

# Rapport sur le Projet Docker

Déploiement d'un Site Web  
Statique avec Docker

école iT  
LES FUTURS TALENTS DU NUMÉRIQUE

Réaliser Par :

Jules NGNIPIBEO

&

OUEDRAOGO Ramata Dite Martine

# Introduction

## Contexte du projet

Dans le cadre de ce projet, l'objectif était de développer et déployer un site web statique personnel en utilisant Docker. Le choix de Docker permet de comprendre l'importance de la conteneurisation dans le déploiement d'applications, même pour des sites simples. Ce projet nous a permis de combiner des compétences en développement web et en gestion de conteneurs Docker pour offrir une solution déployable et facilement transportable.

## Objectif du projet

Le but était de créer un site web statique personnel, de le conteneuriser dans Docker, et de le déployer à l'aide d'un serveur web léger. Le projet visait également à comprendre les concepts fondamentaux de Docker, comme la création d'une image Docker via un Dockerfile, la gestion de conteneurs, et l'exposition d'un service web.

## Choix du site

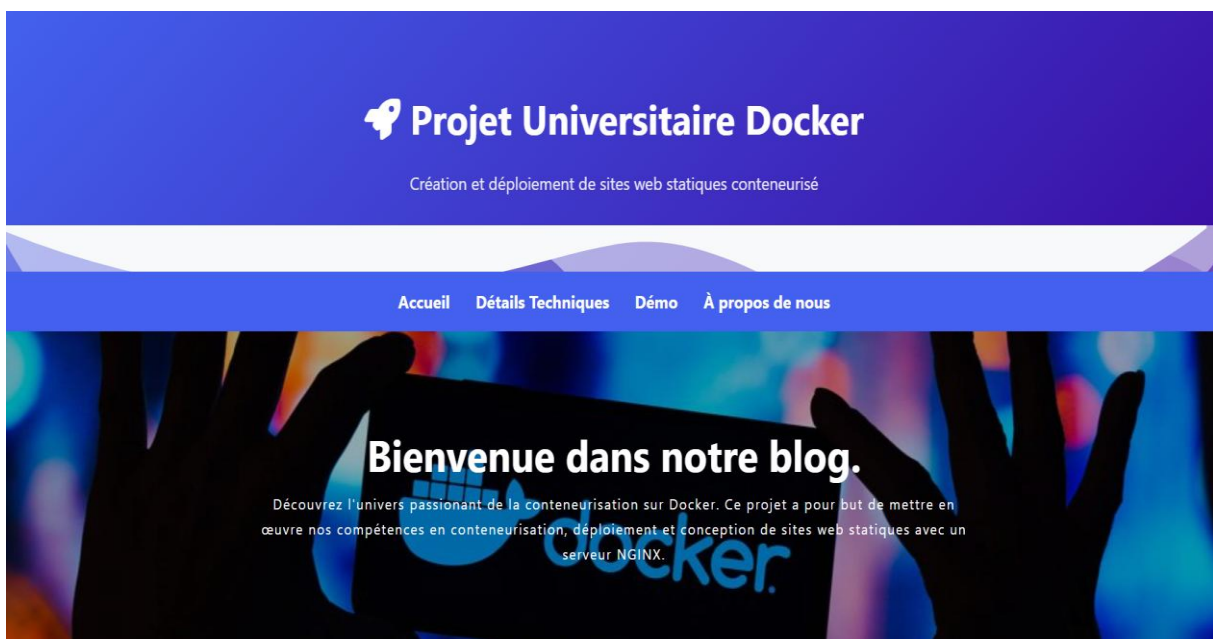
Le sujet du site web étant libre et personnel, nous avons fait le choix de réaliser un site qui présente un projet universitaire mettant en avant notre parcours et nos compétences. L'idée était de concevoir un site portfolio commun à chacun, permettant de valoriser nos compétences personnelles et académiques.

Notre site adopte une approche moderne, épurée et structurée, avec les sections suivantes :

- **Page d'accueil** : introduction au projet, avec une explication synthétique de Docker et du processus de conteneurisation.
- **Détails techniques** : description du Dockerfile, de la configuration du serveur web, des étapes de déploiement, etc.

