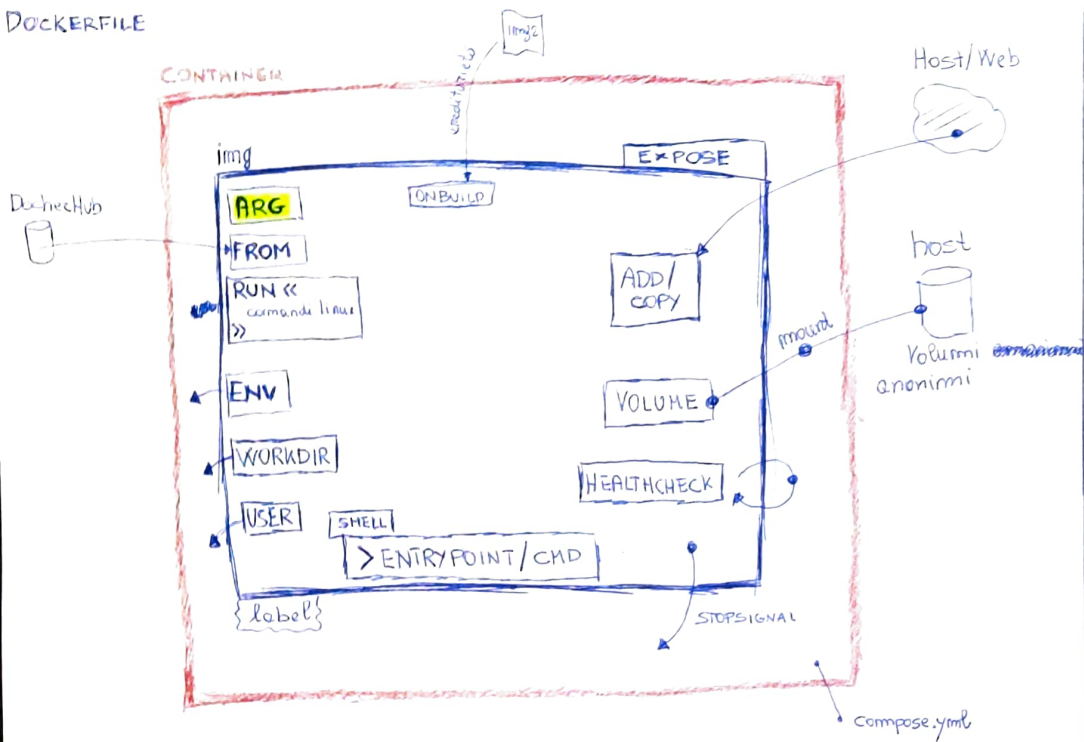



## DOCKERFILE



## RUN

`docker container run`  `img`

Quando parte il container posso sovrascrivere  
vere molte delle configurazioni del  
Dockerfile


`ENV`  `--env`

`WORKDIR`  `--workdir`

`USER`  `--user`

`SHELL`  `comando`

`ENTRYPOINT`  `--entrypoint`

`CMD`  `comando`


`HEALTHCHECK`  `--no-healthcheck`  
(disabilita)

`VOLUME`  `--volume`

`EXPOSE`  `--port`

• A livello di `compose.yml` posso poi  
aggiungere molte configurazioni al  
container

## BUILD

`docker image build`  `--build-arg`

Se `--build-arg` è l'unico modo per poter sovrascrivere  
durante la build le configurazioni varie  
presenti nel Dockerfile.

- Se una istruzione aggiunge contenuto, verrà creato un layer.
- Se aggiunge informazioni su come far partire la app verrà creato un meta-dato.

# DOCKERFILE

**ARG**: Variabile di ambiente che non persiste nel container.  
Ha effetto solo durante la build.

**ENV**: Variabile di ambiente che persiste nel container.  
Ha anche effetto durante la build.

**EXPOSE**: Non ha nessun effetto.  
Serve per documentazione del tipo "cosa dovrebbe esporre il container come porta"

**USER**: Utente per il runtime finale del container

UID[:GID]  
↑  
Username      groupname/root

**VOLUME**: mount point esterno

ENTRYPOINT	CMD	run
Ø	Ø	> .exe [p <sub>i</sub> ]    eseguibile + parametri opzionali
Ø	.exe [p <sub>i</sub> ]	> [.exe [p <sub>j</sub> ]]
.exe .exe	Ø [p <sub>i</sub> ]	> [p <sub>i</sub> ] > [p <sub>j</sub> ]