

Scaletta Reti Avanzate

Fare i container e provarli nello stesso computer

1. **Ambiente in cui eseguire i file:** due Docker container di un'immagine ubuntu inizialmente identici. Nel primo mettiamo il file client.c e nell'altro server.c. Timestamp.c, Makefile e twamp.h in tutti e due i docker (lo faccio con il comando gitClone).
2. Facciamo in modo che i **due Docker** si trovino **nella stessa rete** così riescono a comunicare (se non comunicano modificare il file di configurazione del container).
3. Dovremmo poter eseguire file client e server **passando gli indirizzi IP l'uno dell'altro** e quindi dovrebbero testare la bandwidth da soli.

Proviamo a mettere i container in computer diversi

Test per vedere se va bene. Se il server è sempre online allora il Client può fare quel cazzo che gli va.

1. Nel container del server faccio partire il server con ./server
2. Nel container del client faccio partire il server con ./client
3. Devo dare passare “./client -s [indirizzo IP server]” al client per connetterli. Per averlo bisogna scrivere ifconfig sul terminale (vedo tutte le interfacce che il computer ha; ne avrò solo una e c'è scritto sotto anche l'indirizzo IP).
4. Faccio “make” e “make setcap” (bisogna farli solo la prima volta che creo docker! Questo perché fa i file compilati e poi rimangono lì per sempre. Ci passiamo poi noi tra i computer il container tramite Docker hub).

Creazione interfaccia HTTP

Bisogna usare express tramite Visual Studio Code.

2 interfacce 1 in cui passo gli indirizzi HTTP che servono e 1 in cui visualizzo i risultati. Le metto sul client.

Per aprire l'interfaccia vado sul browser e scrivo il URL che mi da docker.

```
expt(`./client -s ${req.ip} -i 10000`);
```

```
const shell = require('shelljs')  
shell.exec('./path_to_your_file')
```

Posso fare due richieste:

GET: Prendo una pagina WEB (facendo un get su HTTP)

POST: Mando i dati (indirizzi IP a docker in questo caso)

La parte di mostrare i risultati: Dobbiamo mettere anche nel server (mentre prima era solo sul client).

Grafico

Pensiamo alla fine e cerchiamo una libreria online da copiare.

Comandi usati per il primo punto

- **git clone url repo** (per clonare la repo di GitHub ovvero scaricarla)
- **cd twamp-protocol** (per andare dentro alla cartella)
- **./server**
- **apt install make** (installare il make)
- **apt install apt-utils** (libreria di make)
- **make setcup** (perché c'era scritto nella repository)
- **apt install ip**
- **docker inspect [container id]** (guarda l'indirizzo ip del container)
- **useradd server**
- **sudo -u server ./server** (creato il docker di ubuntu che ha solo le funzioni basilari per funzionare senza extra. ha solo un utente sud che è l'amministratore. La repo GitHub non lascia eseguire il server su account amministratore quindi abbiamo creato un nuovo account che si chiama server e abbiamo eseguito server sotto l'account server al posto dell'account amministratore)

- **sudo -u [client]** (il prossimo comando verrà eseguito con il client, ma io rimango sudo)
- **sudo -u client ./client -s 172.17.0.2** (per collegare il client al server e in automatico mi testa la bandwidth)