
Certification Développeuse Data - IA

— Mise en Situation
Professionnelle 2 —

Amandine Yborra



Sommaire

Introduction	p 2
Partie 1 : Les bases du projet	p 2
Le besoin client	
Un état de l'art	
Partie 2 : La mise en oeuvre du projet	p 3 - 7
Traduction et choix techniques	
Partie 3 : Bilan et axes d'améliorations du projet	p 7
Bilan	
Axes d'améliorations	
Conclusion	p 7

Introduction

Ce rapport finalise près de deux ans de formation auprès de Simplon. Après 7 mois de cours intensifs, je termine par ce projet mon année d'alternance en entreprise.

Genapi l'entreprise qui m'a accueillie cette année est un éditeur de logiciel spécialisé dans le notariat depuis 1988 et membre fondateur de Septeo.

Ce projet de certification n'est pas un projet d'entreprise mais un projet personnel. Le notariat ayant des données vraiment sensibles, il m'est impossible de présenter un projet issu de mon travail auprès d'eux. C'est pourquoi j'ai choisi de créer un projet fictif mais ayant pris racine auprès de mon entreprise.

Partie 1 : Les bases du projet

Le besoin client

Aujourd'hui, le marché de l'immobilier est tendu du fait de la baisse des taux de prêts.

Beaucoup vendent et beaucoup achètent. Les deux parties veulent un prix juste.

C'est pourquoi les agents immobiliers doivent estimer au mieux le prix des biens qu'ils vont devoir vendre et ainsi contenter leurs clients.

Il leur faut donc un logiciel ou une API permettant de prédire le prix du bien selon certains critères qui définiront le bien à estimer.

Le client : Agences Immobilières

Le besoin :

- Avoir une API pour entrer les critères permettant de la prédiction et avoir un visuel sur le résultat.

Un état de l'art

L'utilisation du Cloud pour le déploiement représente la facilité avec laquelle l'on adapter les besoins en applications et ressources informatiques. Cette adaptabilité et cette flexibilité donnent aux entreprises un avantage compétitif et permettent également de supprimer les délais et le casse-tête informatique si cela devait se faire dans l'infrastructure informatique interne à votre entreprise.

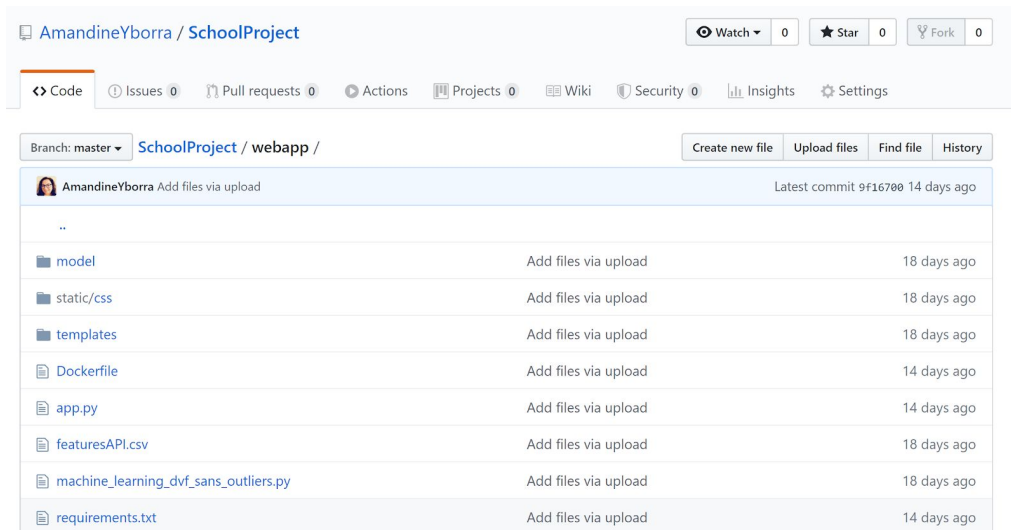
De plus le cloud, du fait de son infrastructure et que les logiciels soient déjà installés, est prêt à être utilisé par tout nouveau client, réduisant ainsi le besoin des entreprises à engager leurs techniciens informatiques internes ou sous-traités. Cela permet non seulement des économies de coûts, mais également de pouvoir commencer à utiliser les applications du cloud bien plus rapidement.

