

Laboratorio Network Security 2

Containerizzazione e networking con Docker

Informazioni utili

L'indirizzo della piattaforma in cui sono presenti le challenge verrà inviato sul canale Telegram.

Solo esercizi 0 e 2 sono caricati sul portale. Gli altri non richiedono di trovare la flag e sono indicati solo sulla slide.

PER FAVORE SCARICATE I FILE DA CLASSROOM NON DALLA PIATTAFORMA!!!!

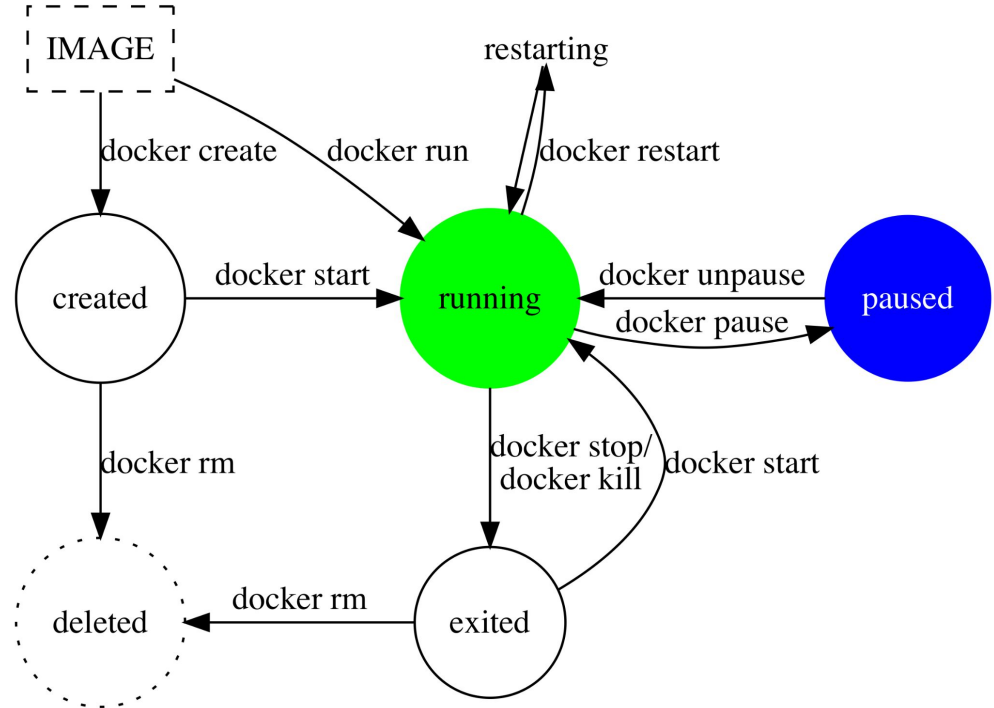
Ripasso Docker

Per non doversi imparare i comandi a memoria si può usare un comodo cheat sheet come questo.

[Docker cheat sheet in formato PDF](#)

In rete se ne trovano anche altri per quanto riguarda Compose e la parte di Networking

Container lifecycle



Esercizio 0 - docker_things_up

A partire dal Dockerfile è necessario creare un'immagine Docker (build) e successivamente istanziare un container (run) che esponga la porta 8888.

Tutto il necessario si trova nel file `docker_things_up.zip`

- It works on my computer

- Yes, but we are not going to give your computer to the client



Esercizio 1 - DockerNetworko101

Si istanzino 2 container Alpine Linux e collegandosi ad uno si provi a “pingare” il secondo usando prima il suo indirizzo ip e poi usando il nome del container.

Successivamente, si crei una rete bridge custom e si colleghino i 2 container Alpine. Ora si riprovi il ping (si usando l'indirizzo ip che il nome host) usando il nome del container.

Alla fine si possono rimuovere i container e la rete.

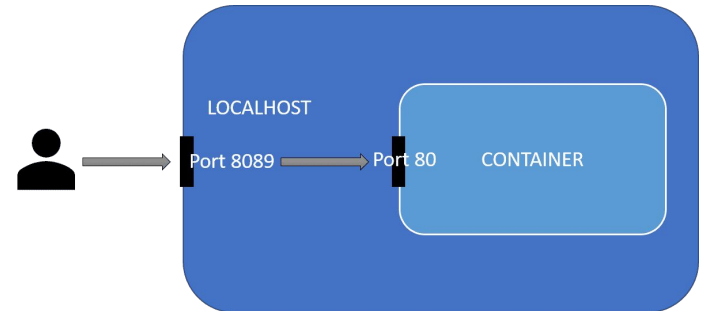
Esercizio senza flag!



Esercizio 1 - DockerNetworko101

Quando di istanzia un container sarà automaticamente collegato alla rete bridge di default, in cui fra le altre cose possiamo fare port forwarding dall'host ai container (opzione -p del comando run)

Creando una rete bridge custom possiamo isolare i container e risolvere i nomi. Il servizio di nomi risulta particolarmente utile quando si creano applicazioni multicontainer (Docker Compose)



Esercizio 2 - compose_things_up

Si metta in esecuzione l'applicazione multi-container definita nel file `docker-compose.yml`. L'applicazione è formata da backend (fastAPI) e frontend (React).

