

PROJET GDELT NOSQL



Adrien SENET - Elliot CANDALE - Remi GENET - Sully DILOU - Vincent POQUET

À PROPOS DU PROJET GDELT ET DE LA PROBLÉMATIQUE

GDELT 2.0 est un index de la société mondiale, un ensemble de données ouverte qui tente de rendre la société humaine

«calculable» et veut réinventer la façon dont nous étudions le monde humain en temps réel à une échelle planétaire.

Essentiellement, dans les 15 minutes suivant un événement GDELT l'a traduit, traité pour en identifier tous les articles, citations,

personnes, organisations, lieux, thèmes, émotions, etc. Problématique : Analyser l'évolution de la pandémie COVID19 via sa

couverture médiatique avec: Un système de stockage distribué, résilient et performant sur AWS Une année entière de données

GDELT Proposer des tables et visualisation pour : Afficher le nombre d'articles/événements qui parlent de COVID Pour un pays

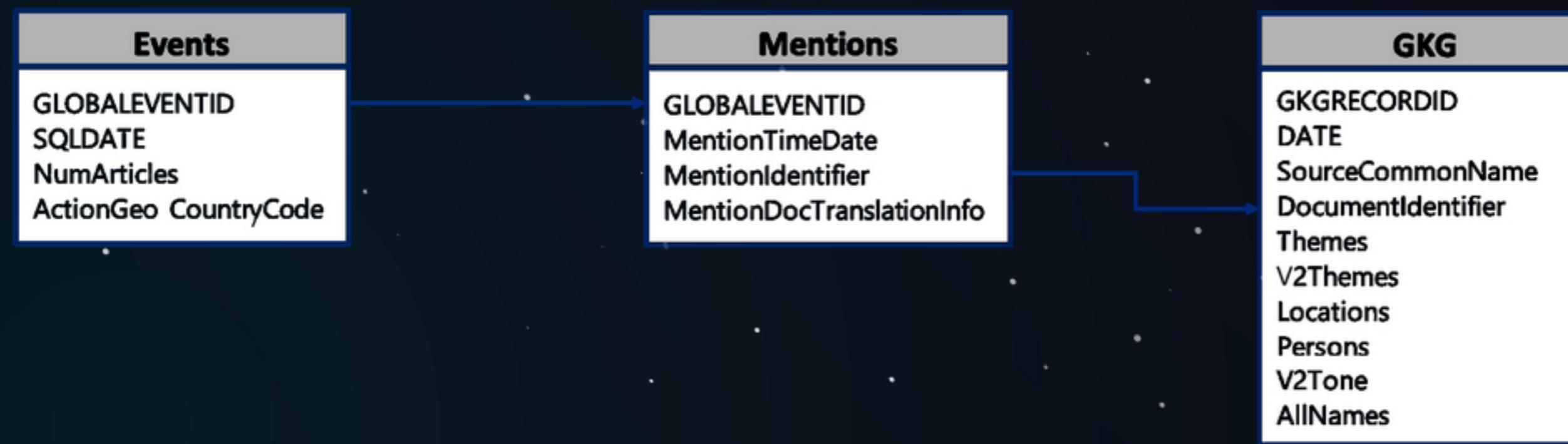
donné en paramètre, affichez les évènements qui y ont eu Pour une source de données affichez les thèmes, personnes, lieux ainsi

que le le nombre d'articles et le ton moyen Observer des patterns dans l'evolution qui pourraient nous permettre d'identifier la

