**Docker:**

Docker allows you to package an application with all of its dependencies into a standardized unit for software development.

[Docker](https://www.docker.com/) is all about making it easier to create, deploy, and run applications by using containers. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

Docker is a tool that is designed to benefit both developers and system administrators, making it a part of many DevOps (developers + operations) toolchains.

In a way, Docker is a bit like a virtual machine. But unlike a virtual machine, rather than creating a whole virtual operating system, Docker allows applications to use the same Linux kernel as the system that they're running on and only requires applications be shipped with things not already running on the host computer. This gives a significant performance boost and reduces the size of the application.

# Delete all docker containers

docker rm $(docker ps -a -q)

# Delete all docker images

docker rmi $(docker images -q)

**Docker works with the following fundamental components:**

* Container – an application sandbox. Each container is based on an image that holds necessary configuration data. When you launch a container from an image, a writable layer is added on top of this image. Every time you commit a container (using the docker commit command), a new image layer is added to store your changes.
* Image – a static snapshot of the containers' configuration. Image is a read-only layer that is never modified, all changes are made in top-most writable layer, and can be saved only by creating a new image. Each image depends on one or more parent images.
* Platform Image – an image that has no parent. Platform images define the runtime environment, packages and utilities necessary for containerized application to run. The platform image is read-only, so any changes are reflected in the copied images stacked on top of it.
* Registry – a repository of images. Registries are public or private repositories that contain images available for download. Some registries allow users to upload images to make them available to others.
* Dockerfile – a configuration file with build instructions for Docker images. Dockerfiles provide a way to automate, reuse, and share build procedures.

