

Docker

The commands you reach for daily with the Docker CLI. `docker compose` (v2, a subcommand) has replaced the old `docker-compose` binary — both are shown where they differ.

Images

Command	Action
<code>docker build -t app:1.0 .</code>	build an image from the Dockerfile here
<code>docker images</code>	list local images
<code>docker pull nginx:alpine</code>	download an image
<code>docker tag app:1.0 app:latest</code>	add another tag
<code>docker push registry/app:1.0</code>	upload to a registry
<code>docker rmi app:1.0</code>	remove an image

Containers

Command	Action
<code>docker run nginx</code>	run a container in the foreground
<code>docker ps</code>	list running containers
<code>docker ps -a</code>	list all containers, including stopped
<code>docker stop <id></code>	graceful stop (SIGTERM)
<code>docker start / restart <id></code>	start / restart a container
<code>docker rm <id></code>	remove a stopped container
<code>docker exec -it <id> sh</code>	open a shell inside a running container
<code>docker logs -f <id></code>	follow a container's logs

Common run flags

Flag	Effect
-d	detached — run in the background
-p 8080:80	map host port 8080 to container port 80
-v ./data:/app/data	bind-mount a host folder
-e KEY=value	set an environment variable
--name web	give the container a stable name
--rm	auto-remove the container when it exits
--network mynet	attach to a user-defined network

A typical one-liner

```
docker run -d --name web --rm \  
  -p 8080:80 -v ./html:/usr/share/nginx/html \  
  nginx:alpine
```

Compose

Command	Action
docker compose up -d	start the stack in the background
docker compose down	stop and remove the stack
docker compose ps	list the stack's containers
docker compose logs -f	follow logs for all services
docker compose build	rebuild service images
docker compose restart <svc>	restart one service

Clean Up

Command	Removes
<code>docker system df</code>	show what Docker is using (safe)
<code>docker container prune</code>	all stopped containers
<code>docker image prune</code>	dangling (untagged) images
<code>docker image prune -a</code>	all images not used by a container
<code>docker volume prune</code>	unused volumes
<code>docker system prune -a</code>	containers, networks, and unused images

prune deletes without asking twice. `docker volume prune` can wipe database data living in an unused volume, and `system prune -a` removes every image no running container references. Check `docker system df` first.