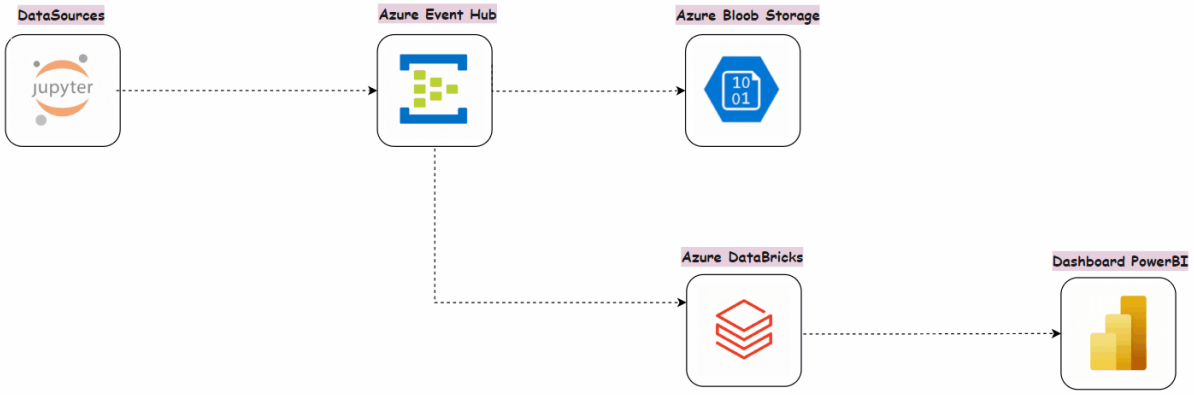
Analyse de logs en temps réel

Squad 3



DBAA OMAR

MOUFLLA FAISSAL

HARRATI YASSINE

SEFDINE NASSUF

Vue d'ensemble

Ce projet consiste à développer un pipeline de données qui ingère, traite et analyse en temps réel les logs générés par une application web ou mobile. Utilisant Azure Event Hub pour l'ingestion de données et Apache Spark sur Azure HDInsight pour le traitement et l'analyse, le système permettra de surveiller la performance de l'application, de détecter les anomalies, d'analyser le comportement des utilisateurs et de renforcer la sécurité.

L’entreprise souhaite surveiller et optimiser la performance de son application ainsi que d’assurer une expérience fluide aux utilisateurs

Objectifs

Impl´ementer un syst`eme pour :

- Surveiller et optimiser la performance de l’application
- Détecter les anomalies
- Analyser le comportement des utilisateurs

Exigences de Données

Nous allons collecter les données depuis notre plateforme. (Dans notre cas: Un script de génération des données.)

Plan

I. Compréhension du Business

Réunion entre l'équipe et les parties prenantes pour une validation du cahier de charge.

II. Compréhension des données

Exploration des données et mise en place d'un catalogue des données montrant la disponibilité des données, leurs sources ainsi que leurs traitements.

III. Préparation des données

Collection, transformation et chargement des données pour une analyse approfondie.

IV. Modélisation

Analyse et visualisation des données pour tirer des informations aidant à la prise de décision.

