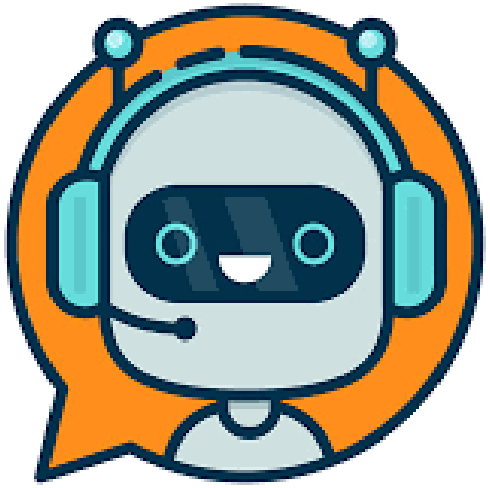
# PROJET 10

**DÉVELOPPEZ UN CHATBOT POUR RÉSERVER DES VACANCES ET PILOTER SA PERFORMANCE**

**#MICROSOFT BOT FRAMEWORK #AZURE LUIS #AZURE APPLICATION INSIGHT #VSCODE #GITHUB #TEST UNITAIRE #KQL KUSTO QUERY LANGAGE #DASHBOARD #AZURE APP SERVICES**



## Ingénieur IA

Développez et intégrez des algorithmes de Deep Learning au sein d'un produit IA

OPENCLASSROOMS

**OUDDANE NABIL**

## SOMMAIRE

Projet 10

# Développez un chatbot pour réserver des vacances


A. **INTRODUCTION**
   1. Contexte
   2. Objectifs
   3. Données

B. **DEVELOPPEMENT**
   1. Création du modèle LUIS
      a. Création d'une ressource Language Understanding LUIS sur AZURE
      b. Modélisation LUIS
   2. Développement du bot pour répondre a notre besoin
      a. prérequis au BOT
      b. Développement du BOT en python
      c. Test du bot sur l'émulateur
      d. Mise en place de 5 tests unitaires
      e. Mise en place des alertes app insights
      f. Création d'un dashboard
      g. monitoring de log bespoke
   3. Déploiement du bot sur azure
      a. workflow github tests unitaires
      b. déploiement de azure web app service
      c. azure bot service

C. **Model UPDATE**

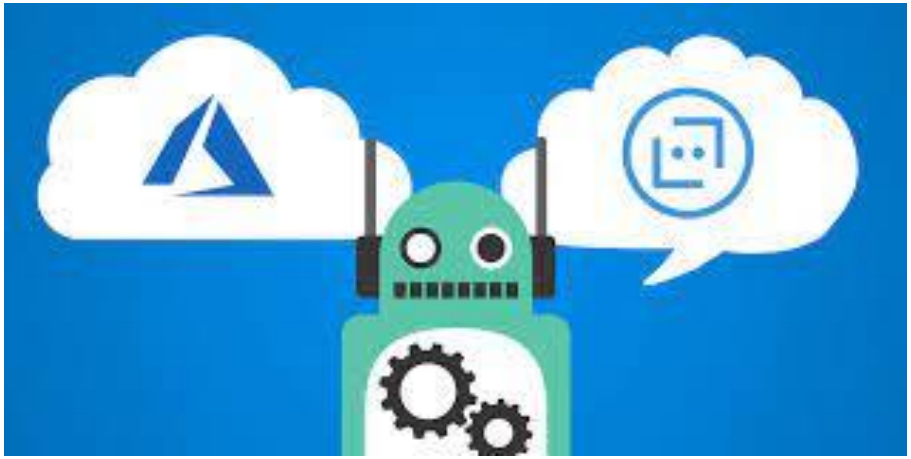
# A INTRODUCTION

# 1. Contexte

FLY ME

- **ENJEU Compétences DU P9:**
  - Chatbot avec **Microsoft Bot framework SDK v4**
  - **Analyse sémantique avec LUIS** d'azure pour identifier les variables nécéssaires
  - Evaluation de la performance du modèle LUIS avec **Azure application insight**

- **ENJEU global:**
  - Développer un chatbot pour aider les utilisateurs à choisir une offre de voyage
  - MVP: support interne à la réservation d'un billet d'avion



# 2. Objectifs

- SCRIPT du pipeline complet  stocké sur github
  - Application web chatbot développé grâce au **microsoft  Bot Builder SDK et au service cognitif LUIS**
    - Demontrer les fonctionnalités de l'application à des futurs utilisateurs
  - Outils de suivi et d'analyse de l'activité du chatbot en prod  avec Azure application insight
    - Pour rassurer les managers

- Méthodologie
  - pilotage de la performance du modèle en production
    - Critère d'évaluation du modèle LUIS
    - Schéma du mécanisme d'évaluation du modèle en prod
    - Modalités de mise à jour du modèle

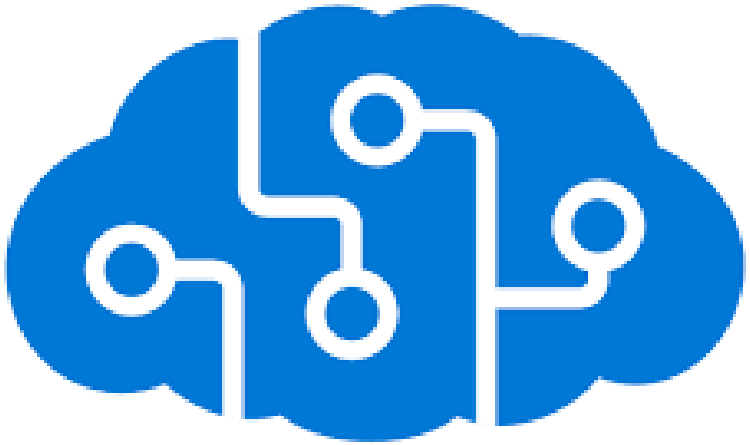
# 3. données

- **Données utilisateur :**
  - https://s3-eu-west-1.amazonaws.com/static.oc-static.com/prod/courses/files/AI+Engineer/Project+10%C2%A0-+D%C3%A9veloppez+un+chatbot+pour+r%C3%A9server+des+vacances/frames.zip
    - Historiques d'échanges entre chatbot et utilisateur
    - JSON format. 5 champs principaux
      - user_id
      - wizard_id
      - Id
      - userSurveyRating
      - turns

