PROJET 10: Open Classrooms

/01

Développez un chatbot pour réserver des vacances

Bartcus Marius | 2023

Somaire

/03

Introduction

Context

- FlyMe: entreprise qui propose des voyage au professioneles et particuliers
- Creation d'une chatbot pour les colaborateurs de l'entreprise FlyMe

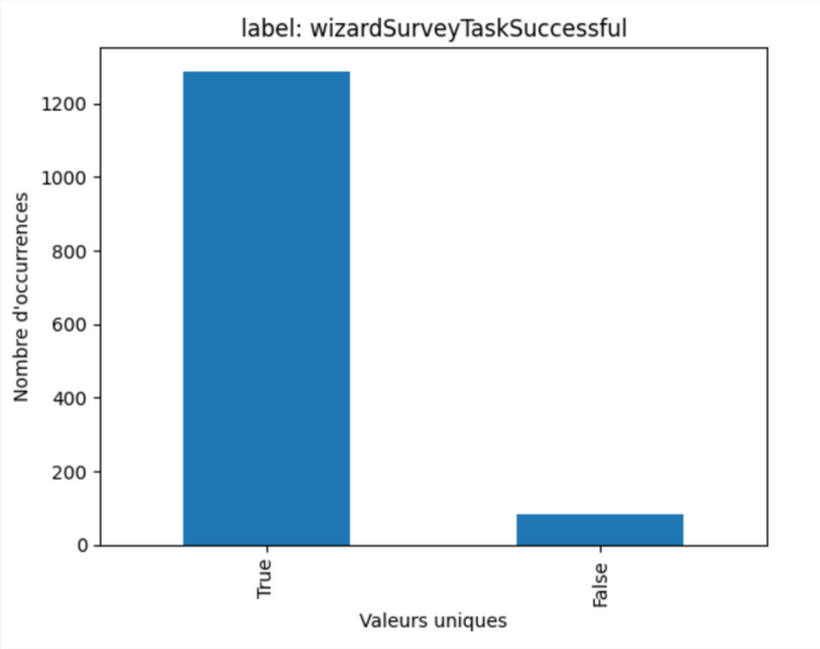
Objective

- Construire une MVP pour aider les employés de FlyMe à réserver des voyages pour leurs vacances. Pour ça on va utilisé des données disponible en ligne.
- Creation des tests unitaires pour testé le chat.
- Identifié dans la demande d'une utilisateur:
  - Ville de départ
  - Ville de destination
  - Date aller souhaitée du vol
  - Date retour souhaitée du vol
  - Budget maximum pour le prix total des billets
- Suivre et d’analyser l’activité du chatbot en production et lever une alerte lorsque le chatbot donnera plusieurs mauvaises réponses à des utilisateurs

Les données

Frames Dataset
https://www.microsoft.com/en-us/research/project/frames-dataset/download/

- Contient un historique d'échange entre un utilisateur et le Bot composés de plusieurs tours.
- Les données sont dans une format json.
- Le dialoge est entre un robot et un utilisateur essayant de réserver un vol.



- L'ensemble de données est composé de **1369 dialogues annotés**. 82 comunication avec le chatbot n'a pas aider l'utilisateur.

/05

MODÈLE D'ANALYSE SÉMANTIQUE: LUIS

Bartcus Marius | 2023|OC

Première étape du projet

- En utilisant le service cognitif Azure LUIS (Language Understanding) - entraîner et deployer une modèle d'analyse sémantique

Étape 1

Creation de ressource LUIS.

Étape 2

Ajouter les intents, entities, prebuilt entities et rendre les données compatible avec LUIS

```
{'text': 'leave whenever. i should be back for September 11',
 'intent_name': 'BookFlight',
 'entity_labels': [{'entity_name': 'end_date',
   'start_char_index': 37,
   'end_char_index': 49}]}
```

37 49

Étape 3

Ajouter les exemples au modèle.

Étape 4

Entraîner et déployer (publier) le modèle

Étape 5

Tester.
query = 'I would like to book a flight from Lyon to Paris tomorrow at 11pm with a budget of 100 euros'

```
{'or_city': 'Lyon',
 'dst_city': 'Paris',
 'geographyV2': {'value': 'Paris', 'type': 'city'},
 'datetimeV2': {'type': 'datetime',
  'values': [{'timex': '2023-02-23T23',
    'resolution': [{'value': '2023-02-23 23:00:00'}]}]},
 'number': 100,
 'budget': '100 euros'}
```