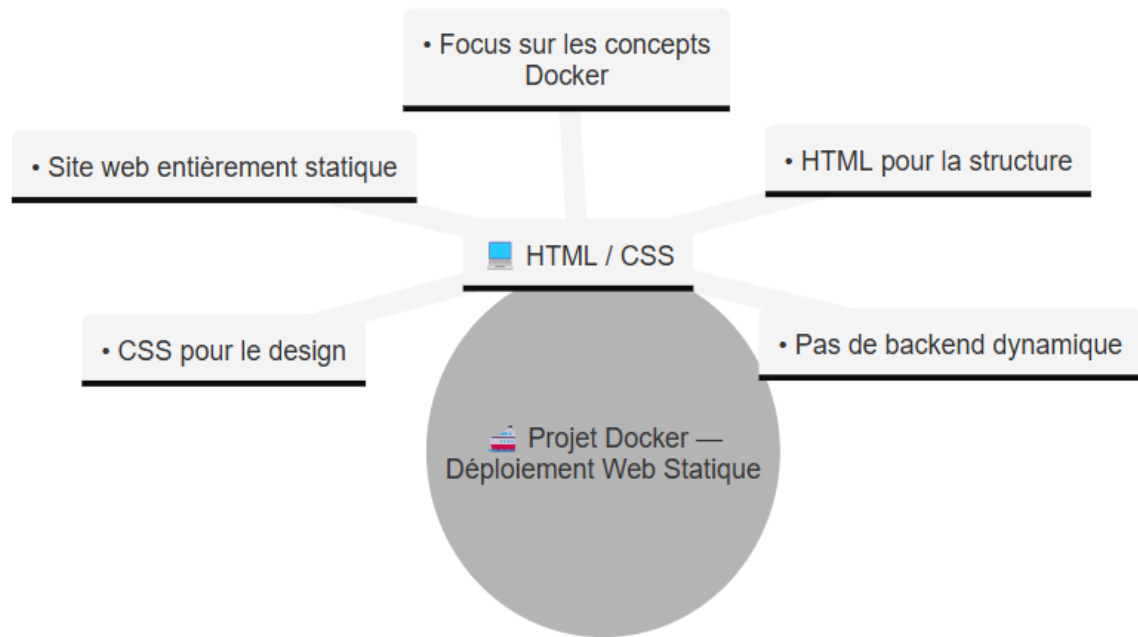
- **Démo** : aperçu du site en fonctionnement, avec capture d'écran et lien vers la version déployée (le cas échéant).
- **À propos de nous** : liens vers nos profils professionnels (GitHub, LinkedIn), nos projets antérieurs et une brève présentation de chacun.

Ce choix s'est imposé naturellement, car il nous permettait non seulement de répondre aux consignes du projet (développement d'un site statique dans Docker), mais aussi de créer un outil concret et réutilisable dans un cadre professionnel ou académique. En résumé, le site remplit une double fonction : support d'apprentissage technique et carte de visite numérique pour nos futurs projets ou candidatures.