B

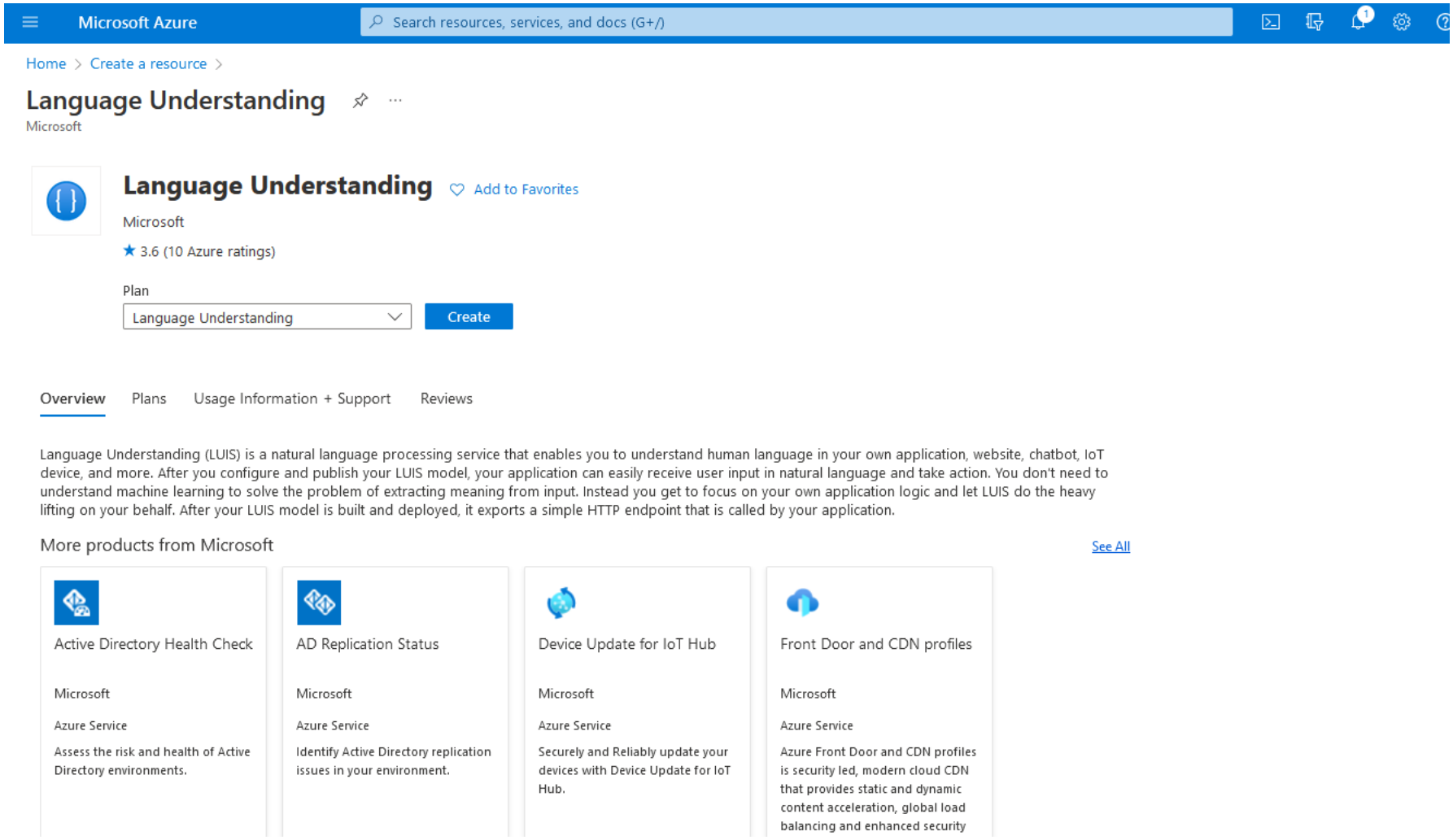
B

# 1- CREATION DU MODELE LUIS

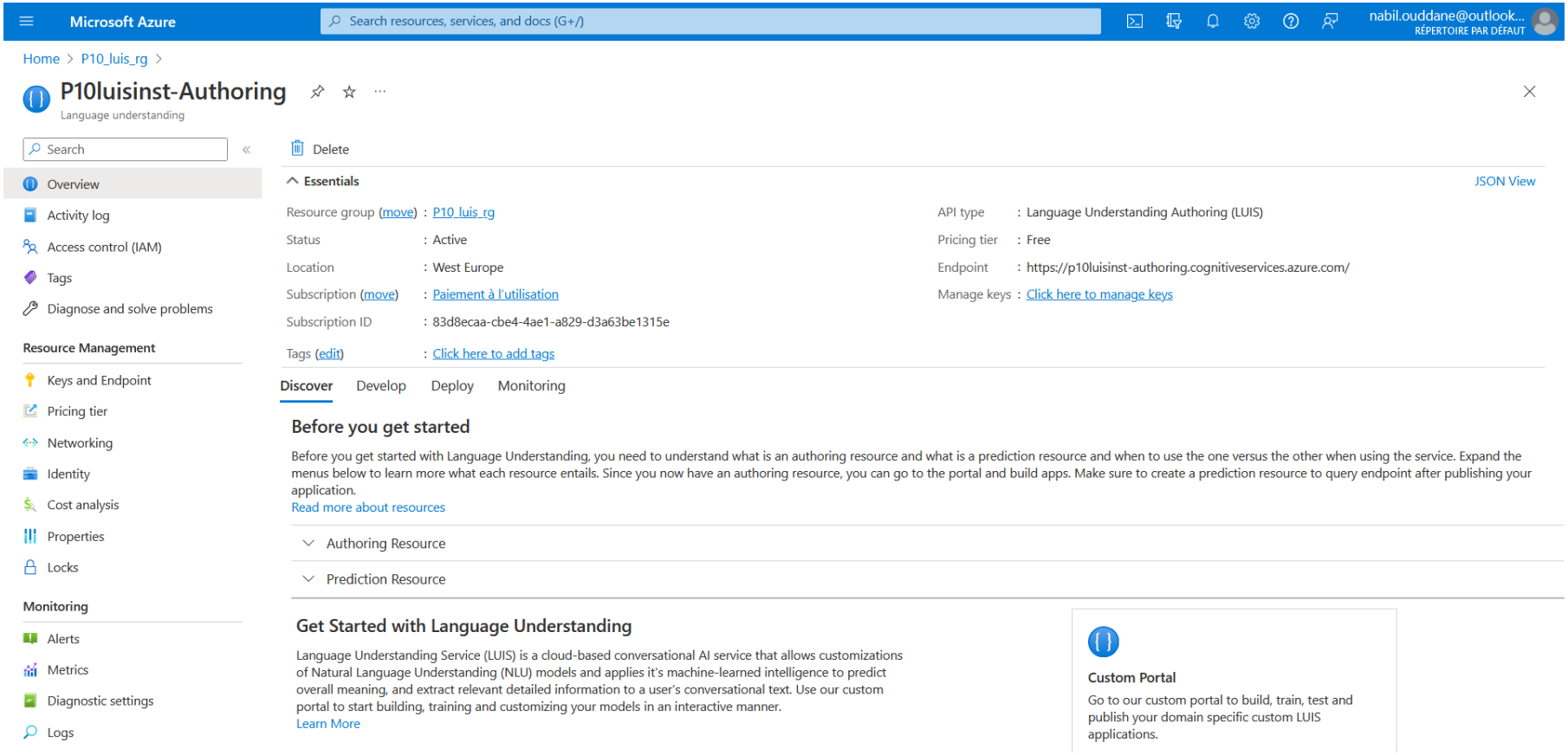


# 1-a: Creation d'une ressource Language Understanding LUIS sur AZURE

***pricing authoring: F0: 5 calls/sec 1M calls /mois - pricing prediction: F0: 5 calls/sec 10K calls /mois***

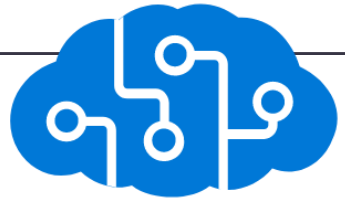


# 1-a: Creation d'une ressource Language Understanding LUIS sur AZURE *(LUIS va être remplacé par CLU)*



# 1-b: Modelisation LUIS *(cf fichier jupyter p10_luis_0.2.ipynb)*

## *3 POSSIBILITES : LIB PYTHON / API / PORTAIL*



**IMPORT et COMPREHENSION**
des fichiers d'exemples de dialogues frames,json

**FORMATAGE pour FEEDING de LUIS**
des fichiers d'exemples de dialogues frames,json

**Analyse et extraction à partir des turns de l'intention et des entités avec leurs positions:**
- intentions
- dst_city
- or_city
- budget
- str_date
- End_date

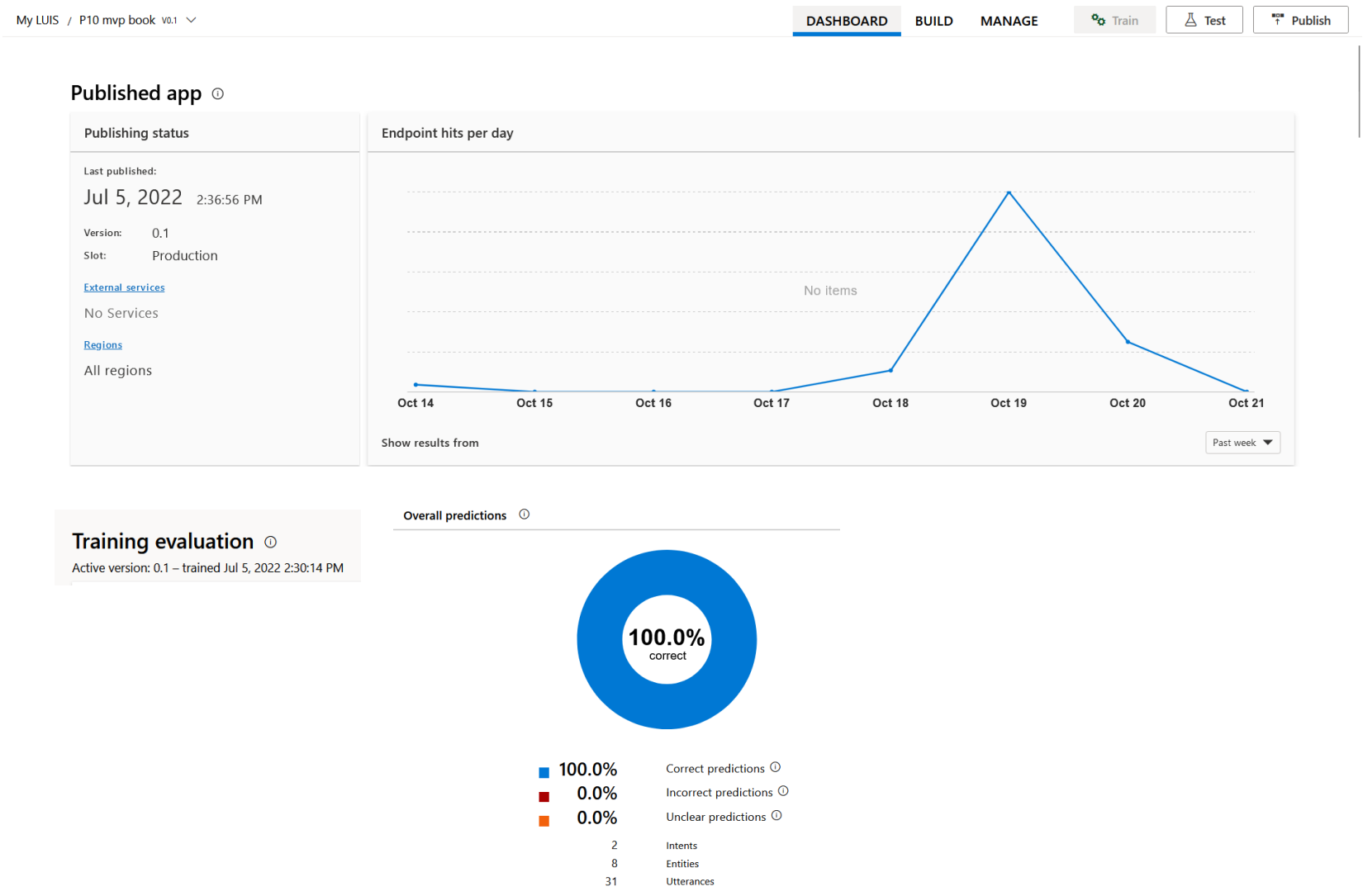
**CREATION de l'APPLICATION LUIS et du CLIENT AUTHORING**

- Modele d'intention
- Prebuilt entity (money, geography v2, datetime v2)
- Entities:
- Budget avec feature « money »
- Or_city avec feature « geography v2 »
- Dst_city avec feature « geography v2 »
- Str_date avec feature « datetime v2»
- end_date avec feature « datetime v2 »

