Docker Cheatsheet

A concise reference for Docker commands and concepts, ideal for quick lookups and efficient Docker workflows.

Basic Commands

Image Management

`docker pull <image>`	Download an image from Docker Hub.
`docker images`	List available images locally.
`docker rmi <image>`	Remove an image.
`docker build -t <image> .`	Build an image from a Dockerfile in the current directory.
`docker tag <source_image> <target_image>`	Tag an image.
`docker push <image>`	Push image to Docker Hub or registry.

Container Management

`docker run <image>`	Create and start a container.
`docker ps`	List running containers.
`docker ps -a`	List all containers (running and stopped).
`docker stop <container>`	Stop a running container.
`docker start <container>`	Start a stopped container.
`docker restart <container>`	Restart a container.
`docker rm <container>`	Remove a stopped container.
`docker exec -it <container> <command>`	Execute a command inside a container.

Networking

`docker network create <network>`	Create a new network.
`docker network ls`	List available networks.
`docker network connect <network> <container>`	Connect a container to a network.
`docker port <container>`	List port mappings for a container.

Dockerfile Instructions

Essential Instructions

`FROM <image>`	Specifies the base image for the Dockerfile.
`RUN <command>`	Executes commands during the image build process.
`CMD <command>`	Specifies the default command to run when the container starts.
`EXPOSE <port>`	Declares the port the container listens on.
`ENV <key> <value>`	Sets environment variables.
`COPY <src> <dest>`	Copies files/directories from the host to the container.
`ADD <src> <dest>`	Similar to COPY, but can also extract archives and fetch URLs.
`WORKDIR <path>`	Sets the working directory for subsequent instructions.

User and Volume Management

`USER <user>`	Sets the user for subsequent RUN, CMD, and ENTRYPOINT instructions.
`VOLUME <path>`	Creates a mount point for persistent storage.
`STOPSIGNAL <signal>`	Signal to be used to stop the container
`ARG <name>[=<default value>]`	Defines a build argument.

Docker Compose

Compose File Structure

A `docker-compose.yml` file defines services, networks, and volumes for a multi-container Docker application.

Key elements include:

- `version`: Specifies the Compose file version.
- `services`: Defines individual containers.
- `networks`: Defines networks used by the services.
- `volumes`: Defines persistent data volumes.

Common Compose Commands

`docker-compose up`	Builds, (re)creates, starts, and attaches to containers defined in a `docker-compose.yml` file.
`docker-compose up -d`	Runs the containers in detached mode (background).
`docker-compose down`	Stops and removes containers, networks, and volumes defined in the Compose file.
`docker-compose ps`	Lists the containers defined in the Compose file.
`docker-compose logs`	View the logs of the containers.
`docker-compose stop`	Stop services
`docker-compose start`	Start services
`docker-compose restart`	Restart services

Service Configuration

`image`	Specifies the image to use for the service.
`build`	Specifies the path to the Dockerfile to build the image.
`ports`	Exposes ports from the container to the host.
`volumes`	Mounts volumes to the container.
`environment`	Sets environment variables for the service.
`depends_on`	Defines service dependencies.
`networks`	Attaches the service to a network.

Docker Swarm

Swarm Initialization and Management

`docker swarm init`	Initializes a new Swarm cluster. Run this on the manager node.
`docker swarm join`	Joins a node to an existing Swarm cluster. Use the token provided by `docker swarm init`.
`docker swarm leave`	Leaves the Swarm cluster. Use `--force` if the node is unreachable.
`docker node ls`	Lists the nodes in the Swarm cluster.
`docker node inspect <node_id>`	Inspect a specific node.
`docker node update <node_id>`	Update node's role and availability.

Service Deployment and Scaling

`docker service create`	Creates a new service in the Swarm cluster.
`docker service ls`	Lists the services running in the Swarm cluster.
`docker service update`	Updates an existing service (e.g., scale, update image).
`docker service scale <service_id>=<number_of_replicas>`	Scales a service to the specified number of replicas.
`docker service inspect <service_id>`	Inspect a specific service.
`docker service logs <service_id>`	View logs for a specific service.
`docker service rm <service_id>`	Removes a service from the Swarm cluster.

Stack Deployment

`docker stack deploy -c <compose_file>.yml <stack_name>`	Deploy a stack based on a compose file.
`docker stack ls`	List deployed stacks.
`docker stack rm <stack_name>`	Remove a deployed stack.
`docker stack ps <stack_name>`	List the tasks in the stack.