




MISE EN PLACE D'UNE CHAÎNE DE STREAMING AVEC DOCKER ET KAFKA

*Collège de Bois-de-Boulogne
420-D24-BB*



Par: François Hébert et David Savard

OBJECTIFS INITIAUX DU PROJET

- *Définir une structure utilisant la conteneurisation*
- *Élaborer une chaîne de traitement (ex. Spark)*
- *Bâtir une pipeline d'ingestion de données*

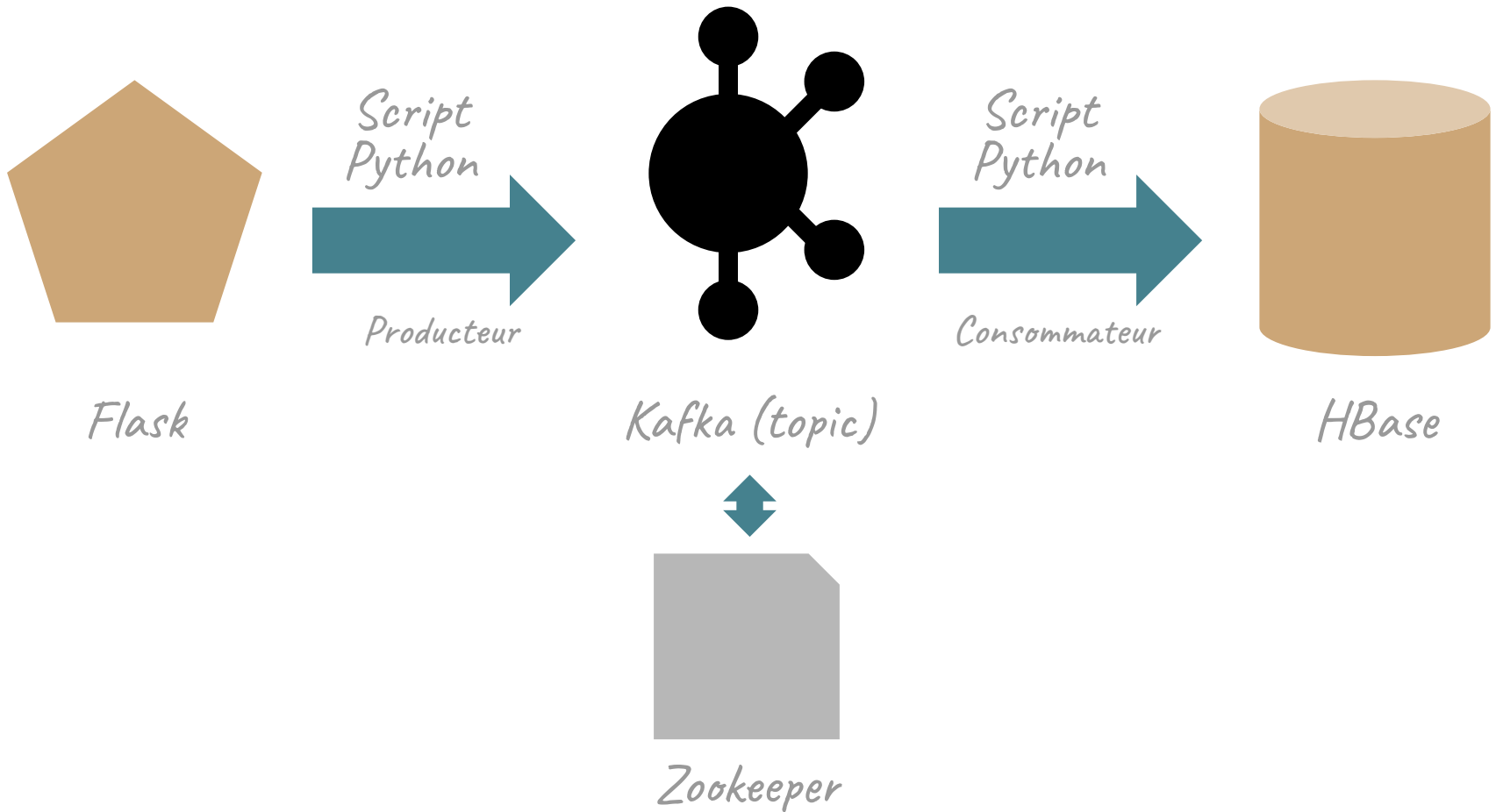
SOLUTION RETENUE

- ❖ *Une chaîne d'ingestion de données en temps réel utilisant Docker, Kafka et HBase.*

SOURCES DE DONNÉES

*Simulation d'un streaming
(par une application Flask)
de données générées par des capteurs
sous format JSON afin de représenter
des événements variés .*

```
{"time":"7:47:20.000 AM","customer":"Ewen Gillson","action":"power off","device":"Amazon Echo"}
```



& ARCHITECTURE IMPLÉMENTATION

DÉPLOIEMENT & GESTION DES SOURCES


- *Nous avons utilisé GitHub comme répertoire pour stocker les fichiers*
- *Les étapes de déploiements sont simples et bien documentées*



github.com/daprogram/projetD24.git



DÉMO!



RÉSULTAT OBTENU

*Établissement d'une chaîne de streaming fonctionnelle
pour le traitement en temps réel, démontrant efficacité et adaptabilité.*

PRINCIPAUX DÉFIS

- ★ *La synchronisation des services Docker et communication avec Kafka a été plutôt simple*
- ★ *Le défi a été de déterminer la méthodologie pour stocker les données dans HBase...*



COMPÉTENCES ACQUISES



Paramétrage d'une structure de conteneurs (ex. docker-compose, Dockerfile)

Approfondissement du langage Python

Expérimentation et compréhension accrue des flux de données (ETL)

- ★ *Le projet illustre la puissance de Kafka pour l'ingestion de données en temps réel*
- ★ *Docker est un outil idéal qui permet un déploiement et une gestion facile tout en maintenant une uniformité des environnements entre les membres d'une équipe*

CONCLUSION