Docker CLI Cheatsheet

A comprehensive cheat sheet for Docker CLI, covering essential commands for managing images, containers, volumes, and networks. Includes examples for quick reference.

Basic Commands

Image Management

`docker build [options] .`	Builds an image from a Dockerfile in the current directory. **Options:** `-t <image_name>`: Tags the image with a name. `--build-arg <var>=<value>`: Sets build-time variables. **Example:** `docker build -t myapp:latest .`
`docker pull <image_name>`	Pulls an image from Docker Hub or a registry. **Example:** `docker pull ubuntu:latest`
`docker images [options]`	Lists available images. **Options:** `-a`: Shows all images, including intermediate image layers. **Example:** `docker images`
`docker rmi <image_id>`	Removes an image. **Example:** `docker rmi b750fe78269d`
`docker image prune [options]`	Removes unused images. **Options:** `-a`: Remove all unused images, not just dangling ones. **Example:** `docker image prune -a`

Container Management

`docker run [options] <image>`	Creates and starts a container from an image. **Options:** `-d`: Detached mode (run in background). `-i`: Interactive mode. `-t`: Allocate a pseudo-TTY. `--name <name>`: Assign a name to the container. `-p <host_port>:<container_port>`: Port mapping. `-v <host_path>:<container_path>`: Volume mounting. `-e <VAR>=<value>`: Set environment variables. **Example:** `docker run -d -p 80:80 nginx`
`docker create [options] <image>`	Creates a container but does not start it. **Example:** `docker create --name my_container ubuntu`
`docker start <container_id>`	Starts a stopped container. **Example:** `docker start my_container`
`docker stop <container_id>`	Stops a running container. **Example:** `docker stop my_container`
`docker restart <container_id>`	Restarts a container. **Example:** `docker restart my_container`
`docker rm [options] <container_id>`	Removes a stopped container. **Options:** `-f`: Force removal of a running container. **Example:** `docker rm my_container`

Inspecting and Interacting with Containers

Container Inspection

`docker ps [options]`	Lists running containers. **Options:** `-a` : Shows all containers, including stopped ones. `-q` : Only display numeric IDs. **Example:** `docker ps -a`
`docker inspect <container_id>`	Displays detailed information about a container. **Example:** `docker inspect my_container`
`docker logs [options] <container_id>`	Fetches the logs of a container. **Options:** `-f` : Follow log output. `--tail <n>` : Output the last n number of lines. **Example:** `docker logs -f my_container`
`docker top <container_id>`	Displays the processes running inside a container. **Example:** `docker top my_container`
`docker stats [options] <container_id>`	Displays resource usage statistics for a container. **Example:** `docker stats my_container`

Docker Volumes and Networking

Volume Management

`docker volume create [options] <volume_name>`	Creates a new volume. **Options:** `--driver <driver_name>` : Specify volume driver name (e.g., local, nfs). **Example:** `docker volume create my_volume`
`docker volume ls`	Lists all existing volumes. **Example:** `docker volume ls`
`docker volume inspect <volume_name>`	Displays detailed information about a volume. **Example:** `docker volume inspect my_volume`
`docker volume rm <volume_name>`	Removes a volume. **Example:** `docker volume rm my_volume`
`docker volume prune`	Removes all unused local volumes. **Example:** `docker volume prune`

Interactive Container Operations

`docker exec [options] <container_id> <command>`	Executes a command inside a running container. **Options:** `-i` : Interactive mode. `-t` : Allocate a pseudo-TTY. `-d` : Detached mode. `-u <user>` : Username or UID (format: `<name|uid>[:<group|gid>]`). **Example:** `docker exec -it my_container bash`
`docker attach <container_id>`	Attaches your terminal's standard input, output, and error streams to a running container. **Example:** `docker attach my_container`
`docker kill <container_id>`	Kills a running container by sending a SIGKILL signal. **Example:** `docker kill my_container`
`docker pause <container_id>`	Pauses all processes within a container. **Example:** `docker pause my_container`
`docker unpause <container_id>`	Unpauses all processes within a paused container. **Example:** `docker unpause my_container`

Network Management

`docker network create [options] <network_name>`	Creates a new network. **Options:** `--driver <driver_name>`: Specify network driver (e.g., bridge, overlay). `--subnet <subnet>`: Assign subnet address. **Example:** `docker network create --driver bridge my_network`
`docker network ls`	Lists all existing networks. **Example:** `docker network ls`
`docker network inspect <network_name>`	Displays detailed information about a network. **Example:** `docker network inspect my_network`
`docker network connect <network_name> <container_id>`	Connects a container to a network. **Example:** `docker network connect my_network my_container`
`docker network disconnect <network_name> <container_id>`	Disconnects a container from a network. **Example:** `docker network disconnect my_network my_container`
`docker network rm <network_name>`	Removes a network. **Example:** `docker network rm my_network`
`docker network prune`	Removes all unused networks. **Example:** `docker network prune`

Advanced Docker Commands

Docker System Commands

`docker system df`	Shows Docker disk usage. **Example:** `docker system df`
`docker system events`	Gets real-time events from the Docker server. **Example:** `docker system events`
`docker system info`	Displays system-wide information about Docker. **Example:** `docker system info`
`docker system prune [options]`	Cleans up unused Docker resources. **Options:** `-a`: Remove all unused images and containers. `--volumes`: Prune volumes as well. **Example:** `docker system prune -a`

Docker Compose (Orchestration)

`docker-compose up [options]`	Builds, (re)creates, starts, and attaches to containers defined in a `docker-compose.yml` file. **Options:** `-d` : Detached mode. `--build` : Force rebuild images. `--scale <service>=<num>` : Scale a service to a specified number of instances. **Example:** `docker-compose up -d`
`docker-compose down [options]`	Stops and removes containers, networks, volumes, and images created by `docker-compose up`. **Options:** `--rmi all` : Remove all images used by the service. **Example:** `docker-compose down`
`docker-compose ps`	Lists the containers created by `docker-compose`. **Example:** `docker-compose ps`
`docker-compose logs [options] <service>`	View output from services. **Options:** `-f` : Follow log output. **Example:** `docker-compose logs -f web`
`docker-compose exec <service> <command>`	Execute arbitrary commands within a service's container. **Example:** `docker-compose exec web bash`