V. Déploiement

Publication du dashboard dans le cloud, validation et livraison du projet

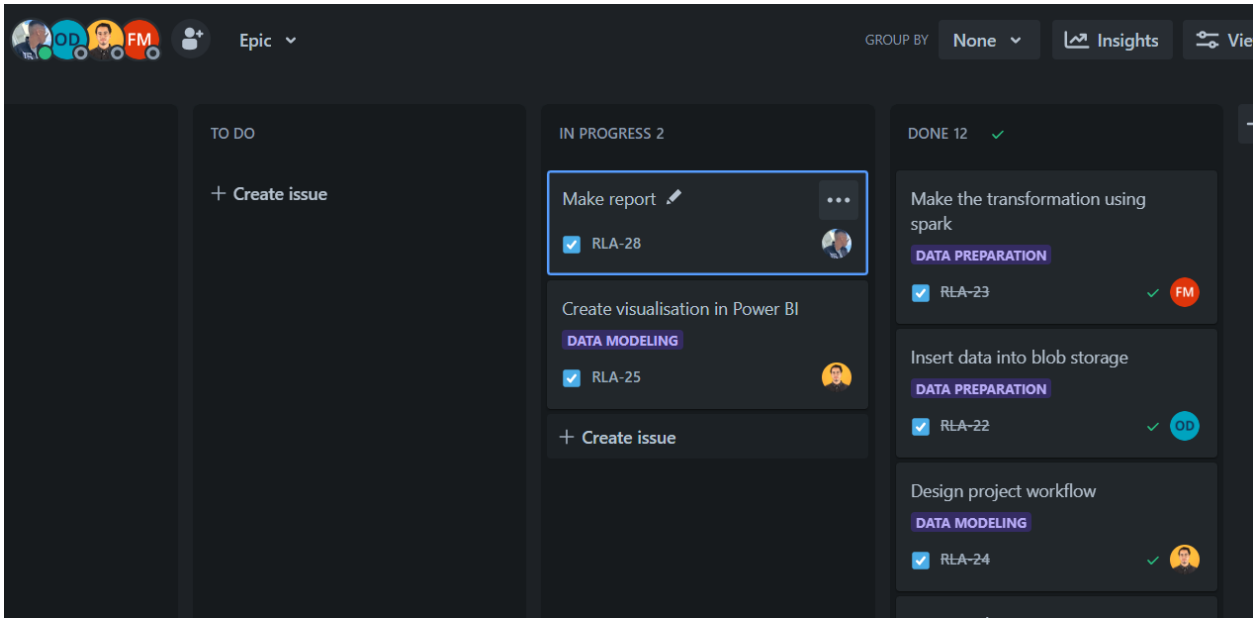
Compréhension du Business

Nous avons commencé avec une session de brainstorming qui nous aidé à comprendre le business, identifier les objectifs du projets et comment ils s'allient avec les finalités de l'entreprise. Nous avons mis en place un cahier de charge disponible dans notre dépôt github dans un dossier appelé "docs".
lien du dépôt : https://github.com/Sefdine/realtime-log-analysis/tree/main/docs

Tache faite par SEFDINE Nassuf

C'est dans cette phase que nous avons officiellement lancé le projet.
Nous avons départager les rôles en choisissant MOUFLLA Faissal comme chef du projet.

Nous avons donc commencé par faire un plan en utilisant Jira :



Tache faite par SEFDINE Nassuf

Compréhension des données

Quand le plan est défini, il faut une compréhension des données existantes pour bien débuter le projet. De ce fait, nous avons identifié nos sources de données, dans notre cas, un script de génération. Nous avons créé un catalogue des données montrant les données existantes, ceux dont on va utiliser pour notre projet et comment on va les traiter pour quelle finalité. Le catalogue est disponible dans le dépôt dans le dossier "docs".

