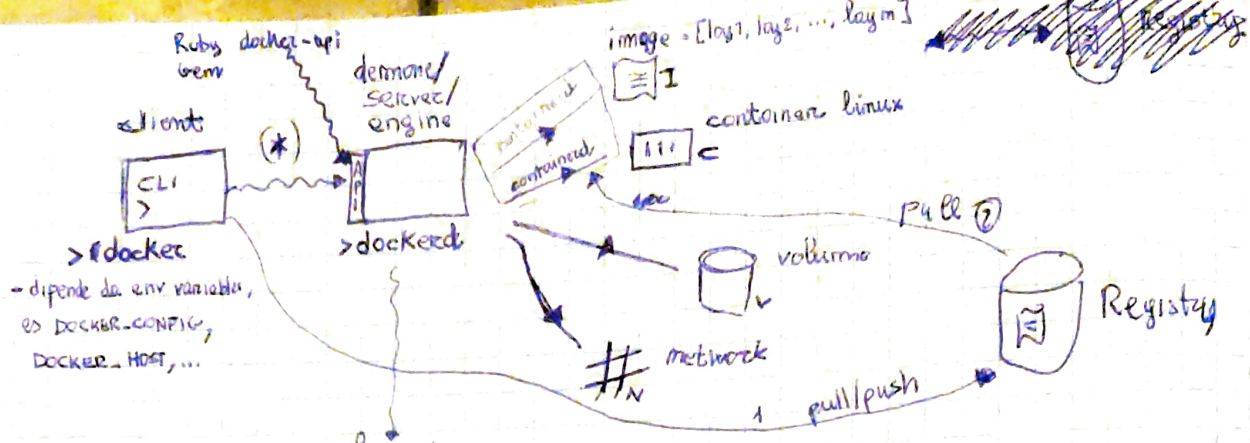


0) docker-help

1) Check CLI variables (docs Docker)

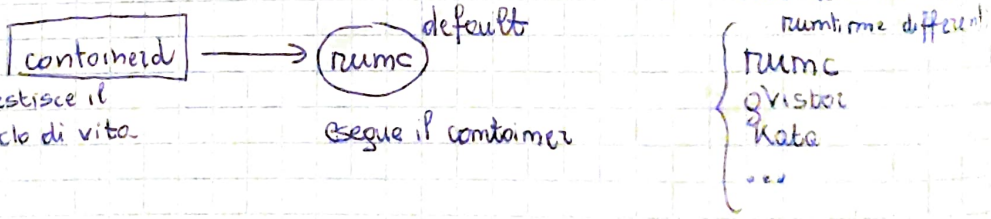
2) Avviare demone nei due modi : Linux, systemctl | docker dockerd → va in foreground e vedo i log

```
dockerd --help
dockerd --debug
```

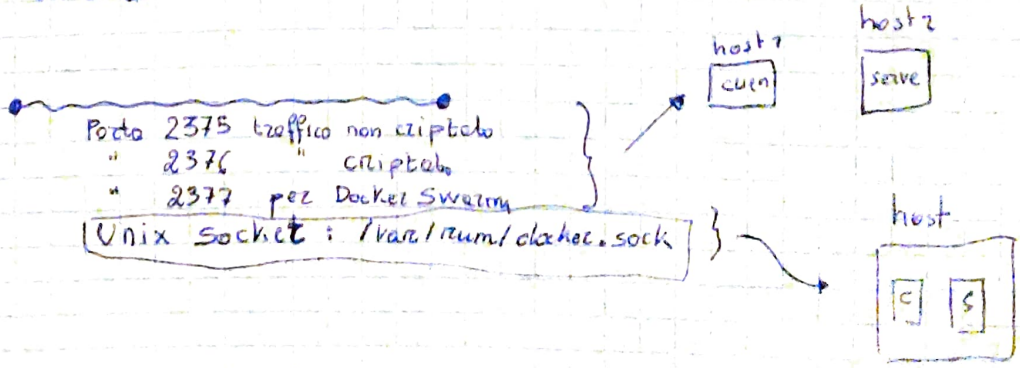
3) OSS

- dockerd demone che gestisce i container
- containerd è componente che si interfaccia col kernel per eseguire i container

