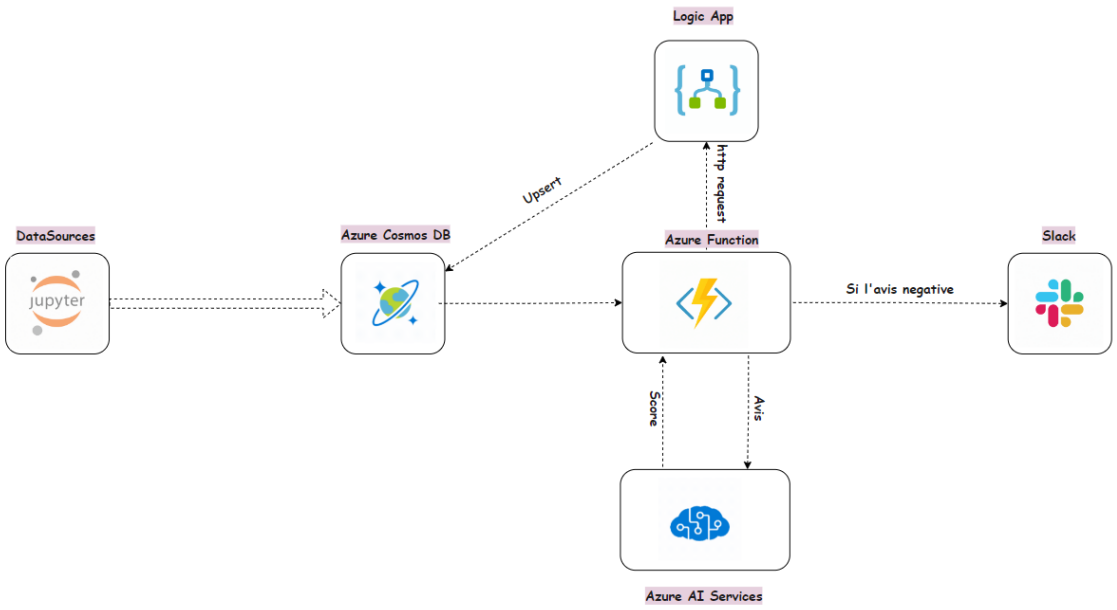
Reviews Restaurant Analysis

Rapport de projet

Pipeline d’analyse de revues des restaurants:



Fait par:

- DBAA Omar
- MOUFLLA Faissal
- SEFDINE Nassuf
- HARRATI Yassine

Le 29 Décembre 2023

PLAN

1. Introduction
2. Contexte du projet
3. Description du projet
4. Architecture du système
5. Mise en oeuvre
6. Conclusion

Introduction:

Le projet "Analyse Dynamique des Avis de Restaurants avec Azure" est une initiative visant à mettre en œuvre un pipeline de données complet utilisant les services d'Azure pour la collecte, le traitement et l'analyse des avis sur les restaurants. L'objectif principal de ce projet est de fournir un système efficace et évolutif pour stocker, analyser et tirer des insights à partir des données d'avis, spécifiquement dans le secteur des restaurants.

Contexte du Projet :

Dans le domaine de la restauration, les avis et les commentaires des clients jouent un rôle critique dans la réputation et le succès des établissements. Ces avis, généralement disponibles sur des plateformes en ligne, offrent un aperçu précieux des expériences des clients, influençant les décisions d'autres consommateurs et impactant directement la notoriété des restaurants.

L'importance de ce projet réside dans la nécessité de comprendre et d'exploiter de manière proactive les avis des clients pour améliorer la qualité des services, identifier les points forts et faibles des restaurants, et ainsi garantir une expérience client optimale.

En adoptant les services Azure tels que Cosmos DB, Functions, Cognitive Services et Logic Apps, notre projet vise à fournir une solution robuste qui non seulement stocke et traite les avis, mais utilise également des analyses de données avancées pour fournir des informations exploitables aux restaurateurs.

