

# Docker

---

example: self-host a website (dataset)

Issue is: to deliver such thing, it's hard to find a uniform way for different OS.

## Docker Image

A Docker image is a lightweight, standalone, executable package that includes everything needed to run a piece of software, including the code, a runtime, libraries, environment variables, and configuration files. Images are the building part of Docker and are used to create Docker containers. They act as a blueprint or a template. Images are immutable, meaning once they are created, they do not change.

## Docker Container

A Docker container is a runnable instance of a Docker image. You can think of a container as the execution of an image, the runtime environment. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

### Key Differences:

- **Images are static.** They are like a class in Object Oriented Programming (OOP), defining what the application and its dependencies need.
- **Containers are dynamic.** They are like an instance of a class in OOP, running and doing the actual work.

### Workflow:

1. **Build:** Create a Docker image based on a Dockerfile. This file contains step-by-step commands on how the image should be built.
2. **Ship:** Once the image is created, you can push it to a Docker registry (like Docker Hub) where it can be shared and accessed by others.
3. **Run:** From the image, Docker can produce one or more containers that can run on any environment supporting Docker, ensuring consistency across different environments.

example

```
# Use an official Python runtime as a parent image
FROM python:3.8-slim

# Set the working directory to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Install any needed packages specified in requirements.txt
RUN pip install --no-cache-dir -r requirements.txt
```

```
# Make port 5000 available to the world outside this container
```

```
EXPOSE 5000
```

```
# Define environment variable
```

```
ENV NAME world
```

```
# Run app.py when the container launches
```

```
CMD ["python", "app.py"]
```