

Sincronizzazione dei Dati in un Cluster Docker Swarm con MQTTS

Michele Montesi

E-Mail: michele.montesi3@studio.unibo.it

June 18, 2023

1 Introduzione

In questo documento, verrà presentato il progetto di sincronizzazione dei dati in un cluster Docker Swarm utilizzando il protocollo MQTTS. Sarà illustrato il contesto applicativo, il problema specifico da risolvere e la soluzione implementata, comprese le tecnologie e le configurazioni adottate.

2 Contesto Applicativo

Il progetto parte da un ambiente Docker Swarm, che è un sistema di orchestrazione di container che permette di gestire un cluster di nodi Docker. Inoltre, si assume che sia stato configurato un servizio di messaggistica MQTT (Message Queuing Telemetry Transport) su tutti i nodi del cluster, noto come MQTTS (MQTT over TLS/SSL). L'utilizzo di MQTTS fornisce una comunicazione sicura e cifrata tra i nodi del cluster Swarm.

L'obiettivo del progetto è quello di ottenere un cluster Docker Swarm in cui il servizio MQTTS sia in esecuzione su tutti i nodi del cluster. Inoltre, è necessario garantire la persistenza dei dati e la loro sincronizzazione tra i nodi, utilizzando l'utilità `lsyncd`.

3 Problema Specifico

Il problema che si presenta è la necessità di sincronizzare i dati tra i nodi del cluster Docker Swarm. A causa della natura distribuita del cluster, con i suoi nodi indipendenti, potrebbero verificarsi situazioni in cui i dati siano inconsistenti o disallineati tra i vari nodi. È fondamentale risolvere questo problema per garantire l'integrità e la coerenza dei dati nell'ambiente Docker Swarm.

4 Soluzione Implementata

Per affrontare il problema della sincronizzazione dei dati tra i nodi del cluster Docker Swarm, è stata adottata la soluzione seguente:

4.1 MQTTS

Il servizio di messaggistica MQTT è stato configurato su tutti i nodi del cluster Swarm utilizzando il protocollo MQTTS, che fornisce una connessione sicura e cifrata. L'utilizzo di MQTTS garantisce la riservatezza e l'autenticazione durante la comunicazione tra i nodi.

4.2 lsyncd

Per garantire la sincronizzazione dei dati tra i nodi del cluster, è stata utilizzata l'utilità `lsyncd` (Live Syncing Daemon). `Lsyncd` è uno strumento che permette di monitorare le modifiche apportate ai file in una directory e di propagarle automaticamente a una o più destinazioni remote.

È stata configurata un'apposita regola di sincronizzazione in `lsyncd` per monitorare le directory dei dati all'interno di ciascun nodo del cluster Docker Swarm e propagare automaticamente le modifiche ai nodi rimanenti. In questo modo, qualsiasi modifica effettuata su un nodo verrà sincronizzata in modo automatico e trasparente a tutti gli altri nodi del cluster.

4.3 Persistenza dei Dati

Per garantire la persistenza dei dati nel cluster Docker Swarm, è stato configurato un volume Docker per le directory dei dati. I volumi Docker consentono di separare i dati dal container, consentendo loro di persistere anche in caso di riavvio del container o del nodo. Ciò garantisce che i dati rimangano disponibili anche in situazioni di fallimento o manutenzione.

5 Conclusioni

In questa relazione è stato presentato il progetto di sincronizzazione dei dati in un cluster Docker Swarm con MQTTS. Il progetto è stato motivato dal problema della sincronizzazione dei dati tra i nodi indipendenti del cluster Swarm. È stata presentata la soluzione implementata, che utilizza il protocollo MQTTS per la comunicazione sicura e l'utilità `lsyncd` per la sincronizzazione automatica dei dati tra i nodi. Si è inoltre sottolineata l'importanza della sincronizzazione dei dati in un ambiente distribuito come Docker Swarm per garantire l'integrità e la coerenza dei dati tra i nodi del cluster.