Description du Projet :

Ce projet repose sur l'intégration de plusieurs services Azure clés, utilisés de manière conjointe pour mettre en place un pipeline de données efficace et robuste pour l'analyse des avis de restaurants. Les composants principaux utilisés sont Azure Cosmos DB, Azure Functions, Azure Cognitive Services et Azure Logic Apps.

Azure Cosmos DB : Ce service joue un rôle central en tant que stockage principal pour toutes les données liées aux restaurants, utilisateurs et avis. Sa capacité à gérer de grands volumes de données avec une faible latence et une haute disponibilité en fait une source de données idéale pour le pipeline.

Azure Functions : Ces fonctions serverless sont déployées pour traiter les nouvelles entrées dans Cosmos DB. Elles sont déclenchées automatiquement lorsqu'un nouvel avis est ajouté, et préparent les données d'avis pour l'analyse de sentiment.

Azure Cognitive Services (API d'Analyse de Texte) : Ce service est utilisé pour effectuer des analyses de sentiment sur le texte des avis. Les données d'avis sont envoyées depuis Azure Functions vers Cognitive Services, qui détermine ensuite le score de sentiment pour chaque avis.

Azure Logic Apps : Cet outil est utilisé pour orchestrer et gérer les flux de travail entre Cosmos DB, Azure Functions et Azure Cognitive Services. Il garantit la cohérence et l'efficacité du flux de données et des processus, en gérant les erreurs, les tentatives de réexécution et les actions basées sur des conditions.

Interconnexion des Composants :

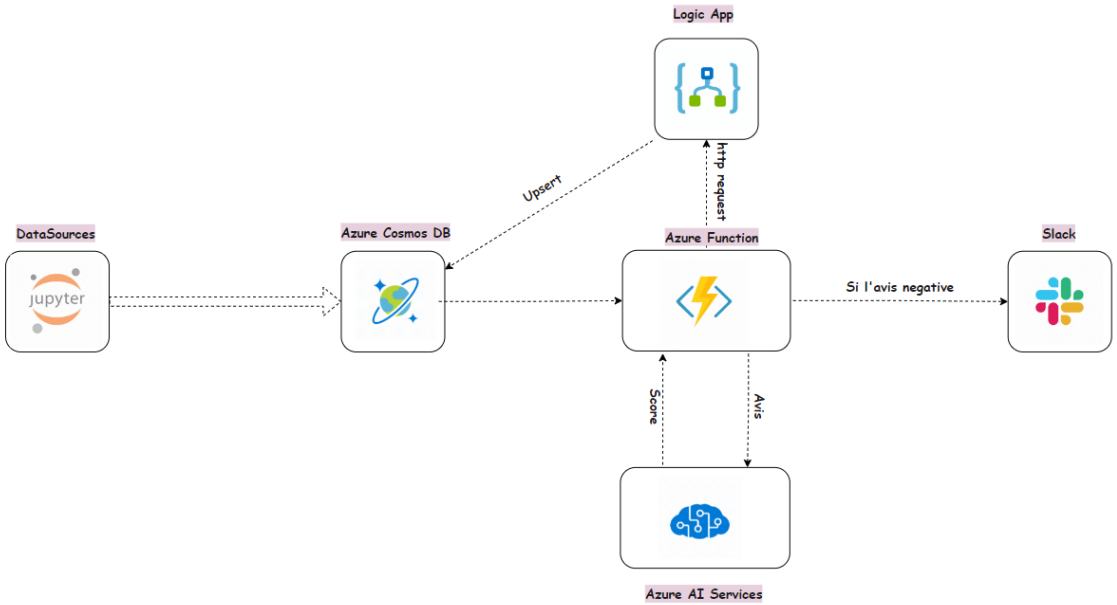
Les composants Azure sont interconnectés pour former un pipeline de données fluides. Lorsqu'un nouvel avis est ajouté dans Cosmos DB, Azure Functions est déclenché pour préparer les données pour l'analyse. Les données d'avis sont ensuite envoyées à Azure Cognitive Services pour l'analyse de sentiment. Une fois l'analyse effectuée, les résultats sont renvoyés aux Azure Functions pour mettre à jour les données d'avis dans Cosmos DB. Tout au long de ce processus, Azure Logic Apps gère et coordonne les flux de données et de processus, assurant ainsi la cohérence et l'efficacité du pipeline.