/06

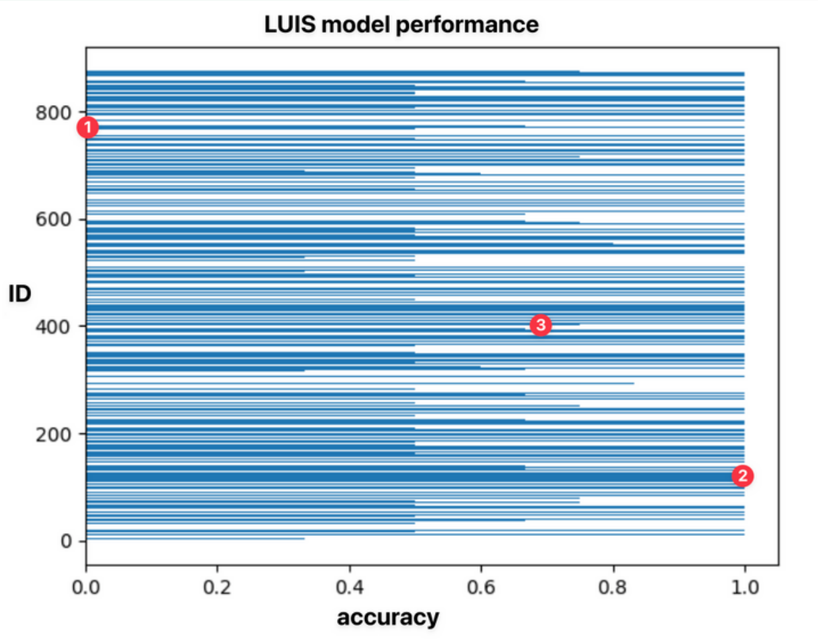
59,71 %

A partir des données de test, on obtient une précision de 59,71% entre le modèle prévu et la libéralisation effective des données originales.

Chaque énoncé a fait l'objet d'une évaluation.

- Un certain nombre d'énoncés ont été évalués comme étant **entièrement exacts**.
- Bon nombre d'entre eux **ne prévoient rien**
- D'autres ont un niveau de **précision moyen**.

On peut examiner ces trois types d'énoncés en les observant.




```
test_data[0]
```

```
{'text': 'how about if I leave from Beijing?',
 'intent_name': 'BookFlight',
 'entity_labels': [{'entity_name': 'or_city',
   'start_char_index': 26,
   'end_char_index': 33}]}
```

```
evaluator.y_pred[0]
```

```
{'geographyV2': {'value': 'Beijing', 'type': 'city'}, 'or_city': 'Beijing?'}
```

```
evaluator.y_true[0]
```

```
{'or_city': 'Beijing'}
```

Tester LUIS

Bartcus Marius | 2023|OC

/07

CHATBOT

Étape 1

Clonner et utilisé https://github.com/microsoft/BotBuilder-Samples/tree/main/samples/python/**21.corebot-app-insights**

Étape 2

Connecter le Bot au service LUIS pour semantiser les messages d'utilisateur et recuperer les 5 unité demandé.

Étape 3

Implementer les dialogues: demande d'information, annulation, confirmation

Étape 4

Implementer les tests unitaire: luis et dialogues.

Étape 5

Déployé sur Azure Web App.

