LOGS APIs

**Architecture technique :**

Schema

API Talend -> Apache ActiveMQ -> middleware -> (Logstash -> ElasticSearch -> Kibana)ELK

Définir chaque modules de l’architecture

**Infrastructure proposée**

1. **API Talend** :
   * Capture les informations de log (requêtes, timestamp, utilisateur, etc.).
   * Envoie les logs formatés en JSON à une queue ActiveMQ de manière asynchrone.
2. **ActiveMQ** :
   * Reçoit et stocke les messages de log provenant de l'API Talend.
   * Assure la persistance et la fiabilité des messages en cas de panne.
3. **Script de traitement des logs** :
   * Un script (par exemple, en Python ou Java) est développé pour consommer les messages de la queue ActiveMQ.
   * Le script traite les logs et les envoie à Logstash.
4. **Logstash** :
   * Reçoit les logs du script de traitement, les filtre, et les envoie à Elasticsearch.
5. **Kibana** :
   * Utilisé pour visualiser et analyser les logs.

**Avantages de l'utilisation d'ActiveMQ :**

Couplage lâche : ActiveMQ permet de découpler la génération de logs dans l’API de leur traitement et de leur stockage. Cela signifie que ton application Talend n'a pas besoin d'attendre que les logs soient stockés avant de continuer à traiter les requêtes.

Fiabilité : ActiveMQ offre des fonctionnalités de persistance et de redondance, ce qui peut être utile pour s'assurer que les logs ne sont pas perdus. ACID ?

Scalabilité : Il est possible de traiter les messages de la queue de manière distribuée, ce qui peut aider à gérer un volume important de logs.

**Prérequis** :

A compléter later.

Server Linux avec Java et Docker

**Mise en place POC and POV**

1. Installation de Apache Active MQ