Cette architecture offre une solution évolutive et cohérente pour stocker, traiter et analyser les avis de restaurants de manière dynamique et efficace.

Architecture du Système :

Le système repose sur une architecture cloud avec une série de services Azure interconnectés pour gérer les avis de restaurants.

Diagramme de l'Architecture :



Entrées Utilisateur : Les avis des utilisateurs sur les restaurants sont générés et stockés dans Azure Cosmos DB.


```python
# Function to insert data into Cosmos DB
def insert_into_cosmos(data):
    try:
        # Encrypt specific fields
        data['user']['name'] = encrypt_field(data['user']['name'])
        data['user']['email'] = encrypt_field(data['user']['email'])
        data['user']['location'] = encrypt_field(data['user']['location'])
        data['user']['dob'] = encrypt_field(data['user']['dob'])

        container.create_item(body=data)
        print("Inserted data into Cosmos DB.")
    except Exception as e:
        print(f"Failed to insert data into Cosmos DB: {str(e)}")
```

Les données sont cryptées avant d'être insérées dans Cosmos DB.

Tache faite par MOUFLLA Faissal

Azure Functions : Lorsqu'un nouvel avis est ajouté à Cosmos DB, Azure Functions est déclenché. Ces fonctions serverless préparent les données d'avis pour l'analyse de sentiment.

Azure Cognitive Services : Les données d'avis préparées sont envoyées à Azure Cognitive Services (API d'Analyse de Texte) pour l'analyse de sentiment. Ce service détermine le score de sentiment pour chaque avis.

Rétroaction à Azure Functions : Une fois l'analyse de sentiment effectuée, les résultats sont renvoyés aux Azure Functions pour mettre à jour les données d'avis dans Cosmos DB.

Azure Logic Apps : Azure Logic Apps orchestre et gère les flux de données entre Cosmos DB, Azure Functions et Azure Cognitive Services. Il assure la cohérence du pipeline, gère les erreurs et les actions basées sur des conditions.

Intégration de l'API Slack : En cas d'avis négatif, Azure Functions déclenche une notification sur Slack à l'aide de l'API Slack, informant les utilisateurs concernés.

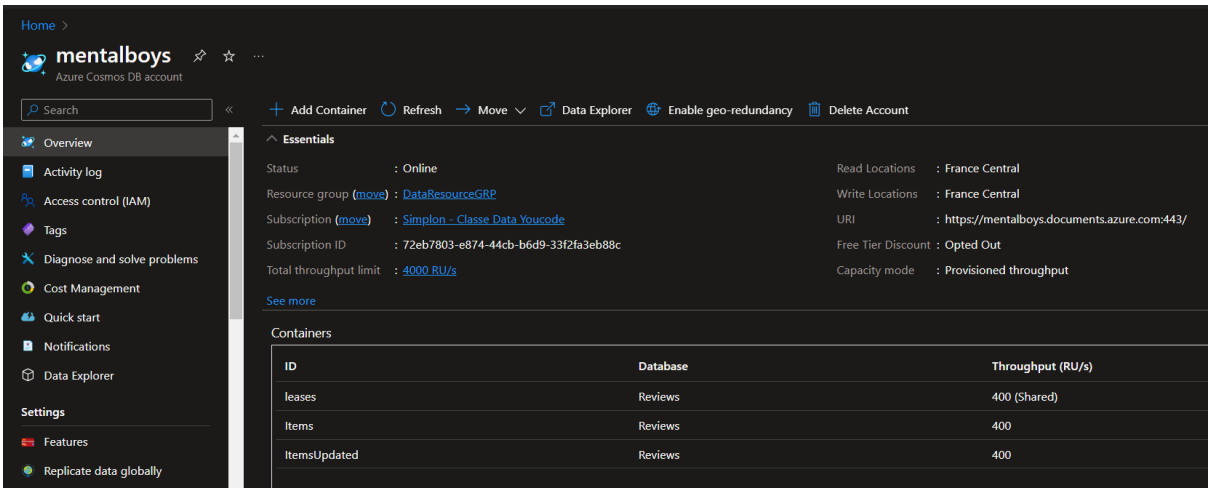
Ce schéma représente un flux continu de données entre les différents services Azure, depuis la collecte des avis jusqu'à l'analyse de sentiment, la mise à jour des données et les

