Projet de groupe : une base de données analytique

En tant que nouvelle recrue du département Informatique, on vous a confié la tâche de concevoir une petite infrastructure analytique pour une direction de l'entreprise. Pour des raisons de coût, il a en effet été décidé de **déplacer un cas d'utilisation tournant aujourd'hui dans Snowflake** sur une plafeforme déployée en interne sur un des **clusters Kubernetes**. Le budget est, vous vous en doutez, **extrêmement** limité.

Le modèle de données qui doit être stocké est un schéma classique en étoile / flocon, pesant un peu moins de 1 To non compressé. La croissance en volume prévue est de **+10% par an**.

> L'ingestion des données depuis les systèmes de production (essentiellement des bases PostgreSQL) sera réalisée par une autre équipe à l'aide d'un outil d'intégration de données. Ce n'est pas de votre ressort.

Les utilisateurs principaux sont des ***Business Analysts*** qui consomment des données à l'aide d'un outil de *dashboarding* open source, **Apache Superset**. Ils s'attendent à ce que les temps de réponse des requêtes soient inférieurs à 1 seconde en moyenne.

Une autre population devra utiliser la plateforme, les ***Data Scientists***. Ils écrivent des requêtes plus complexes à partir de notebooks **Jupyter**. Ils acceptent des temps de réponse plus lents car leurs requêtes scannent beaucoup de données et sont extrêmement gourmandes en ressources.

Si ce projet est un succès, il est imaginé de déplacer de nouveaux cas d'utilisation sur cette plateforme.

Plan d'action conseillé

1. Documentation

Renseignez-vous sur ces technologies et essayez de bien comprendre ce que vous devez remplacer.

2. Exploration

1. Sélectionnez quelques bases de données qui, selon vous, répondraient à ces besoins. Vous pourriez par exemple:
    - Opter pour un SGBDR classique, en utilisant quelque chose comme PostgreSQL ou MariaDB. Mais cela va-t-il correctement monter à l'échelle? Y a t-il des indexes, des extensions appropriés à ce cas précis d'utilisation?
    - Prendre quelque chose qui *scale* horizontalement, comme ClickHouse, Druid jusqu'à des choses comme Trino. Plus difficile à mettre en oeuvre, mais peut être une bonne idée pour la suite? Our "sur-ingéniéré"?
    - Vous pourriez également concevoir votre infra autour de choses plus exotiques, comme une stack DuckDB + MinIO

    > Autre ? Il y a tellement d'options!

2. Déployez votre base de données localement. Utilisez une abstraction comme `docker`, `docker-compose`, ou une implémentation locale de k8s comme `minikube`, `k3s` ou `microk8s` car elles

permettent de reproduire facilement votre infrastructure.

3. Insérez des données dans votre instance pour vos tests
    - **Astuce** : vous pouvez créer vos propres jeux de données, ou utiliser des jeux de données gratuits disponibles en ligne, tels que TPC-H, ou le célèbre *New York Taxi Dataset*. Pensez également que la plupart des bases de données ont leur propre base de données exemple que vous pouvez charger, généralement mentionnée sur leur site web. Par exemple le wiki Postgres liste toutes les bases d'exemples disponibles à l'adresse https://wiki.postgresql.org/wiki/Sample_Databases
4. Testez! Lancez Superset (faites un petit dashboard joli!), construisez un notebook Jupyter ou une petite application qui écrit du SQL : le choix est le votre. Assurez-vous que ces tests soient pertinents et intéressants : ils feront partie de votre livrable et doivent convaincre l'auditoire (fictif).

3. Demande d'avis

Enfin, rédigez une RFC détaillant votre processus de réflexion avec **au moins** les parties suivantes :

- **Introduction** : faites un récapitulatif du problème que vous devez résoudre et des principaux objectifs
- **Contraintes** : quelles sont les contraintes dans lesquelles vous opérez pour résoudre ces besoins
- **Design** : pourquoi avez-vous choisi cette solution, à quoi elle ressemble et comment elle s'intègre dans l'écosystème de l'entreprise
- **Implémentation** : mettez en évidence certaines parties importantes de votre design, illustrées par les tests que vous avez effectués
- **Points négatifs** : les aspects pas si géniaux de votre solution. quels sont les problèmes que vous imaginez surgir dans les semaines/mois/années suivant le déploiement
- **Alternatives envisagées** : ce que vous avez examiné et rejeté (et pourquoi !)

Livrables

- Un **RFC** avec une structure appropriée comme mentionné ci dessus, exporté au format PDF. Le document devrait faire au moins 6000 signes, illustré par des diagrammes d'architecture.
- Une **PoC de de vorte infrastructure**, partagée via un repo Git, contenant:
    - votre infrastructure packagée au choix comme un **Dockerfile**, un **docker-compose**, ou des **manifestes Kubernetes**
    - un fichier README.md au format markdown, expliquant comment exécuter les quelques exemples d'utilisation de votre platforme. Rappellez vous qu'ils peuvent prendre à peu près **n'importe quelle** forme (soyez créatif !)

Date limite

Ce projet est à rendre pour le 30 mars (date à confirmer)

Dernier conseil

Il n'y a pas de bon choix, ce qui importe le plus ici est de voir que vous avez compris le problème et constater quel a été votre cheminement.