**ENTRAINEMENT:**
À partir d'une 30taine d'exemples formatés issus de frames,json

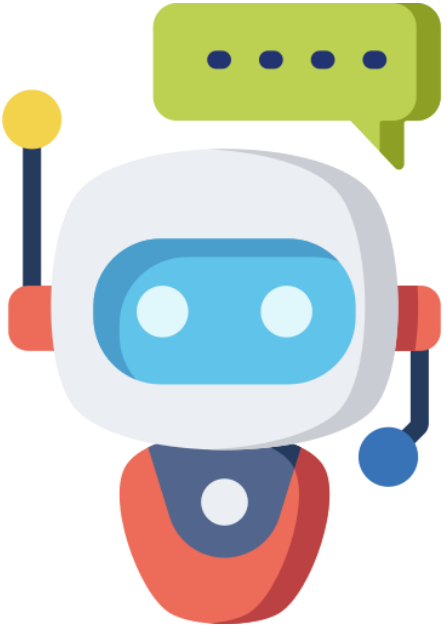
**PUBLICATION**

# 1-b: Modelisation LUIS Dashboard

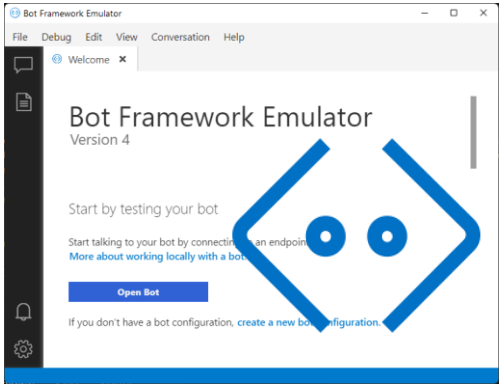


B

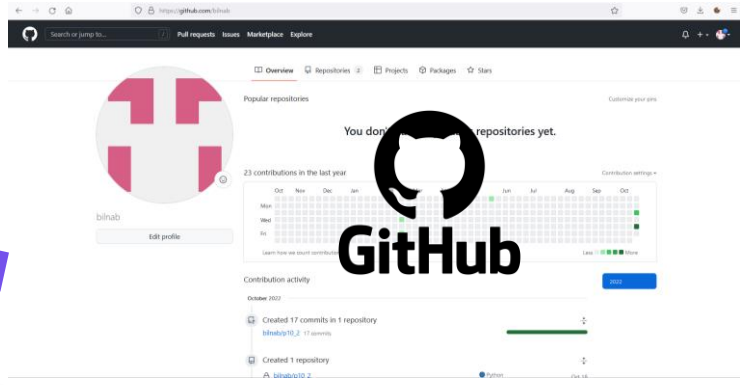
# 2- DEVELOPPEMENT DU BOT POUR REPONDRE A NOTRE BESOIN



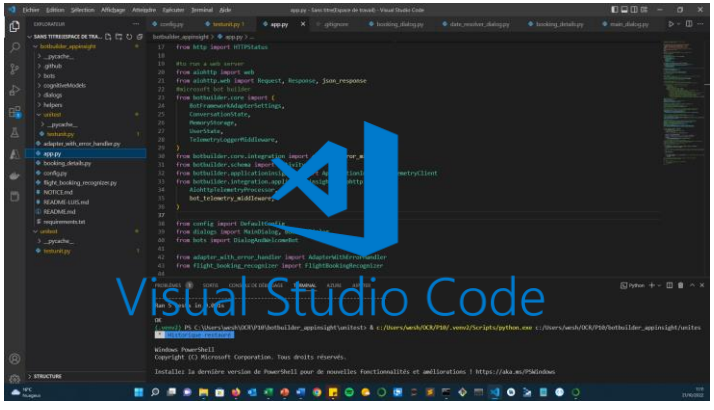
# 2-a: prérequis au BOT

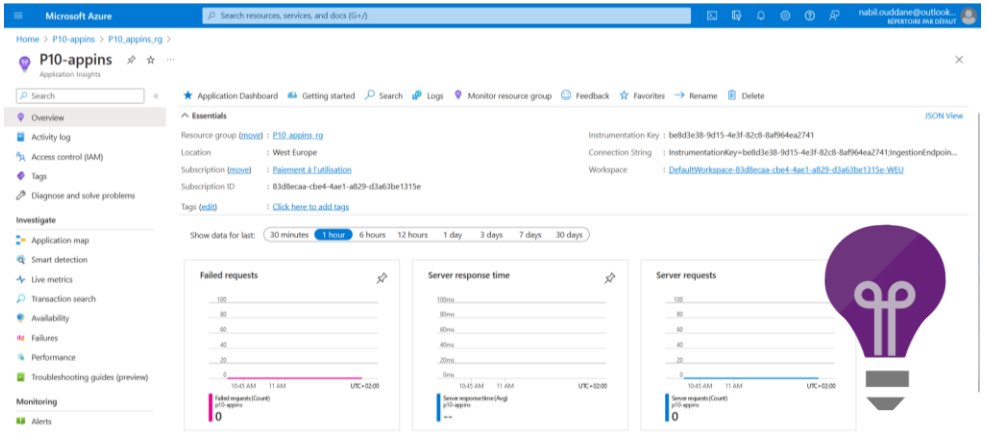







- Bot Framework Emulator
- Github
- Vscode
- Azure App insight



# 2-b : Developpement du BOT en python

**CLONAGE DU REPO MICROSOFT BOTBUILDER SAMPLE**
contenant un exemple de mise en œuvre du botbuilder en python avec prise en charge du monitoring app insights

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

**PARAMETRAGE DE CONFIG.PY**

PORT, LUIS_APP ID, KEY & HOSTNAME, APPINSIGHTS KEY

**ADAPTATION DE BOOKING_DETAILS.PY**

DESTINATION, ORIGIN, DEPARTURE_DATE, RETURN_DATE, BUDGET

**LEGERE ADAPTATION DE APP.PY**

- client de telemetry pour LUIS à ajouter sinon pas de log dans appinsight
- host='0.0.0.0' OU 'LOCALHOST' suivant cloud ou local

**ADAPTATION DE HELPERS/LUIS_HELPER.PY**

- Recuperation des données de notre modèle LUIS en faisant les bons GET
- Prise en compte d'un daterange si besoin

