

Generazione di numeri random dentro un cluster Docker Swarm

Luca Rapolla

August 2023

1 Introduzione

L'applicazione consiste nella semplice generazione di numeri random di lunghezza prefissata. L'infrastruttura è composta da un client ed un server di cui il primo gestisce l'interfaccia utente ed il secondo controlla la parte logica dell'applicativo. Il tutto è sorretto da 3 macchine virtuali autogenerate su cui è stato costruito un cluster basato su *Docker Swarm*.

2 Tecnologie Utilizzate

2.1 Vagrant

Per la costruzione delle 3 macchine virtuali, mi sono servito di Vagrant, uno strumento *Open Source* per la costruzione di ambienti virtualizzati. Tramite l'utilizzo di un file chiamato **VagrantFile** è possibile descrivere le caratteristiche di una o più macchine virtuali che il tool andrà a creare.

2.2 Ansible

Ansible è un framework *Open Source* per la gestione e configurazione di sistemi. Tramite il suo utilizzo, un amministratore ha la possibilità di descrivere delle regole, tramite sintassi YAML, all'interno di un file chiamato **playbook** cui, tramite il suo utilizzo e quello di un altro file chiamato **inventory**, al cui interno sono contenuti gli ip delle macchine da configurare, esegue il setup automatico e senza agenti delle macchine target.

2.3 Docker / Docker swarm

Docker è una piattaforma di *containerizzazione* che permette di creare, distribuire e gestire applicazioni in ambienti isolati chiamati "container". Questi contengono tutto il necessario per eseguire un'applicazione, rendendo il deploy consistente tra diversi ambienti. Docker Swarm, incluso in Docker, è un tool per creare e gestire cluster di container. Consente di scalare orizzontalmente le applicazioni, distribuendole tra i nodi del cluster migliorandone l'affidabilità e la disponibilità.

2.4 Python Flask

Flask è un framework per lo sviluppo di applicazioni web utilizzando Python. Offre strumenti essenziali per costruire applicazioni web in modo semplice ed efficiente consentendo la gestione delle richieste HTTP, la creazione di rotte URL, e l'integrazione di template per la generazione dinamica di pagine HTML.

2.5 Nginx

Nginx è un server web *Open Source* noto per la sua velocità, scalabilità ed efficienza. Agisce come proxy inverso gestendo le richieste HTTP tra client e server di backend. La sua struttura modulare permette l'integrazione di funzionalità aggiuntive, rendendolo una scelta popolare per migliorare l'affidabilità e la distribuzione delle applicazioni web.

2.6 NFS (Network File System)

NFS è un network di configurazione file che consente ai computer su una rete di accedere e condividere risorse di archiviazione come se fossero locali. Tramite NFS possono montare risorse remote come se fossero locali consentendo di accedere ai dati in modo efficiente.

3 Funzionamento dell'Applicazione

Come prima cosa bisogna creare e mettere in funzione le macchine virtuali su cui poggerà l'intera applicazione. Per fare ciò ho creato un **VagrantFile** su cui ho descritto 3 macchine con i requisiti minimi per supportare Docker basate su debian. Le 3 macchine hanno i seguenti indirizzi IP:

- Macchina manager: 10.133.7.101
- Macchina worker 1: 10.133.7.111
- Macchina worker 2: 10.133.7.112

Dopo aver completato il file di configurazione, ho utilizzato il comando **vagrant up** per far partire la routine di creazione delle macchine virtuali. In seguito ho creato il file **inventory.ini**, utile per la configurazione delle macchine tramite Ansible, ed il **playbook.yml**, il quale andrà a descrivere tutte le regole di configurazione. In particolare, il **playbook.yml** è diviso in 5 macro sezioni, una per attività:

- Installazione di Docker e Join delle macchine al cluster swarm.
- Installazione della directory condivisa.
- Pulizia della directory condivisa.
- Inserimento dei file necessari all'interno della directory condivisa.
- Deploy dei servizi all'interno del Cluster

Di seguito, eseguendo il comando **ansible-playbook playbook.yml -i inventory.ini** avverrà l'installazione ed il setup di tutto ciò che serve all'applicazione per funzionare. In particolare, la cartella condivisa è mappata, tramite il file docker compose, sia alla cartella *html* di nginx sia alla cartella di lavoro del container

del webserver. In seguito non ci resta altro che utilizzare l'applicazione, raggiungendo uno degli indirizzi ip sopra elencati e scegliendo il numero di cifre che il numero dovrà avere. In particolare, il numero massimo di cifre generabili, viene scelto andando ad impostare la variabile `MAXBITS` all'interno del `Dockerfile`