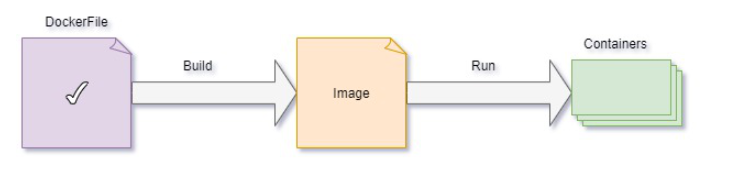
# What is Docker Image?

Docker Image is an executable package of software that includes everything needed to run an application.

This image informs how a container should instantiate, determining which software components will run and how.

Docker Container is a virtual environment that bundles application code with all the dependencies required to run the application. The application runs quickly and reliably from one computing environment to another.



Docker images are built using the Dockerfile which consists of a set of instructions that are required to containerize an application.

# Components of Docker Image

The following are the terminologies and components related to Docker Image:

* **Layers:** Immutable filesystem layers stacked to form a complete image.
* **Base Image:** The foundational layer, often a minimal OS or runtime environment.
* **Dockerfile:** A text file containing instructions to build a Docker image.
* **Image ID:** A unique identifier for each Docker image.
* **Tags**: Labels used to manage and version Docker images.

# SubCommands of Docker Image

The following are the some of the sub commands that are used with Docker Image:

| **Command** | **Description** |
| --- | --- |
| **docker image build** | This command is used for building an image from the Dockerfile |
| **docker image history** | It is used for knowing the history of the docker image |
| **docker image inspect** | It is used for displaying the detailed information on one or more images |
| **docker image prune** | It used for removing unused images that are not associated with any containers |
| **docker image save** | This command helps in saving the docker images into a tar archived files |
| **docker image tag** | It helps in crating a tag to the target image that refers to the source image. |

## **Uses of Docker Images :**

The following are the uses of Docker Images:

1. We can easily and effectively run the containers with the aid of docker images.
2. All the code, configuration settings, environmental variables, libraries, and runtime are included in a Docker image.
3. Docker images are platform-independent.
4. Layers are the building blocks of an image.
5. WithWhen using the build command, the user has the option of completely starting from scratch or using an existing image for the first layer.

## Difference between Docker Image Vs Docker Container

The following are the difference between Docker Image and [Docker Container](https://www.geeksforgeeks.org/containerization-using-docker/" \t "_blank):

| **Docker image** | **Docker container** |
| --- | --- |
| The Docker image is the Docker container’s source code. | The Docker container is the instance of the Docker image. |
| Dockerfile is a prerequisite to Docker Image. | Docker Image is a pre-requisite to Docker Container. |
| Docker images can be shared between users with the help of the Docker Registry. | Docker containers can’t be shared between the users. |
| To make changes in the docker image we need to make changes in Dockerfile. | We can directly interact with the container and can make the changes required. |

Structure Of Docker Image

The layers of software that make up a Docker image make it easier to configure the dependencies needed to execute the container.

Base Image: The basic image will be the starting point for the majority of Dockerfiles, and it can be made from scratch.

Parent Image: The parent image is the image that our image is based on. We can refer to the parent image in the Dockerfile using the FROM command, and each declaration after that affects the parent image.

Layers: Docker images have numerous layers. To create a sequence of intermediary images, each layer is created on top of the one before it.

Docker Registry: Refer to this page on the Docker Registry for further information.

# ****How To Create A Docker Image And Run It As Container?****

Follow the below steps to create a Docker Image and run a Container:

**Step 1:**[Create a Dockerfile.](https://www.geeksforgeeks.org/docker-concept-of-dockerfile/)

**Step 2:**Run the following command in the terminal and it will create a docker image of the application and download all the necessary dependencies needed for the application to run successfully.

docker build -t <name>:<tag>

This will start building the image.

**Step 3:**We have successfully created a [Dockerfile](https://www.geeksforgeeks.org/what-is-dockerfile/" \t "_blank) and a respective Docker image for the same.

**Step 4:**Run the following command in the terminal and it will create a running container with all the needed dependencies and start the application.

docker run -p 9000:80 <image-name>:<tag>