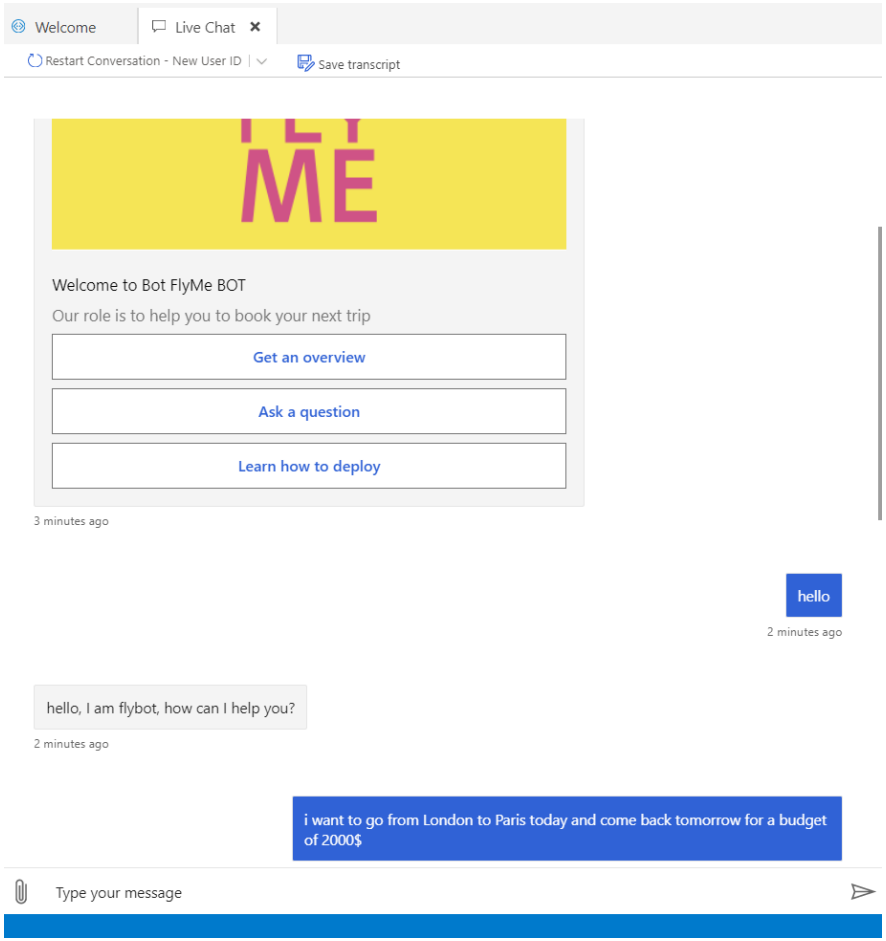
**ADAPTATION DES DIALOGUES DU REPERTOIRE DIALOGS**

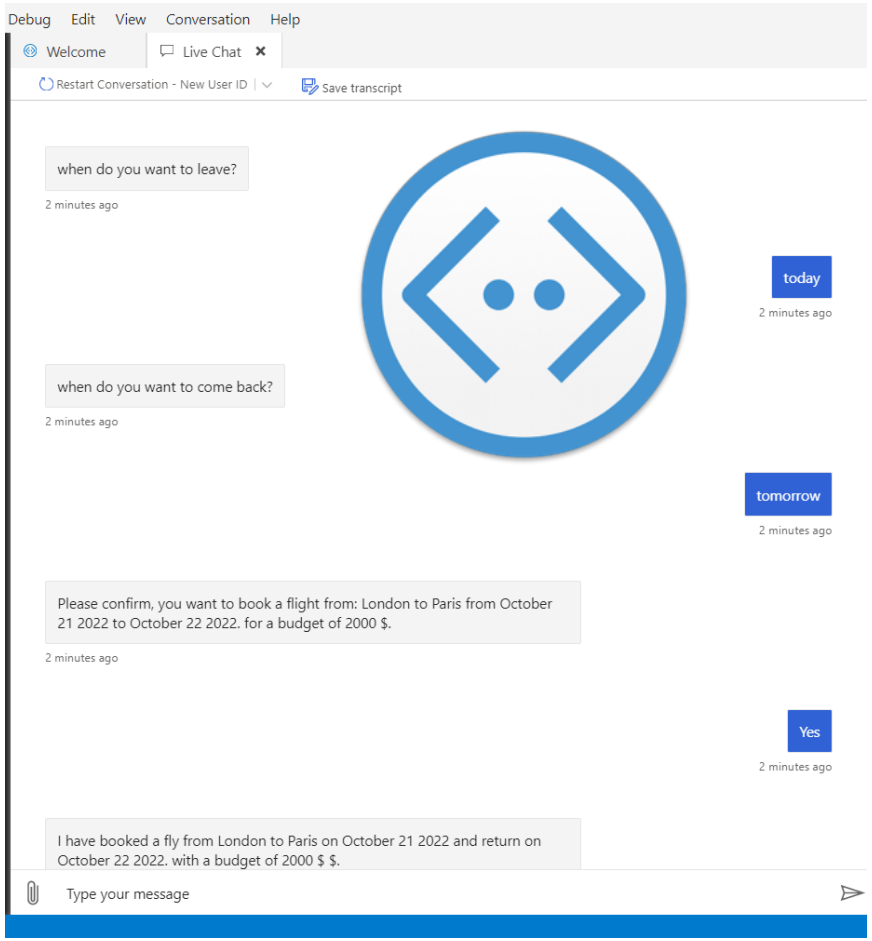
- Main_dialog
- Booking_dialog
- Date_resolver_dialog

**ECRITURE DE TEST UNITAIRES**

- LIBRAIRIE UNITTEST
- 5 tests basés sur le bon fonctionnement de LUIS

# 2-c : Test du bot sur l'émulateur



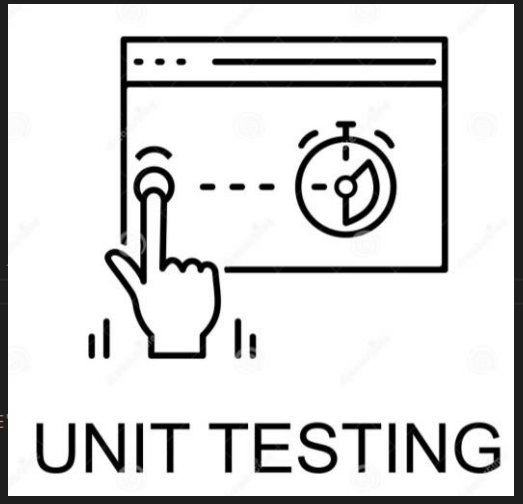



# 2-d : Mise en place de 5 tests unitaires

config.py M | testunit.py 1 | app.py M | .gitignore | booking_dialog.py | date_resolver_dialog.py | booking_details.py | main_dialog.py

unitest > testunit.py > TestLuisReco

```
14  request ='I want to go from Paris to London from the 2022-10-22 until the 2022-10-24 for a budget of 1000 $'
15  response = clientRuntime.prediction.resolve(CONFIG.LUIS_APP_ID, query=request)
16
17  #for c, entity in enumerate(response.entities):
18  #    print(entity)
19
20  # classe scenario de test qui est une classe fille de unittest.TestCase
21  class TestLuisReco(unittest.TestCase):
22
23      def test_intent(self):
24          """intention testing"""
25          #self.assertEqual(response.top_scoring_intent.intent,'book')
26          self.assertTrue(response.top_scoring_intent.intent == 'book', "intention testing
27
28      def test_origin(self):
29          """origin testing."""
30          for c, entity in enumerate(response.entities):
31              if entity.type == 'or_city':
32                  self.assertTrue(entity.entity == 'paris', "origine city testing - FAILURE
33
34      def test_dest(self):
35          """destination testing."""
```




PROBLÈMES 1 | SORTIE | CONSOLE DE DÉBOGAGE | TERMINAL | AZURE | JUPYTER — Python

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\Users\wesh\OCR\P10\botbuilder_appinsight\unitest> & c:/Users/wesh/OCR/P10/.venv2/Scripts/Activate.ps1
(.venv2) PS C:\Users\wesh\OCR\P10\botbuilder_appinsight\unitest> & c:/Users/wesh/OCR/P10/.venv2/Scripts/python.exe c:/Users/wesh/OCR/P10/botbuilder_appinsight/unites
t/testunit.py
.....
----------------------------------------------------------------------
Ran 5 tests in 0.001s

