

# RAPPORT DE STAGE

Mai - Juin 2025



Pi-Topo

Préparé par:

Vhann Danielle KIBAMBA-WILFRIDE

## SOMMAIRE

REMERCIEMENTS .....	3
I. INTRODUCTION .....	4
II. PRESENTATION DE L'ENTREPRISE .....	5
III. CONTEXTE ET OBJECTIFS DU STAGE .....	6
IV. TRAVAUX EFFECTUÉS .....	7
V. ANNEXES .....	8
VI. CONCLUSION .....	9

Je tiens à remercier chaleureusement l'ensemble de l'équipe de LIJE-Technologies pour m'avoir accueillie au sein de leur structure durant mon stage de 5 semaines.

Ce stage a été une véritable opportunité d'apprentissage, tant sur le plan technique que professionnel. LIJE-Technologies développe et commercialise l'application mobile iTopo, ce qui m'a permis de découvrir un environnement de travail dynamique, centré sur l'innovation et les technologies de pointe.

Durant cette période, j'ai eu la chance de manipuler et de mieux comprendre des outils essentiels comme Docker et QGIS. Cette immersion m'a permis d'approfondir mes compétences en administration de systèmes, en gestion de conteneurs, et en exploitation de solutions SIG (Systèmes d'Information Géographique).

Je remercie tout particulièrement Monsieur S. THARAUD, MATHIEU BECHADE, G. VERNADAT, pour leur accompagnement, patience et leurs conseils précieux, ainsi que tous les membres de l'équipe pour leur disponibilité, leur bienveillance et l'ambiance agréable qu'ils ont su instaurer.

Ce stage a renforcé mon intérêt pour le monde de l'informatique et m'a conforté dans mon choix de spécialisation en SISR. Je garderai un excellent souvenir de mon passage chez LIJE-Technologies.

# INTRODUCTION

Dans le cadre de ma formation en BTS Services Informatiques aux Organisations (SIO), spécialité Solutions d'Infrastructure, Systèmes et Réseaux (SISR), j'ai effectué un stage d'une durée de cinq semaines au sein de l'entreprise LIJE-Technologies, située à COUZEIX.

Ce stage s'inscrit dans le prolongement de ma formation théorique, avec pour objectif principal de me confronter aux réalités du métier et de mettre en pratique les compétences techniques acquises en cours. Il représente une étape essentielle dans mon parcours, me permettant de mieux comprendre le fonctionnement d'une entreprise informatique et de me professionnaliser.

LIJE-Technologies est une entreprise spécialisée dans le développement et la commercialisation d'iTopo, une application mobile et web innovante dédiée à la gestion et à la visualisation de données géographiques. Ce contexte professionnel m'a permis d'évoluer dans un environnement stimulant, en lien direct avec les nouvelles technologies et de me familiariser avec le monde professionnel.

Durant ce stage, j'ai notamment eu l'occasion de manipuler des outils comme Docker qui était le but de mon stage, en renforçant ainsi mes compétences en virtualisation et en déploiement d'application.

Ce rapport a pour but de présenter l'entreprise d'accueil, les missions qui m'ont été confiées, ainsi que les compétences mobilisées et développées au cours de cette expérience.

## **II. PRESENTATION DE L'ENTREPRISE :**

Lije-Technologies entreprise Française, située à Couzeix, est un éditeur de logiciels applicatifs crée le 07 mai 2014 par Julien DELAYE, développe et commercialise l'application web et mobile **i-Topo**. Elle a pour forme Juridique la SASU (société par action simplifiée unipersonnelle), c'est une entreprise privée et autonome.

C'est une petite entreprise dont l'effectif vari de 3 à 5 salariés.

### **1) Nature de L'entreprise :**

i-Topo est une solution qui est née dans la tranchée.... Développée et commercialisée par la société Lije Technologies, i-Topo a été imaginée par Julien DELAYE, un entrepreneur de Travaux Publics canalisateur qui voulait révolutionner le suivi de chantier par la photo.

« Nous avons mis en place i-Topo pour répondre à une problématique, commune à tous les exploitants, à savoir l'archivage et le stockage des photos de chantiers (branchement, réparations de fuites...). Cela nous permet de compléter les DT et DICT. Etant donné que nous ajoutons des commentaires sur les prises de vue géolocalisées, nous créons notre base de données avec des informations techniques comme le diamètre ou la nature des matériaux qui ne figurent pas toujours sur les plans de réseau. La fonction itinéraire sur l'application mobile permet aux équipes terrain de se rendre directement sur le chantier » Dit Julien DELAYE.

En 2017 Julien s'associe avec Gérald BERNADAT, Directeur opérationnel afin de commercialiser l'application. Cette application déjà étendue dans toute la France aujourd'hui après une analyse de la CIC de Limoges, ils décident de se lancer sur le marché espagnol.