Partie 2 : La mise en oeuvre du projet

Traduction et choix techniques

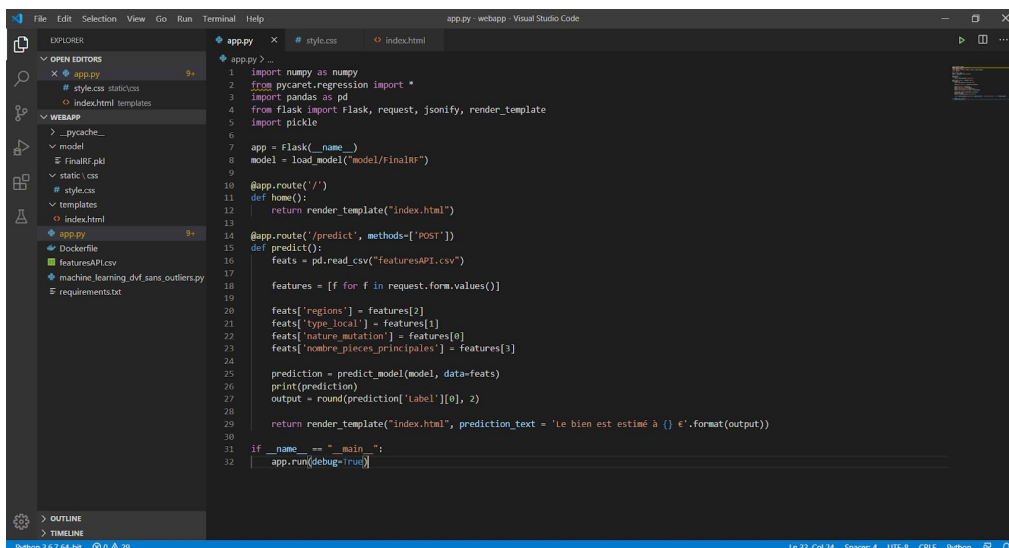
Création d'une API

L'API va être créé avec Flask, framework léger qui permet d'éditer des API en python. De plus il est associé à des templates HTML et CSS pour la mise en forme. Le dossier pour la création de l'API devra être composé comme suit.



Avec en racine le script de l'API, le fichier *requirement.txt*, regroupant l'environnement pour le bon fonctionnement du modèle et de l'API. Un fichier csv avec seulement les colonnes pour récupérer les données du formulaire qui seront les inputs du modèle prédictif Le dossier static contient le CSS, le dossier templates a le fichier HTML qui sert de page d'accueil et de formulaire. Et enfin le dossier model contient le modèle sauvegarder (pipeline de preprocessing avec l'algorithme entraîné).

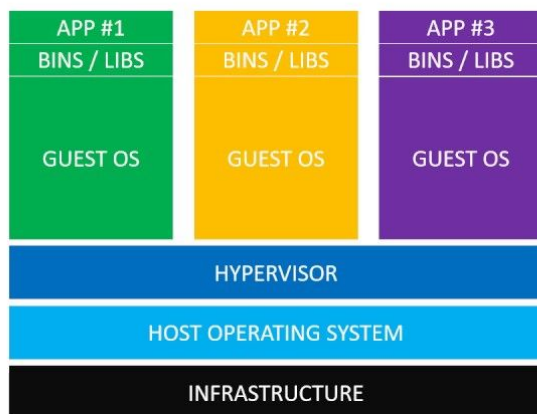
Le script *app.py*, va à l'aide du fichier *index.html*, charger une page d'accueil qui sera un formulaire.



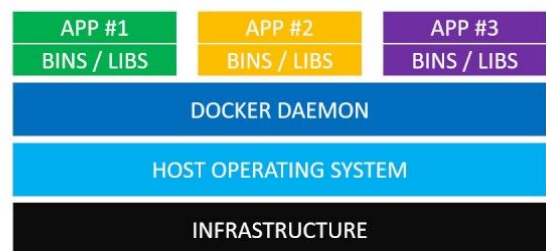
Une fois les données validées en appuyant sur le bouton "Estimation" elle sont récupérées en ajoutées au fichier CSV. Ensuite la prédiction est faite par le modèle chargé avec le csv en input . L'API charge ensuite la page avec la prédiction.

