

Comandi Docker

Comandi Docker e Compose collaudati per l'operatività – cercabili e copiabili con un clic.
Con avvisi sui comandi che eliminano dati.

daniele.michelin //
cheat sheet

Immagini

Elenca le immagini locali

```
docker images
```

Scarica un'immagine con tag fisso

```
docker pull nginx:1.27
```

Costruisce un'immagine dal Dockerfile in questa cartella

```
docker build -t myapp:1.0 .
```

Rinomina un'immagine per un registry

```
docker tag myapp:1.0 registry.example.com/myapp:1.0
```

! Elimina un'immagine

```
docker rmi myapp:1.0
```

Avviare e gestire i container

Elenca i container in esecuzione

```
docker ps
```

Elenca tutti i container, anche quelli fermi

```
docker ps -a
```

Avvia un container in background con mappatura porte

```
docker run -d --name web -p 8080:80 nginx
```

Container usa e getta con shell interattiva

```
docker run --rm -it alpine sh
```

Riavvia il container dopo reboot o crash

```
docker run -d --restart unless-stopped nginx
```

Ferma un container in modo ordinato (SIGTERM, poi SIGKILL)

```
docker stop web
```

! Ferma ed elimina forzatamente un container

```
docker rm -f web
```

Log, exec e diagnostica

Ultime 100 righe di log, poi segui in tempo reale

```
docker logs -f --tail 100 web
```

Apre una shell in un container in esecuzione

```
docker exec -it web sh
```

Esegue un singolo comando nel container

```
docker exec web nginx -t
```

Uso di CPU e RAM di tutti i container (istantanea)

```
docker stats --no-stream
```

Volumi e reti

Monta un volume con nome in un container

```
docker run -v data:/var/lib/mysql mariadb
```

Esegue il backup di un volume in un archivio tar

```
docker run --rm -v data:/d -v $(pwd):/b alpine tar czf /b/data.tgz -C /d .
```

! Elimina un volume – i dati vanno persi

```
docker volume rm data
```

Elenca le reti

```
docker network ls
```

Crea una rete bridge dedicata (DNS tramite nome container)

```
docker network create appnet
```

Docker Compose

Avvia lo stack in background

```
docker compose up -d
```

Ricostruisce le immagini e avvia lo stack

```
docker compose up -d --build
```

Segui i log di un servizio in tempo reale

```
docker compose logs -f web
```

Scarica nuove immagini, ricrea i servizi modificati

```
docker compose pull && docker compose up -d
```

Shell nel container di un servizio

```
docker compose exec web sh
```

Ferma lo stack e rimuove container e rete

```
docker compose down
```

! Come down, ma elimina anche i volumi

```
docker compose down -v
```

Pulizia con prune

Spazio usato da immagini, container, volumi

```
docker system df
```

! Elimina le immagini senza tag (dangling)

```
docker image prune
```

! Elimina tutti i volumi inutilizzati – possibile perdita di dati

```
docker volume prune
```

! Elimina container fermi, reti inutilizzate e immagini dangling

```
docker system prune
```

! Radicale: tutto ciò che è inutilizzato, immagini e volumi inclusi

```
docker system prune -a --volumes
```

Limiti di risorse

Limita la memoria a 512 MB

```
docker run -d --memory 512m nginx
```

Limita a 1,5 core CPU

```
docker run -d --cpus 1.5 nginx
```

Modifica i limiti di un container in esecuzione

```
docker update --memory 1g --memory-swap 1g web
```

Imposta la rotazione dei log per un container

```
docker run -d --log-opt max-size=10m --log-opt max-file=3 nginx
```

Trova gli OOM kill dell'ultima ora

```
docker events --since 1h --filter event=oom
```