OK
(.venv2) PS C:\Users\wesh\OCR\P10\botbuilder_appinsight\unitest>
```

# 2-e: Mise en place des alertes app insights

- Création d'**alertes** déclenchant **l'envoi de mails** à partir de requêtes sur les logs de la table **CustomEvents** de application insights:
  - *Score moyen LUIS de l'intention < 0,5 sur n jours glissants*
  - *Nombre d'intention « book » / Nombre d'intention « none » < 1 sur 4 jours glissants*
  - *Nombre de waterfall bookdialogs debutés / nombre de waterfall bookdialogs terminés < 0,5 sur n jours glissants => indication d'interruption en cours de dialogue*
  - *Nombre de waterfall Maindialogs debutés / nombre de waterfall Maindialogs terminés < 0,5 sur n jours glissants => indication de booking finalisés*
  - *Nombre anormal (trop grand) d'activités d'un meme utilisateur*

Home > P10-appins | Alerts >

## Alert rules ···

Create | Columns | Refresh | Export to CSV | Open query | Delete | Enable | Disable

Search | Target resource type : all | Target scope : P10-appins | Subscription : all | Signal type : all | Severity : all | Status : Enabled

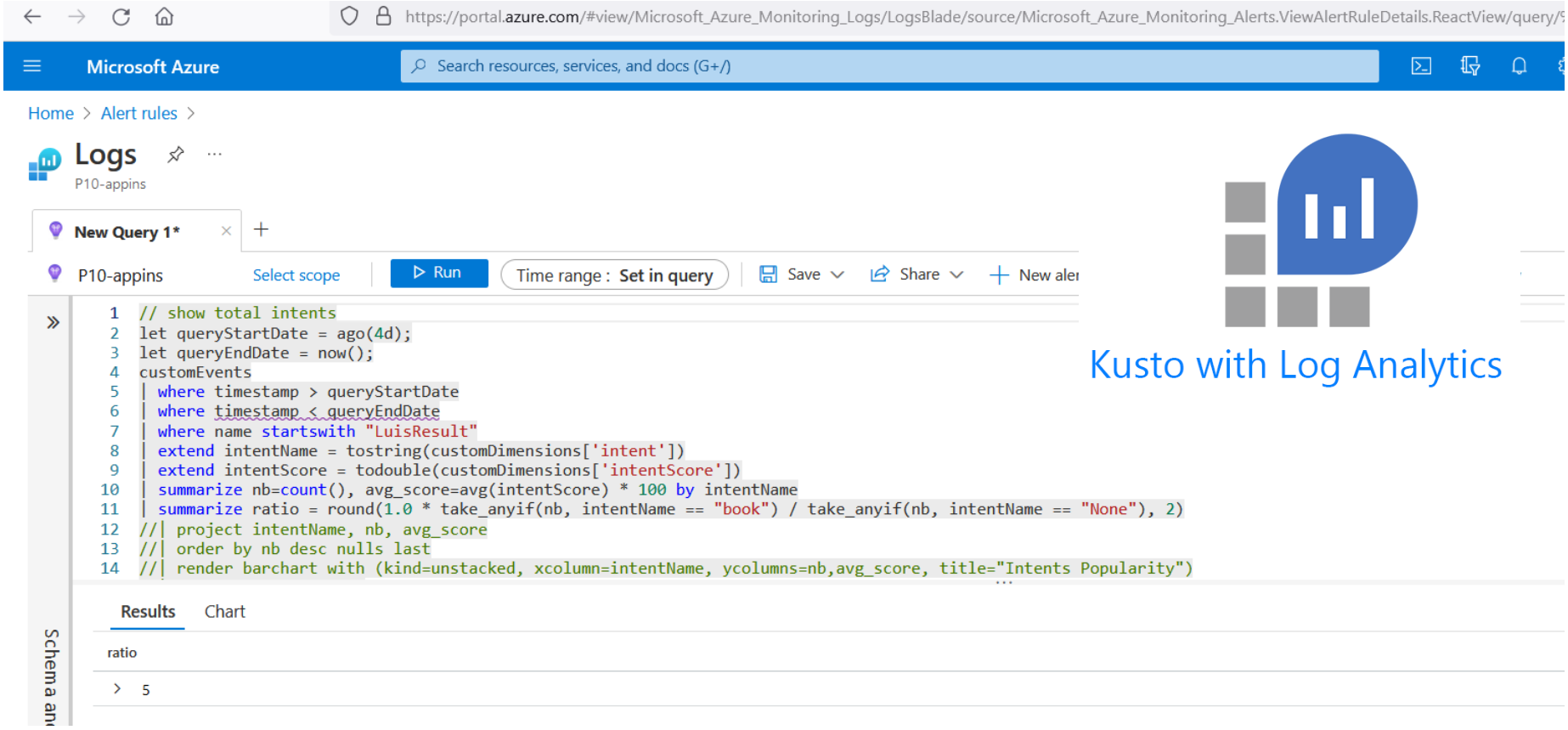
Showing 1 to 6 of 6 Alert rules.

No grouping

| Name | Condition | Severity | Target scope | Target resource type | Signal type | Status |