prochaine vague



NOS CHAMPS CHOISI ET LA MODÉLISATION DE GDELT



DES BESOINS ET UNE STACK

Ingestion



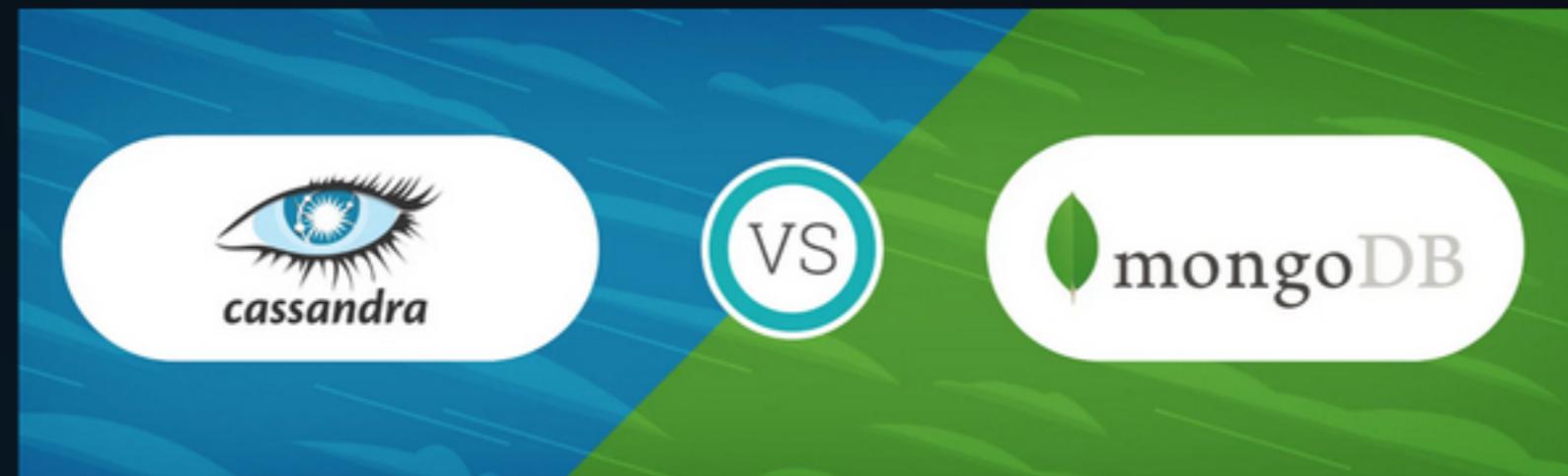
Stockage



Récupération
ETL
Visualisation



CASSANDRA VS MONGODB



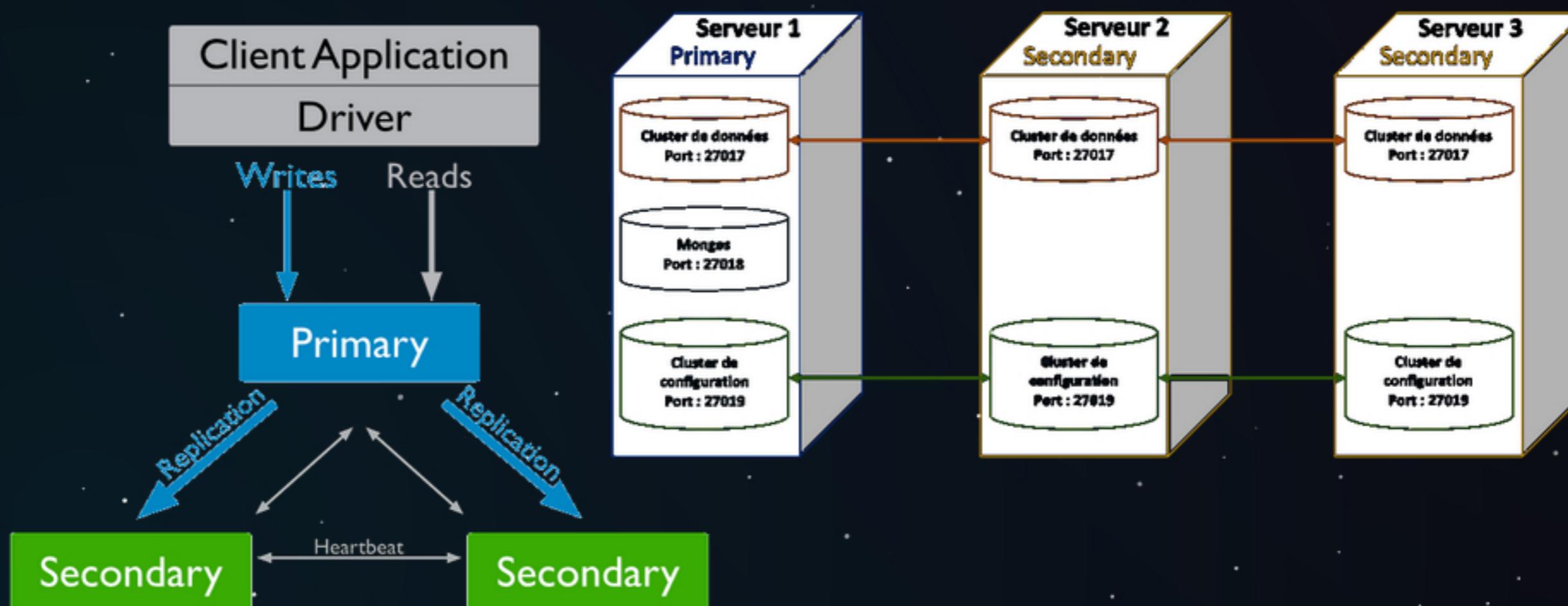
Cassandra :

