

**DT/NT** : DT

**LESSON** : DevOps

**SUBJECT:** Docker 4  
Image

**BATCH: 149**

**19/10/2023**



**TECHPRO**  
EDUCATION



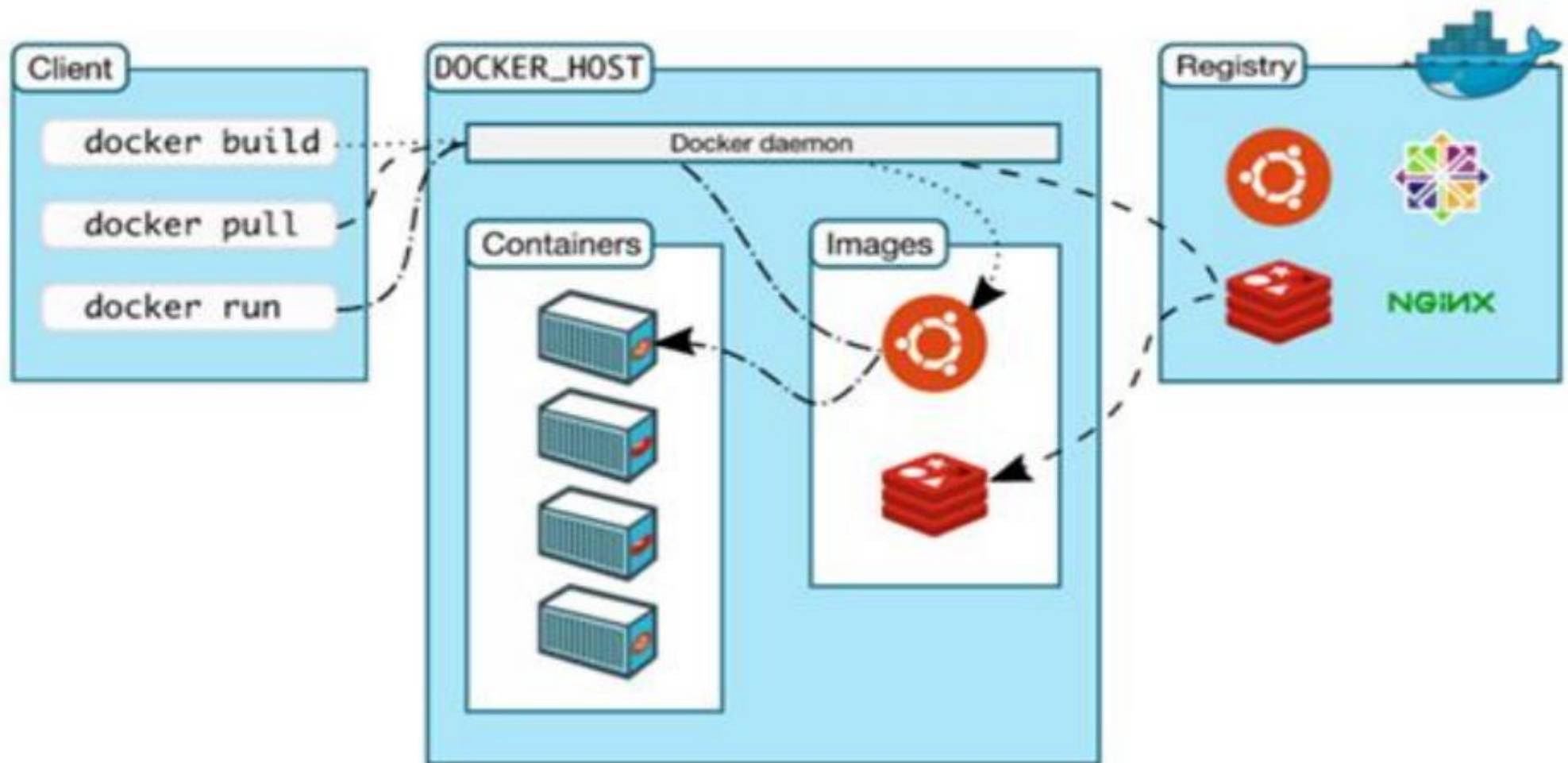
techproeducation.com



+1 (585) 304 29 59



# Docker Architecture



# Docker Image



# Table of Contents

- ▶ What is a Docker image?
- ▶ Dockerfile
- ▶ Dockerfile Instructions
- ▶ Docker Image Naming
- ▶ docker image Commands