### **2) Sur Qui l'application s'adresse :**

L'application i-Topo s'adresse principalement aux professionnels du secteur des travaux publics et de la gestion de chantiers.

Voici plus précisément les profils concernés :

- **Exploitants d'infrastructures** : routes, voiries, canalisations, etc.
- **Collectivités locales** : services techniques municipaux, intercommunaux ou départementaux
- **Maîtres d'ouvrage** : ceux qui commanditent les travaux (collectivités, promoteurs, services techniques...) etc.

### **Objectif**

Faciliter :

- Le **suivi collaboratif** des chantiers
- La  **traçabilité photo** géolocalisée des interventions
- L'**archivage structuré** pour les preuves et contrôles qualité

### **III. CONTEXTE ET OBJECTIFS DU STAGE :**

Dans le cadre de ma formation en BTS SIO, spécialité SISR, j'ai effectué un stage au sein de Lije-Technologies du 25 mai au 26 juin 2025. Ce stage s'inscrit dans une démarche professionnalisaante visant à me confronter à des problématiques concrètes d'administration système et de déploiement de solutions informatiques, en lien direct avec les technologies actuelles.

L'objectif principal de mon stage était de contribuer à la mise en place d'un environnement de développement géographique, en installant et configurant l'outil QGIS (Quantum GIS) dans un conteneur Docker. Le but de cette démarche était de faciliter le déploiement d'une solution SIG (Systèmes d'Information Géographique) au sein de l'organisation, en s'appuyant sur l'automatisation, la portabilité et l'isolation offertes par Docker.

Docker a joué un rôle central dans ce projet. J'ai utilisé cette technologie de conteneurisation pour créer un environnement cohérent, reproductible et indépendant du système hôte. Cette approche permet non seulement de simplifier la gestion des dépendances logicielles de QGIS, mais aussi de faciliter sa mise en œuvre sur d'autres machines ou serveurs sans avoir à répéter manuellement toute la procédure d'installation.

En parallèle, j'ai rédigé une documentation technique complète décrivant toutes les étapes nécessaires à la création et à la gestion de cet environnement : téléchargement de l'image Docker, configuration du conteneur, accès à l'interface QGIS, persistance des données, ainsi que les bonnes pratiques de maintenance. Cette documentation vise à servir de guide pour les futurs utilisateurs ou administrateurs souhaitant déployer rapidement une solution SIG fonctionnelle via Docker.

Ce travail s'inscrit dans une volonté globale de modernisation et d'automatisation des outils utilisés par l'organisation, et m'a permis de renforcer mes compétences dans le domaine des conteneurs, tout en appliquant mes connaissances sur les infrastructures et les systèmes.

## **IV. TRAVAUX EFFECTUÉS :**

### **❖ Semaine 1 :**

Durant la première semaine de mon stage (lundi , mardi, mercredi) , j'ai commencé par me familiariser avec l'outil de conteneurisation **Docker**, en explorant ses concepts fondamentaux tels que les **images**, **conteneurs**, **Dockerfile**, **volumes** et **réseaux**. En parallèle, j'ai mené des recherches pour comprendre l'utilité et les fonctionnalités de **QGIS**, un système d'information géographique (SIG) libre, afin de cerner les enjeux de son installation dans un environnement conteneurisé.

J'ai ensuite consulté plusieurs **documentations techniques** en ligne concernant l'installation de QGIS dans un conteneur Docker. Cela m'a permis d'identifier les fichiers et dépendances nécessaires à sa mise en place (ex. bibliothèques système, environnement graphique, variables d'environnement, etc.).

Pour assurer un bon suivi de mes travaux et faciliter la collaboration avec mon tuteur, j'ai également mis à jour mon **dépôt GitHub** afin de versionner toutes les étapes de mon projet, en créant des **commits réguliers** et en documentant les modifications effectuées.

Enfin, j'ai débuté l'installation de QGIS en élaborant un premier **Dockerfile** pour définir l'image, ainsi qu'un fichier **docker-compose.yml** afin de simplifier le déploiement et la gestion du conteneur.

### **❖ Semaine 2 : Exploration du dépôt GitHub de QGIS Docker et personnalisation des fichiers de configuration**

Lors de la deuxième semaine, j'ai approfondi mes recherches sur le déploiement de QGIS en environnement conteneurisé. J'ai découvert et exploré le **dépôt GitHub officiel de QGIS Docker**, qui propose une structure complète pour l'installation de QGIS à l'aide de Docker.

