Docker Hello Example – Complete Notes (with save & load)

This document explains the Docker workflow using a simple C program example. It includes Dockerfile, image building, container execution, lifecycle behavior, and image sharing using docker save and docker load.

1. Dockerfile Explanation

- FROM ubuntu:22.04 – Uses Ubuntu 22.04 as the base image.
- RUN apt update && apt install -y gcc nano – Installs GCC compiler and Nano editor.
- COPY hello.c /hello.c – Copies the C source file into the image.
- RUN gcc hello.c -o hello – Compiles the C program inside the image.
- CMD ["/hello"] – Default command executed when the container starts.

2. Building the Docker Image

Build the image using the Dockerfile in the current directory:

```
docker build -t hello-22.04:v1.0 .
```

This creates a Docker image named hello-22.04 with tag v1.0. The image contains Ubuntu 22.04, GCC, Nano, and the compiled hello binary.

3. Running the Container (Interactive)

The following command runs a container interactively and overrides the default CMD:

```
docker run -it --name hello22 hello-22.04:v1.0 bash
```

Here, bash becomes the main process instead of /hello. When bash exits, the container stops because its main process has ended.

4. Restarting the Container

```
docker start -ai hello22
```

Docker restarts the container using the same main process (bash). Since bash exits immediately without a proper interactive terminal, the container stops again.

5. Sharing Docker Images (docker save & docker load)

Docker allows you to export images as .tar files so they can be shared with others without using Docker Hub or the internet.

5.1 Saving a Docker Image

Use docker save to export an image into a .tar file:

```
docker save -o hello-22.04-v1.0.tar hello-22.04:v1.0
```

This creates a file named hello-22.04-v1.0.tar which contains the full image. You can copy this file to another system using USB, SCP, or any file transfer method.

5.2 Loading a Docker Image

On another system, load the image using:

```
docker load -i hello-22.04-v1.0.tar
```

After loading, the image will appear in docker images and can be used to run containers immediately.

6. Key Docker Concepts Summary

- Dockerfile defines how an image is built.
- Docker image is a read-only template.
- Docker container is a running instance of an image.
- Container lifetime depends on its main process.
- docker run creates a new container.
- docker start restarts an existing container.
- docker save exports an image to a .tar file.
- docker load imports an image from a .tar file.

Conclusion: This example demonstrates the full Docker workflow from image creation to sharing images with others, which is commonly used in embedded systems and offline environments.