|---|---|---|---|---|---|---|
| Failure Anomalies - P10-appins | Failure Anomalies detected | 3 - Informational | p10-appins | Application Insights | Smart detector | Enabled |
| P10_Book_intent_score_tresh_Warning | avg_score <= 0.5 In selected dim... | 2 - Warning | P10-appins | Application Insights | Log search | Enabled |
| P10_Book_none_intent_ratio_tresh_Warning | ratio <= 1 | 2 - Warning | P10-appins | Application Insights | Log search | Enabled |
| P10_BookDialog_tresh_Warning | Percentage <= 0.5 In selected di... | 2 - Warning | P10-appins | Application Insights | Log search | Enabled |
| P10_MainDialog_tresh_Warning | Percentage <= 0.5 In selected di... | 2 - Warning | P10-appins | Application Insights | Log search | Enabled |
| P10_user_abnormal_activity_attack_Warning | Count > 10 | 2 - Warning | P10-appins | Application Insights | Log search | Enabled |

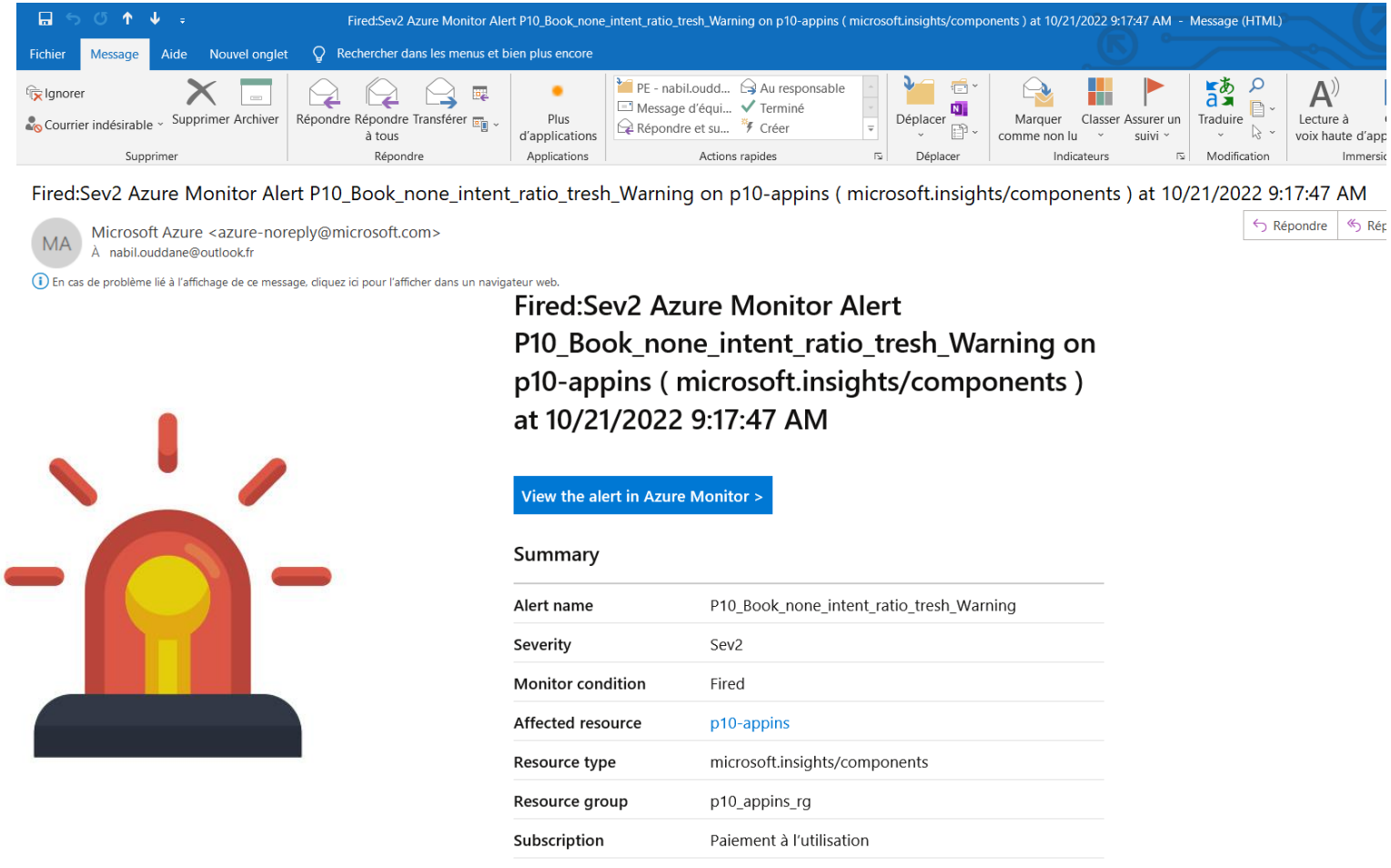
# 2-e: Mise en place des alertes app insights

- Des ajustements sont évidemment nécessaires en fonction de la volumétrie
- Les requetés sont écrites en **KQL** kusto query langage



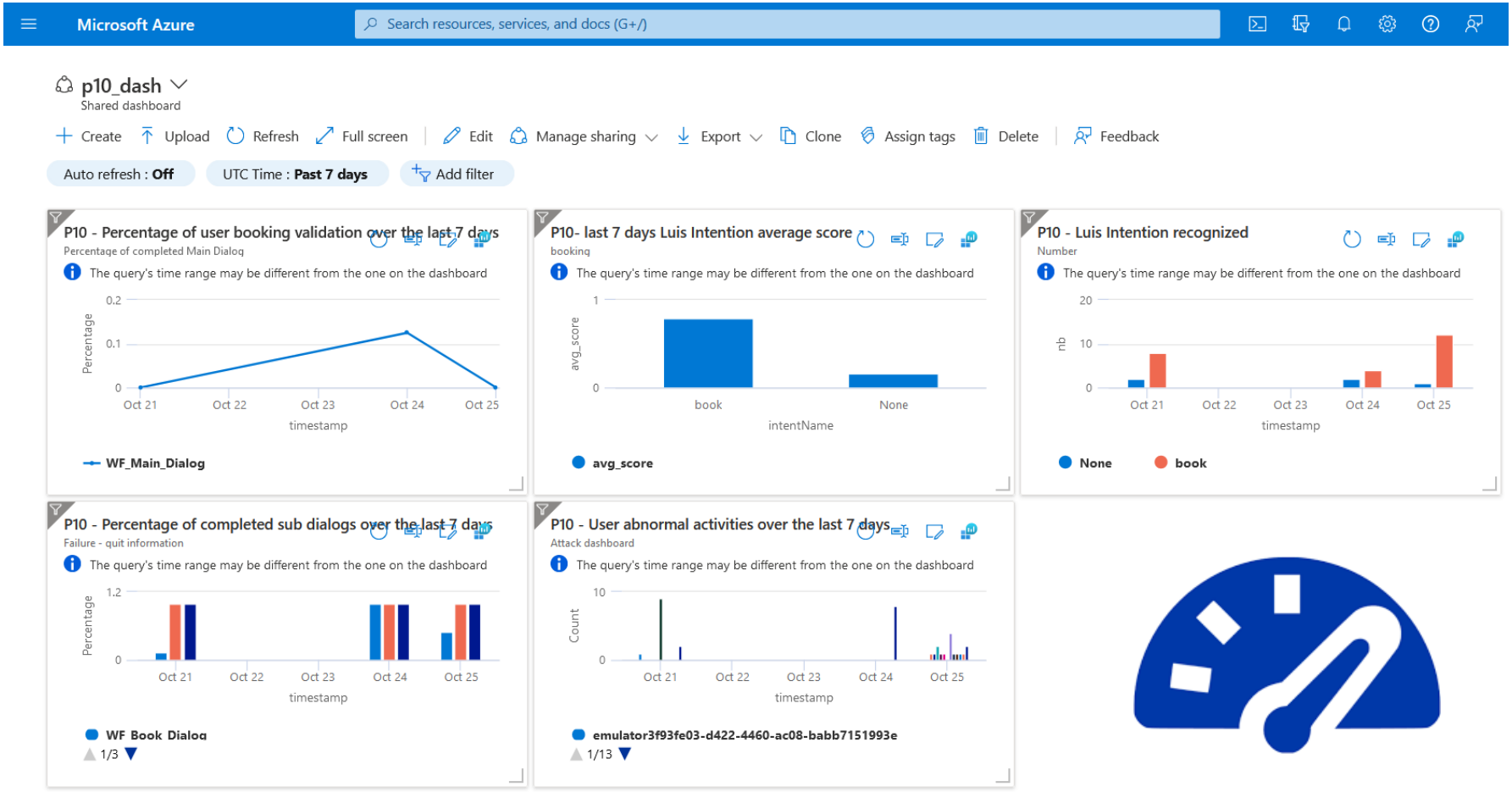
# 2-e: Mise en place des alertes app insights

- Réception de mails