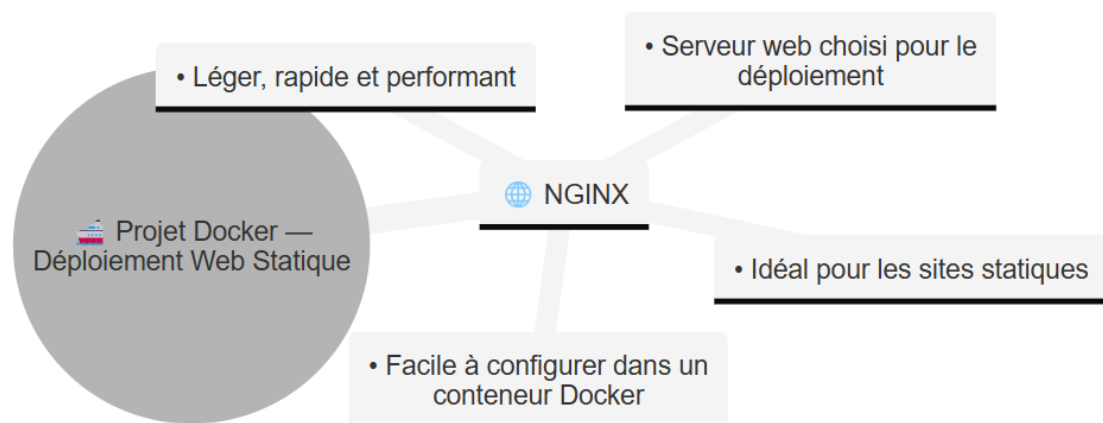


## Technologies Utilisées

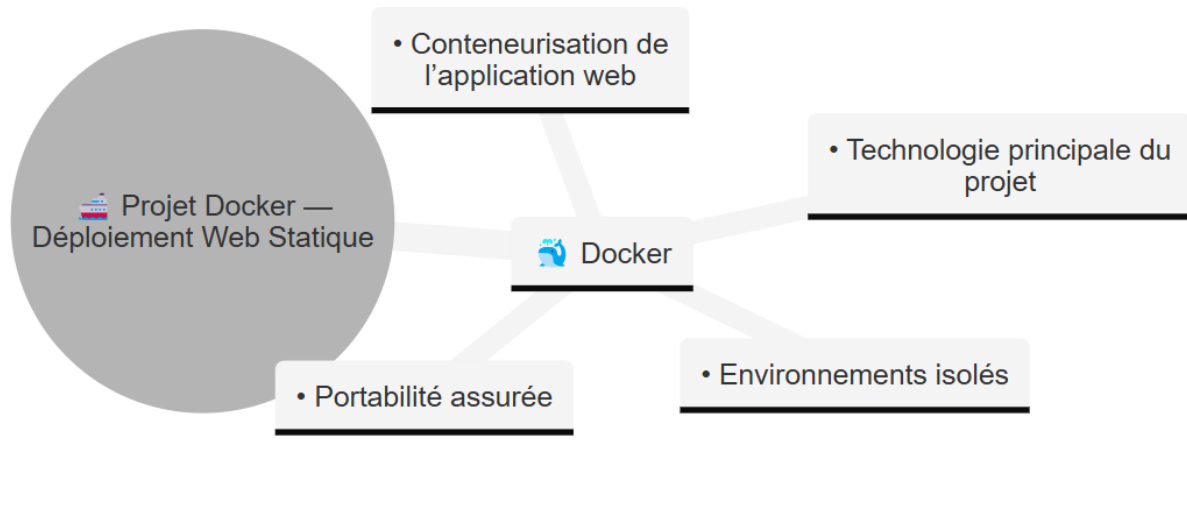
- **HTML/CSS**



## • NGINX



- Docker



## Étapes de réalisation

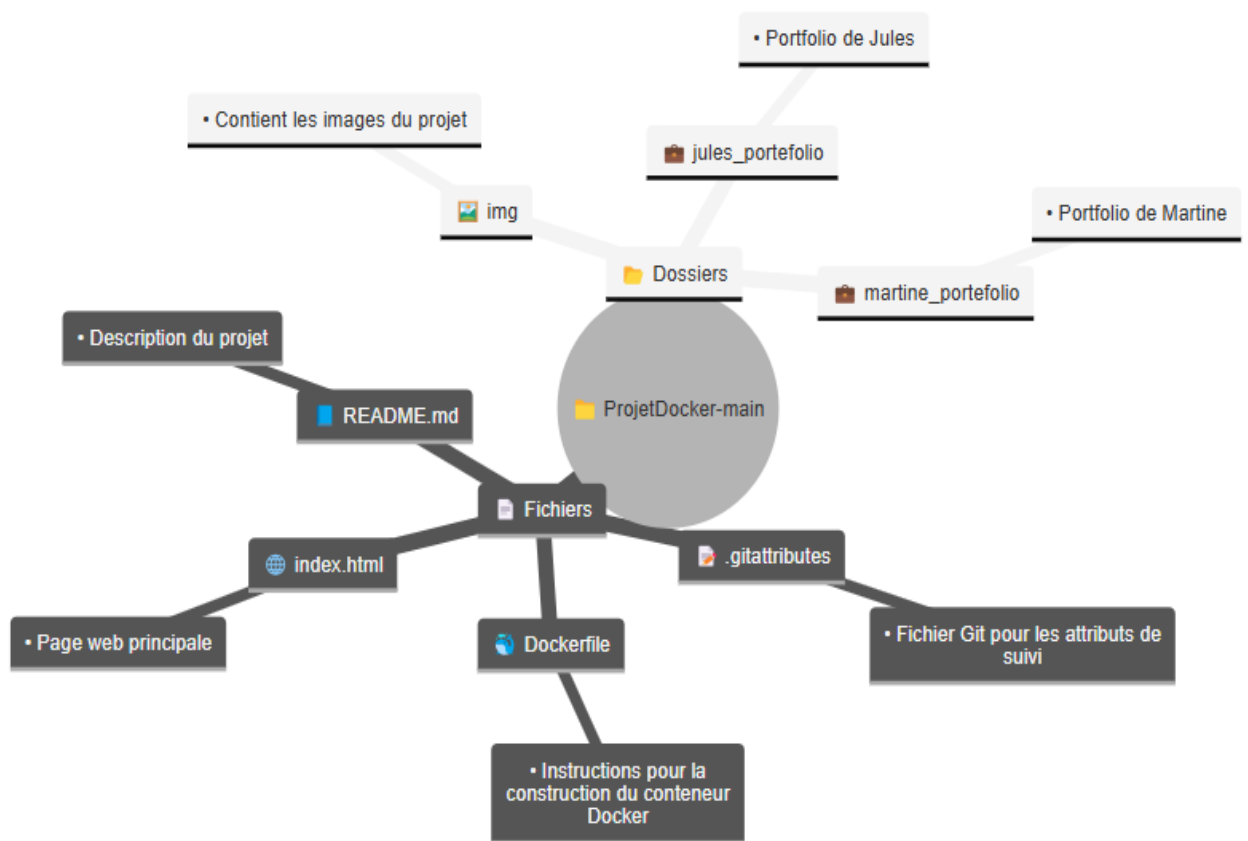
### ❖ Préparation du projet

La première étape a été de mettre en place la structure de projet. Cela consistait à organiser les fichiers du site web (HTML et CSS) ainsi qu'à préparer un fichier Dockerfile pour la création de l'image Docker.

### ✓ Structure du projet

Un répertoire contenant les fichiers HTML et CSS du site.

Un fichier Dockerfile pour construire l'image Docker.



## ✓ Création du site web

Le site web a été conçu de manière simple, en utilisant **HTML** pour la structure de la page et **CSS** pour la mise en forme et la personnalisation du design. Le choix du sujet était personnel, ce qui a permis de créer un site web pertinent et fonctionnel.