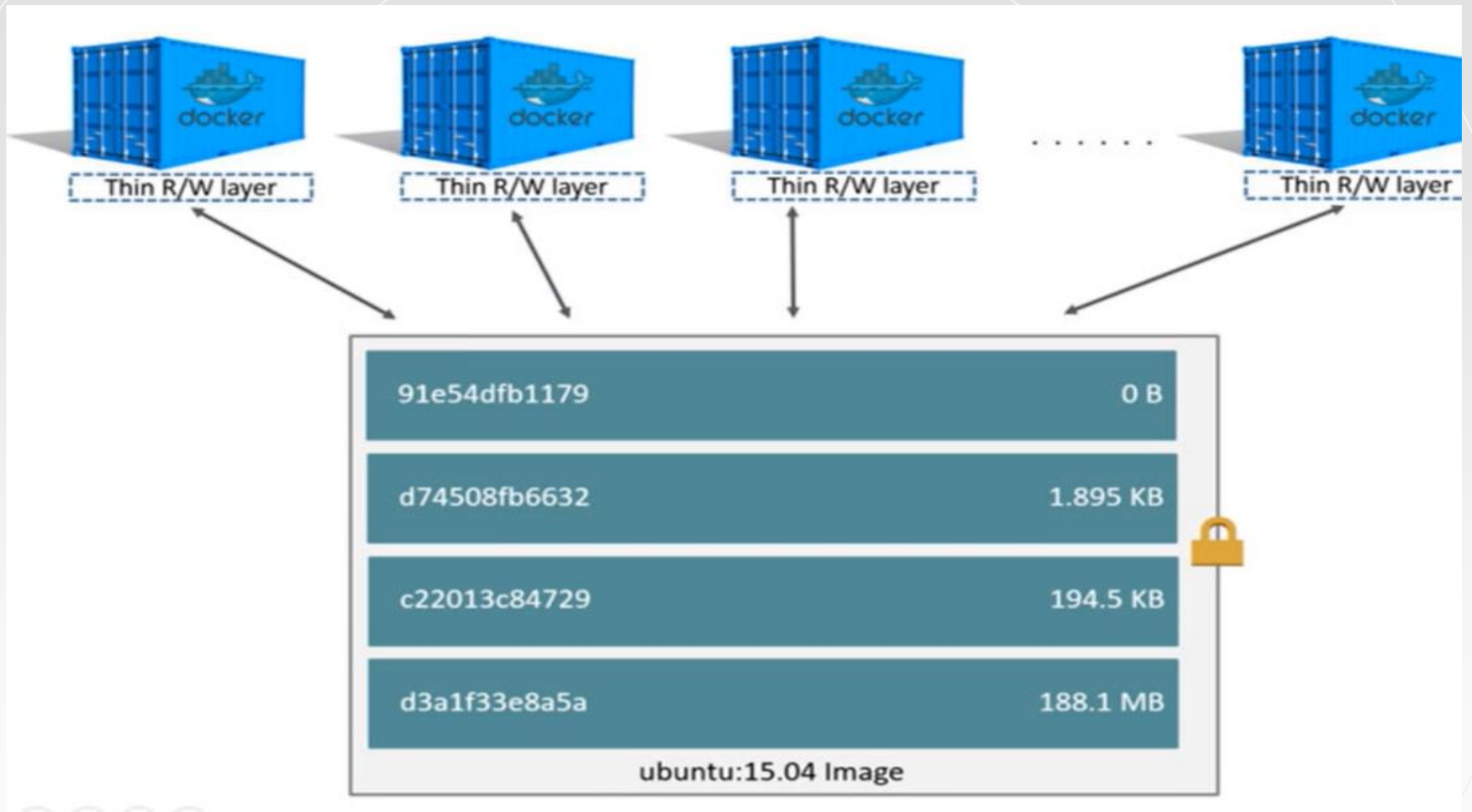
▶ What is a Docker image?

# Images and Containers

- An image is a read-only template with instructions for creating a Docker container.