Lien du dépôt : https://github.com/Sefdine/realtime-log-analysis/tree/main/docs

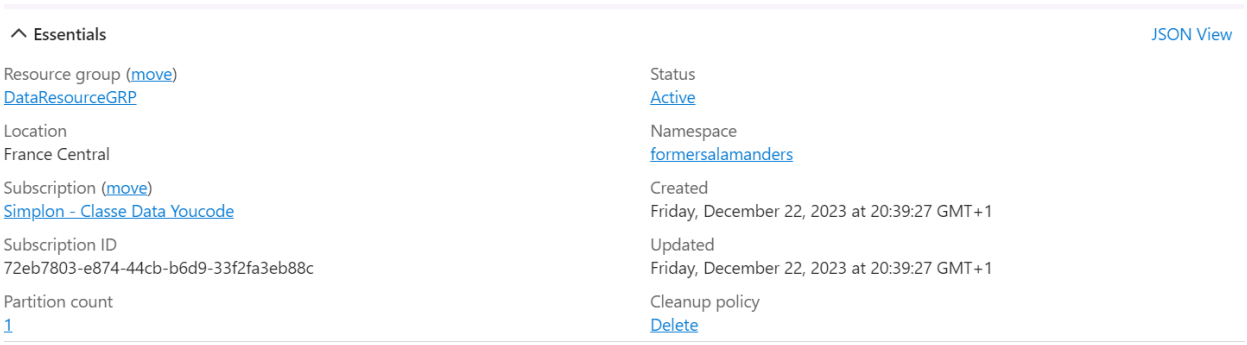
Tache faite par SEFDINE Nassuf

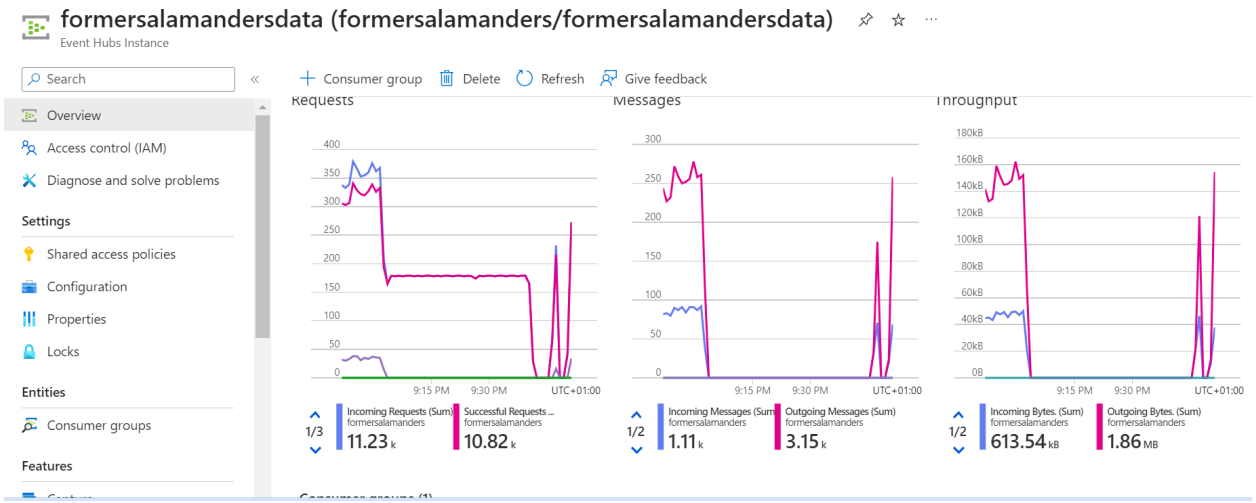
Préparation des données

Dans cette phase, nous avons commencé par créer et configurer les services nécessaires pour l'implémentation du processus. Nous avons créé et commencé l'implémentation direct.

1. Azure Event Hub : Nous avons créé un service Azure Event Hub qui nous permet de lire les données en tant réel depuis le local. Pour ce faire, nous avons téléchargé les identifiants ainsi que les librairies nécessaires pour se connecter avec le service. Et avons implémenté un script pour envoyer les données en temps réel dans Event Hub.

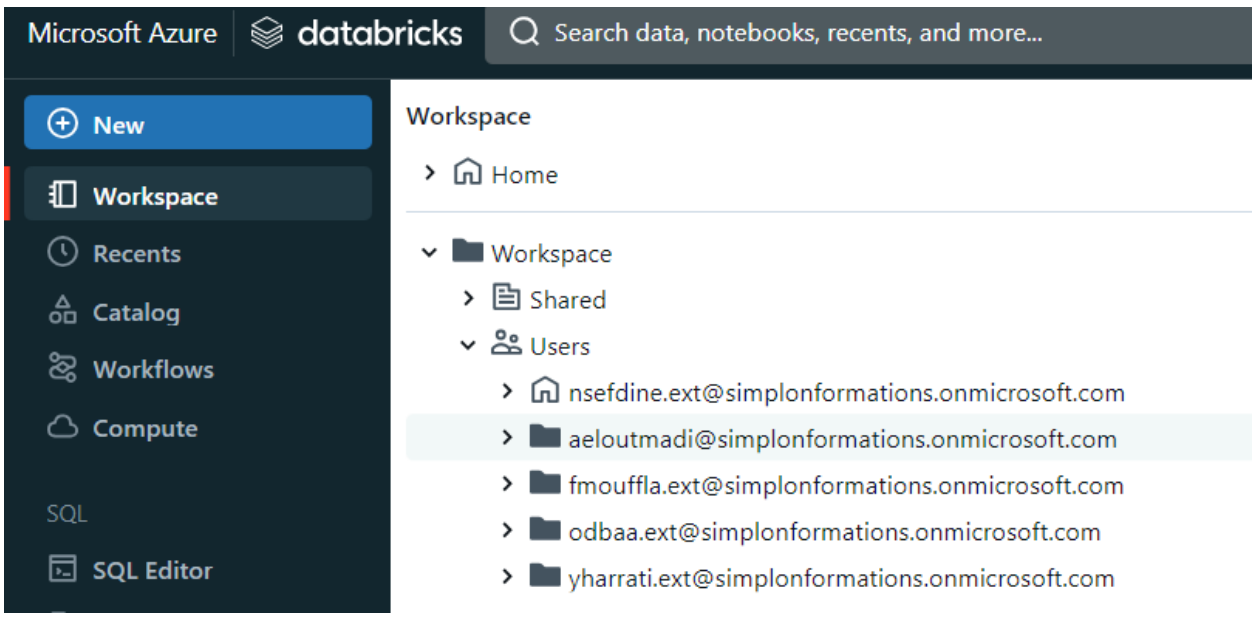
```python
import os
from azure.eventhub import EventHubProducerClient, EventData
from dotenv import load_dotenv

load_dotenv()

access_key = os.getenv('access_key')



CONNECTION_STR = f"Endpoint=sb://formersalamander.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey={access_key}"
EVENTHUB_NAME = "formersalamandersdata"

producer_client = EventHubProducerClient.from_connection_string(CONNECTION_STR, eventhub_name=EVENTHUB_NAME)
# Continuously generate log entries and send them to the Event Hub
try:
    while True:
        log_entry = generate_log_entry()
        print(f"Sending log entry: {log_entry}")  # Print the log entry being sent
        event_data_batch = producer_client.create_batch()
        event_data_batch.add(EventData(log_entry))
        producer_client.send_batch(event_data_batch)
        time.sleep(random.uniform(0.2, 1.0))  # Simulate delay between log entries
except KeyboardInterrupt:
    print("Interrupted. Sending has been stopped.")
finally:
    # Close the producer client
    producer_client.close()
```