- Si l'application requiert une availability élevée et nécessite des temps de réponse instantanés alors Cassandra est à privilégier
- Peu flexible car très dépendant du partitionnement des données
- Moins de fonctionnalités par rapport à MongoDb.
- Plusieurs fonctionnalités standards type SQL ne sont pas disponibles.
- Indexation moins riche que MongoDb.

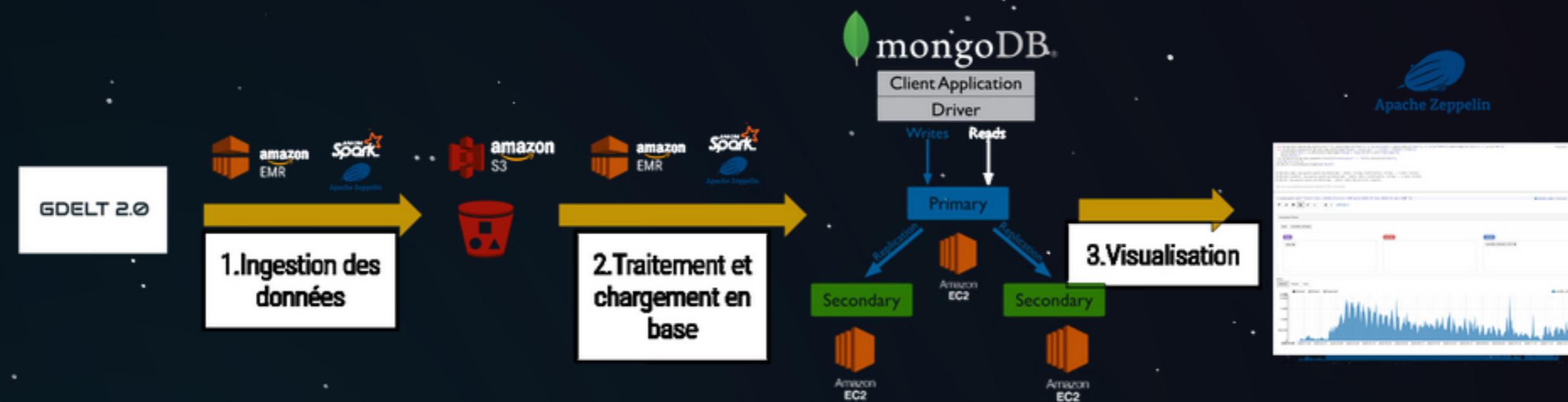
Mongo DB :

- Un master qui dirige plusieurs slaves
- Pas de schéma des données nécessaire
- Facilité de prise en main
- Adapté pour les grand volume de données
- Scalabilité et performance.
- Souplesse d'évolution de l'architecture.
- Tolérance aux pannes.
- Sharding – Multi-indexing

UTILISATION DE MONGODB



ARCHITECTURE



INGESTION DES DONNÉES

- Une équipe chargé du job d'ingestion
- Première itération en chargeant une journée de données en local des 6 bases ; events, mentions, gkg (english + translingual)
- Lancement du job sur un cluster EMR
- Près de 700 Go stocké sur S3

TRAITEMENT DES DONNÉES ET CHARGEMENT EN BASE

- Une personne en charge de l'exploration de données (préparation requêtes)
- Cluster EMR pour traiter les données via spark dans un notebook zeppelin
- Traitement par batch de mois
- Construction de dataframe de pre-traitement des données
- Une personne en charge du setup du Cluster MongoDB
- Chargement des dataframes dans la base MongoDB