Mise en place d'un conteneur :

Une fois l'API Flask fonctionnelle je la mets dans un conteneur pour pouvoir par la suite la déployer plus facilement. Pour cela j'utilise Docker avec lequel on peut avoir rapidement des conteneurs embarquant tous les environnements des API facilitant le partage et le déploiement et permettant surtout l'isolation des processus.



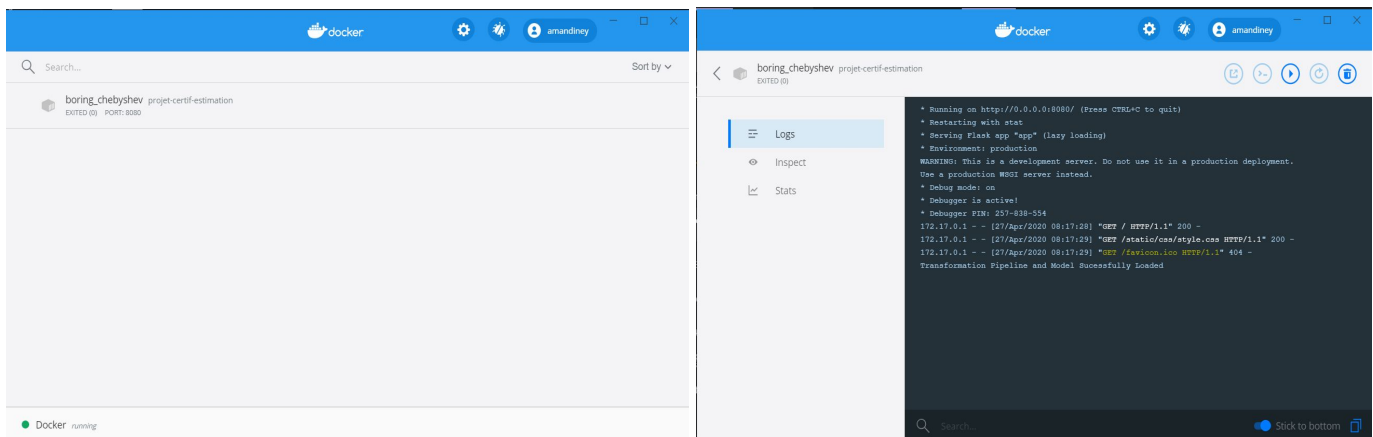
Virtual Machines



Docker Containers

Pour mettre en place un conteneur il faut créer une image qui doit être mise à la racine du dossier de l'API. Puis faire un build de cette image soit avec Visual Studio Code soit avec PowerShell.

Résultat sur Docker Dashboard :



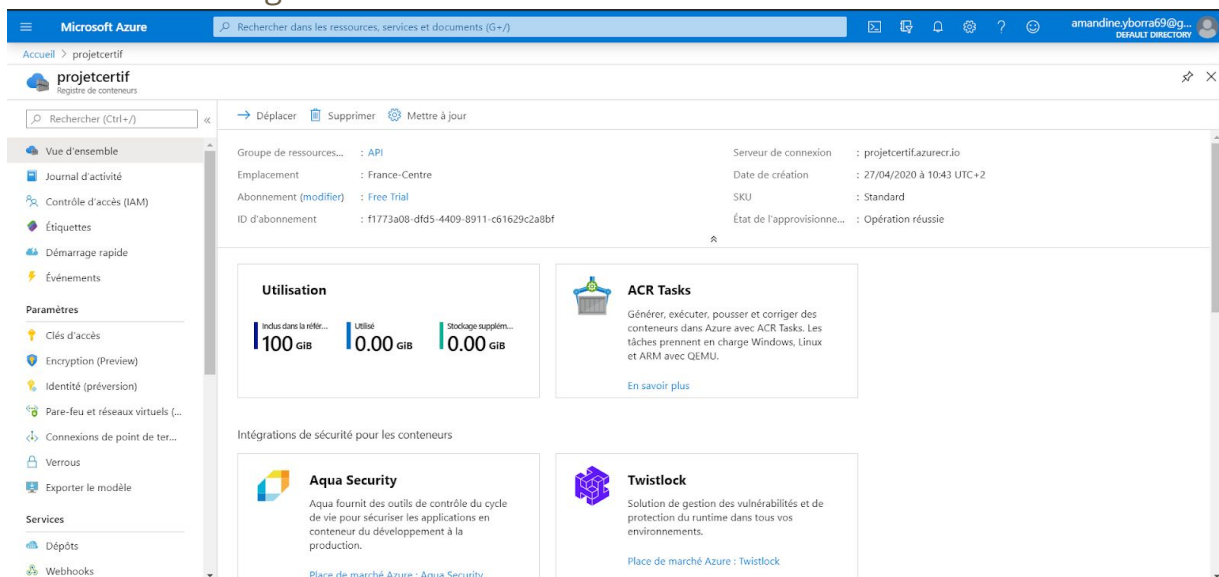
On peut tester que tout fonctionne bien en suivant le lien dans les logs du dashboard.