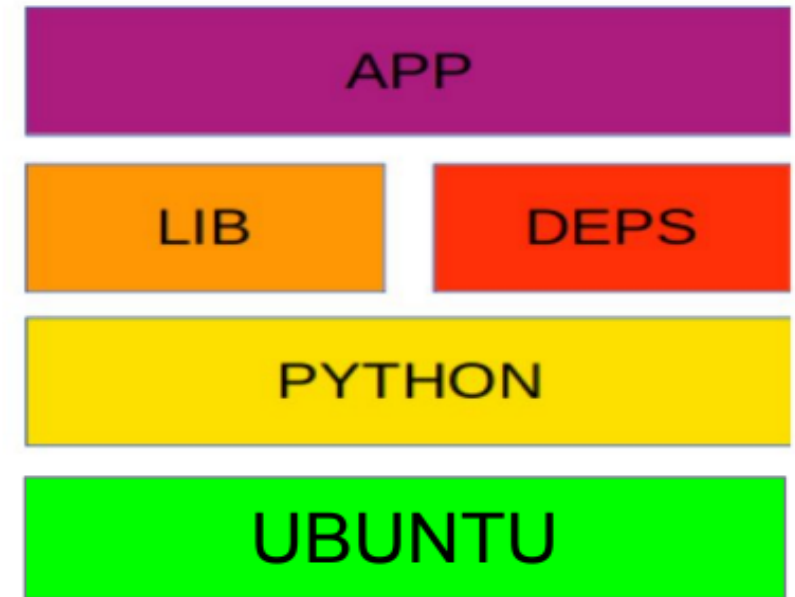






# What is a Docker image?

- An image is based on another image, with some additional customization.
- For example, we may build an image which is based on the ubuntu image, but installs the python and our application, as well as the configuration details needed to make our application run.

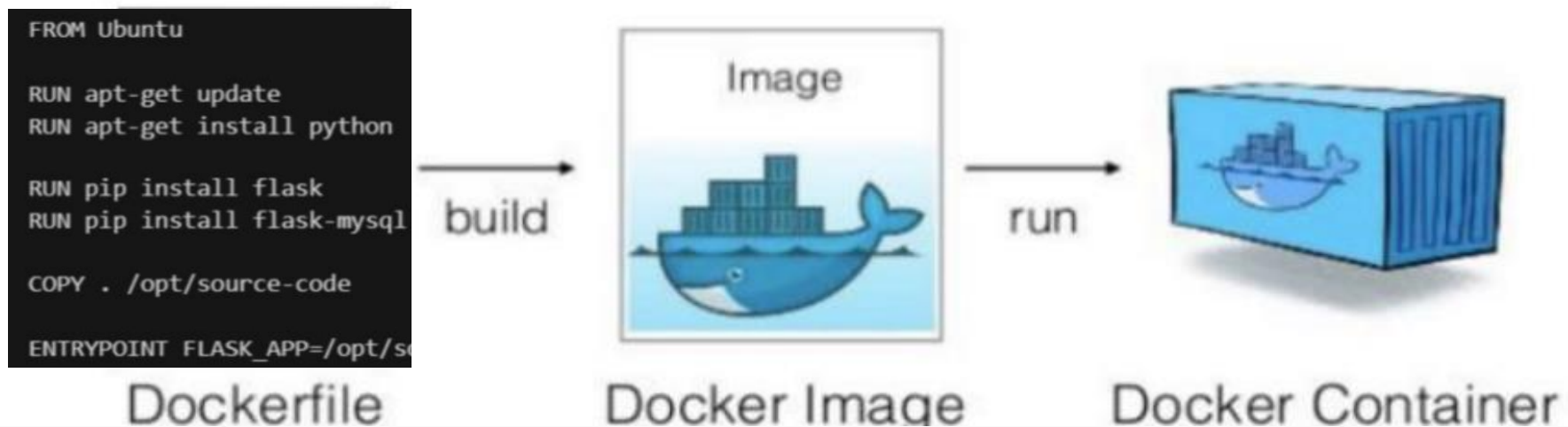


Dockerfile kendine özgü kuralları olan bir dille yazılan ve bizlerin Docker Imageleri oluşturmamızı sağlayan dosyalardır.



# Dockerfile

- A **Dockerfile** is a text document that define the steps needed to create the image and run it.
- We might create our own images with a **Dockerfile**.
- Each instruction in a Dockerfile creates a layer in the image. When we change the Dockerfile and rebuild the image, only those layers which have changed are rebuilt.
- This is part of what makes images so lightweight, small, and fast, when compared to other virtualization technologies.



FROM | Oluşturulacak imajın hangi imajdan oluşturulacağını belirten talimat. Dockerfile içerisinde geçmesi mecburi tek talimat budur. Mutlaka  
Ör: FROM ubuntu:18.04

LABEL | İmaj metadata'sına key=value şeklinde değer çiftleri eklemek için kullanılır. Örneğin team=development şeklinde bir etiket eklenerek t  
Ör: LABEL version:1.0.8

RUN | İmaj oluşturulurken shell'de bir komut çalıştırmak istersek bu talimat kullanılır. Örneğin apt-get install xxx ile xxx isimli uygulamayı  
Ör: RUN apt-get update

WORKDIR | cd xxx ile istediğimiz folder'a geçmek yerine bu talimat kullanılarak istediğimiz klasöre geçer ve oradan çalışmaya devam ederiz.  
Ör: WORKDIR /usr/src/app

