1 Il Load Balancer NGINX

A questo servizio si collegheranno tutti i client che necessitano di accedere alla risorsa.

1.1 Dockerfile

```
1  FROM nginx:1.20.0
2  RUN echo -e "\t\tCopying Config"
3  COPY Contents/nginx.conf /etc/nginx/nginx.conf
4  RUN echo -e "\t\Setting Permissions"
5  RUN chmod 555 /etc/nginx/nginx.conf
```

Listing 1: Dockerfile Load Balancer NGINX

1.1.1 Analisi Dockerfile

Il file è uguale a quello visto nella **??** tranne che per il fatto che non carichiamo i file del webserver.

1.2 Servizio docker compose

```
1 LoadBalancer:
2   build: Dockerfiles/load-balancer/.
3   image: loadbalancer
4   container_name: LB
5   networks:
6     - Internal
7     - External
8   ports:
9     - "80:80"
```

Listing 2: Load Balancer Docker Compose

La prima riga indica il nome univoco del servizio.
Riga 2 è opzionale e indica il percorso in cui effettuare la build dell'immagine se questa non è presente.
Riga 3 indica il nome dell'immagine. Se non è presente in locale verrà o presa dalla repo remota o buildata (se è presente l'istruzione build).
Riga 4 indica un nickname per il servizio.
Riga 6 impone al container di collegarsi al network *Internal*, il quale non ha accesso alla rete esterna.
Riga 7 Consente al container di collegari al network *External*, il quale ha accesso alla rete esterna.
Riga 9 Specifica il mapping delle porte in ingresso, la porta del container deve coincidere con quella specificata nel config del Load Balancer.

Per implementare un sistema multi host La riga 6 non è necessaria, dato che le macchine saranno collegate tramite rete esterna.

1.3 Configurazione

```
1  events {}
2  http {
3    upstream balanceGroup1 {
4      server WebServer1:80;
5      server WebServer2:80;
6      server WebServer3:80;
7      server WebServer4:80;
8    }
9
10   server {
11     listen 80;
12
13     location / {
14       proxy_pass http://balanceGroup1;
15     }
16   }
17 }
```

Listing 3: File di configurazione Load Balancer NGINX

Alla riga 3 specifichiamo un gruppo di server di nome *balanceGroup1*.
Dalla riga 4 alla riga 7 specifichiamo gli indirizzi dei server appartenenti a *balanceGroup1*, in questo caso vengono utilizzati degli indirizzi appartenenti alla rete *Internal* ma possono essere sostituiti con normalissimi indirizzi HTTP(S).
Alla riga 11 specifichiamo su che porta ascoltare.

Per implementare un sistema multi host Sostituire gli indirizzi alle righe 4-7 con gli indirizzi effettivi dei propri host su cui gira webserver (**??**), prestare particolare attenzione alle porte su cui rispondono i webserver (**??**).

1.3.1 Modalità di load balancing

NGINX supporta 3 modalità di load balancing:

1. Round Robin
   - Round Robin standard (Default)
   - Round Robin pesato
2. Least Connected
3. Session Persistence (ip-hash)

Round Robin La tecnica Round Robin consiste nel dividere il carico proporzionalmente tra i vari host.

Round Robin standard (Default) In questo caso tutti gli host hanno lo stesso peso → il carico viene distribuito equamente tra tutti.

Round Robin pesato In questo caso vengono specificati dei pesi per alcuni o tutti i server, il round robin distribuisce le richieste tenendo conto dei pesi specificati.
In caso di omissione del peso questo viene considerato pari a 1.

```
upstream balanceGroup1 {
  server WebServer1:80 weight=3;
  server WebServer2:80;
  server WebServer3:80 weight=2;
  server WebServer4:80;
}
```

Listing 4: Round Robin pesato

In questo caso specifico i server 1 e 3 riceveranno rispettivamente il triplo e il doppio delle richieste rispetto ai server 2 e 4.

Least Connected La tecnica Least Connected tiene traccia del carico di ogni server e reindirizza al server con meno carico al momento della richiesta.
Questa tecnica è molto utile nel caso in cui il tempo di risposta ad una richiesta vari significativamente in base alla richiesta, in questo caso con Least Connected evitiamo di caricare un server con molte richieste in elaborazione.

```
upstream balanceGroup1 {
  least_conn;
  server WebServer1:80;
  server WebServer2:80;
  server WebServer3:80;
  server WebServer4:80;
}
```

Listing 5: Least Connected

Session Persistence (ip-hash) Questa tecnica assegna ad ogni ip sempre lo stesso server usando una funzione di hash per mappare ad ogni ip un server.

```
upstream balanceGroup1 {
  ip_hash;
  server WebServer1:80;
  server WebServer2:80;
  server WebServer3:80;
  server WebServer4:80;
}
```

Listing 6: Session Persistence

Informazioni più dettagliate sulle tipologie di load balancing di NGINX si possono trovare nella documentazione di NGINX.