Basic Docker Commands by Hungry Coders

Essential Commands for Java Spring Boot Developers

Docker is a popular containerization platform that simplifies application deployment and scaling. This cheat sheet provides Java Spring Boot developers with essential Docker commands to build, run, and manage containers efficiently.

1. General Docker Information

- Check Docker Version: docker --version

- Display Docker System Info: docker info

- List All Docker Commands: docker --help

2. Building and Managing Images

- Build an Image: docker build -t <image-name>:<tag> .

- (Example: docker build -t myapp:1.0 .)

- List All Images: docker images

- Remove an Image: docker rmi <image-id>

- (Force remove: docker rmi -f <image-id>)

- Tag an Image: docker tag <image-id> <repository>:<tag>

- (Example: docker tag myapp:1.0 myrepo/myapp:latest)

- Push an Image to Docker Hub: docker push <repository>:<tag>

- (Example: docker push myrepo/myapp:latest)

3. Running Containers

- Run a Container: docker run -p <host-port>:<container-port> <image-name>

- (Example: docker run -p 8080:8080 myapp:1.0)

- Run a Container in Detached Mode: docker run -d -p <host-port>:<container-port> <image-name>

- Run a Container with a Volume: docker run -v <host-path>:<container-path> <image-name>

- (Example: docker run -v $(pwd):/app myapp:1.0)

- Run a Container with Environment Variables: docker run -e <key>=<value> <image-name>

- (Example: docker run -e SPRING_PROFILES_ACTIVE=dev myapp:1.0)

- Restart a Stopped Container: docker start <container-id>

- Stop a Running Container: docker stop <container-id>

4. Debugging Containers

- View Logs from a Container: docker logs <container-id>

- (Follow logs in real-time: docker logs -f <container-id>)

- Access a Running Container (Interactive Mode): docker exec -it <container-id> /bin/bash

- Inspect Container Details: docker inspect <container-id>

- Monitor Resource Usage (Stats): docker stats <container-id>

5. Managing Containers

- List All Running Containers: docker ps

- List All Containers (Running and Stopped): docker ps -a

- Stop All Running Containers: docker stop $(docker ps -q)

- Remove a Stopped Container: docker rm <container-id>

- (Remove all stopped containers: docker container prune)

6. Networking

- List Docker Networks: docker network ls

- Create a Network: docker network create <network-name>

- Run a Container in a Specific Network: docker run --network <network-name> <image-name>

- Inspect a Network: docker network inspect <network-name>

7. Docker Compose

- Start Services Defined in docker-compose.yml: docker-compose up

- (Run in detached mode: docker-compose up -d)

- Stop Services: docker-compose down

- Rebuild and Restart Services: docker-compose up --build

- View Logs of All Services: docker-compose logs -f

8. Cleaning Up

- Remove Dangling Images: docker image prune

- Remove Unused Containers, Networks, and Images: docker system prune