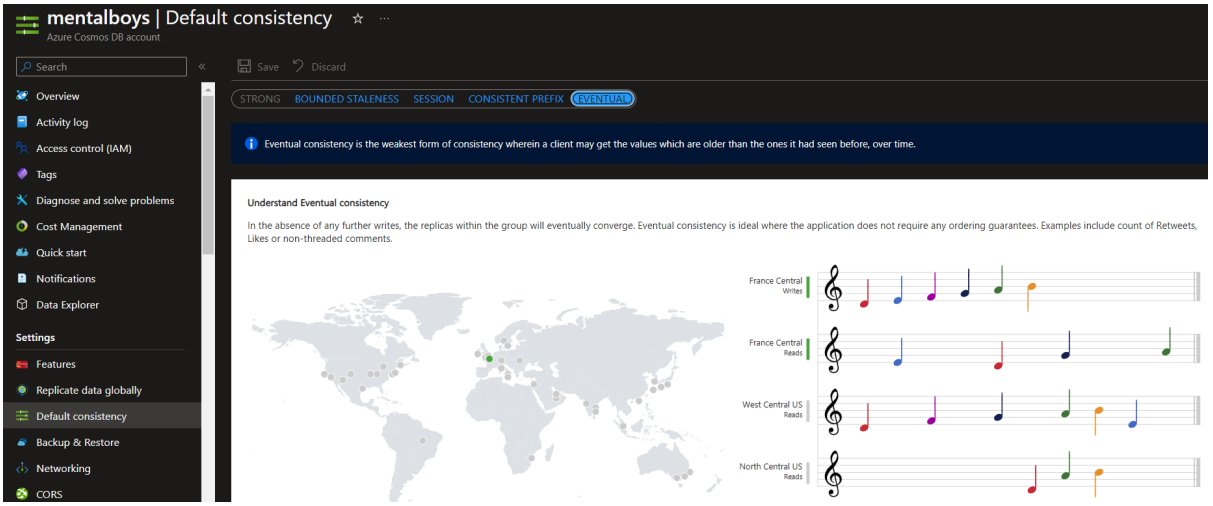
notifications. Chaque service joue un rôle spécifique dans le traitement et la gestion des avis de restaurants, créant ainsi un pipeline de données dynamique et efficace.

Mise en Œuvre :

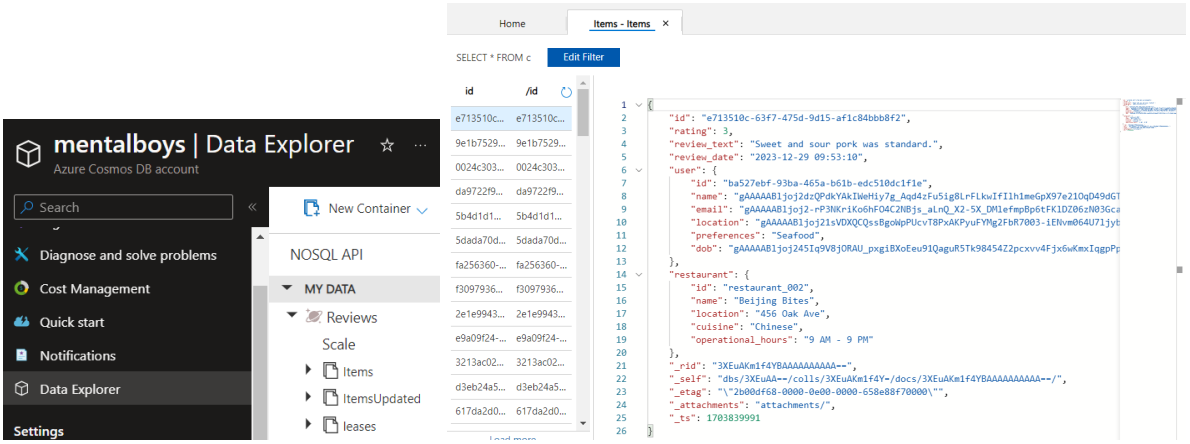
Azure Cosmos DB :

-Créez une instance de Cosmos DB.





