Docker images downloaded from a public registry (like Docker Hub or another container registry) are typically immutable, meaning you cannot directly modify or edit them once they are downloaded. These images are designed to be consistent and unchangeable to ensure that the software running inside them behaves predictably.

However, you have several options if you need to customize a Docker image or create a new image based on an existing one:

1. \*\*Dockerfile and Build\*\*:

- Create a `Dockerfile` that starts with the base image you want to modify.

- Add instructions in the `Dockerfile` to customize the image by installing software, configuring settings, or making other changes.

- Build a new Docker image using the `docker build` command, which will create a new image with your modifications.

Example `Dockerfile`:

```Dockerfile

# Use the base image you want to customize

FROM ubuntu:20.04

# Install packages or make other customizations

RUN apt-get update && apt-get install -y some-package

```

2. \*\*Use a Multi-Stage Build\*\*:

- If you want to create a smaller final image, you can use multi-stage builds. These allow you to use one image for building and another for the final runtime image.

Example Multi-Stage Dockerfile:

```Dockerfile

# Stage 1: Build environment

FROM node:14 as build

WORKDIR /app

COPY package\*.json ./

RUN npm install

COPY . .

RUN npm run build

# Stage 2: Runtime environment

FROM nginx:alpine

COPY --from=build /app/build /usr/share/nginx/html

```

3. \*\*Use Docker Compose\*\*:

- Docker Compose allows you to define multi-container applications and specify how they should interact. You can customize containers by providing a `Dockerfile` for your custom image.

4. \*\*Create a New Image Tag\*\*:

- You can pull an existing image, make modifications in a new container based on that image, and then commit those changes to create a new image. However, this is not a recommended practice as it can result in non-reproducible and less maintainable images.

Example:

```bash

docker run -it --name my\_temp\_container original\_image bash

# Make your modifications inside the container

docker commit my\_temp\_container my\_custom\_image

```

Remember that best practices generally involve creating custom Docker images through Dockerfiles and build processes to ensure consistency, maintainability, and traceability. Directly editing images is discouraged in most cases.