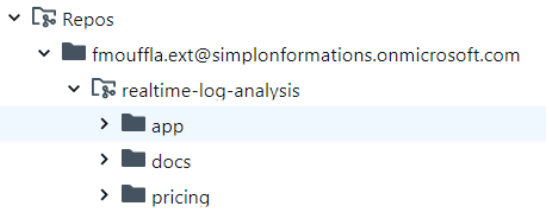


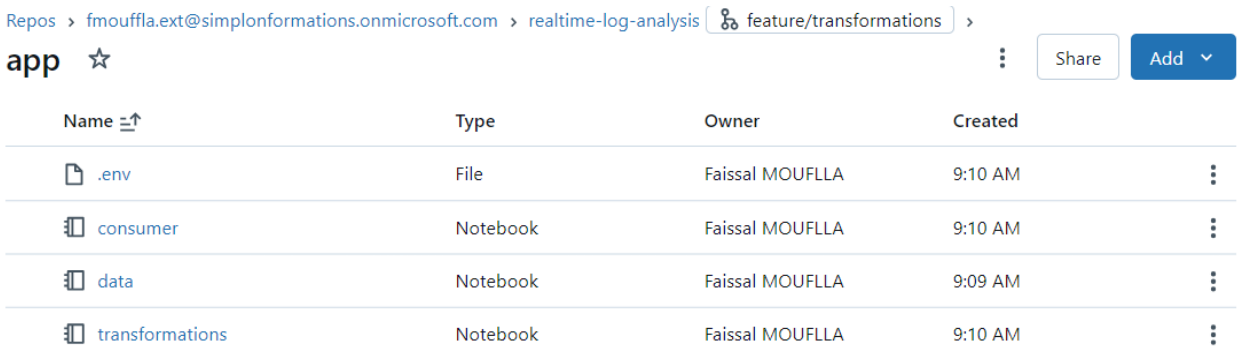
Tâche faite par DBAA Omar.

2. Azure Databricks : Nous avons créé un service databricks et nous avons ajouté tout le monde dans l'équipe en créant un cluster en multi-nodes. L'intérêt est de pouvoir travailler dans le même notebook.