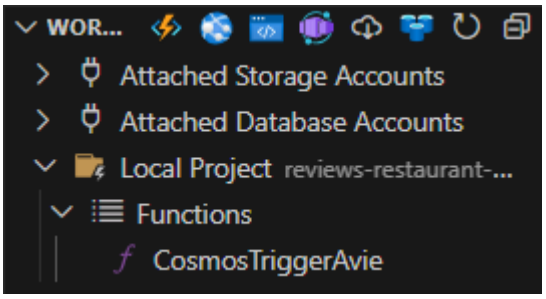
-Définissez les conteneurs pour stocker les données des restaurants, des utilisateurs et des avis.

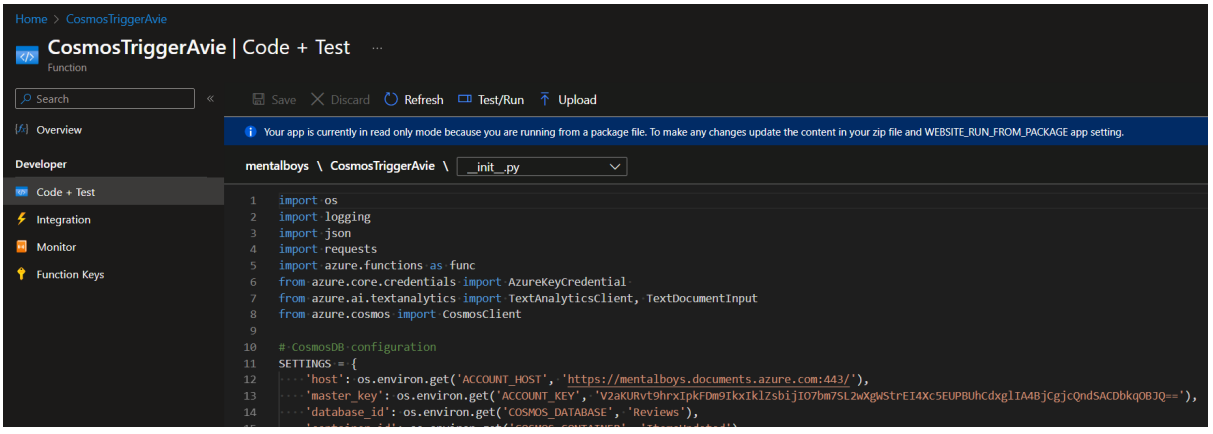


Tache faite par MOUFLLA Faissal et DBAA Omar

Azure Functions :

-Créez des fonctions déclenchées par les événements dans Cosmos DB pour traiter les nouveaux avis.