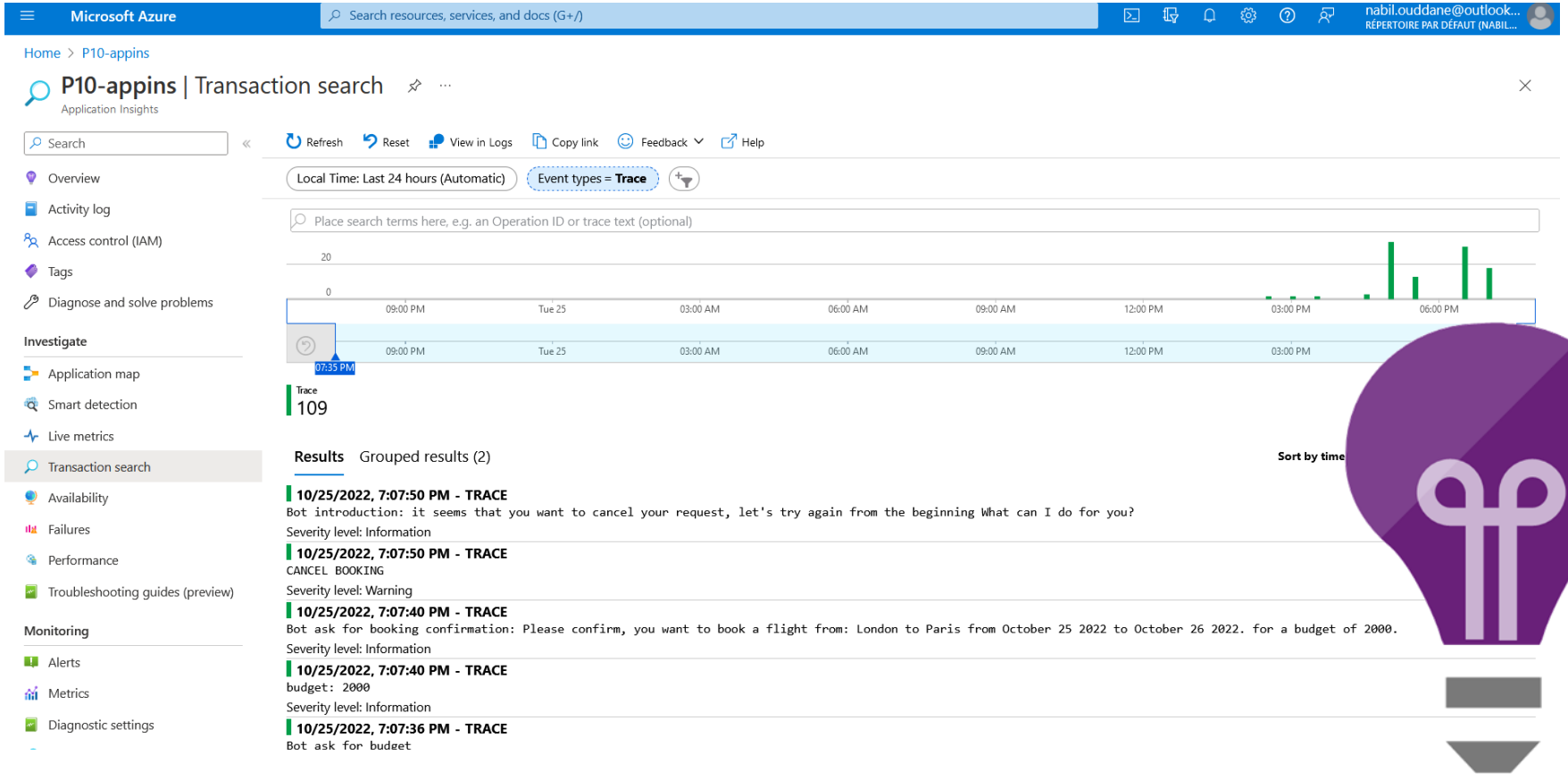
# 2-f: Creation d'un dashboard

- Suivi de plusieurs métriques importantes pour le modèle grâce aux requetés KQL
- Possibilité de rajouter d'autres métriques si besoin



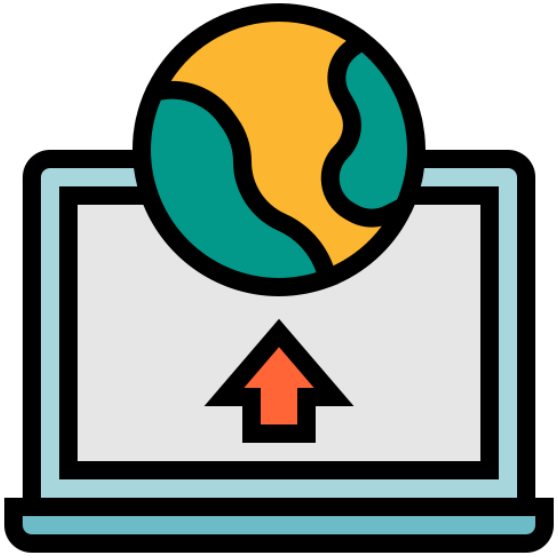
# 2-g: monitoring de log bespoke

- Export des logs vers app insights
- Plusieurs types de messages stockés dans la table TRACE: info / warning / error
  - Permettant une investigation plus fine sur les problèmes rencontrés



B

# 3- DÉPLOIEMENT DU BOT SUR AZURE



# 3 : Deploiement

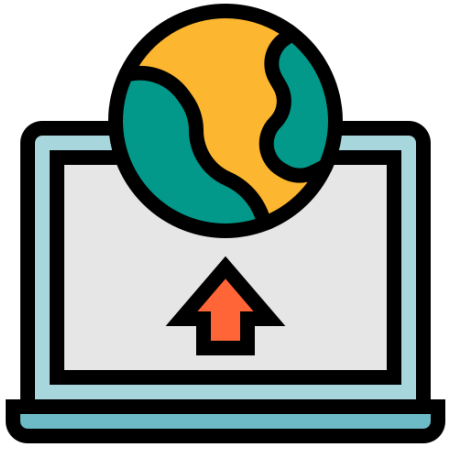
**PUSH DU CODE SUR GITHUB**

- CREATION D'UN FICHIER YAML POUR EXECUTER LES TESTS UNITAIRES
- SUR EVENEMENT DE PULL REQUEST

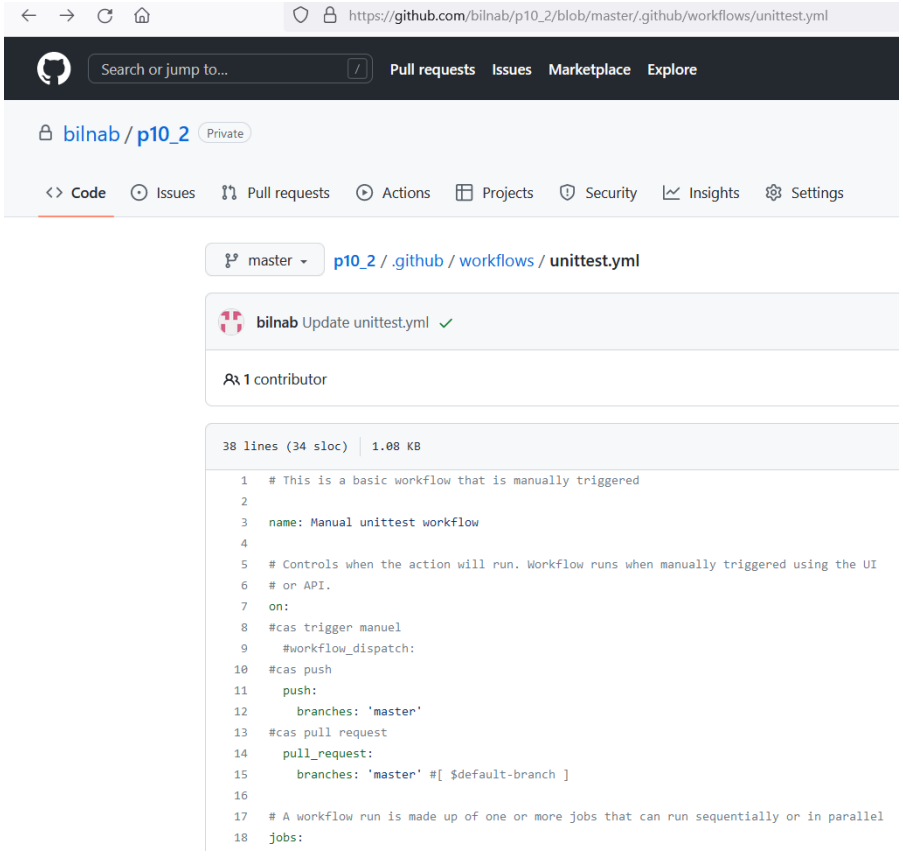
**CREATION D'UNE RESSOURCE AZURE APP**
SERVICE QUI HEBERGERA L'APPLICATION WEB DU BOT

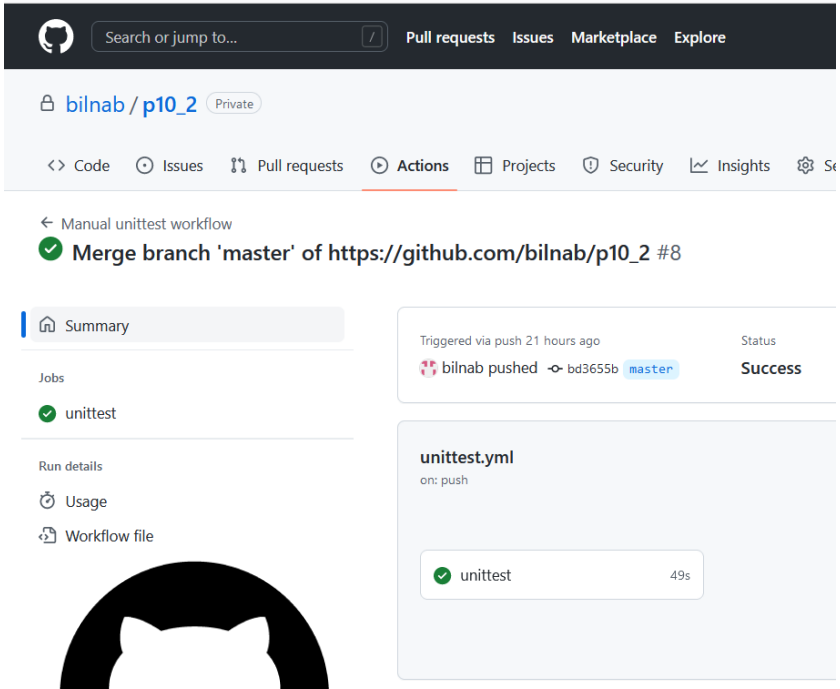
- PARAMETRAGE DE DEPLOIEMENT CONTINU A PARTIR DE GITHUB
- Attention à avoir un fichier requirements.txt propre (pas de conflit de lib )

**CREATION D'UN BOT SERVICE EN ROLE DE CHANNEL**



# 3-a: workflow github tests unitaires

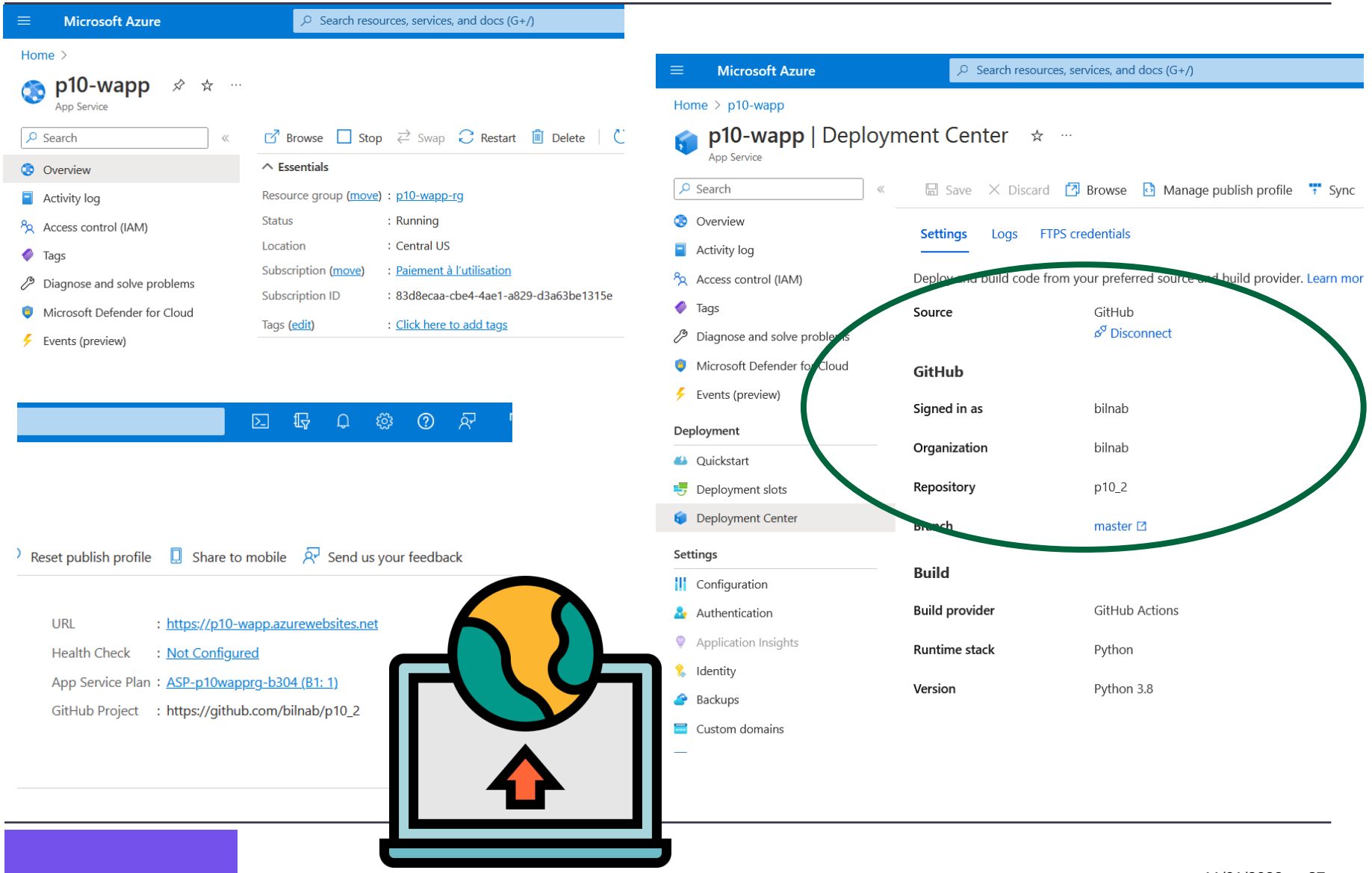


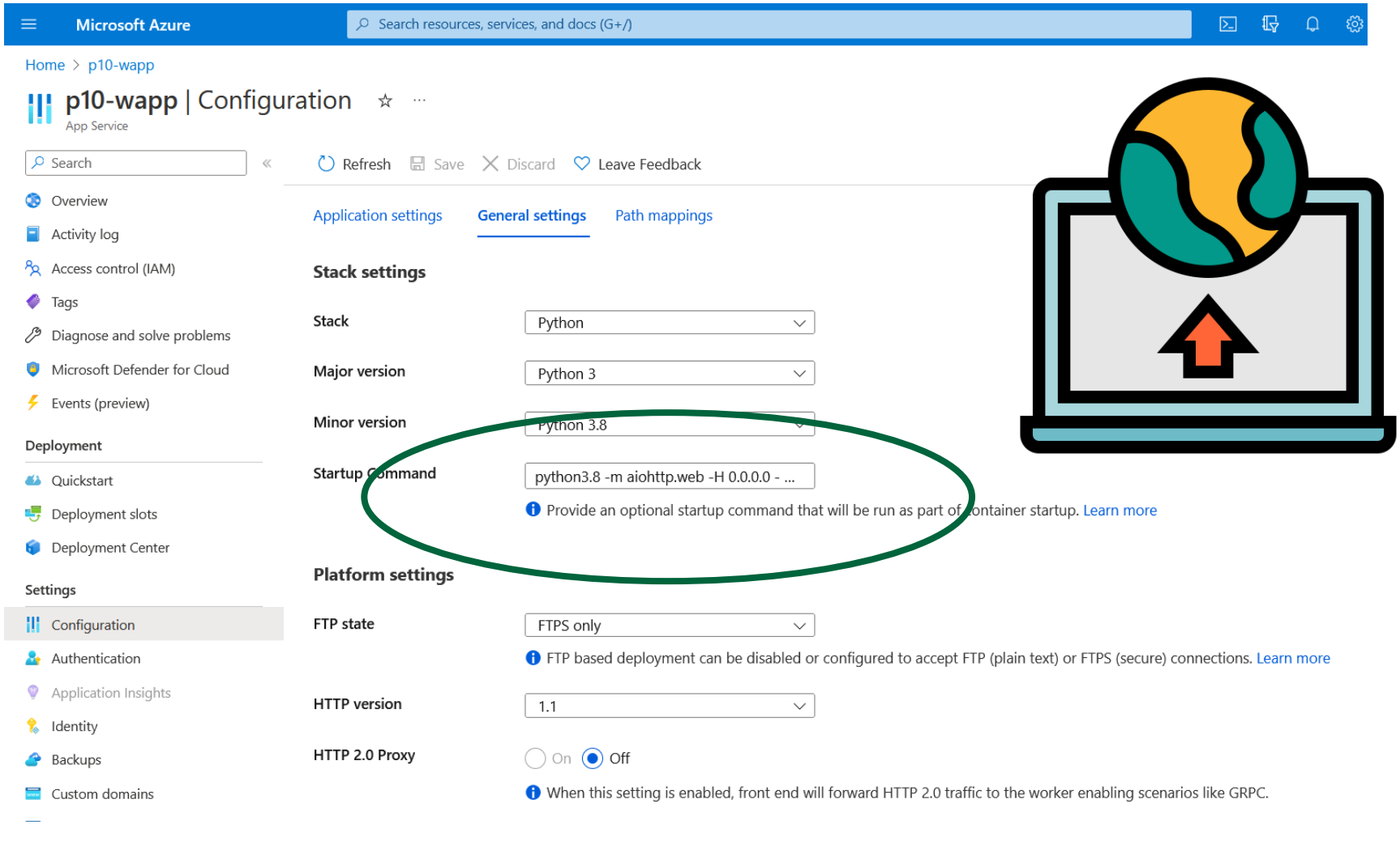

# 3-a: workflow github tests unitaires: fichier YAML


