container image

A **container image** is a lightweight, stand-alone, and executable software package that includes everything needed to run a piece of software in a container.

This includes the application code, libraries, dependencies, system tools, settings, and runtime. Container images serve as the blueprint for creating containers, which are instances of the image running on a container runtime such as Docker, containerd, or others.

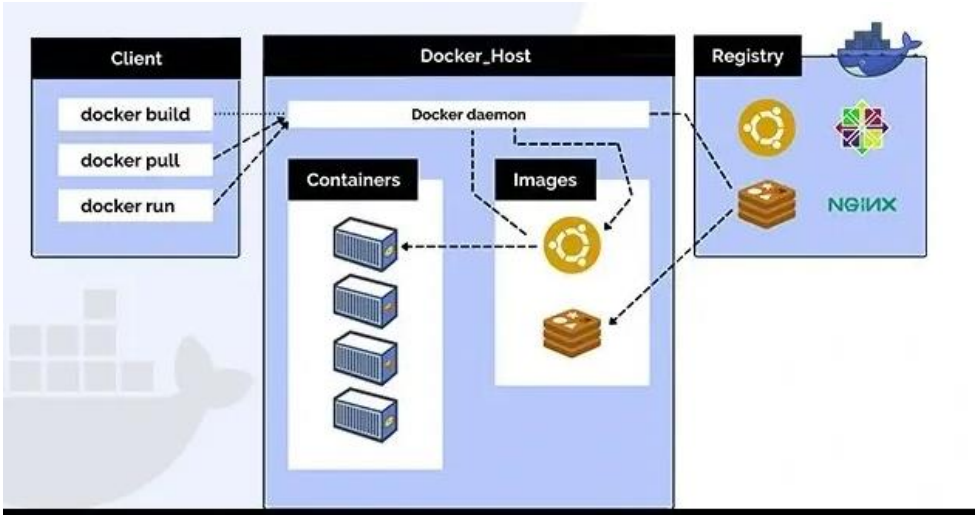
Key Concepts Related to Container Images:

1. **Container Image vs. Container**:
    - **Container Image**: A read-only template that includes the application and everything it needs to run.
    - **Container**: An instance of a container image that runs in an isolated environment with its own filesystem, networking, and process space.
2. **Layers in a Container Image**: Container images are built in layers, each layer representing a set of changes (such as installing a package, adding files, or modifying configurations). When a container image is updated or built, only the changes (new layers) are added, which makes them efficient and cacheable.

   For example:
    - Base Image Layer (e.g., an OS like Ubuntu or Alpine)
    - Application Layer (e.g., the web application)
    - Dependency Layer (e.g., required libraries or binaries)
3. **Image Manifest**: Every container image includes an **image manifest**, which describes the image's layers, the default configuration (such as entry points, environment variables), and other metadata about the image.

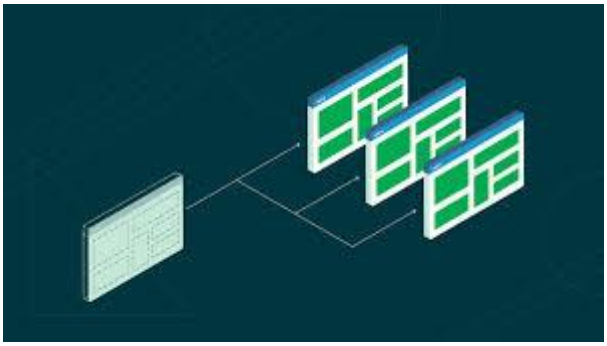
Components of a Container Image:

1. **Filesystem Layers**: These are read-only layers that represent the filesystem for the container, such as the base OS, libraries, application code, etc. These layers are stacked together to create the image.
2. **Metadata**: This includes configuration data, such as the image's author, version, environment variables, exposed ports, and entrypoint instructions (what command to run when the container starts).
3. **Entrypoint & CMD**:
    - **Entrypoint**: Defines the executable that will be run when the container starts.
    - **CMD**: Defines the default arguments for the entrypoint or provides a default command if the entrypoint is not defined.



Types of Container Images:

1. **Base Images**: These are minimal images that provide a basic operating system or runtime environment without any additional applications or tools. Examples include:
    - ubuntu: A base Ubuntu operating system image.
    - alpine: A small, lightweight Linux distribution image.
2. **Application-Specific Images**: These images come with a specific application installed. For example:
    - nginx: An image with the Nginx web server installed.
    - node: An image with Node.js installed, ready for running JavaScript applications.
3. **Language-Specific Images**: These images provide the environment for specific programming languages or frameworks. For example:
    - python: A base Python image for running Python applications.
    - openjdk: An image with the OpenJDK runtime for running Java applications.
4. **Multi-Stage Images**: These are container images that are built using multiple stages to optimize the final image size. For example, a development stage may compile or build an application, and a final stage will create a minimal runtime environment for the application.



Common Container Image Repositories:

Container images are typically stored and shared through container image registries, which are centralized repositories where images can be pulled, pushed, and stored. The most popular registries include:

1. **Docker Hub**: The default registry for Docker images. It hosts both public and private repositories, including official images from Docker.
2. **Google Container Registry (GCR)**: A Google Cloud platform service for storing Docker images.
3. **Amazon Elastic Container Registry (ECR)**: AWS's managed container image registry service.
4. **Azure Container Registry (ACR)**: A Microsoft Azure service for managing private container image repositories.
5. **Quay.io**: A container image registry by Red Hat, known for its focus on security and enterprise use cases.