Docker Command Cheat Sheet and Guide

# 1. Docker Image Building

Build an image from a Dockerfile in the current directory (.):

Command:

docker build -t <image-name>:<tag> .

Usage:

- Builds an image from a Dockerfile in the current directory.  
- <image-name> is the custom name for the image, and <tag> is an optional version label, e.g., myapp:1.0.

Example:

docker build -t spring-boot-app:latest .

# 2. Docker Run and Manage Containers

Run a container from an image:

Command:

docker run -d -p <host-port>:<container-port> --name <container-name> <image-name>

Usage:

- -d: Runs the container in detached mode (background).  
- -p: Maps the port on the host to the port in the container, e.g., 8080:8080.  
- --name: Provides a custom name for easy reference.

Example:

docker run -d -p 8080:8080 --name my-spring-app spring-boot-app:latest

# 3. Listing, Stopping, and Removing Containers

List all running or stopped containers:

Commands:

docker ps # Shows running containers  
docker ps -a # Shows all containers (including stopped ones)

Stop a container:

docker stop <container-name>

Remove a container:

docker rm <container-name>

Example:

docker stop my-spring-app  
docker rm my-spring-app

# 4. Docker Volumes

Creating and using Docker volumes for persistent data:

Create a volume:

docker volume create <volume-name>

Run a container with an attached volume:

docker run -d -p <host-port>:<container-port> --name <container-name> -v <volume-name>:<container-path> <image-name>

Usage:

- Volumes allow data persistence by storing data outside the container.  
- <container-path> is where the volume is mounted in the container.

Example:

docker volume create my-volume  
docker run -d -p 8080:8080 --name my-spring-app -v my-volume:/data spring-boot-app:latest