```yaml
# This is a basic workflow that is manually triggered

name: Manual unittest workflow

# Controls when the action will run. Workflow runs when manually triggered using the UI
# or API.
on:
#cas trigger manuel
  #workflow_dispatch:
#cas push
  push:
    branches: 'master'
#cas pull request
  pull_request:
    branches: 'master' #[ $default-branch ]

# A workflow run is made up of one or more jobs that can run sequentially or in parallel
jobs:
  # This workflow contains a single job called "unittest"
  unittest:
    # The type of runner that the job will run on
    runs-on: ubuntu-latest

    # Steps represent a sequence of tasks that will be executed as part of the job
    steps:
    # Runs a single command using the runners shell
    - uses: actions/checkout@v3
    - name: Set up Python
      uses: actions/setup-python@v4
      with:
        python-version: '3.8'
    - name: Install dependencies
      run: |
        python -m pip install --upgrade pip
        pip install -r requirements.txt
    - name: Test with unittest
      run: |
        python -m unittest -v unitest/testunit.py
```

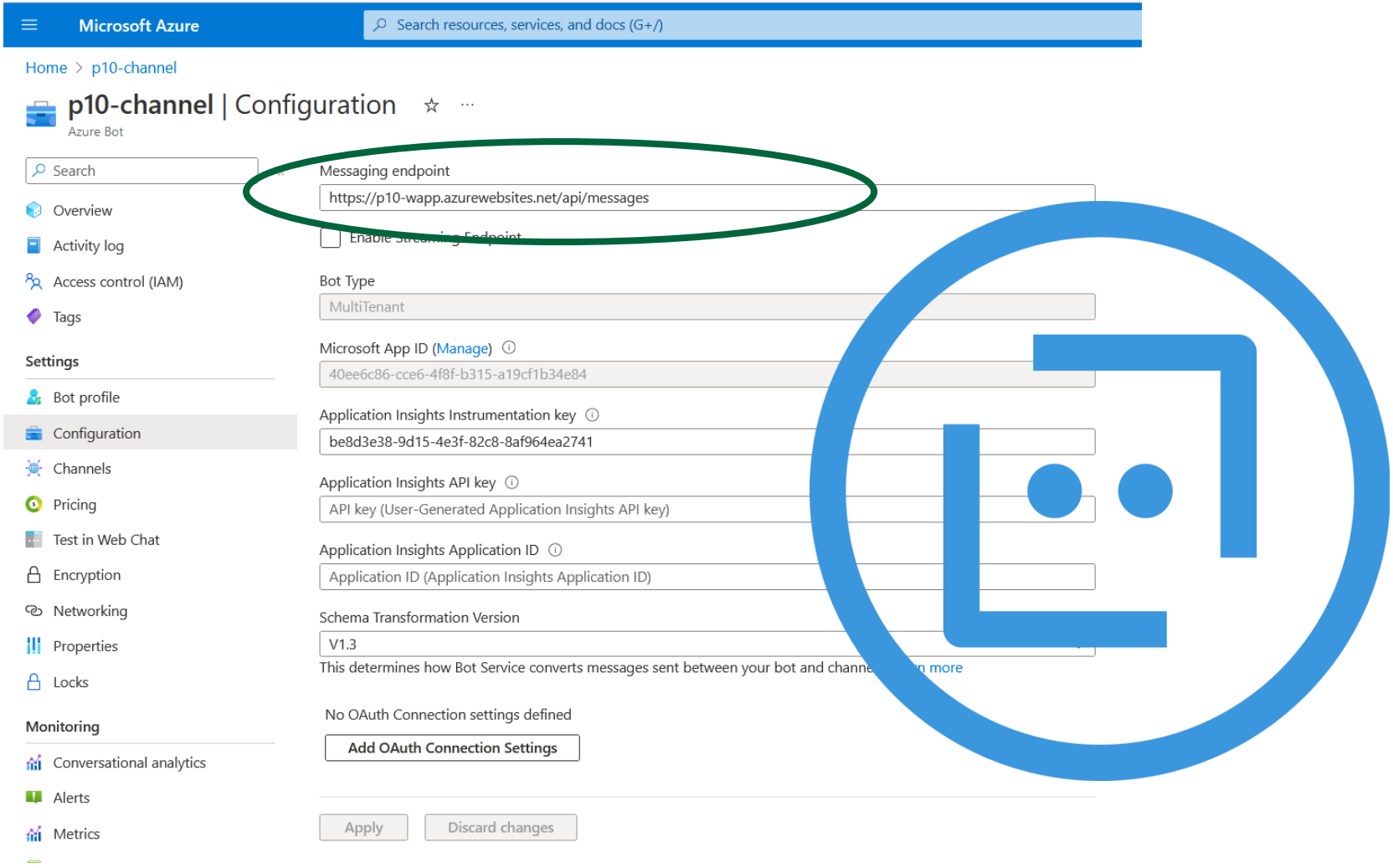
# 3-b: déploiement de azure web app service paramétrage github



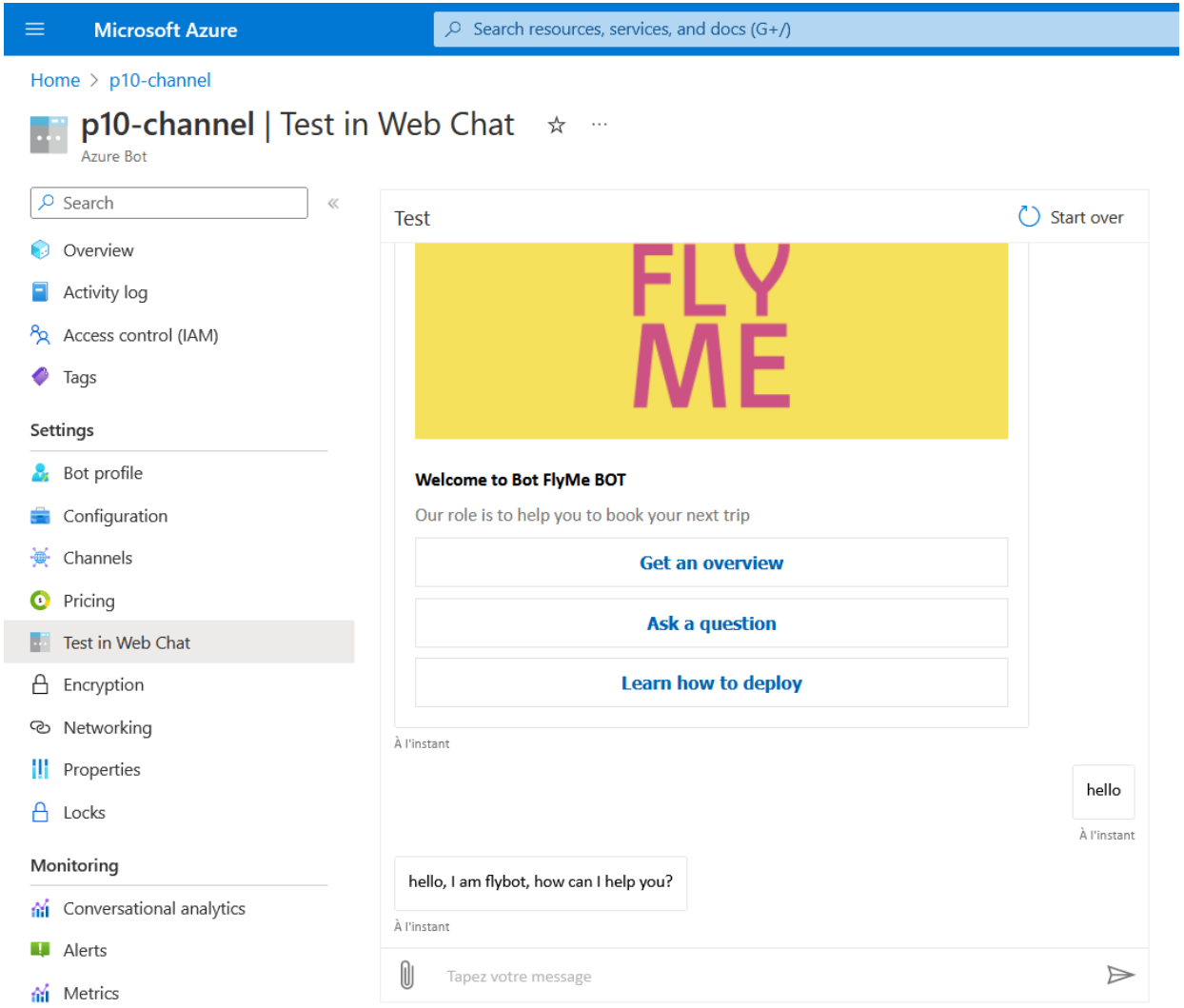
# 3-b: deploiement de azure web app service paramétrage de la startup commande

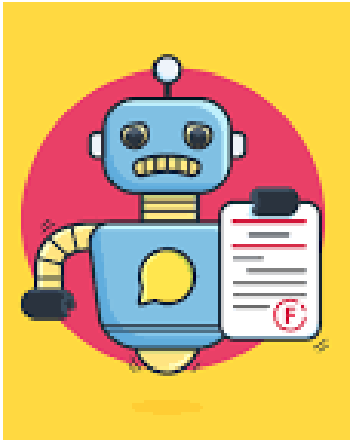


# 3-c: azure bot service
# parametrage du endpoint et des id/keys



# 3-c: azure bot service
## web bot channel fonctionnel



C

# MODEL UPDATE

# Savoir ou on se trouve dans le cycle de vie

**MODELE EN DEBUT DE VIE**

- *MVP : peu de fonctionnalités*
- *Dialogues peu evolués*
- *Peu mature*
- *Peu de volumétrie*
- *De nombreux ajustements à faire*

**MODELE EN MILIEU DE VIE**

- Mature
- *Nécessite le développement de nouvelles fonctionnalités*
- *Nécessite des dialogues plus évolués*
- *Volumétrie conséquente*
- *Peu d'ajustements à faire si on garde le même niveau de service*

# Modèle UPDATE: Cas du modèle en début de vie

MONITORING

ALERT

DASHBOARD

LOG

**ANALYSE & EVALUATION:**

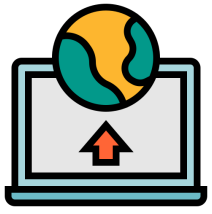
- *Score moyen book intention LUIS < seuil*
- *Proportion de booking CANCEL > seuil*
- *Proportion de QUIT > seuil*
- *Proportion d'entités mal reconnues > seuil*

**REENTRAINEMENT DU MODELE LUIS:**

*Avec de nouvelles utterances habilement choisies*

**RECONSTRUCTION D'UN NOUVEAU MODELE LUIS PLUS ADAPTE:**

*Avec plus d'intentions possibles et un meilleur choix d'entités*



**LEGERE ADAPTATION DU CODE DU BOT:**
*Permet un meilleur dialogue sans être plus évolué*
*Permet une meilleure gestion des exceptions*

**REDEPLOIEMENT CONTINU**