VISUALISATION

- Deux personnes en charge de la visualisation des résultats
 - Requête via Zeppelin
 - Connexion via MongoDB + SparkSQL

DIFFICULTÉS RENCONTRÉES 1/2

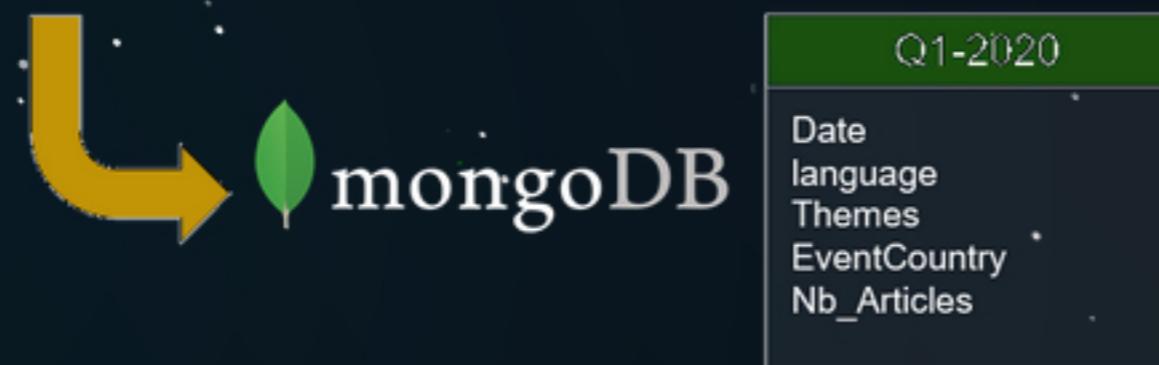
- Problèmes liés aux droits des comptes AWS EDUCATES
 - Bug connexion AWS S3a
 - Impossibilité de créer un USER IAM avec ID et Clé (necessite un token qui expire au bout de 3h)
 - Problème partage des données bucket S3 (seul le compte qui a créé le bucket parvient à accéder aux données)
 - On ne peut pas créer d'organisation/groupe

DIFFICULTÉS RENCONTRÉES 2/2

- Problèmes liés au passage à l'échelle :
 - Erreur “Timeout waiting for connection from pool”
 - Erreur “Unexpected end of ZLIB input stream”
 - Partage de crédits impossible

REQUÊTE 1

- Q1 : afficher le nombre d'articles/événements qui parlent de COVID qu'il y a eu pour chaque triplet (jour, pays de l'évènement, langue de l'article)



REQUÊTE 2



REQUÊTE 3



REQUÊTE 4

- Le nombre d'articles a tendance à augmenter avant une accélération du nombre de cas
- Difficulté pour appliquer un algorithme de ML pour manque de temps
- Il pourrait être intéressant de recouper par pays les articles sur diverses épidémies afin d'observer si avec GDELT un modèle peut prédire les futures vagues.
- Une analyse en temps réel selon les sorties d'articles pourrait prévoir les risques d'une épidémie
- Il y a un décalage de 7 à 15 jours avant la vague ce qui est trop tard avec l'incubation mais permet des décisions plus rapides



BUDGET

- ETL et Traitement des données :
 - EMR : m4.large (ingestion en base) : 0,03 USD par heure par instance =>
 - S3 : (stockage des données brutes) : 0,023 USD par Go par mois =>
 - Total: ~120\$
- Stockage en base de données :
 - EC2 : m4.large : 0,111 USD par heure par instance
 - EBS : 3*200Go
 - Total: ~60\$

AMÉLIORATIONS POSSIBLES 1/2

- Instances spots pour les machines EC2
- Déploiement automatique du cluster MongoDB via un script ou via une autre techno type Ansible
- Meilleur utilisation du sharding et des indexes
- ETL des données aggregées de MongoDb vers une RDBMS type Redshift
- Utiliser un outil de visualisation (Quicksight, Tableau, PowerBi, Mongo Charts) ou une webApp (Metabse, Streamlite, Dash by Plotly)
- Automatiser une pipeline d'ingestion des données toutes les 15mn

AMÉLIORATIONS POSSIBLES 2/2

Alternative

- Fichiers Parquets + Athena (équivalent AWS de Big Queery)
- Serverless (donc résilient)
- Facturation à l'usage (par volume requêté)



DÉMONSTRATION

