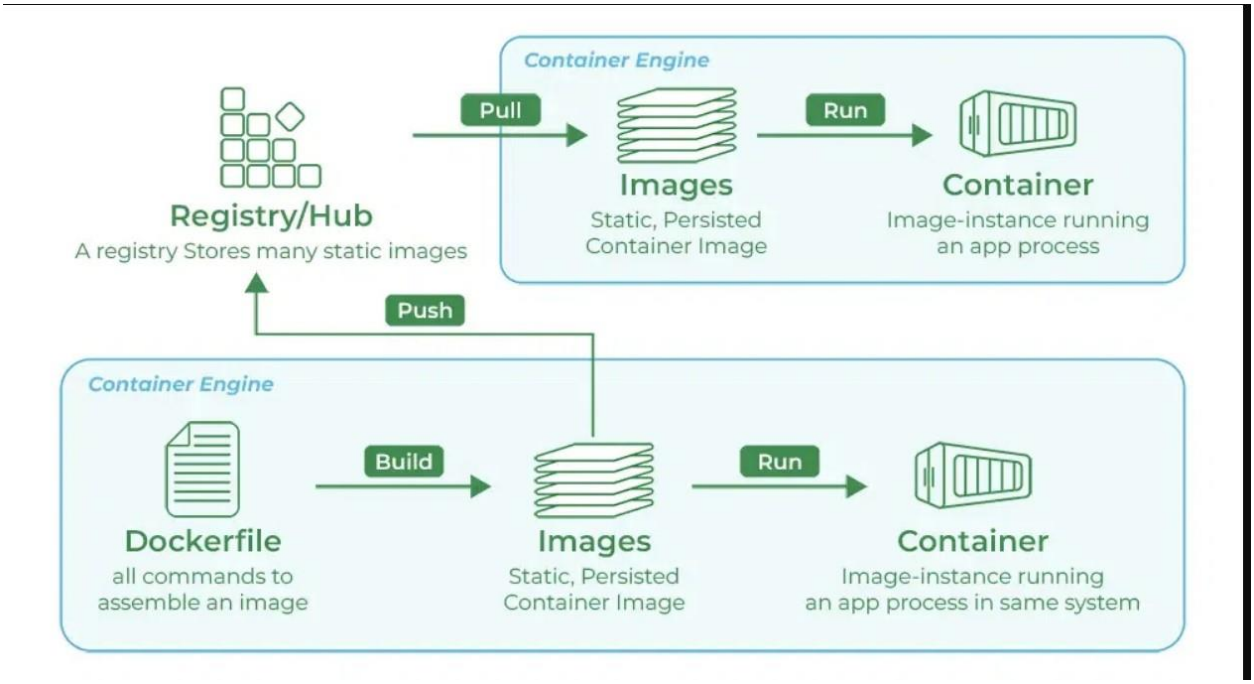
Docker: Build, Run, Push & Pull



1. **Dockerfile**: A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image[1]. Using docker build, users can create an automated build that executes several command-line instructions in succession[1].
2. **Images**: Docker images are the basis of containers. An Image is an ordered collection of root filesystem changes and the corresponding execution parameters for use within a container runtime[1]. An image typically contains a union of layered filesystems stacked on top of each other. An image does not have state and it never changes[1].
3. **Containers**: A container is a runtime instance of an image—what the image becomes in memory when executed (that is, an image with state, or a user process)^[1]. You can see a list of your running containers with the command, docker ps, just as you would in Linux[1].
4. **Docker Hub**: This is a cloud-based registry service which allows you to link to code repositories, build your images and test them, stores manually pushed images, and links to Docker Cloud so you can deploy images to your hosts[1]. It provides a centralized resource for container image discovery, distribution and change management, user and team collaboration, and workflow automation throughout the development pipeline[1].

5. **Containerization**: This is OS-based virtualization that creates multiple virtual units in the userspace, known as Containers[1]. Containers share the same host kernel but are isolated from each other through private namespaces and resource control mechanisms at the OS level[1]. This isolation is achieved by using namespaces and cgroups. Container-based Virtualization provides a different level of abstraction in terms of virtualization and isolation when compared with hypervisors[1].