Étape 6

Utiliser Azure App Insights pour l'analyse de l'activité des utilisateurs

Étape 7

Utiliser Bot Emulator en local pour tester la communication avec le chatbot

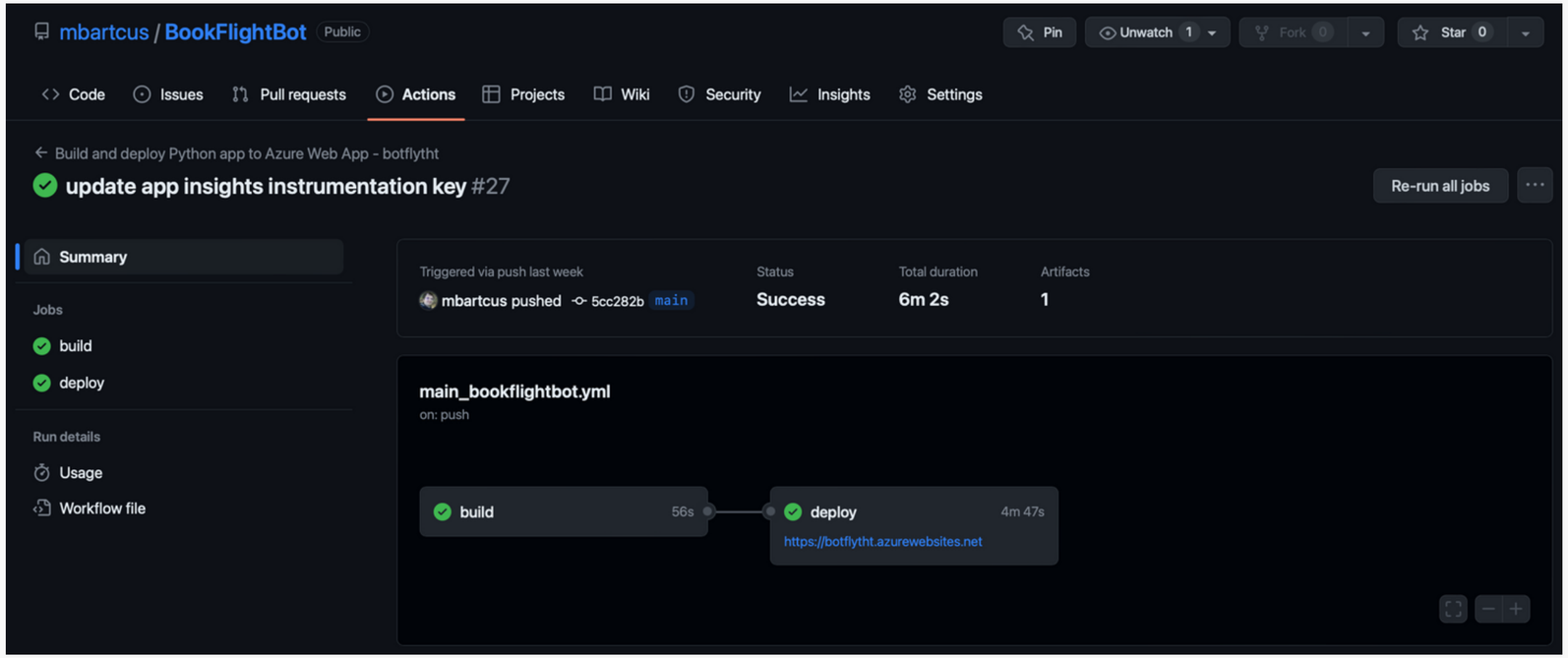
Seconde étape du projet

- On doit construire une chatbot en utilisant Microsoft Bot Framework SDK (python)
- Et le déployé sur Azure Web App