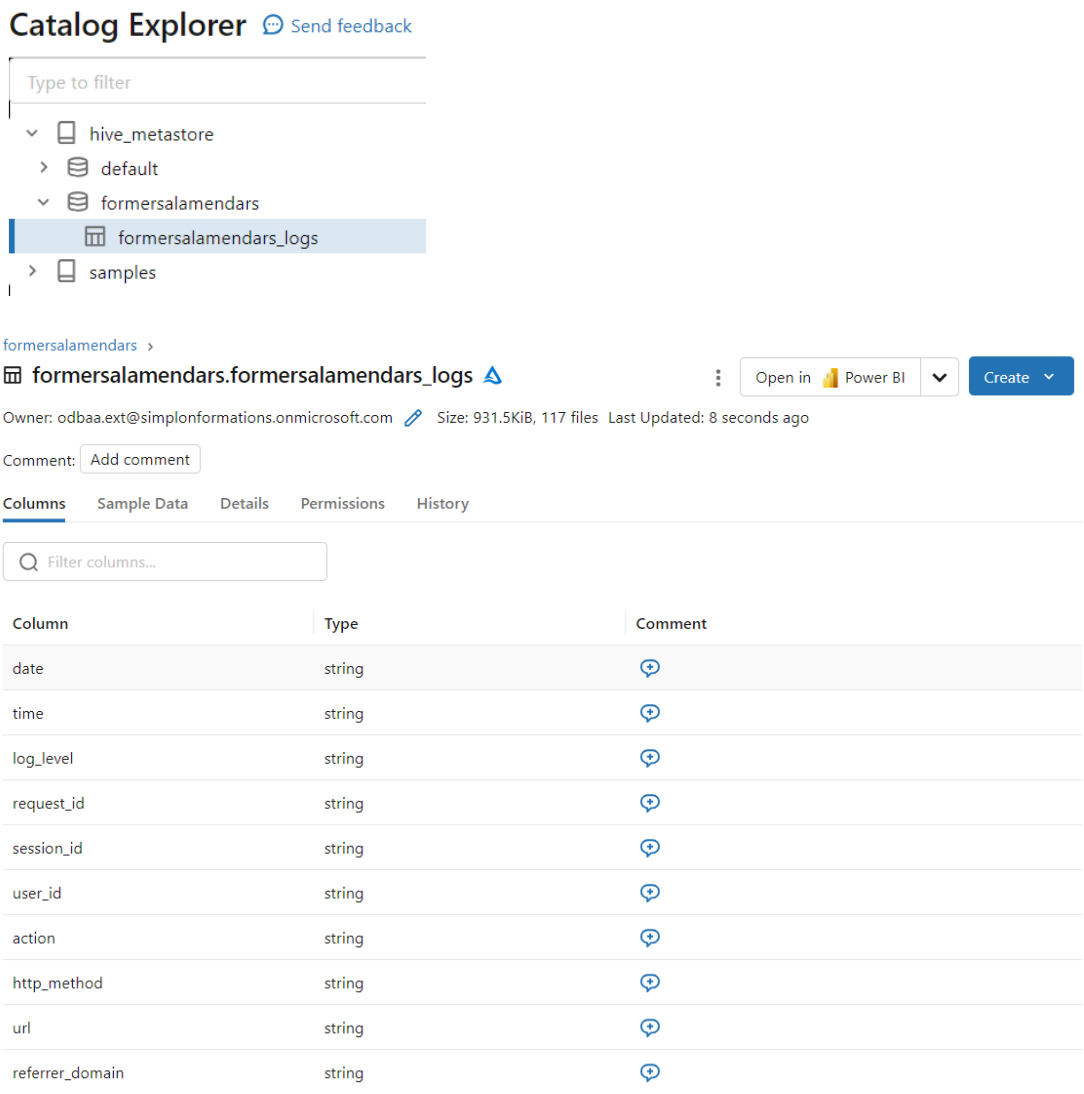


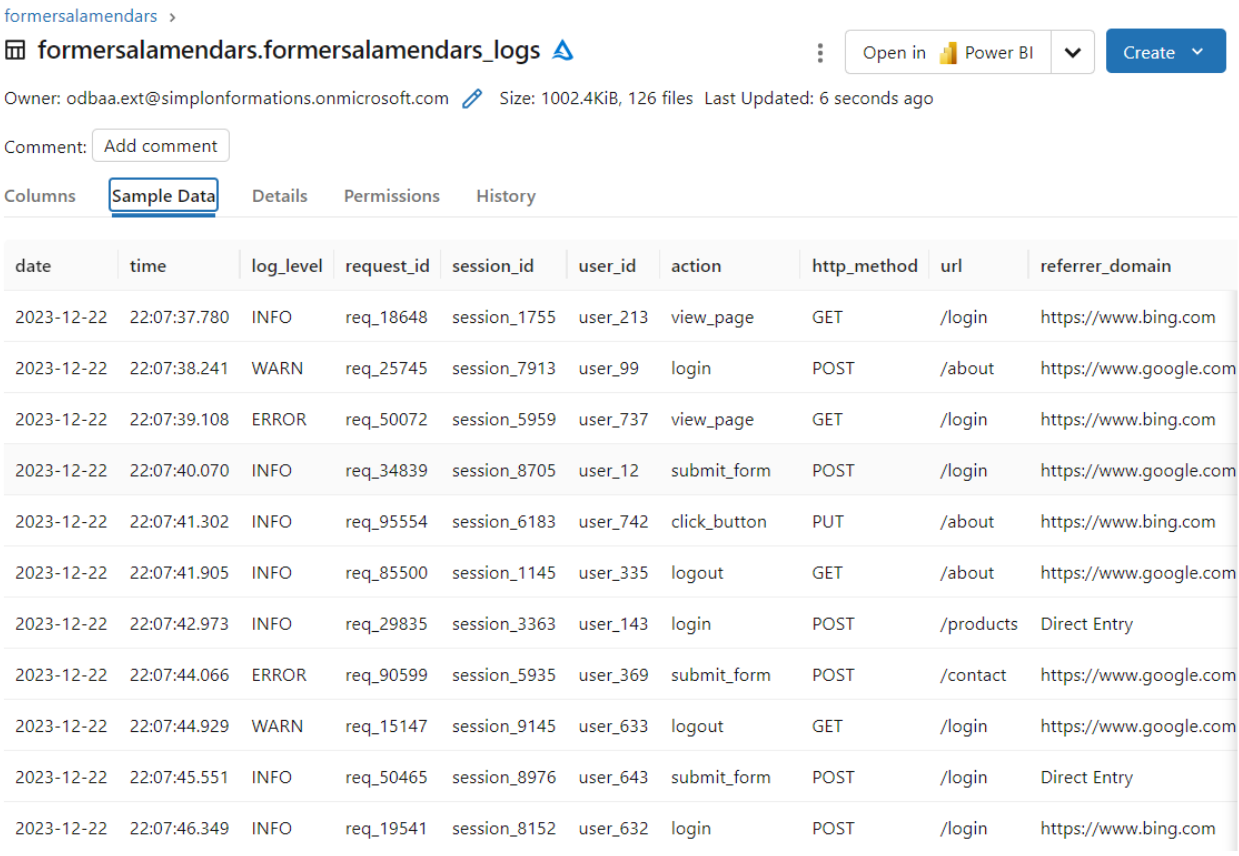
Nous avons le dépôt github, pour faire le versionnement d'une manière adaptée.