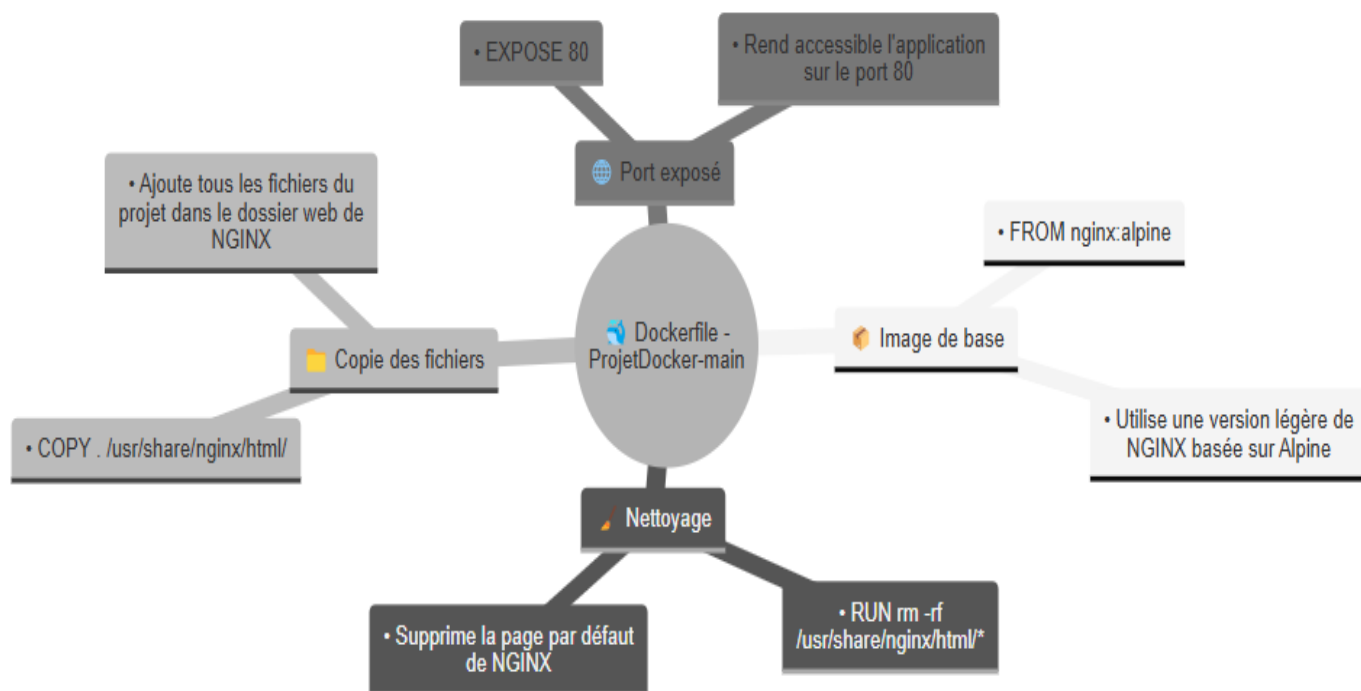
- **Sujet du site** : Un site de présentation personnelle.
- **Design** : Utilisation de CSS pour gérer les styles, la mise en page et la présentation visuelle.

## Configuration Docker

L'étape suivante a consisté à préparer l'image Docker. Nous avons choisi d'utiliser nginx, un serveur web léger et performant, comme base pour l'image Docker.

Un **Dockerfile** est un fichier texte qui contient une liste d'instructions permettant de construire une image Docker. Autrement dit, c'est un plan de construction pour créer un conteneur avec tout ce qu'il faut pour faire tourner une application.

Ainsi le **Dockerfile** a été configuré de la manière suivante :



## Déploiement du site

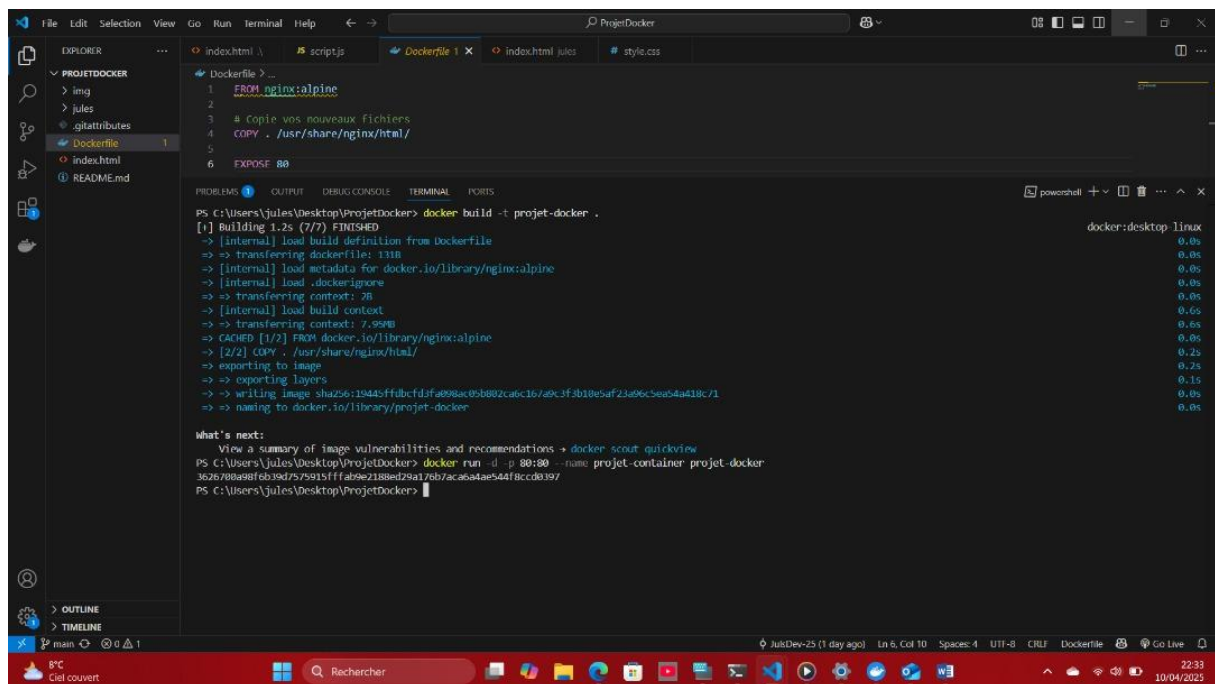
Une fois le Dockerfile prêt, les étapes suivantes ont été réalisées :