Déploiement :

Il se fait via le portail d'Azure pour que l'API soit accessible à tous et à tout moment. Azure favorise les déploiements rapides avec une interface facile à comprendre et son lien avec Docker permet de pusher directement l'image dans Azure Containers Registry.

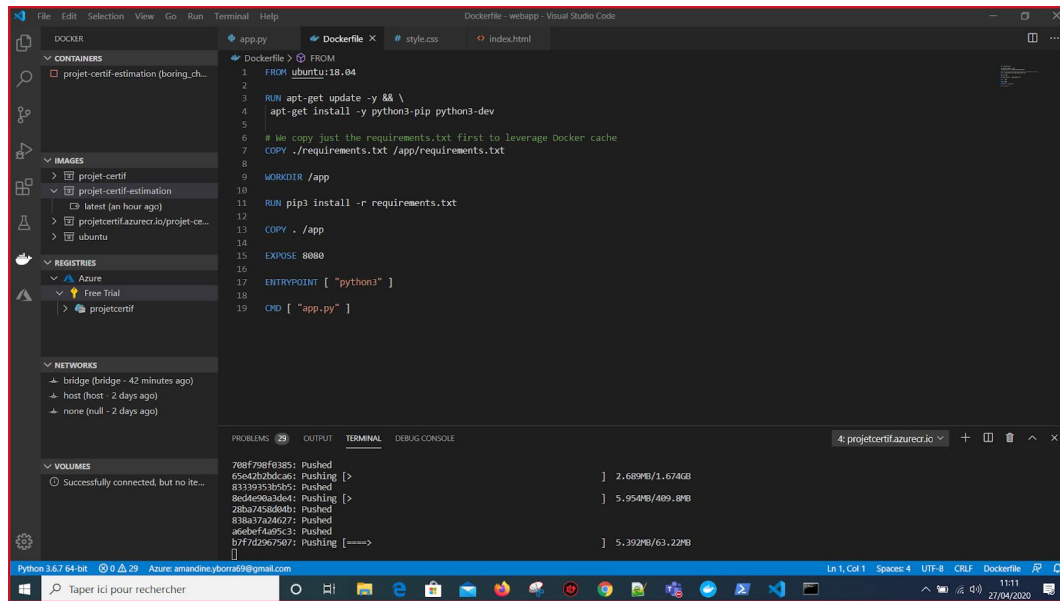
Les étapes pour déployer dans Azure :

- Paramétrage d'un environnement Azure



- Push de l'image Docker dans le Azure Container Registry

Avec la création sur Azure d'un Azure Container Registry on peut créer, stocker, sécuriser, numériser, répliquer et gérer des images. Et bien sûr connecter différents environnements comme des services Azure tel qu'App Service.



- Création d'une WebApp Azure et exécution

Permet de créer et déployer des applications web. Le déploiement continu est facilité et peut être automatisé.

Microsoft Azure

Rechercher dans les ressources, services et documents (G+7)

amandine.yborra69@g...
DEFAULT DIRECTORY

Accueil > Nouveau > Application web

Application web

De base **Docker** Surveillance Balises Vérifier + créer

Extrayez des images conteneurs d'Azure Container Registry, de Docker Hub ou d'un dépôt Docker privé. App Service déploie en production l'application conteneurisée avec vos dépendances préférées en quelques secondes.

