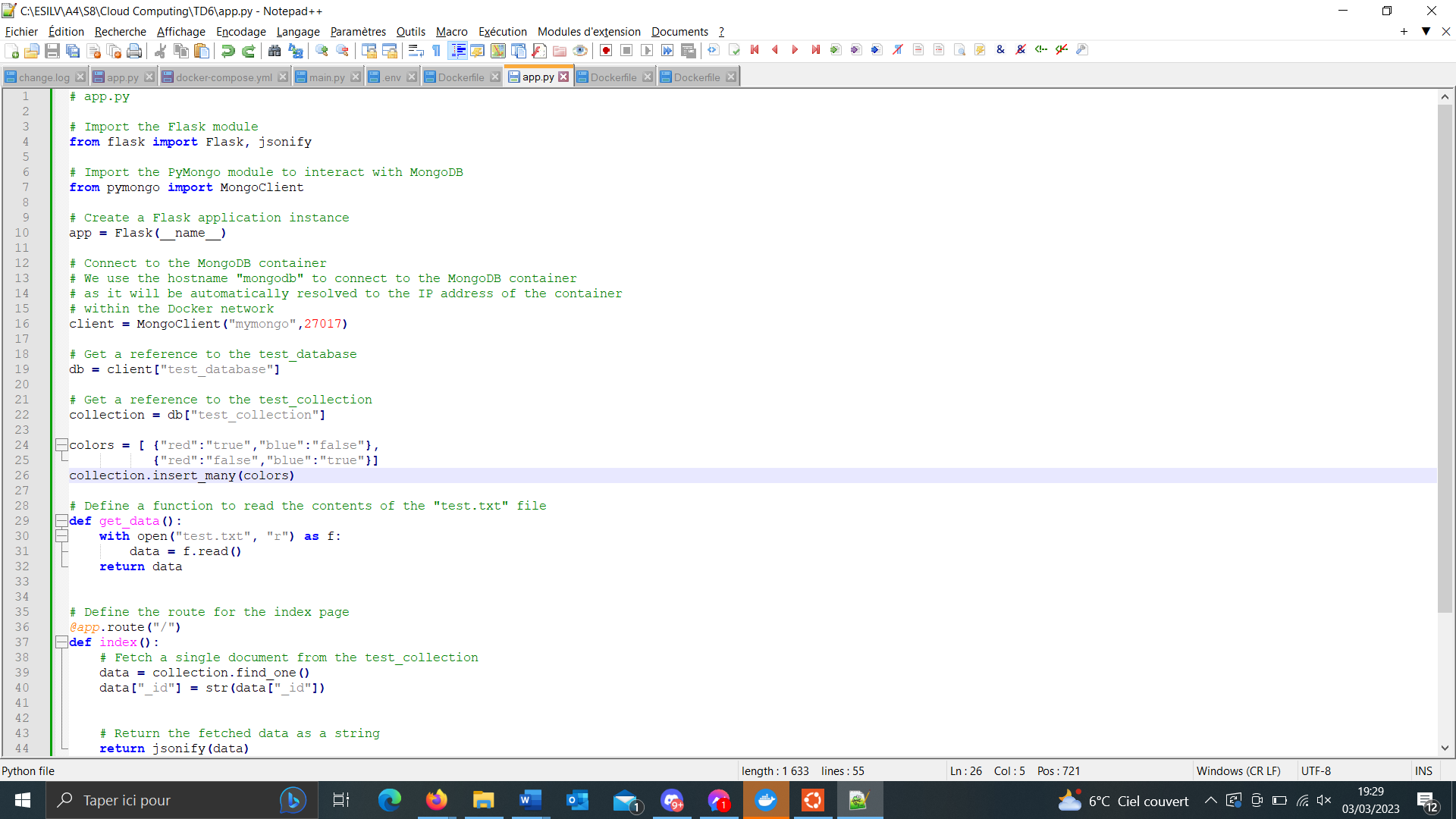
On crée le network

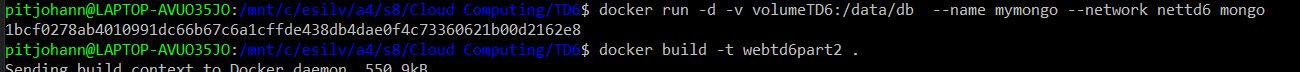


On run le container mongo en indiquant l’utilisation du volume avec la commande :

-v <volume Name : database place in the container>

Puis on build le container pour le web (même que partie 1 sauf pour le app.py)

