

# 1 Docker Swarm

## 1.1 `--scale`

Avviando il docker compose con il flag `--scale nome_servizio=n` docker crea automaticamente  $n$  istanze del servizio `nome_servizio` e gestisce automaticamente il load balancing con tecnica Round Robin.

Da notare che per effettuare uno scaling automatico, docker richiede che nel compose del servizio non sia presente l'attributo `container_name`.

Questa era un'opzione disponibile anche come flag dentro a docker compose v2 ma è stata deprecata in favore di docker swarm.

Nel caso di questo progetto, per avviare molteplici istanze del webserver il comando sarebbe dovuto essere `docker compose up WebServer1 --scale WebServer1=4`

## 1.2 Docker Swarm

Docker Swarm è una feature che consente l'unione di più istanze docker su macchine differenti sotto un controllo centralizzato.

Ogni istanza di docker, chiamata *nodo*, può avere uno dei due ruoli:

- Manager
- Worker

Il nodo manager conosce in ogni momento lo stato dei nodi worker e si occupa di dividere le task (e, di conseguenza, il carico) fra questi.

Il nodo manager è anche responsabile di gestire i vari errori dei singoli nodi, reindirizzando le task ad altri nodi worker.

Nel momento in cui si ha un docker swarm funzionante, per replicare il sistema realizzato in questo progetto (ignorando il server log) è sufficiente aggiungere al docker compose del webserver l'attributo `scale: 4`.

Questo attributo, disponibile solo se si usa docker in modalità swarm, crea in automatico 4 istanze del webserver e gestisce il load balancing tra queste.

Per maggiori informazioni su docker swarm visitare la [documentazione ufficiale](#).