Bartcus Marius | 2023|OC

/08

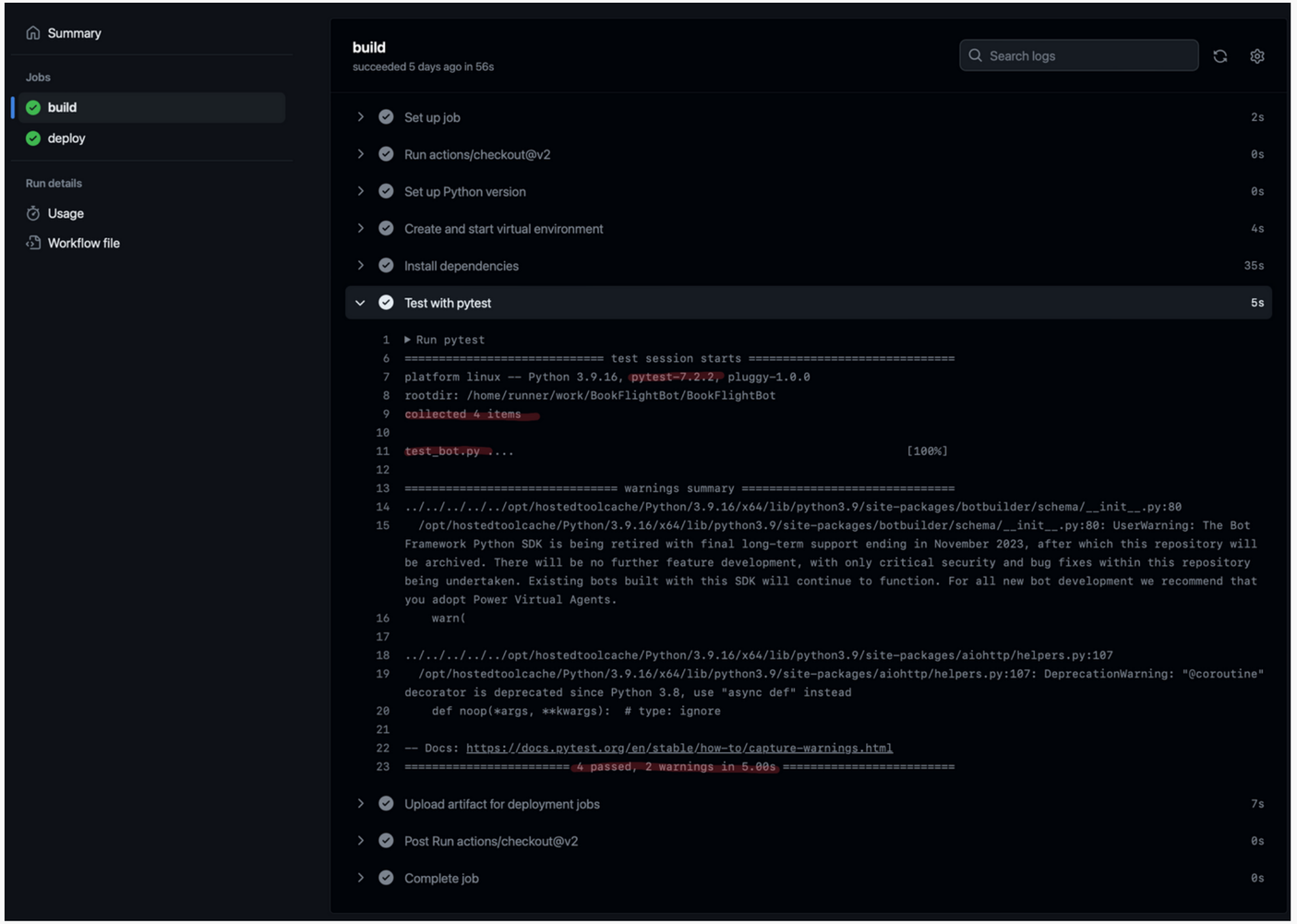
GitHub Actions



Service d'automatisation de workflows qui permet aux développeurs de créer, tester et déployer du code de manière continue directement depuis leur dépôt Github.

