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 che utilizza 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, utilizzando gli stessi dati. Inoltre, è necessario garantire la persistenza dei dati e la loro sincronizzazione tra i nodi, utilizzando l'utility 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 lsyncd

Per garantire la sincronizzazione dei dati tra i nodi del cluster, è stata utilizzata l'utility 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.

4.2 Ansible

Per automatizzare l'installazione di lsyncd su tutti i nodi del cluster Docker Swarm, è stato utilizzato Ansible. Ansible è un framework di automazione IT che semplifica la distribuzione, la configurazione e la gestione delle applicazioni e delle infrastrutture. Attraverso l'utilizzo di playbook Ansible, è stato possibile definire e eseguire in modo dichiarativo le operazioni di installazione e configurazione di lsyncd su tutti i nodi del cluster.

5 Sviluppo degli Script Ansible

Per lo sviluppo degli script Ansible utilizzati in questo progetto, sono stati seguiti i seguenti passaggi:

5.1 Struttura del Progetto

Il progetto Ansible è stato organizzato seguendo una struttura standard, con i seguenti elementi principali:

- `inventory.yml`: Un file di inventario che elenca tutti i nodi del cluster Docker Swarm su cui verrà eseguita l'installazione di lsyncd.
- `lsyncdInstall.yml`: Il playbook che gestisce l'installazione del pacchetto lsyncd su tutti i nodi del cluster Docker Swarm.
- `lsyncdConf.yml`: Il playbook principale che definisce le operazioni da eseguire per la configuraizone di lsyncd.
- `roles/`: Una directory che contiene i ruoli Ansible. Ogni ruolo rappresenta una specifica funzionalità o configurazione da applicare sui nodi del cluster.
- `roles/setup-lsyncd/`: Il ruolo che definisce la configurazione di lsyncd su ogni nodo del cluster Docker Swarm.
- `roles/start-lsyncd/`: Il ruolo che definisce l'avvio del servizio lsyncd su ogni nodo del cluster Docker Swarm.

5.2 Definizione dei Ruoli

Per definire i ruoli Ansible per la configurazione e l'avvio di lsyncd, sono state eseguite le seguenti attività:

- Creazione delle directory:
  - `roles/setup-lsyncd/`
  - `roles/start-lsyncd/`
- Definizione di un file `tasks/main.yml` all'interno del ruoli. Questo file contiene l'elenco delle attività da eseguire per installare e configurare lsyncd.
- Definizione di file di configurazione di lsyncd all'interno del ruolo `setup-lsyncd`: `templates/lsyncd.conf.j2`.
- Definizione di file di variabili all'interno di `setup-lsyncd` per definire i percorsi da sincronizzare.

5.3 Utilizzo dei Playbook

Per eseguire l'installazione, e poi la configurazione, di lsyncd su tutti i nodi del cluster Docker Swarm, è sufficiente eseguire i seguenti comandi:

```
ansible-playbook -i inventory.ini -vv lsyncdInstall.yml
ansible-playbook -i inventory.ini -vv lsyncdConf.yml
```

6 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. Questi dati saranno sincronizzati attraverso lo script ansible descritto in precedenza.

7 Conclusioni

In questa relazione è stato presentato il progetto di sincronizzazione dei dati in un cluster Docker Swarm con MQTTS tramite lsyncd. Il progetto è stato motivato dal problema della sincronizzazione dei dati tra i nodi indipendenti del cluster Swarm. È stata presentata la soluzione implementata, l'utility 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.