Dans le dossier, vous pouvez voir un fichier notebook transformations qui se charge de faire la connexion avec Event Hub, récupérer les données, les traiter et les charger dans Hive metastore. Les données sont traitées et sauvegardées en temps réel.





date	time	log_level	request_id	session_id	user_id	action	http_method	url	referrer_domain
2023-12-22	22:07:37.780	INFO	req_18648	session_1755	user_213	view_page	GET	/login	https://www.bing.com
2023-12-22	22:07:38.241	WARN	req_25745	session_7913	user_99	login	POST	/about	https://www.google.com
2023-12-22	22:07:39.108	ERROR	req_50072	session_5959	user_737	view_page	GET	/login	https://www.bing.com
2023-12-22	22:07:40.070	INFO	req_34839	session_8705	user_12	submit_form	POST	/login	https://www.google.com
2023-12-22	22:07:41.302	INFO	req_95554	session_6183	user_742	click_button	PUT	/about	https://www.bing.com
2023-12-22	22:07:41.905	INFO	req_85500	session_1145	user_335	logout	GET	/about	https://www.google.com
2023-12-22	22:07:42.973	INFO	req_29835	session_3363	user_143	login	POST	/products	Direct Entry
2023-12-22	22:07:44.066	ERROR	req_90599	session_5935	user_369	submit_form	POST	/contact	https://www.google.com
2023-12-22	22:07:44.929	WARN	req_15147	session_9145	user_633	logout	GET	/login	https://www.google.com
2023-12-22	22:07:45.551	INFO	req_50465	session_8976	user_643	submit_form	POST	/login	Direct Entry
2023-12-22	22:07:46.349	INFO	req_19541	session_8152	user_632	login	POST	/login	https://www.bing.com

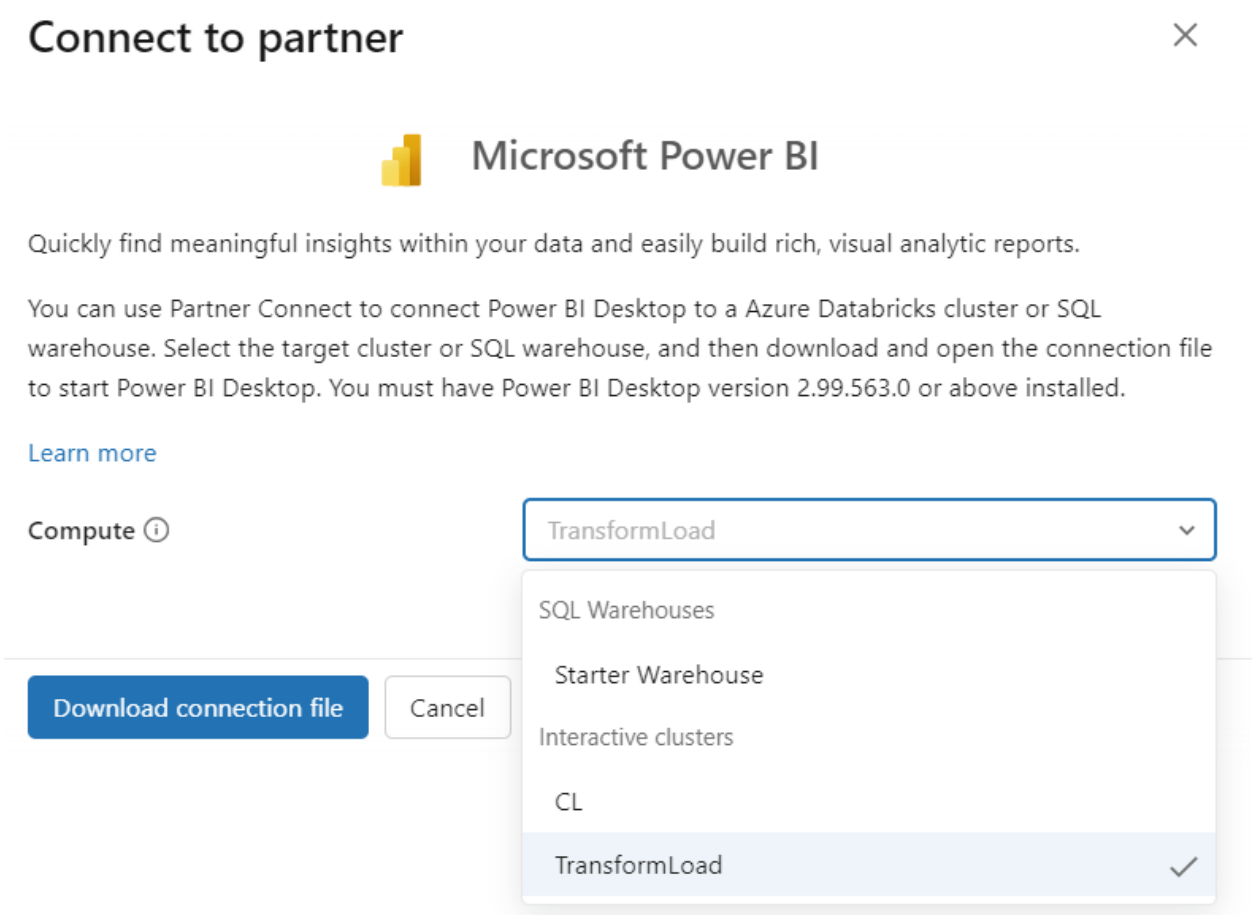
La connexion entre Databricks et Event Hub a été faite avec DBAA Omar.