- Construction de l'image Docker avec : **docker build -t project-docker .**

Cette commande construit l'image à partir du Dockerfile et lui attribue le nom mes-sites.

- Lancement du conteneur Docker : **docker run -d -p --name project-contnainer 8080:80 project-docker**

Cette commande exécute le conteneur en arrière-plan et mappe le port 8080 de l'hôte vers le port 80 du conteneur. Le site devient ainsi accessible via <http://localhost:8080>.

A screenshot of a Windows terminal window with a dark theme. The terminal shows the execution of Docker commands. The first command is 'docker build -t project-docker .' which builds the image successfully. The second command is 'docker run -d -p 8080:80 --name project-contnainer project-docker' which starts the container. The terminal output includes progress bars for building the image and a list of ports. The Windows taskbar at the bottom shows the date as 10/04/2023 and the time as 22:33.

```
PS C:\Users\jules\Desktop\ProjectDocker> docker build -t project-docker .
[+] Building 1.2s (7/7) FINISHED
-> [internal] load build definition from Dockerfile
-> => transferring dockerfile: 331B
-> [internal] load metadata for docker.io/library/nginx:alpine
-> [internal] load .dockerignore
-> => transferring context: 2B
-> [internal] load build context
-> => transferring context: 7.59kB
-> CACHED [1/2] FROM docker.io/library/nginx:alpine
-> [2/2] COPY . /usr/share/nginx/html/
-> exporting to image
-> exporting layers
-> writing image sha256:19445ff8b6fd3fa898ac05b882cabc167ab9c3f3b10e5af23a96c5ea54a418c71
-> naming to docker.io/library/project-docker

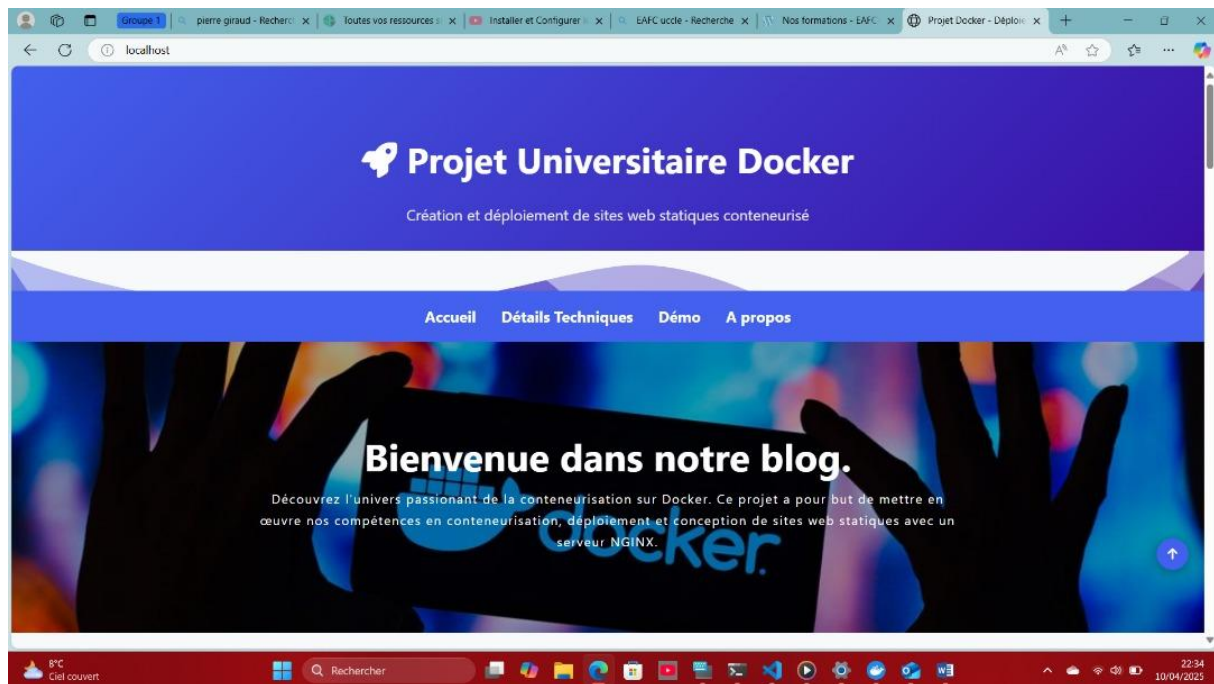
What's next:
View a summary of image vulnerabilities and recommendations + docker scout quickview
PS C:\Users\jules\Desktop\ProjectDocker> docker run -d -p 8080:80 --name project-contnainer project-docker
3626788a98fb39d7575915ffab9e2188ed29a176b7acab4ae544f8ccdb397
PS C:\Users\jules\Desktop\ProjectDocker>
```