Options

Conteneur unique

Source d'image

Azure Container Registry

Options Azure Container Registry

Registre *

projetcertif

Image *

webapp

Balise *

latest

Commande de démarrage

Vérifier + créer < Précédent Suivant : Surveillance >

Microsoft Azure

Rechercher dans les ressources, services et documents (G+7)

amandine.yborra69@g...
DEFAULT DIRECTORY

Accueil > Nouveau > Application web

Application web

Sélectionnez un abonnement pour gérer les coûts et les ressources déployées. Utilisez les groupes de ressources comme des dossiers pour organiser et gérer toutes vos ressources.

Abonnement *

Free Trial

Groupe de ressources *

API

Créer nouveau

Détails de l'instance

Nom *

Estimation-prix-immobilier

Publier *

Code Conteneur Docker

Système d'exploitation *

Linux Windows

Région *

France Central

Vous ne trouvez pas votre plan App Service ? Essayez une autre région.

Plan App Service

Le niveau tarifaire du plan App Service détermine l'emplacement, les fonctionnalités, le coût et les ressources de calcul associés à votre application. En savoir plus

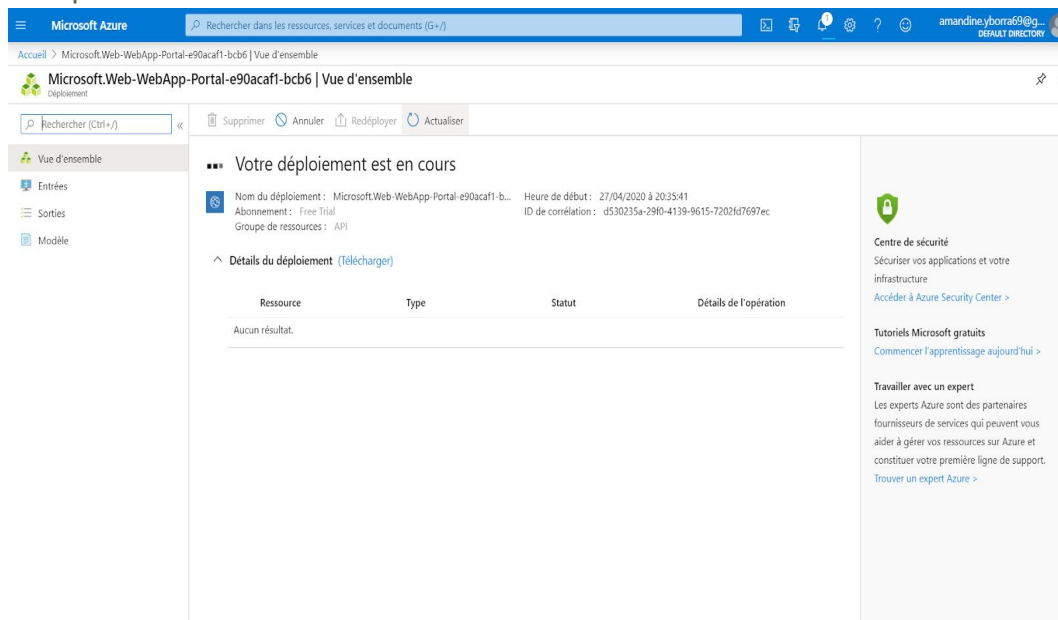
Plan Linux (France Central) *

ASP-API-a190 (F1)

Créer

Vérifier + créer < Précédent Suivant : Docker >

- Déploiement



Partie 3 : Bilan et axes d'améliorations du projet

Bilan du projet

Point négatif :

Les abonnements d'Azure sont cher et la version d'essai ne dure que 1 mois

Points positifs :

Déploiement facilité par Azure et ses connecteurs

Axes d'améliorations

- Création d'un système de login/password
- Création de l'application web directement sur Azure

Conclusion

L'application a bien été déployée sur le Cloud Azure et utilise une web API de chez Azure également. On peut penser à programmer les mises à jours à venir.