La transformation a été faite par MOUFLLA Faissal.

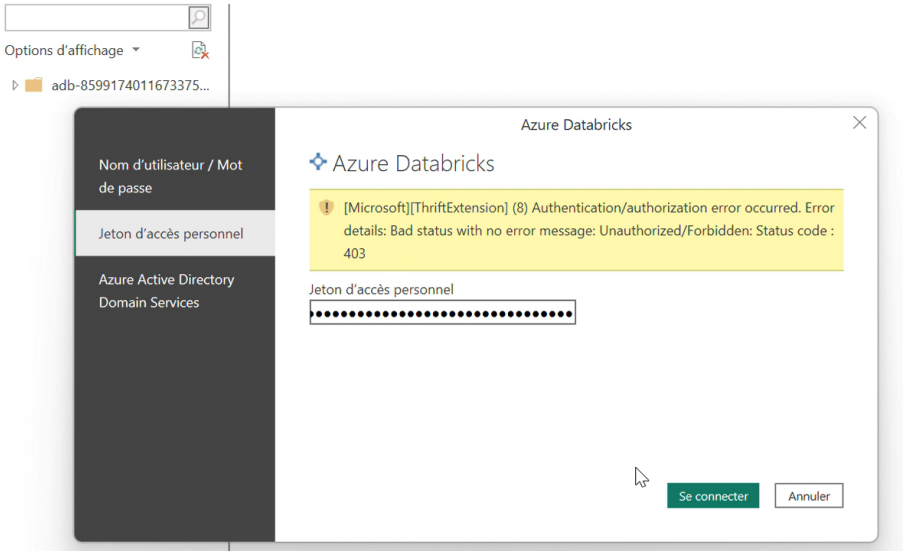
Le chargement a été fait par SEFDINE Nassuf.

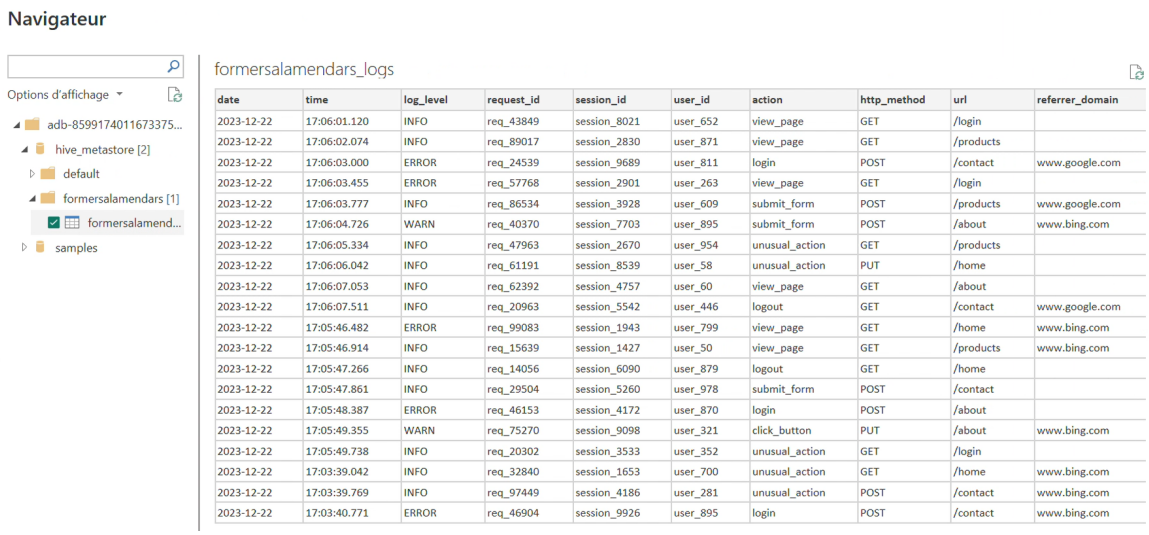
Modélisation

Dans cette phase, nous avons effectué les visualisations dans un tableau de bord Power BI en se basant sur les analyses nécessaires.
Nous avons donc fait la connexion entre Databricks et Power BI Desktop afin de récupérer les données en temps réel. Il nous a fallu créer un token, puis créer une connexion avec les partenaires, dans notre cas Power BI Deskto.



