DATE : 28.03.2025

DT/NT : NT

LESSON : DOCKER

SUBJECT: IMAGE

BATCH : B 303

AWS-DEVOPS

TECHPRO EDUCATION

techproeducation.com +1 (585) 304 29 59

Docker Architecture

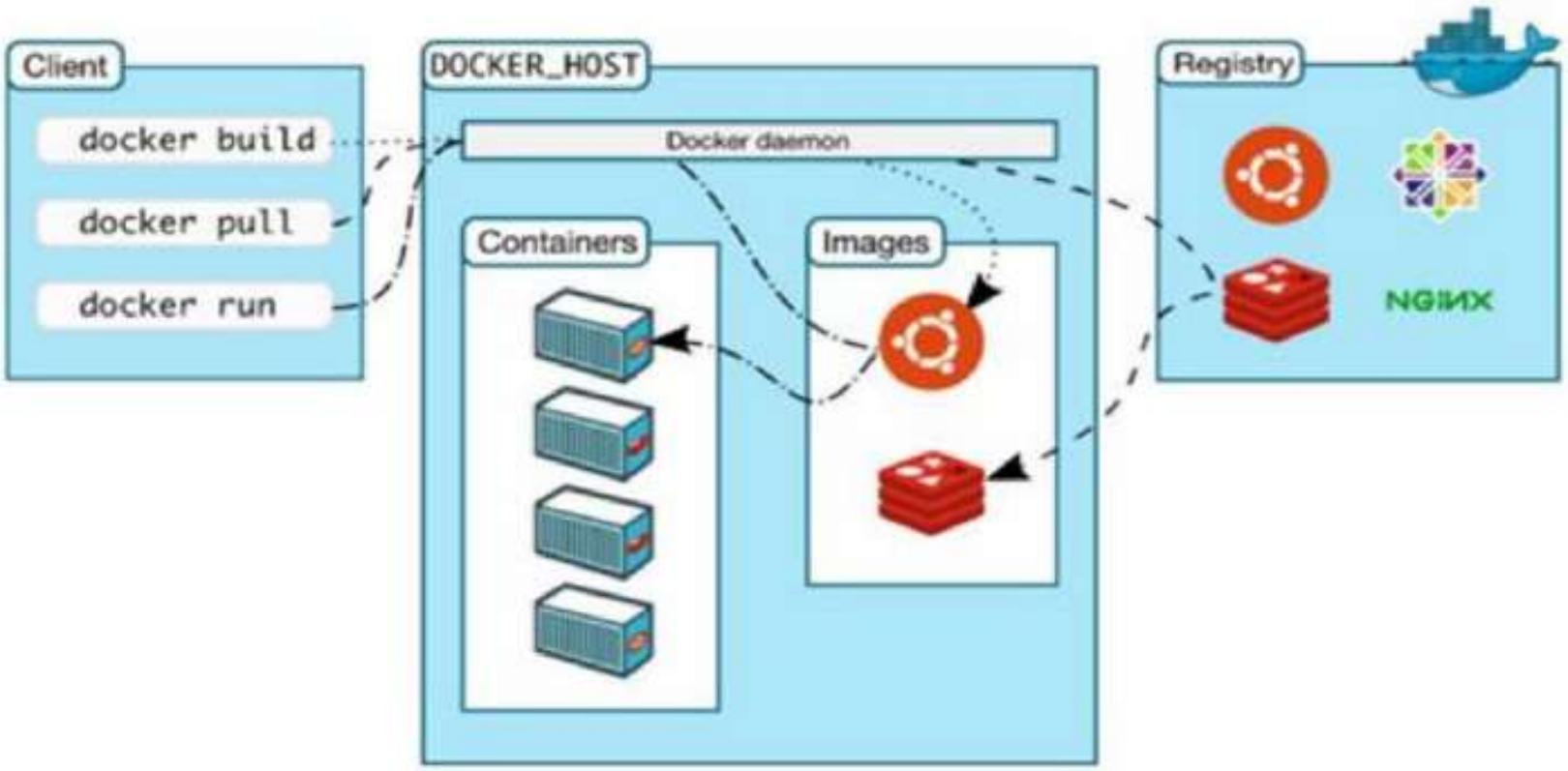


Table of Contents

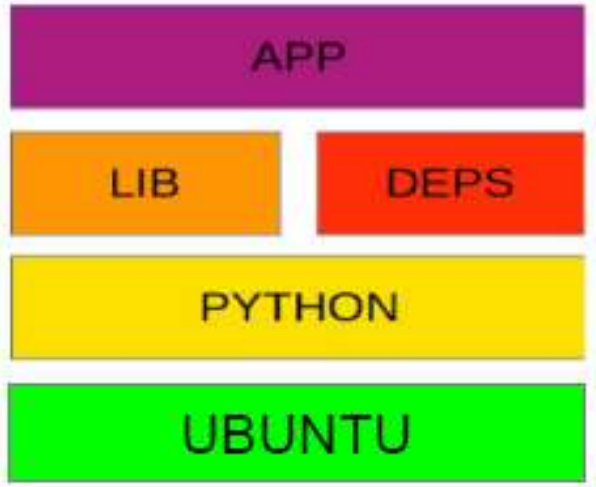
What is a Docker image?

