**INTRODUCTION**

L'objectif de ce rapport est de décrire le processus de création d'une application Node.js qui utilise l'API OpenWeatherMap pour récupérer les données météorologiques d'un lieu spécifié par les coordonnées de latitude et de longitude. L'application permet également de spécifier ces coordonnées via des variables d'environnement.

**1.CHOIX TECHNIQUES**

**Langage de programmation**

Nous avons choisi d'utiliser JavaScript avec Node.js comme langage de programmation pour créer l'application.

**Framework Express.js**

Nous avons utilisé le framework Express.js pour créer l'application web. Express.js est un framework Web minimaliste et flexible pour Node.js, qui simplifie le processus de création de serveurs Web et de routage.

**Bibliothèque Axios**

Nous avons utilisé Axios comme client HTTP pour effectuer des requêtes vers l'API OpenWeatherMap. Axios est une bibliothèque JavaScript pour effectuer des requêtes HTTP depuis Node.js ou le navigateur, qui offre une syntaxe simple et une gestion avancée des erreurs.

**Variables d'environnement**

Nous avons utilisé des variables d'environnement pour permettre la configuration dynamique des coordonnées de latitude et de longitude, ainsi que de la clé API OpenWeatherMap. Cela permet une plus grande flexibilité et facilite le déploiement de l'application dans différents environnements sans avoir à modifier le code source.

**2. ÉTAPES DE CREATION DE L'APPLICATION**

1. **Initialisation du projet** : Nous avons initialisé un nouveau projet Node.js en utilisant **npm init** pour créer un fichier **package.json** et gérer les dépendances du projet.
2. **Installation des dépendances** : Nous avons installé les dépendances nécessaires, à savoir Express.js et Axios, en utilisant **npm install express axios**.
3. **Création du fichier JavaScript** : Nous avons créé un fichier JavaScript (**weather.js**) dans lequel nous avons écrit le code de notre application.
4. **Écriture du code de l'application** : Nous avons écrit le code de notre application en utilisant Express.js et Axios. Le code se compose de la configuration du serveur Express, de la définition des routes et de l'appel à l'API OpenWeatherMap pour récupérer les données météorologiques.
5. **Utilisation de variables d'environnement** : Nous avons utilisé des variables d'environnement pour récupérer les coordonnées de latitude et de longitude ainsi que la clé API OpenWeatherMap. Si aucune valeur n'est fournie pour ces variables d'environnement, des valeurs par défaut sont utilisées.
6. **Lancement du serveur** : Nous avons lancé le serveur en écoutant sur le port spécifié (port 3500 dans notre cas) en utilisant **app.listen(PORT)**.
7. **Création de l’image Docker**: Nous avons tout d’abord créer un DockerFile qui contient les configurations de l’image Docjer et ensuite build l’image Docker.

**3. COMMANDES UTILISEES**

* Initialisation du projet :

**npm init -y**

* Installation des dépendances (Express.js et Axios) :

**npm install express axios**

* Lancement de l'application :

**node weather.js**

* Création de l’image Docker

**Docker build -t weather\_image**

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

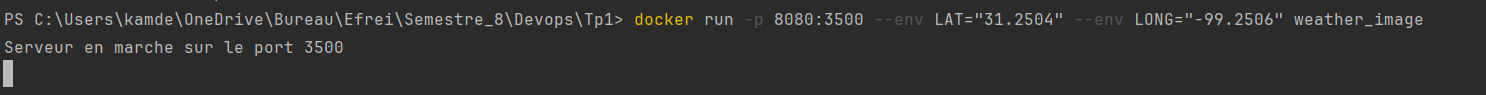
* Login et depot de l’image sur dockerhub

**Docker push**

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

* Lancement de l’application à travers l’image Docker



**Résultat**Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

**CONCLUSION**

Ce rapport décrit le processus de création d'une application Node.js pour récupérer la météo en utilisant l'API OpenWeatherMap. En utilisant Express.js, Axios et des variables d'environnement, nous avons créé une application flexible et extensible qui peut être déployée dans différents environnements.