USER | gireceğimiz komutları hangi kullanıcı ile çalıştırmasını istiyorsak bu talimat ile onu seçebiliriz.  
Ör: USER poweruser

COPY | İmaj içine dosya veya klasör kopyalamak için kullanırız  
Ör: COPY /source /user/src/app

ADD | COPY ile aynı işi yapar yani dosya ya da klasör kopyalarsınız. Fakat ADD bunun yanında dosya kaynağının bir url olmasına da izin verir.  
Ör: ADD https://wordpress.org/latest.tar.gz /temp

ENV | İmaj içinde environment variable tanımlamak için kullanılır  
Ör: ENV TEMP\_FOLDER="/temp"

ARG | ARG ile de variable tanımlarsınız. Fakat bu variable sadece imaj oluşturulurken yani build aşamasında kullanılır. İmajın oluşturulmuş ha  
Ör: ARG VERSION:1.0

VOLUME | İmaj içerisinde volume tanımlanmasını sağlayan talimat. Eğer bu volume host sistemde varsa container bunu kullanır. Yoksa yeni volume  
Ör: VOLUME /myvol

EXPOSE | Bu imajdan oluşturulacak containerların hangi portlar üstünden erişilebileceğini yani hangi portların yayınlanacağını bu talimatla be  
Ör: EXPOSE 80/tcp

ENTRYPOINT | Bu talimat ile bir containerın çalıştırılabilir bir uygulama gibi ayarlanabilmesini sağlarsınız.  
Ör: ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]

# Dockerfile

```
FROM ubuntu
RUN apt-get update -y
RUN apt-get install python3 -y
RUN apt-get install python3-pip -y
RUN pip3 install flask
COPY . /app
WORKDIR /app
CMD python3 ./welcome.py
```

1. Start from a base OS or another image

2. Install all dependencies

3. Copy source code

4. Specify command

**FROM Ubuntu** # Bu Dockerfile hangi işletim sisteminde çalışacak, aşağıdakiler Ubuntu komutları. Butun Docker dosyaları FROM ile başlamak zorundadır.

**RUN apt-get update** # En güncel paketlerle yoluna devam eder

**RUN apt-get install python** # Python'u yükler

**RUN pip install flask** # Flask'i yükler

**RUN pip install flask-mysql** # Flask-mysql'i yükler

**COPY . /opt/source-code** # Dosyaları Lokalden Docker Image'a kopyalar

**ENTRYPOINT FLASK\_APP=/opt/source-code/app.py flask run** # Updates Endpoint



# Dockerfile Instructions

Instruction	Description
ENV	The ENV instruction sets the environment variable <key> to the value <value>.
ADD	The ADD instruction copies new files, directories or remote file URLs from <src> and adds them to the filesystem of the image at the path <dest>.
COPY	The COPY instruction copies new files or directories from <src> and adds them to the filesystem of the container at the path <dest>.
ENTRYPOINT	An ENTRYPOINT allows you to configure a container that will run as an executable.
VOLUME	The VOLUME instruction creates a mount point with the specified name and marks it as holding externally mounted volumes from native host or other containers.

# Dockerfile Instructions

Instruction	Description
WORKDIR	The WORKDIR instruction sets the working directory for any RUN, CMD, ENTRYPOINT, COPY and ADD instructions that follow it in the Dockerfile. If the WORKDIR doesn't exist, it will be created even if it's not used in any subsequent Dockerfile instruction.
ARG	The ARG instruction defines a variable that users can pass at build-time to the builder with the docker build command using the --build-arg <varname>=<value> flag. If a user specifies a build argument that was not defined in the Dockerfile, the build outputs a warning.
HEALTHCHECK	The HEALTHCHECK instruction tells Docker how to test a container to check that it is still working. This can detect cases such as a web server that is stuck in an infinite loop and unable to handle new connections, even though the server process is still running.



# docker image Commands

Command	Description
<a href="#"><u>docker image build</u></a>	Build an image from a Dockerfile
<a href="#"><u>docker image history</u></a>	Show the history of an image
<a href="#"><u>docker image inspect</u></a>	Display detailed information on one or more images
<a href="#"><u>docker image ls</u></a>	List images
<a href="#"><u>docker image prune</u></a>	Remove unused images
<a href="#"><u>docker image pull</u></a>	Pull an image or a repository from a registry
<a href="#"><u>docker image push</u></a>	Push an image or a repository to a registry
<a href="#"><u>docker image rm</u></a>	Remove one or more images
<a href="#"><u>docker image tag</u></a>	Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE