

NE PAS EDITER CE PPT

EDITER LA VERSION → [ICI](#) ←

Mise en Place d'une Chaîne de Streaming avec Docker et Kafka

Collège de Bois de Boulogne -
420-D24-BB

Équipe de Réalisation

- François Hébert
- David Savard

Objectifs initiaux du Projet

- Utiliser Docker et les containers dockers
- Utiliser un framework de traitement comme spark streaming (installation de l'outil seulement)
- Utiliser un framework d'ingestion de données

Solution retenue

- Développer une chaîne de traitement de données en temps réel utilisant Docker et Kafka, en simulant l'ingestion de données de capteurs pour préparer à l'analyse avec (Spark Streaming??) et à la sauvegarde avec (?!)

Sources de Données

- Utilisation de données simulées de capteurs au format JSON pour représenter des événements variés, traitées par notre chaîne de streaming.

```
[{"time": "3:57:09.000 PM", "customer": "Alexi Barts", "action": "power off", "device": "GreenIQ Controller"},  
{"time": "1:29:05.000 AM", "customer": "Nicolle Pargetter", "action": "power off", "device": "August Doorbell Cam"},  
{"time": "6:02:06.000 AM", "customer": "Concordia Muck", "action": "power on", "device": "Footbot Air Quality Monitor"},  
{"time": "5:40:19.000 PM", "customer": "Kippar McCaughen", "action": "power off", "device": "ecobee4"},  
{"time": "4:54:28.000 PM", "customer": "Sidney Jotham", "action": "power off", "device": "GreenIQ Controller"},  
{"time": "3:12:48.000 PM", "customer": "Fanya Menzies", "action": "power off", "device": "ecobee4"},  
{"time": "5:39:47.000 PM", "customer": "Jeanne Gresch", "action": "low battery", "device": "ecobee4"},  
{"time": "2:45:44.000 PM", "customer": "Chen Cuttellar", "action": "power on", "device": "August Doorbell Cam"},  
{"time": "9:23:41.000 PM", "customer": "Merwyn Mix", "action": "power off", "device": "Amazon Echo"},  
{"time": "4:53:13.000 AM", "customer": "Angelico Conrath", "action": "power off", "device": "Amazon Echo"},  
{"time": "12:32:29.000 AM", "customer": "Gilda Emmett", "action": "power on", "device": "August Doorbell Cam"},  
{"time": "3:35:12.000 AM", "customer": "Austine Davsley", "action": "low battery", "device": "ecobee4"},  
{"time": "1:26:13.000 PM", "customer": "Zackariah Thoday", "action": "low battery", "device": "Amazon Echo"},  
{"time": "7:47:20.000 AM", "customer": "Ewen Gillson", "action": "power off", "device": "Amazon Echo"},  
{"time": "4:45:55.000 AM", "customer": "Itch Durnill", "action": "power on", "device": "ecobee4"},  
{"time": "4:12:54.000 AM", "customer": "Winni Dow", "action": "power off", "device": "GreenIQ Controller"},  
{"time": "7:35:23.000 PM", "customer": "Talbot Valentelli", "action": "power on", "device": "August Doorbell Cam"},  
{"time": "1:17:30.000 PM", "customer": "Vikki Muckeen", "action": "low battery", "device": "August Doorbell Cam"},  
{"time": "9:38:13.000 PM", "customer": "Christie Karran", "action": "power off", "device": "Footbot Air Quality Monitor"},  
{"time": "8:02:21.000 AM", "customer": "Evonne Guest", "action": "low battery", "device": "Amazon Echo"},  
{"time": "4:47:50.000 PM", "customer": "Ashia Whittet", "action": "low battery", "device": "Amazon Echo"},  
{"time": "4:00:23.000 AM", "customer": "Glen Purser", "action": "power on", "device": "Footbot Air Quality Monitor"},
```

Architecture et Implémentation

- Chaîne implémentée via Docker:
- Python pour l'ingestion
- Kafka pour l'ingestion de données,
- KafkaConsumer+Python pour le customer

Déploiement et Gestion

- Nous avons utilisé GIT pour stocker les fichiers
- Les étapes de déploiements sont simple et bien documentées

####Commandes:

```
docker-compose -f docker-compose.yml up -d
```

```
-- Pour valider que la communication est ok:
```

```
kafka-console-consumer.sh --topic sensordatainputfrancois  
--from-beginning --bootstrap-server localhost:9092
```


Principaux Défis

- La synchronisation des services Docker et communication avec Kafka a été plutôt simple
-
- Le déficit a résidé en la méthodologie pour stocker les données dans HBASE...

Compétences Acquises

- Approfondissement des connaissances en Docker (gestion des packages, déploiement, etc)
- Approfondissement du langage Python
- Meilleure compréhension des 'flu' de données (extract, transform, load)

Résultats Obtenus

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

Conclusion

- Le projet illustre la puissance de Kafka pour...
- Spark Streaming(!??)
- Docker est aussi un outil génial pour permettre un déploiement(et une gestion) facile et uniforme de la solution entre les membres de l'équipe

Questions et Discussion

- Nous vous remercions pour votre attention.
Des questions ?