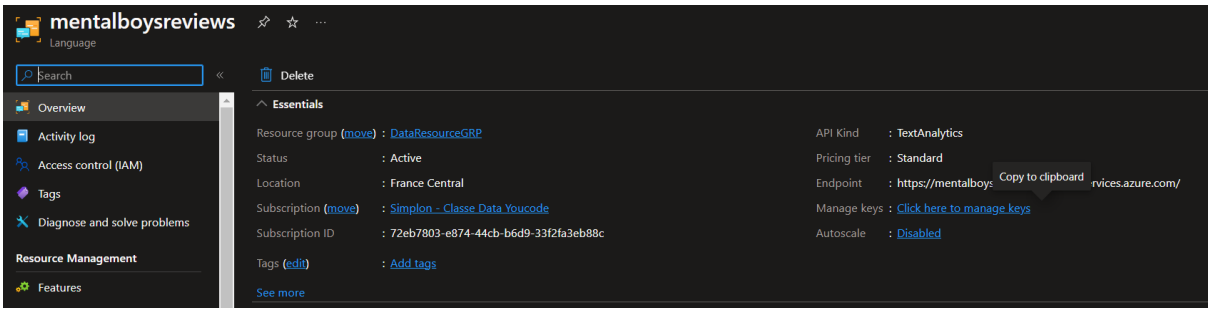
-Implémentez la logique de traitement des avis pour préparer les données avant l'analyse de sentiment.

```python
# Send a HTTP request to Logic App
def send_http_request_logic_app(data):...

# functions to send message into slack
def send_message_to_slack(message):...

# Function to analyze sentiment
def analyse_sentiment(review_text):...

def main(documents: func.DocumentList) -> str:      You, 23 hours ago • feat: add function that trigger cosmos db. Closed…
    if documents:

        # Retrieve document
        item_azure = documents[0]
        # Convert Azure Document to JSON
        item_data = json.dumps(item_azure, default=lambda o: o.__dict__)
        item = json.loads(item_data)['data']

        review_text = item["review_text"]

        result = analyse_sentiment(review_text)[0]

        sentiment = result.sentiment

        if sentiment == 'negative':
            message = f'''Hi, the user with id {item['user']['id']} send a negative review to the restaurant with id {item['restaurant']['id']}.\n
                The review is "{item['review_text']}"
            '''
            send_message_to_slack(message)

        item['sentiment'] = sentiment

        # Send request to logic app
        send_http_request_logic_app(item)
        logging.info(f"Review texts : {review_text}, \t Score : {sentiment}")
```

Tache faite par SEFDINE Nassuf et DBAA Omar

<u>Azure Cognitive Services :</u>

- Configurez et utilisez l'API d'Analyse de Texte pour l'analyse de sentiment.

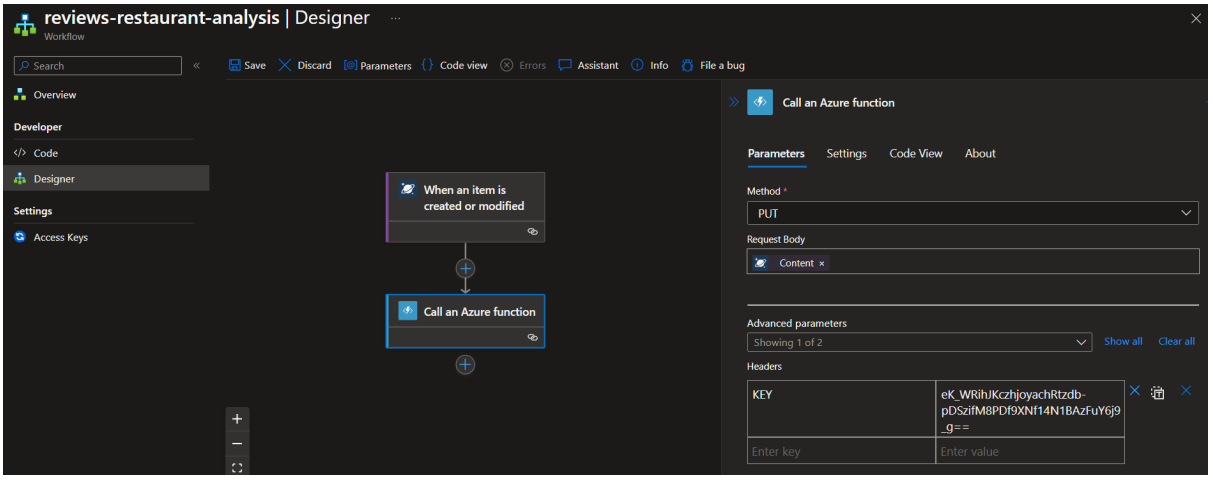


- Intégrez cette API avec Azure Functions pour l'envoi et la réception des données d'avis.