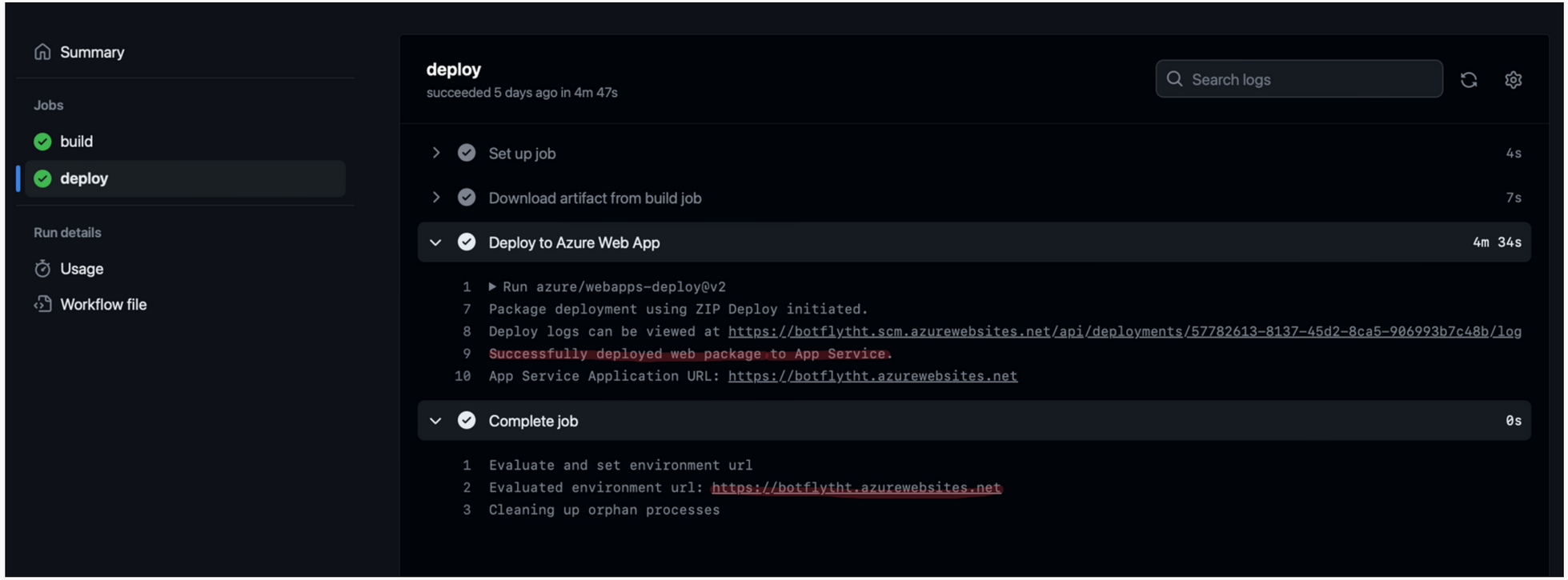
/09

Tests Unitaire



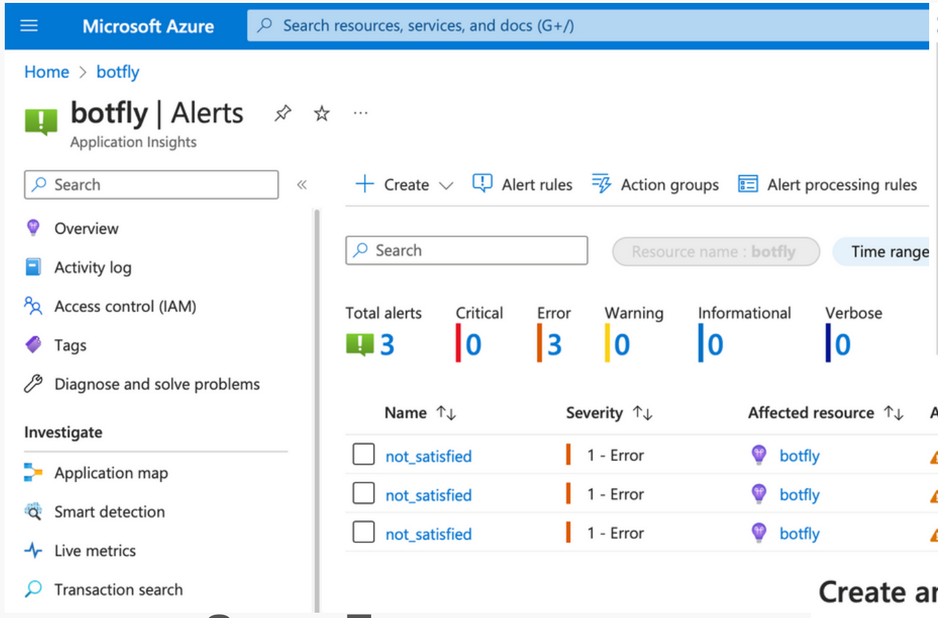
Les tests unitaire sont realisé pour chaque push sur github et à chaque build.

- Tester LUIS
- Tester une réservation étape par étape
- Tester une annulation de réservation
- Tester une réservation en fournissant toutes les informations en une seule fois



Déployment

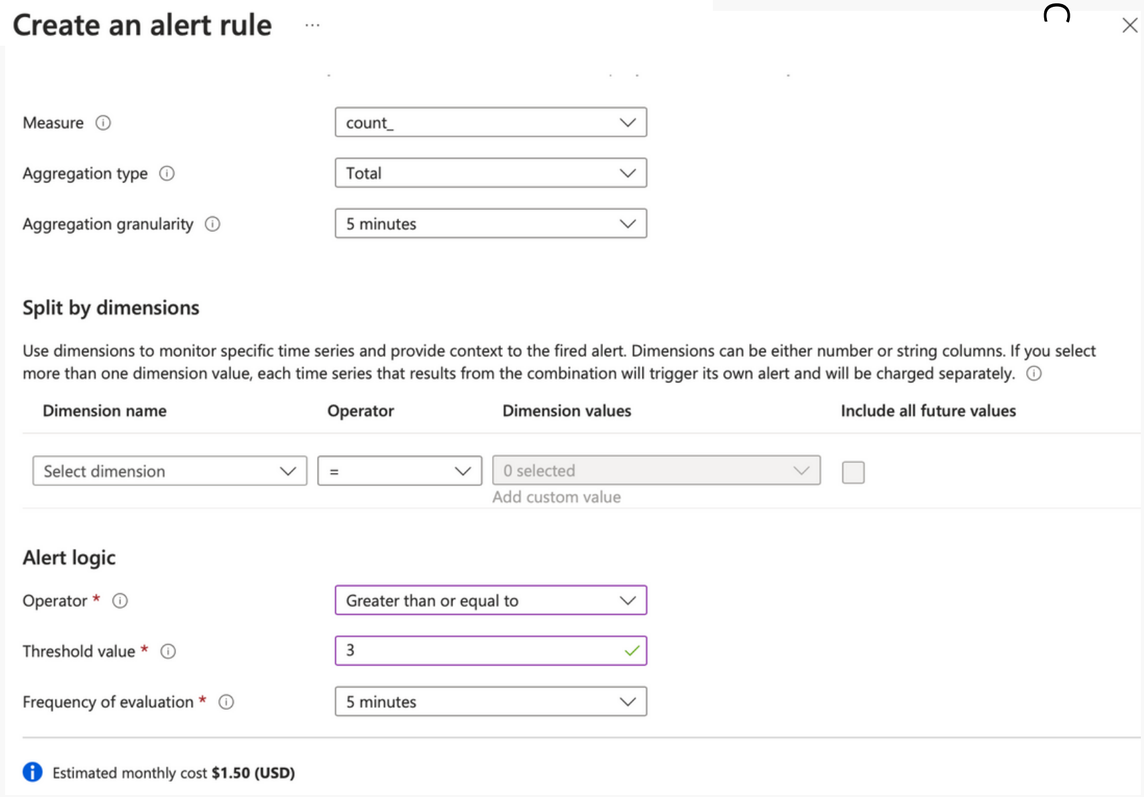
Le déployment est realisé automatiquement pour chaque push sur github.