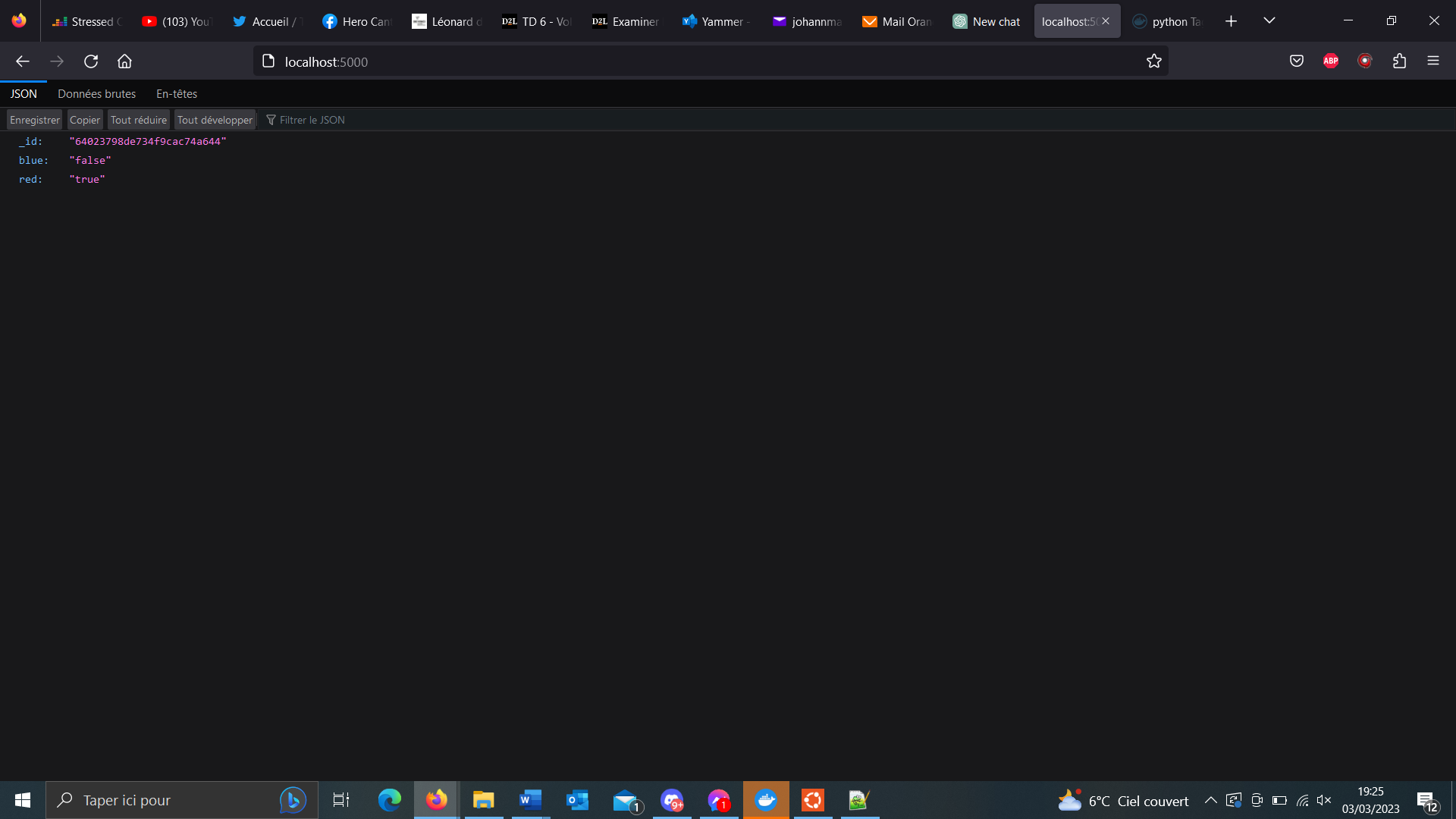




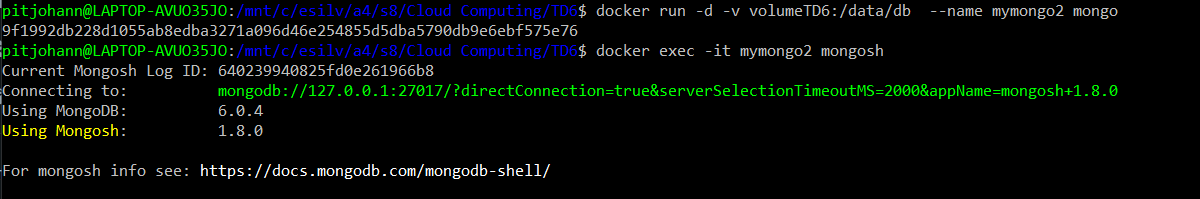
Puis on run ce container sur le même network

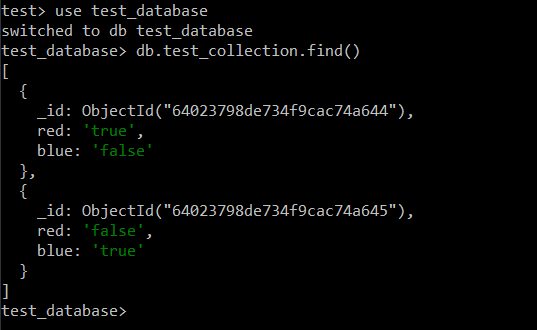


Sur le port 5000, ça fonctionne !



Maintenant on teste sur un autre container mongo et on voit qu’on a accès à la database précédemment créée !