Tache faite par DBAA Omar et SEFDINE Nassuf

<u>Azure Logic Apps :</u>

- Créez des workflows pour gérer les flux de données entre les services.



- Mettez en place la gestion des erreurs, des actions basées sur des conditions et les notifications via Slack en cas d'avis négatif.

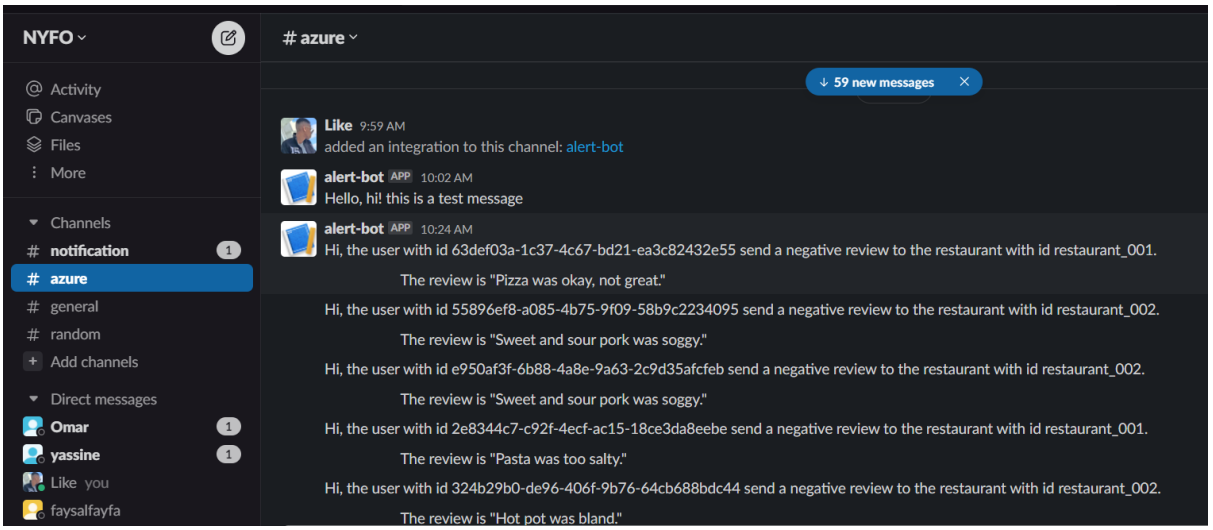
Tache faite par DBAA Omar et SEFDINE Nassuf

Intégration de l'API Slack :

- Utilisez les fonctions d'Azure pour déclencher les notifications Slack en cas d'avis négatif.

App Name	Workspace	Distribution Status
alert-bot	NYFO	Not distributed

- Configurez l'API Slack pour envoyer des notifications détaillées aux utilisateurs concernés.



Tache faite par HARRATI Yassine et SEFDINE Nassuf

Conclusion :

L'implémentation d'un pipeline de données complet utilisant les services Azure pour stocker, traiter et analyser les avis sur les restaurants a été une réalisation cruciale de ce projet. En intégrant Azure Cosmos DB, Azure Functions, Azure Cognitive Services et Azure Logic Apps, nous avons orchestré avec succès un système qui gère efficacement les données des avis, analyse les sentiments et offre une analyse approfondie des sentiments des utilisateurs. Cette mise en œuvre a considérablement simplifié la collecte et l'analyse des avis de restaurants, améliorant notre compréhension des sentiments des utilisateurs tout en respectant rigoureusement les principes de gouvernance des données et la conformité au RGPD. En envisageant l'avenir, des améliorations potentielles pourraient impliquer l'utilisation d'algorithmes d'apprentissage automatique plus avancés pour une compréhension contextuelle plus approfondie et l'intégration de sources de données supplémentaires pour enrichir l'analyse des sentiments.
En conclusion, le pipeline de données alimenté par Azure a démontré la valeur des services cloud dans la facilitation de l'analyse dynamique des avis sur les restaurants, tout en respectant des normes strictes de gouvernance des données et de conformité réglementaire.