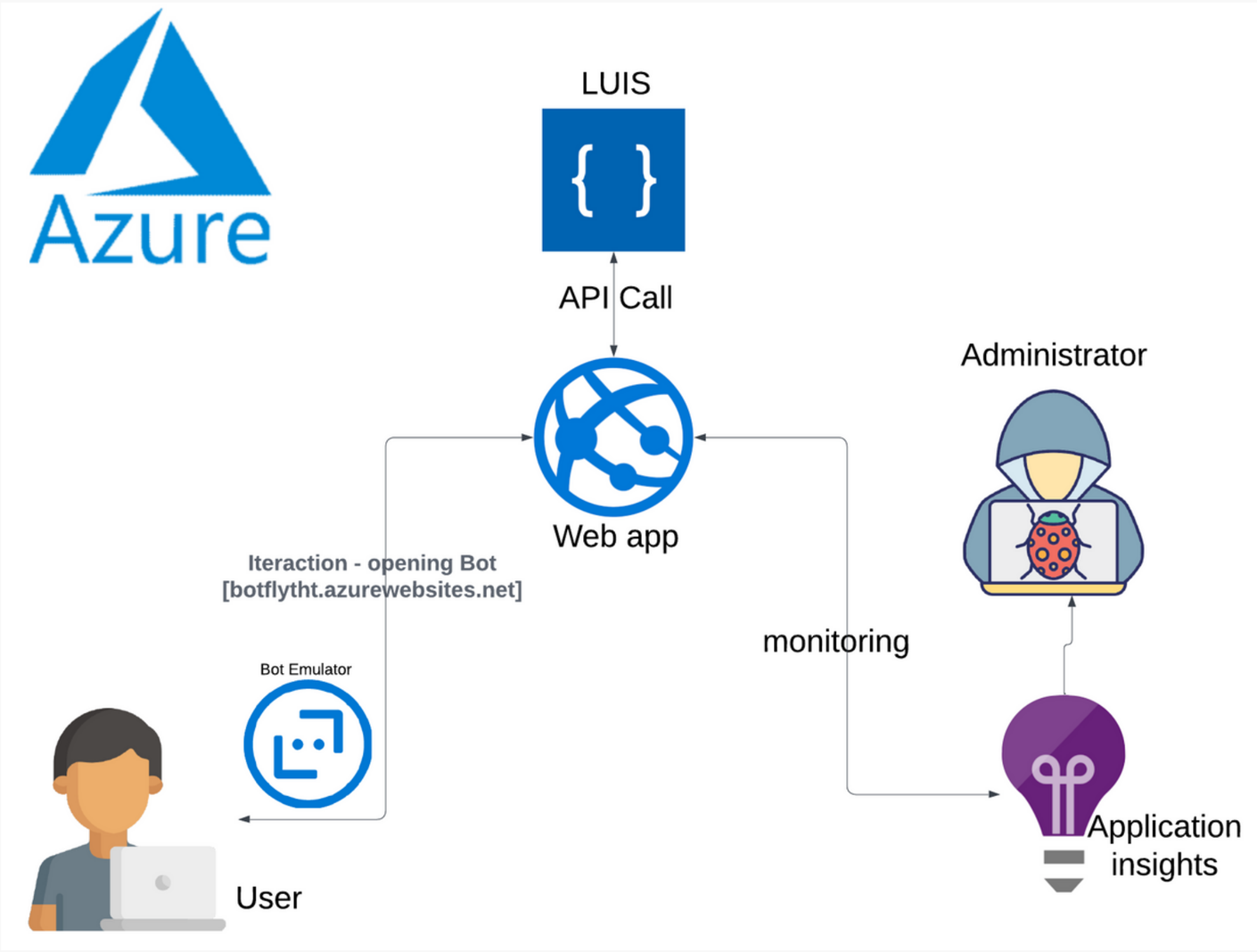
App Insights

- Pour détecter les mauvaises interactions (expérience) entre le chat bot et l'utilisateur.
- On traque deux infos qui remontent au App Insights.
  - INFO (lorsque l'utilisateur est satisfait de la réservation proposée par le chatbot.)
  - ERROR (quand l'utilisateur est insatisfait)

Si l'utilisateur ne valide pas 3 fois la réservation sur une période de 5 minutes, une alerte est déclenchée.