Funziona tutto correttamente o bisogna modificare qualcosa nel file di configurazione?



Esercizio 2 - compose_things_up

Si metta in esecuzione l'applicazione multi-container definita nel file `docker-compose.yml`. L'applicazione è formata da backend (fastAPI) e frontend (React).

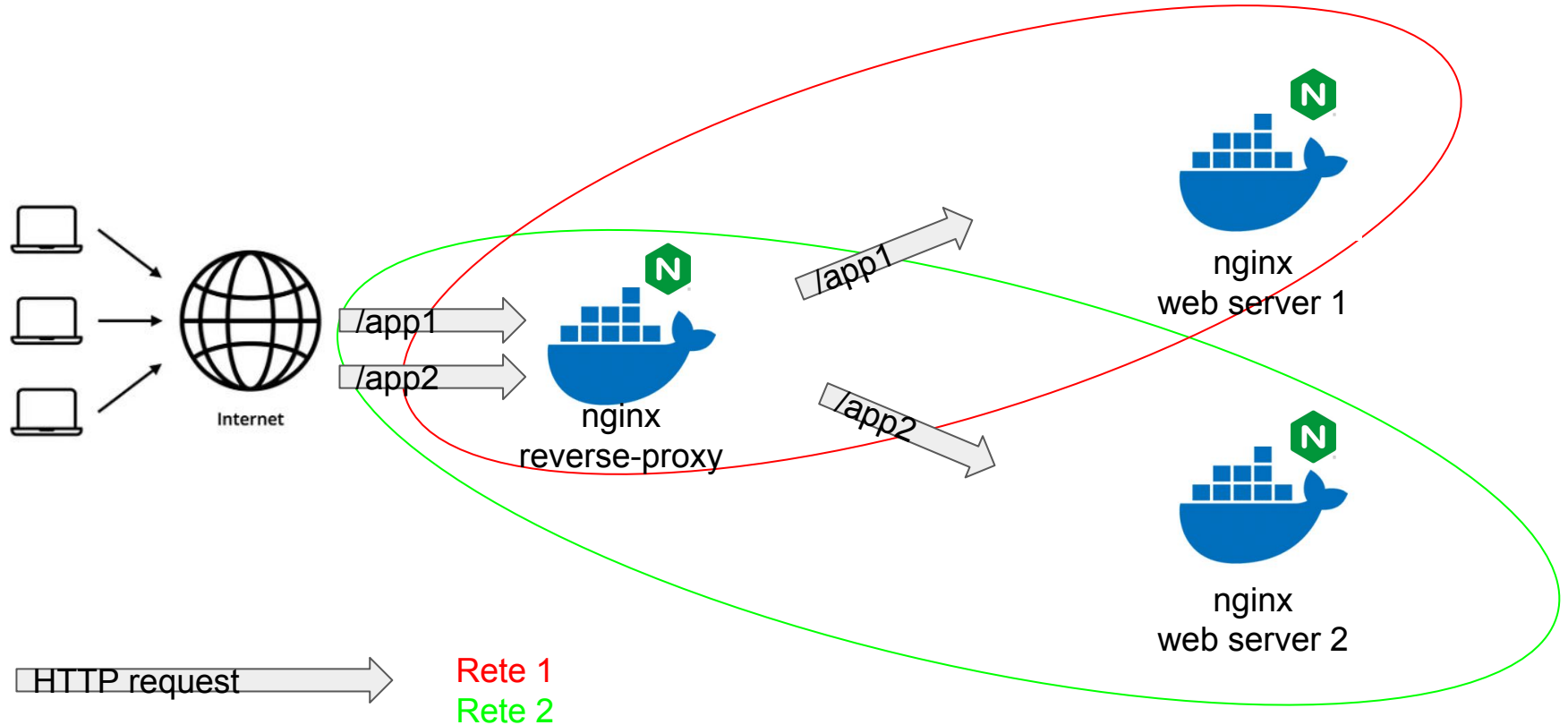
Funziona tutto o bisogna modificare qualcosa nel file di configurazione?

Aiuti:

- 1) Siamo sicuri che i container possano comunicare fra loro?



Esercizio 3 - BetterCallProxy



Esercizio 3 - BetterCallProxy

Si istanzino 2 web server collegati a 2 reti bridge custom in modo da essere tra loro isolati. Ogni server dovrà rispondere alle richieste HTTP inviando una pagina index.html personalizzata.

I 2 web server però non dovranno esporre porte sull'host ma saranno solo raggiungibili attraverso un reverse proxy che espone la porta 80 e che è collegata ad entrambe le reti dei web server.



Le richieste all'endpoint del proxy /app1 dovranno essere inoltrate al primo web server, e così per il secondo.

Sia per i web server che per il proxy si può usare l'immagine Docker Nginx.