- **Vérification du site**

Le site web est accessible via <http://localhost:8080> dans un navigateur. L'utilisateur peut voir la page d'accueil du site avec des liens vers des sections spécifiques.

- **Résultats**





Le projet a permis de démontrer la facilité d'utilisation de Docker pour le déploiement d'applications web statiques. Le site web statique a été correctement conteneurisé et déployé dans un environnement NGINX avec Docker. Il est accessible depuis n'importe quel environnement, garantissant la portabilité et la cohérence.

## Problèmes rencontrés

Au cours du projet, plusieurs défis ont été rencontrés :

### ❖ Problèmes liés à WSL et à Docker Desktop :

- La configuration de WSL2 et la virtualisation ont été sources de confusion à la fin du projet. Après avoir suivi plusieurs étapes pour activer la virtualisation dans le BIOS, des problèmes ont persisté, principalement liés à la configuration de Docker Desktop.
- Des erreurs liées à **HYPERV** et **PlatformVirtualMachine** ont empêché le déploiement de Docker. Finalement, après avoir mis à jour la configuration et réinstallé Docker, tout est revenu à la normale.

## **2. Problèmes d'accès aux fichiers dans Docker :**

- Parfois, la copie des fichiers dans le conteneur ne fonctionnait pas correctement, ce qui nécessitait de revérifier la structure des dossiers et s'assurer que le Dockerfile était correctement configuré pour copier les fichiers au bon endroit.

## **Conclusion**

Ce projet nous a permis de combiner nos compétences techniques et créatives pour concevoir un site web à la fois fonctionnel, esthétique et personnel. En choisissant de créer un portfolio partagé, nous avons répondu aux attentes pédagogiques tout en réalisant un outil valorisable dans un cadre académique ou professionnel.

Grâce à l'utilisation de Docker, nous avons pu mettre en œuvre un déploiement moderne et maîtrisé, tout en consolidant nos connaissances sur la conteneurisation et la gestion d'environnements de développement. En somme, ce projet fut une expérience enrichissante, nous permettant non seulement de progresser techniquement, mais aussi de mieux nous projeter dans notre future vie professionnelle.