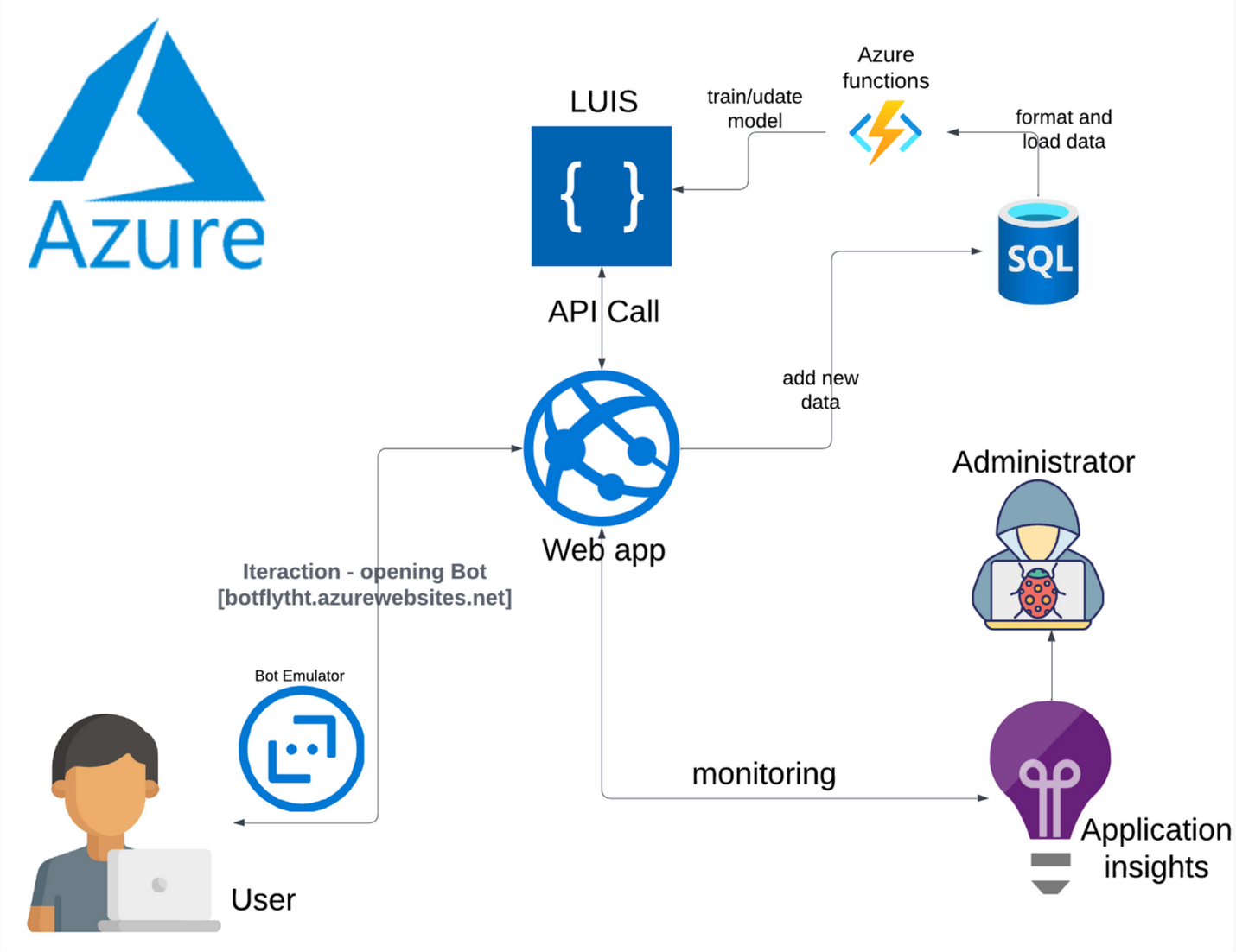


Architecture Actuelle



Cette architecture est destinée au MVP et n'est pas définitive. Une prochaine version pourra enrichir notre modèle en fonction des nouvelles données.