J'ai cloné ce dépôt en local via la commande :

```
git clone <lien_du_dépôt>
```

Cela m'a permis de récupérer l'ensemble des fichiers de configuration fournis, incluant le Dockerfile, les fichiers docker-compose.yml, les scripts de lancement, et diverses ressources nécessaires à la construction de l'image.

### **❖ Semaine 3 :**

Durant la troisième semaine, j'ai poursuivi le travail entamé à partir du dépôt GitHub de QGIS Docker. Mon objectif principal était d'adapter en profondeur la configuration existante afin d'obtenir un déploiement fonctionnel et stable de QGIS dans un conteneur.

J'ai analysé et modifié plusieurs fichiers clés, notamment :

- le Dockerfile, pour y intégrer les dépendances nécessaires à l'exécution correcte de QGIS,
- le fichier docker-compose.yml, pour gérer les services, les ports exposés et les volumes de données.

J'ai rencontré plusieurs difficultés liées à des erreurs d'exécution, à des fichiers manquants ou à des incompatibilités logicielles, ce qui m'a amenée à effectuer un important travail de débogage. Pour cela, j'ai utilisé les logs Docker, ainsi que des recherches en ligne et des tests successifs.

## ❖ Semaine 4 : Finalisation de l'installation de QGIS en conteneur Docker sans Docker Compose

Au cours de la quatrième semaine, j'ai poursuivi le déploiement de QGIS, cette fois-ci en simplifiant l'architecture de mon environnement Docker. J'ai abandonné l'utilisation de **Docker Compose** pour me concentrer uniquement sur un **Dockerfile**, dans une logique de configuration plus légère et ciblée.

Dans cette nouvelle version de mon stack, le service **NGINX** a été retiré, l'objectif étant de faire tourner uniquement **QGIS** dans un conteneur isolé. Afin d'assurer le bon fonctionnement de l'application, j'ai modifié le Dockerfile en y ajoutant une ligne essentielle permettant d'exécuter un fichier précédemment introuvable. Cette étape a été cruciale pour corriger une erreur liée à un chemin d'accès ou à une dépendance manquante au moment du lancement de QGIS.

Une fois le Dockerfile corrigé et fonctionnel, j'ai procédé à la création d'une image personnalisée via la commande suivante :

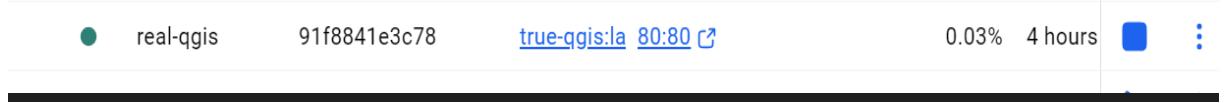
**docker build -t vhannqgis25:latest .**

Enfin, j'ai lancé le conteneur en exposant le port 80, grâce à la commande suivante :

**docker run --name vhannqgis -p 80:80 vhannqgis25:latest**

Cette étape m'a permis de mieux comprendre le processus complet de construction d'image Docker personnalisée et de déploiement de l'application SIG en conteneur tout en gérant les ports d'accès nécessaire.

Avant j'avais un problème, après avoir lu les logs un fichier semblait introuvable, j'ai dû rajouter une ligne supplémentaire pour l'exécution de ce fichier, dans le Dockerfile sur VS code :



```
PS C:\projet_Qgis\qgis-docker> docker build -t true-qgis:latest .
[+] Building 1.2s (13/13) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 2.38kB
=> [internal] load metadata for docker.io/library/ubuntu:noble
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/8] FROM docker.io/library/ubuntu:noble@sha256:b59d21599a2b151e23eea5f6692f4af4d7d31c4e236d22bf9b62b86d2e386bb8f
=> => resolve docker.io/library/ubuntu:noble@sha256:b59d21599a2b151e23eea5f6692f4af4d7d31c4e236d22bf9b62b86d2e386bb8f
=> [internal] load build context
=> => transferring context: 341B
=> CACHED [2/8] RUN apt update && apt install -y gnupg wget software-properties-common && wget -qO - https://qgis.org/downloads/qgis-2022.gpg.key | gpg --no-dst
=> CACHED [3/8] ADD server/conf/qgis-server-nginx.conf /etc/nginx/nginx.conf
=> CACHED [4/8] ADD server/start-xvfb-nginx.sh /usr/local/bin/start-xvfb-nginx.sh
=> CACHED [5/8] RUN sed -i 's/\r//g' /usr/local/bin/start-xvfb-nginx.sh
=> CACHED [6/8] COPY /server/test/data/test_project/test_project.qgs /var/lib/qgis/test_project.qgs
=> [7/8] RUN mkdir -p /var/lib/qgis && chmod 1777 /var/lib/qgis
=> [8/8] WORKDIR /var/lib/qgis
=> exporting to image
```

```
PS C:\projet_Qgis\qgis-docker> mv \server\Dockerfile .
PS C:\projet_Qgis\qgis-docker> docker build -t true-qgis:latest .
[+] Building 1.7s (11/12)
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 2.35kB
=> [internal] load metadata for docker.io/library/ubuntu:noble
=> [internal] load .dockerignore
```

```
PS C:\projet_Qgis> git clone https://github.com/qgis/qgis-docker.git
Cloning into 'qgis-docker'...
remote: Enumerating objects: 1216, done.
remote: Counting objects: 100% (241/241), done.
remote: Compressing objects: 100% (126/126), done.
remote: Total 1216 (delta 141), reused 199 (delta 113), pack-reused 975 (from 1)
Receiving objects: 100% (1216/1216), 306.32 KiB | 2.07 MiB/s, done.
Resolving deltas: 100% (657/657), done.
PS C:\projet_Qgis> cd .\qgis-docker\
PS C:\projet_Qgis\qgis-docker> mv .\server\Dockerfile .
```