il containerd poi ha un runtime



4) (*) come comunicano



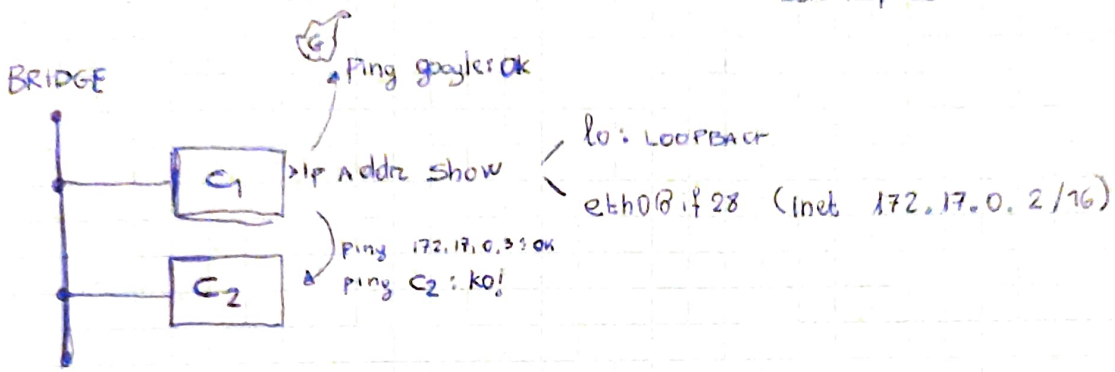
NETWORK

- Se non specifico nulla di default vanno sulla bridge

> docker network ls

- BRIDGE (default) (no per produzione)
- HOST connesso alla rete del docker host
- NONE container senza interfacce di rete (isolamento) solo loopback

ES1



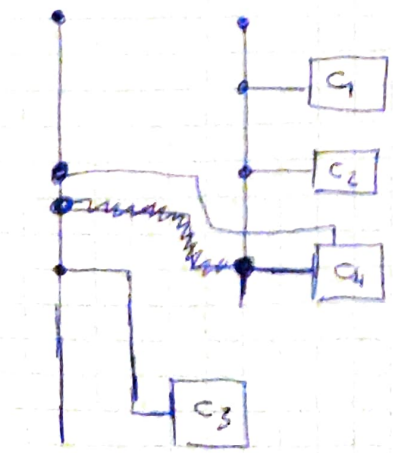
> docker network inspect bridge

SUBNET: 172.17.0.0/16
GATEWAY: 172.17.0.1
...

tra il Docker-Host e la bride network
Punto di accesso o porta verso altre reti

CONTAINER:
- C1
  IPV4: 172.17.0.2/16
- C2
  IPV4: 172.17.0.3/16

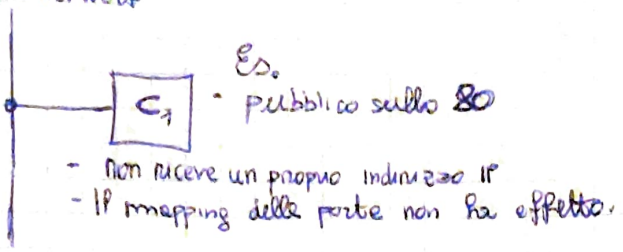
ES2



Su una rete custom come la alpine-net i container si possono parlare sia per IP che per NOME.
(automatic service discovery)
Anche se la risoluzione per nome avviene solo per i container custom non per quelli autogenerati

