

Questions middleware

Traiter par Oumaima fekih
FIA02-GL-03

● *Apache Kafka*



- **Apache Kafka** était un système de messagerie distribué, initialement créé et développé par LinkedIn, passé en open source en 2011, il est actuellement développé par la Apache Software Foundation. Au fil du temps, cet outil a beaucoup évolué. Il s'agit aujourd'hui d'une plateforme centralisée pour le stockage et l'échange en temps réel de toutes les données émises par les entreprises qui l'utilisent.

● *Apache ActiveMQ™*



- **Apache ActiveMQ™** est le serveur de messagerie libre, multiprotocole et basé sur Java le plus populaire. Il prend en charge les protocoles standard de l'industrie afin que les utilisateurs tirent profit des choix des clients dans un large éventail de langues et de plateformes.. Échangez des messages entre vos applications web à l'aide de STOMP sur des websockets. Gérez vos appareils IoT à l'aide de MQTT

● *MQTT*



MQTT (Message Queuing Telemetry Transport) est un protocole de messagerie publish-subscribe basé sur le protocole TCP/IP. Il a été initialement développé par Andy Stanford-Clark (IBM) et Arlen Nipper (EuroTech). Il est conçu pour les connexions avec des sites distants où la bande passante du réseau est limitée.



ACTIVE**MQ**

VS



kafka



MQTT

Etude comparative entre les middlewares MOM suivants : ActiveMQ, MQTT et Kafka

ActiveMQ	MQTT	Kafka
<p>Il s'agit d'un système de messagerie traditionnel qui traite une petite quantité de données. Il a les cas d'utilisation suivants:</p> <ul style="list-style-type: none">•Messagerie transactionnelle•Distribution de données de marché hautes performances•Clustering et modèle de messagerie asynchrone à usage général•Streaming Web de données•API reposante pour la messagerie via HTTP	<p>Il s'agit d'un un protocole de messagerie standard OASIS pour l'Internet des objets . Idéal pour connecter des périphériques distants avec un faible encombrement de code et une bande passante réseau minimale. Il a les cas d'utilisation suivants:</p> <ul style="list-style-type: none">•Transport•Automobile• Télécommunications•Industrie	<p>Il s'agit d'un système distribué destiné à traiter une énorme quantité de données. Il a les cas d'utilisation suivants:</p> <ul style="list-style-type: none">•Messagerie•Suivi des activités du site Web•Métrique•Agrégation de journaux•Traitement de flux•Sourcing événementiel•Journal de validation

<p>Il prend en charge les transactions. Les deux niveaux de prise en charge des transactions sont les suivants:</p> <p>Transactions JMS</p> <p>Transactions XA</p> <p>Il utilise TransactionStore pour gérer les transactions.</p> <p>TransactionStore mettra en cache tous les messages et ACKS jusqu'à ce que la validation ou la restauration se produise.</p>	<p>MQTT est un protocole plus simple avec seulement cinq API conçues pour connecter des appareils. Il ne peut pas être utilisé comme middleware pour les systèmes transactionnels.</p>	<p>Kafka ne supportait initialement pas les transactions, mais depuis sa version 0.11, il prend en charge les transactions dans une certaine mesure.</p>
<p>Il maintient l'état de livraison de chaque message, ce qui entraîne un débit inférieur.</p>	<p>La fiabilité de la livraison des messages est importante pour de nombreux cas d'utilisation de l'IoT. C'est pourquoi MQTT a 3 niveaux de qualité de service définis: 0 - au plus une fois, 1 - au moins une fois, 2 - exactement une fois</p>	<p>Les producteurs de Kafka n'attendent pas les remerciements des courtiers. Ainsi, les courtiers peuvent écrire des messages à un taux très élevé, ce qui entraîne un débit plus élevé</p>

Il ne fournit pas de somme de contrôle pour détecter la corruption des messages hors de la boîte.	Il facilite le chiffrement des messages à l'aide de la couche de sécurité TLS et l'authentification des clients à l'aide de protocoles d'authentification modernes, tels que OAuth.	Il comprend des sommes de contrôle pour détecter la corruption des messages dans le stockage et dispose d'un ensemble complet de fonctionnalités de sécurité.
Il ne peut pas garantir que les messages sont reçus dans le même ordre qu'ils ont été envoyés		Il peut garantir que les messages sont reçus dans l'ordre où ils ont été envoyés au niveau de la partition.
Il s'agit d'une plate-forme de messagerie de type push où les fournisseurs envoient les messages aux consommateurs.		Il s'agit d'une plate-forme de messagerie de type pull où les consommateurs tirent les messages des courtiers.
il est de la responsabilité des producteurs de s'assurer que les messages ont bien été livrés.		il est de la responsabilité des consommateurs de consommer tous les messages qu'ils sont censés consommer.

Est-il possible de porter une solution middleware au niveau du cloud ?

- Oui , il est possible de porter une solution middleware au niveau du cloud car il présente plusieurs avantages:
- Gain d'exploitation
- Dimensionnement
- Redondance simplifiée
- Sécurité des données

Est-ce possible à l'aide de Openshift ?

- OpenShift est une plateforme de Containers-as-a-Service dont l'objectif est de simplifier la gestion des environnements de production, qu'ils soient métiers ou IT, hébergés en mode Cloud ou non.
- Dans ce cas il est possible de porter une solution middleware au niveau du cloud à l'aide de openshift
- OpenShift est un excellent outil pour provisionner et gérer les environnements de développement et d'exécution des applications.
- La solution proposée par Red Hat offre de nombreux avantages :
- Productivité , simplicité et évolutivité