Rapport de projet Ely SENE : Développement d'un wrapper météo avec Dockerisation

Introduction

Dans le cadre de ce projet, j'ai entrepris de créer un wrapper météo permettant de récupérer les données météorologiques d'un lieu donné en utilisant l'API OpenWeather. J'ai ensuite dockerisé cette application et publié l'image Docker sur DockerHub. Ce rapport présentera en détail les étapes que j'ai suivies, les choix techniques que j'ai faits et les difficultés rencontrées tout au long du processus.

Sommaire

1. Création du wrapper météo

Pour créer le wrapper météo, j'ai utilisé le langage de programmation Python. Le wrapper récupère les données météorologiques d'un lieu donné en utilisant les coordonnées de latitude et de longitude, ainsi que la clé d'API OpenWeather. J'ai choisi d'utiliser des variables d'environnement pour stocker la clé d'API, afin de ne pas exposer de données sensibles dans le code source.

Commandes utilisées :

- export LAT=31.2504 : **Définition de la latitude comme variable d'environnement.**

- export LONG=-99.2506 : **Définition de la longitude comme variable d'environnement.**

- export API_KEY=242A1186E124BFB7AF44537B1593BCDD : **Définition de la clé d'API comme variable d'environnement.**

- python weather_wrapper.py : **Exécution du wrapper météo.**

2. Packaging du code dans une image Docker

Après avoir développé le wrapper météo, je l'ai Dockerisé en créant un Dockerfile. Le Dockerfile utilise l'image Python officielle comme image de base, installe les dépendances nécessaires et définit la commande par défaut pour exécuter le script Python.

Commandes utilisées :

- docker build -t maregistry/api:1.0.0 . : **Construction de l'image Docker.**

- docker run --env LAT=31.2504 --env LONG=-99.2506 --env API_KEY=242A1186E124BFB7AF44537B1593BCDD maregistry/api:1.0.0 : **Exécution de l'image Docker avec les variables d'environnement.**

Difficultés rencontrées :

Je n'ai pas rencontré de difficultés majeures lors de la création de l'image Docker. Cependant, j'ai dû m'assurer que la clé d'API OpenWeather n'était pas stockée dans l'image Docker en utilisant des variables d'environnement.

3. Publication de l'image Docker sur DockerHub

Une fois l'image Docker créée, je l'ai publiée sur DockerHub pour la rendre accessible à d'autres utilisateurs.

Commandes utilisées :

- docker login : **Connexion à DockerHub.**

- docker tag maregistry/api:1.0.0 elordiner/maregistry:1.0.0 : **Renommage de l'image Docker.**

- docker push elordiner/maregistry:1.0.0 : **Publication de l'image Docker sur DockerHub.**

4. Mise à disposition du code sur GitHub

Enfin, j'ai mis à disposition le code source du projet sur GitHub pour permettre à d'autres personnes de contribuer et de collaborer.

Commandes utilisées :

- git init : **Initialisation d'un nouveau dépôt Git local.**

- git add . : **Ajout de tous les fichiers au suivi de Git.**

- git commit -m "Initial commit" : **Validation des modifications.**

- git remote add origin <URL _GitHub> : **Ajout de l'URL du référentiel distant.**

- git push -u origin master : **Publication du code sur GitHub.**

Difficultés rencontrées :

Je n'ai pas rencontré de difficultés lors de la mise à disposition du code sur GitHub. Je suis d'ailleurs passé par Github desktop.

Bonus :

Hadolint

J'ai rencontré des problèmes lors de l'installation de Hadolint en utilisant Chocolatey et en téléchargeant le fichier exécutable depuis leur site internet. Malgré plusieurs tentatives, je n'ai pas réussi à résoudre ces problèmes et n'ai donc pas pu effectuer la vérification des erreurs de lint sur le Dockerfile. Concernant le fichier .exe, lorsque j'essayais de l'exécuter, une fenêtre noire apparaissait puis plus rien. Concernant chocolatey avec la commande « choco install trivy », j'ai un message d'erreur stipulant que les ressources étaient introuvables.

Trivy

Trivy a détecté deux vulnérabilités dans les dépendances de l'application Python, à savoir les bibliothèques pip et setuptools. Voici un résumé de chaque vulnérabilité :

Vulnérabilité CVE-2023-5752 dans la bibliothèque pip :

- Gravité : Moyenne

- Statut : Corrigée

- Lien CVE : [CVE-2023-5752](https://avd.aquasec.com/nvd/cve-2023-5752)

Vulnérabilité CVE-2022-40897 dans la bibliothèque setuptools :

- Gravité : Haute

- Statut : Corrigée

- Lien CVE : [CVE-2022-40897](https://avd.aquasec.com/nvd/cve-2022-40897)

Des captures d'écran ont été ajouté sur mon repository Github concernant les vulnérabilités

Actions entrepris :

Mis à jour des dépendances (modification du Dockerfile)

Après avoir réexécuter le scan Trivy, toutes les vulnérabilités ont été corrigées.

Aucunes données sensibles stockées dans l'image.