

Comandos Docker — Cheat Sheet Essencial (2025)

Resumo rápido dos comandos mais úteis do dia a dia. Exemplos prontos para copiar e colar.

Básico

```
docker version
docker info
docker help

# Ver ajuda de um comando específico
docker <comando> --help
```

Imagens

```
# Buscar imagem
docker pull nginx:latest

# Listar imagens locais
docker images

# Remover imagem
docker rmi nginx:latest

# Construir imagem a partir de Dockerfile
docker build -t meuapp:1.0 .

# Marcar (tag) uma imagem
docker tag meuapp:1.0 usuario/meuapp:1.0

# Enviar/baixar do Docker Hub
docker push usuario/meuapp:1.0
docker pull usuario/meuapp:1.0
```

Containers — ciclo de vida

```
# Criar e iniciar (interativo, auto-remove, porta 8080→80)
docker run --rm -it -p 8080:80 --name web nginx:latest

# Executar em segundo plano (detached)
docker run -d --name db -e POSTGRES_PASSWORD=senha -p 5432:5432 postgres:16

# Listar containers
docker ps          # ativos
docker ps -a       # todos

# Start/Stop/Restart
docker start web
docker stop web
docker restart web

# Remover container
docker rm web
```

Execução dentro do container, logs e inspeção

```
# Entrar em um shell dentro do container
docker exec -it web sh          # ou bash conforme a imagem

# Executar um comando pontual
docker exec web ls /usr/share/nginx/html

# Logs (seguir em tempo real)
docker logs web
docker logs -f --tail=100 web
```

```
# Inspecionar (JSON com detalhes)
docker inspect web

# Ver processos, uso e eventos
docker top web
docker stats          # monitoramento de uso
docker events         # eventos em tempo real
```

Volumes e Bind Mounts (dados persistentes)

```
# Volume gerenciado pelo Docker
docker volume create dados
docker volume ls
docker volume rm dados

# Usar volume em um container
docker run -d --name db -e POSTGRES_PASSWORD=senha \
  -v dados:/var/lib/postgresql/data postgres:16

# Bind mount (pasta do host → container)
# Linux/macOS
docker run --rm -it -v $(pwd)/site:/usr/share/nginx/html -p 8080:80 nginx

# Windows (PowerShell)
docker run --rm -it -v ${PWD}/site:/usr/share/nginx/html -p 8080:80 nginx
```

Redes

```
# Listar/criar redes
docker network ls
docker network create minha-rede

# Conectar container a uma rede
docker network connect minha-rede web

# Iniciar já conectando a uma rede
docker run -d --name api --network minha-rede meuapp:1.0
```

Copiar arquivos entre host e container

```
# Host → container
docker cp ./config.yaml web:/etc/nginx/conf.d/config.yaml

# Container → host
docker cp web:/var/log/nginx/access.log ./access.log
```

Limpeza de recursos

```
# Espaço usado
docker system df

# Remover recursos não utilizados (atenção!)
docker system prune          # containers/imagens (dangling)/redes não utilizadas
docker system prune -a      # inclui imagens não utilizadas por nenhum container
docker volume prune
```

Docker Compose (v2: 'docker compose')

```
# Subir serviços em segundo plano
docker compose up -d

# Ver logs
docker compose logs -f --tail=100

# Listar serviços/containers
docker compose ps
```

```
# Executar comando em um serviço
docker compose exec api sh

# Recriar após alterações
docker compose up -d --build

# Derrubar serviços e redes
docker compose down
docker compose down -v      # remove volumes (cautela)
```

Exemplos práticos rápidos

```
# 1) Servir uma pasta estática local na porta 8080
docker run --rm -it -p 8080:80 -v $(pwd)/public:/usr/share/nginx/html nginx

# 2) Banco PostgreSQL com volume persistente
docker volume create pgdata
docker run -d --name pg -e POSTGRES_PASSWORD=senha -v pgdata:/var/lib/postgresql/data -p 5432:5432

# 3) Build e execução de uma app customizada
docker build -t usuario/minha-api:1.0 .
docker run -d --name minha-api -p 3000:3000 usuario/minha-api:1.0
```

Dicas

- Prefira tags de imagem específicas (ex.: postgres:16) para ambientes reprodutíveis.
- Use variáveis de ambiente via arquivos .env (Compose lê automaticamente).
- No Windows, paths em bind mounts funcionam melhor no PowerShell com \${PWD}.
- Verifique permissões ao usar bind mounts (UID/GID dentro do container).
- Para produção, minimize a imagem (ex.: multi-stage build e Alpine/Distroless quando possível).