## ❖ Semaine 5 : Poursuite de l'intégration de QGIS en environnement Docker

Au cours de la cinquième semaine, j'ai poursuivi le travail commencé précédemment sur l'installation de QGIS dans le conteneur Docker sans utiliser Docker Compose ni services externes comme NGINX.

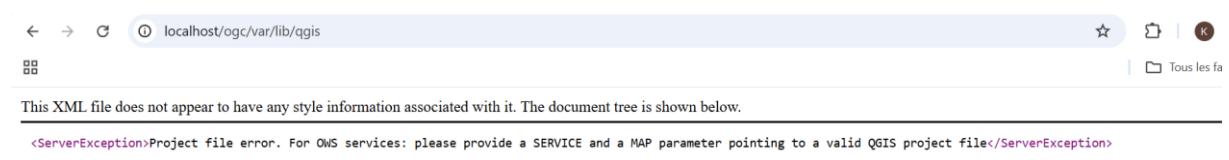
L'objectif de cette semaine a été de stabiliser l'environnement, en testant plusieurs fois si l'interface de Qgis pouvait s'afficher grâce à un des projets Qgis. J'ai continué à ajuster le Dockerfile en fonction des retours d'exécution et des erreurs rencontrées, notamment en ce qui concerne le chemin du projet Qgis qu'on devait mettre sur l'URL.

J'ai créé un dossier sur lequel on pourrait mettre les projets Qgis :

**/var/lib/qgis/test\_project.qgs**

Malheureusement un message d'erreur s'est affiché en disant que le service tourne et répond mais le projet Qgis est invalide, je ne disposais pas d'autres projets en ma possession pour faire ainsi afficher l'interface tant attendu .

Résultat :

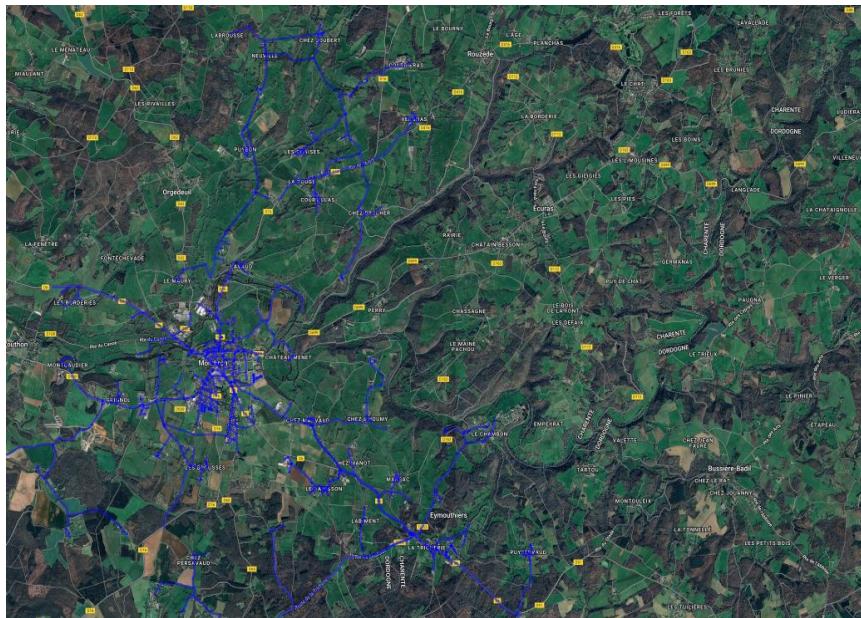


## V. ANNEXES : Visualisation de l'application i-Topo

### Prises de vue de quelques chantiers



Différents types de formats que l'on peut générer



Représentation de la carte

Réseaux



Filtrer

Réinitialiser