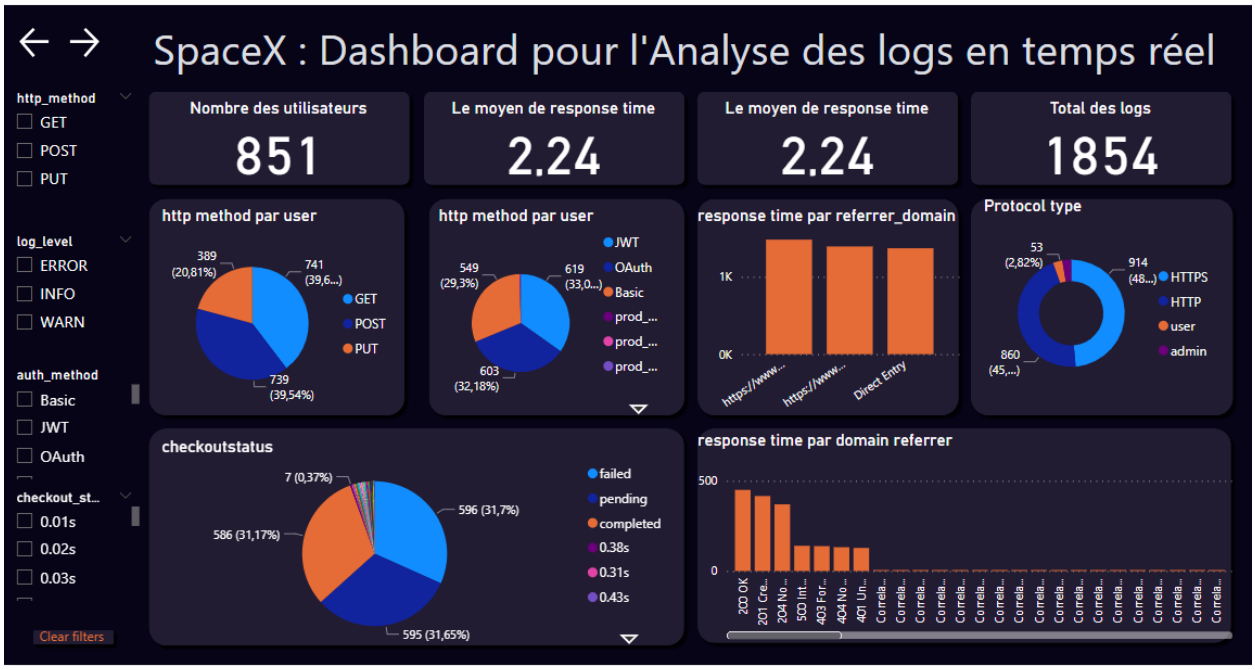
Après avoir choisi cette option, cela nous télécharge un fichier Power BI qu'on utilise pour faire la connexion avec databricks.





date	time	log_level	request_id	session_id	user_id	action	http_method	url	referrer_domain
2023-12-22	17:06:01.120	INFO	req_43849	session_8021	user_652	view_page	GET	/login	
2023-12-22	17:06:02.074	INFO	req_89017	session_2830	user_871	view_page	GET	/products	
2023-12-22	17:06:03.000	ERROR	req_24539	session_9689	user_811	login	POST	/contact	www.google.com
2023-12-22	17:06:03.455	ERROR	req_57768	session_2901	user_263	view_page	GET	/login	
2023-12-22	17:06:03.777	INFO	req_86534	session_3928	user_609	submit_form	POST	/products	www.google.com
2023-12-22	17:06:04.726	WARN	req_40370	session_7703	user_895	submit_form	POST	/about	www.bing.com
2023-12-22	17:06:05.334	INFO	req_47963	session_2670	user_954	unusual_action	GET	/products	
2023-12-22	17:06:06.042	INFO	req_61191	session_8539	user_58	unusual_action	PUT	/home	
2023-12-22	17:06:07.053	INFO	req_62392	session_4757	user_60	view_page	GET	/about	
2023-12-22	17:06:07.511	INFO	req_20963	session_5542	user_446	logout	GET	/contact	www.google.com
2023-12-22	17:05:46.482	ERROR	req_99083	session_1943	user_799	view_page	GET	/home	www.bing.com
2023-12-22	17:05:46.914	INFO	req_15639	session_1427	user_50	view_page	GET	/products	www.bing.com
2023-12-22	17:05:47.266	INFO	req_14056	session_6090	user_879	logout	GET	/home	
2023-12-22	17:05:47.861	INFO	req_29504	session_5260	user_978	submit_form	POST	/contact	
2023-12-22	17:05:48.387	ERROR	req_46153	session_4172	user_870	login	POST	/about	
2023-12-22	17:05:49.355	WARN	req_75270	session_9098	user_321	click_button	PUT	/about	www.bing.com
2023-12-22	17:05:49.738	INFO	req_20302	session_3533	user_352	unusual_action	GET	/login	
2023-12-22	17:03:39.042	INFO	req_32840	session_1653	user_700	unusual_action	GET	/home	www.bing.com
2023-12-22	17:03:39.769	INFO	req_97449	session_4186	user_281	unusual_action	POST	/contact	www.bing.com
2023-12-22	17:03:40.771	ERROR	req_46904	session_9926	user_895	login	POST	/contact	www.bing.com

Après avoir fait la connexion entre Power BI Desktop et Azure Databricks, nous avons fait des visualisations qui permettent de faire des analyses pertinentes et aident à la prise de décision.



Tache faite par HARRATI Yassine

Déploiement

Nous avons fait une vidéo montrant tout le travail effectué de l'implémentation aux visualisations.

Tache faite par HARRATI Yassine