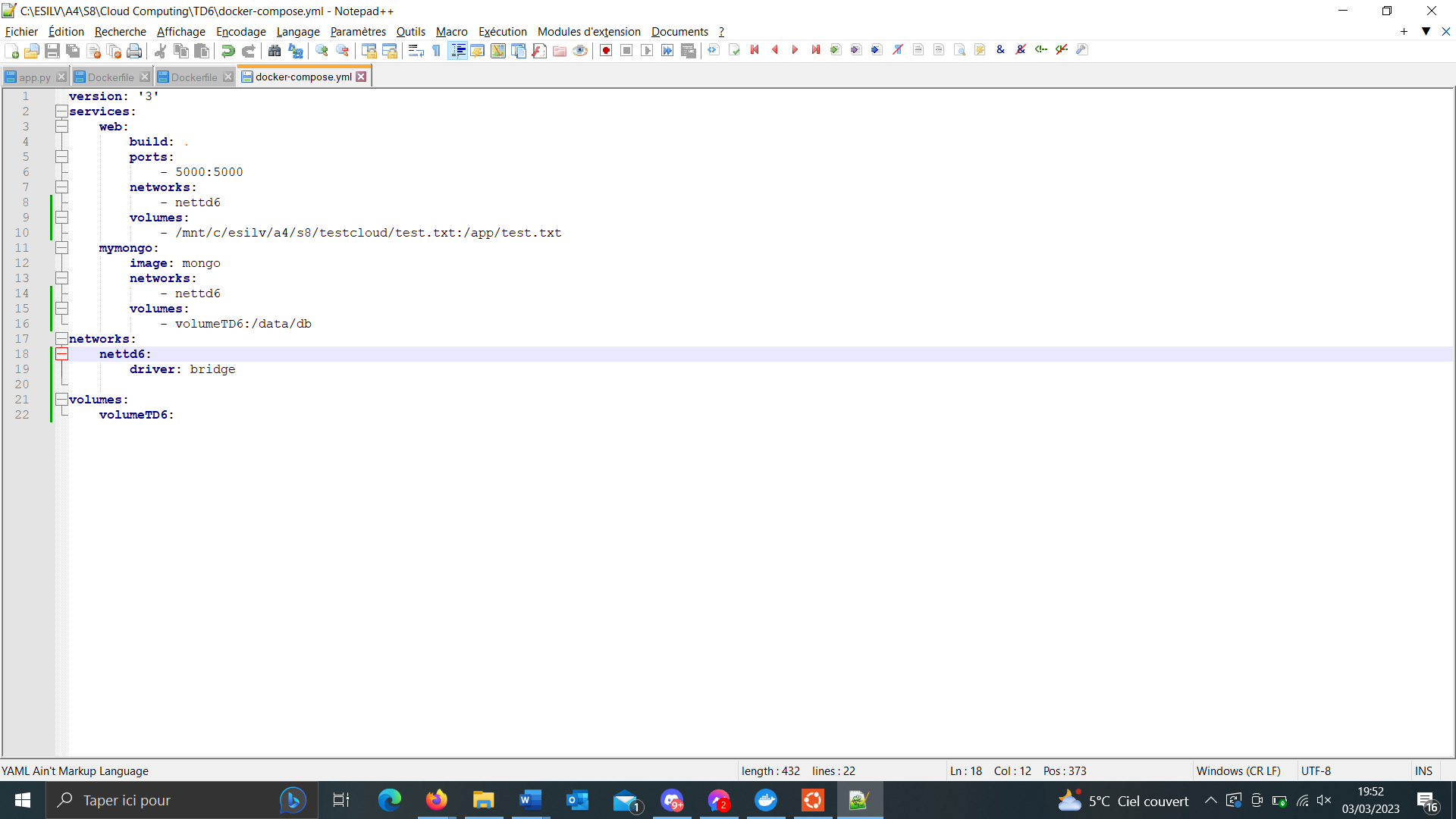




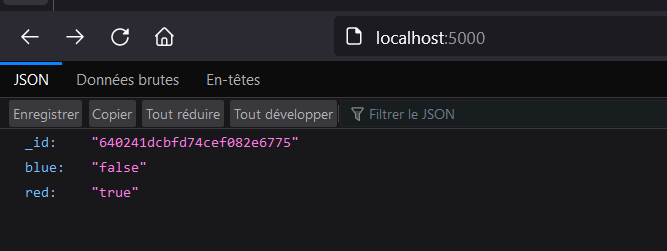
**Part 2: By using docker-compose**

Do both steps by using docker-compose and show your docker-compose.yaml and its changes!

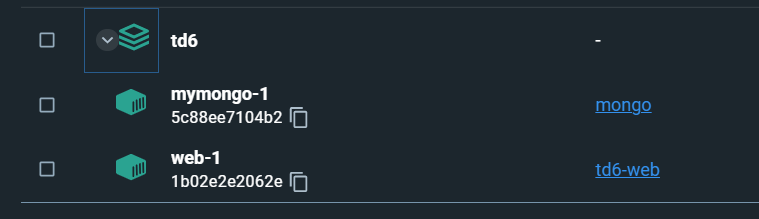
Voici le docker-compose.yml avec le volume et le bind mount.



Ensuite on run le docker compose 



To verify if it worked with volume we check the name of the now created mongo container.



And we check that the data is inside.