Architecture Cible



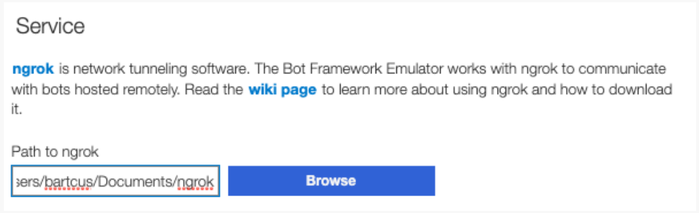
Une architecture d'une prochaine version

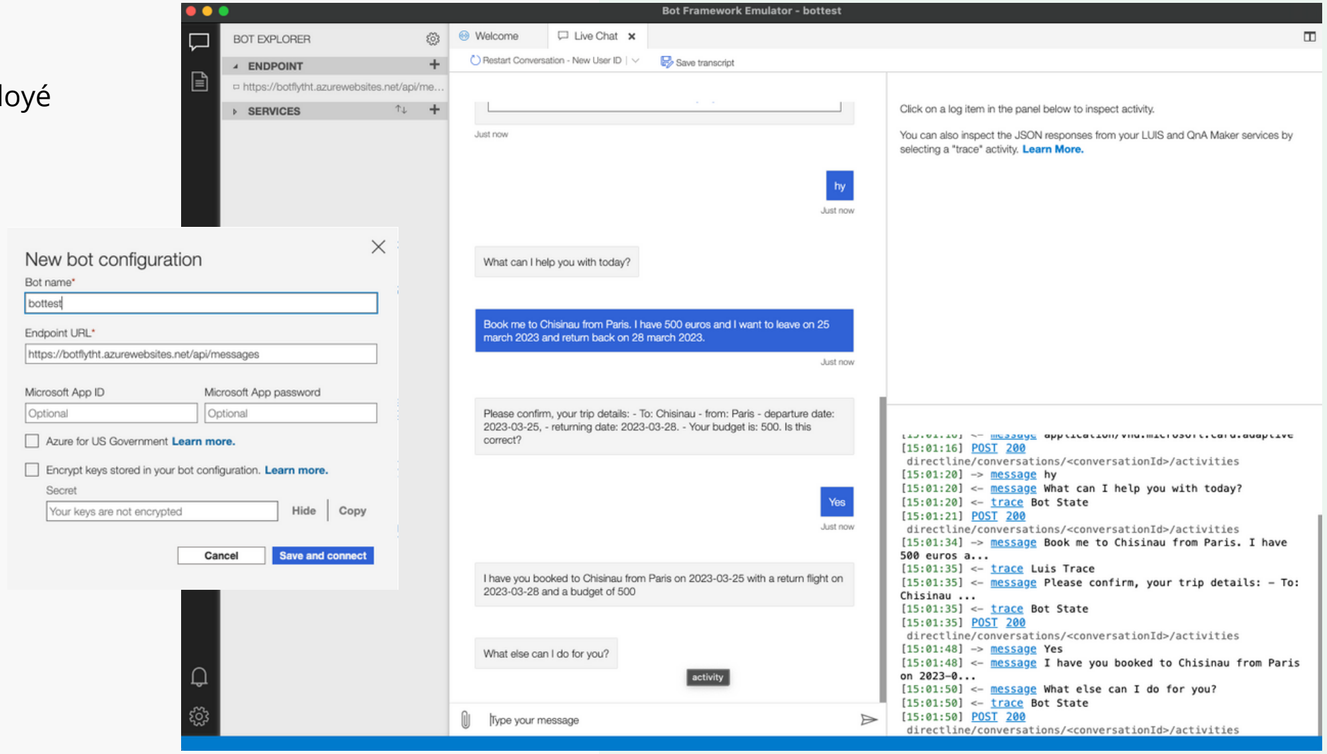
- Récupération des échanges entre les employés de l'entreprise et le Bot afin d'enrichir les données.
- Sur une fréquence d'un mois apprenait un nouveau modèle sur les nouvelles données recueillies et si ses résultats sont mieux, déployés un nouveau modèle.

On parle avec le Bot

On doit installer ngrok pour tester en local le chatbot déployé

```
ngrok -version
ngrok version 2.3.41
```





Conclusions

Ce chatbot est un MVP. Des versions ulterieurs doit être dévéloppé.

Il faudra mettre en place l'intégration continue comme dans l'architecture cible.

De nouvelles entités peuvent être ajoutées (l'aéroport, le nombre de passagers et leurs informations, les services supplémentaires, etc.)

Ajouté d'autre surveillances comme par exemple en observant le temps reponce d'utilisateur.

Merci !