Es 3 host

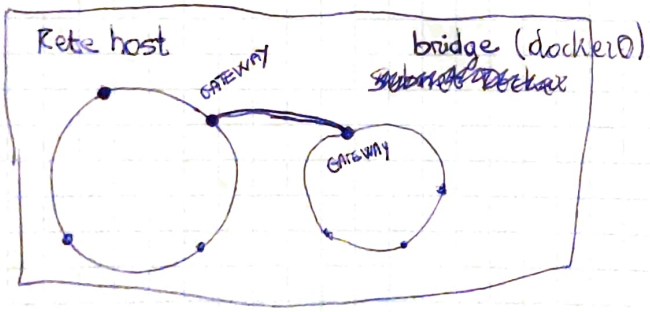
host network



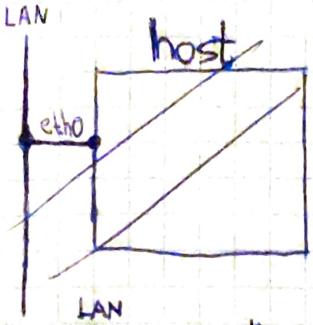
- Lo vedo direttamente sulla porta 80 del localhost
- Non richiede la traduzione degli indirizzi di rete (NAT) e non viene creato un "userland-proxy" per ogni porto.

Es 4 NONE

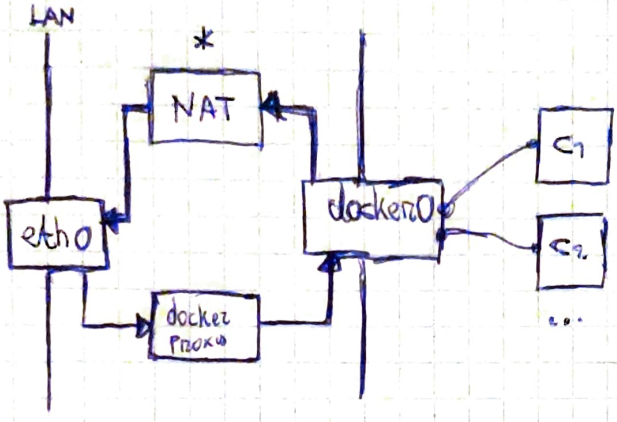
Isolazione completa, non c'è accesso alla rete.



docker0 è la rete di tipo bridge.



* tutti i container escono con lo stesso IP



OR

IMMagine

- Ci infilo il più possibile per rendermi leggerla esattamente per ciò che mi serve

MA con la stessa immagine devo poter runnare i container in DEV, LAB, PRO.
quindi le variabili e le url si possono da fuori

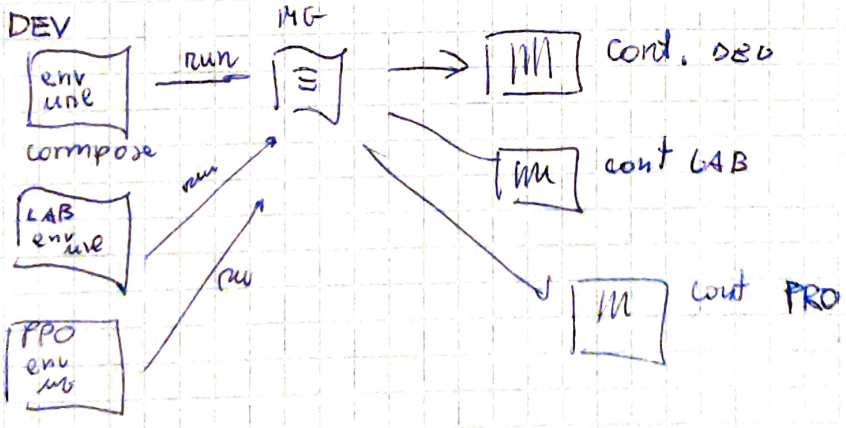


IMMAGINE: Abbastanza generica per essere usata in DEV/LAB/PRO

" specifica per fare tutto ciò che serve.