Images and Containers

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

What is a Docker image?

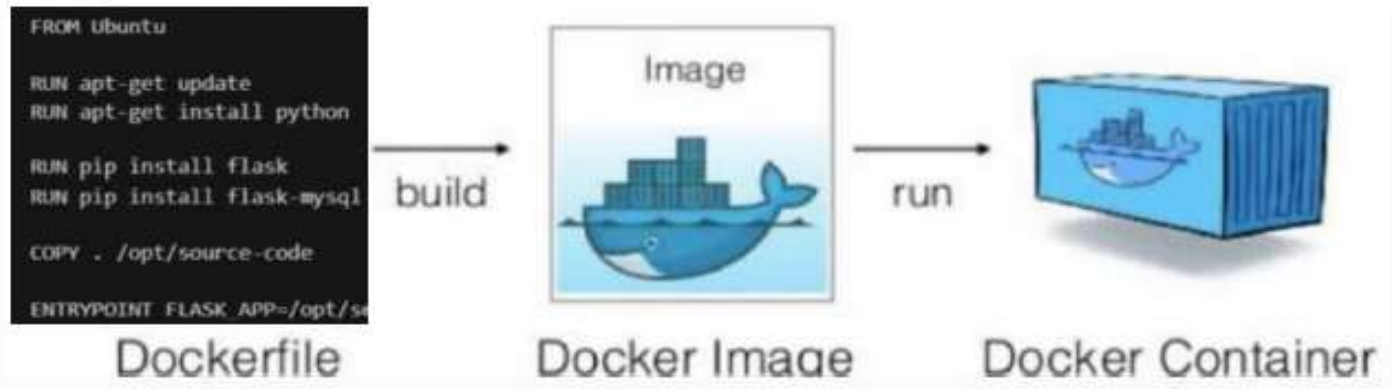
- An image is based on another image, with some additional customization.
- For example, we may build an image which is based on the ubuntu image, but installs the python and our application, as well as the configuration details needed to make our application run.



Dockerfile

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.



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

Dockerfile

```
FROM Ubuntu # Bu Dockerfile hangi isletim sisteminde calisacak, asagidakiler Ubuntu komutlari. Butun Docker dosyalari FROM ile baslamak zorundadir.

RUN apt-get update # En guncel paketlerle yoluna devam eder
RUN apt-get install python # Pyhton'u yukler

RUN pip install flask # Flask'i yukler
RUN pip install flask-mysql # Flask-mysql'i yukler

COPY . /opt/source-code # Dosyalari lokalden Docker Image'a kopyalar

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run # Updates Endpoint
```

Dockerfile Instructions

Instruction	Description
FROM	The FROM instruction initializes a new build stage and sets the *Base Image* for subsequent instructions. As such, a valid Dockerfile must start with a FROM instruction.
RUN	The RUN instruction will execute any commands in a new layer on top of the current image and commit the results. The resulting committed image will be used for the next step in the Dockerfile.
CMD	The main purpose of a CMD is to provide defaults for an executing container. There can only be one CMD instruction in a Dockerfile. If you list more than one CMD then only the last CMD will take effect.
EXPOSE	The EXPOSE instruction informs Docker that the container listens on the specified network ports at runtime. You can specify whether the port listens on TCP or UDP, and the default is TCP if the protocol is not specified.

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 Naming

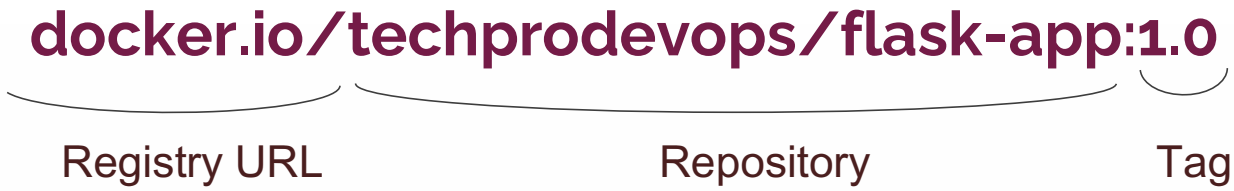
docker.io/ubuntu:20.04

Registry URL — Repository — Tag

docker.io/techprodevops/flask-app:1.0

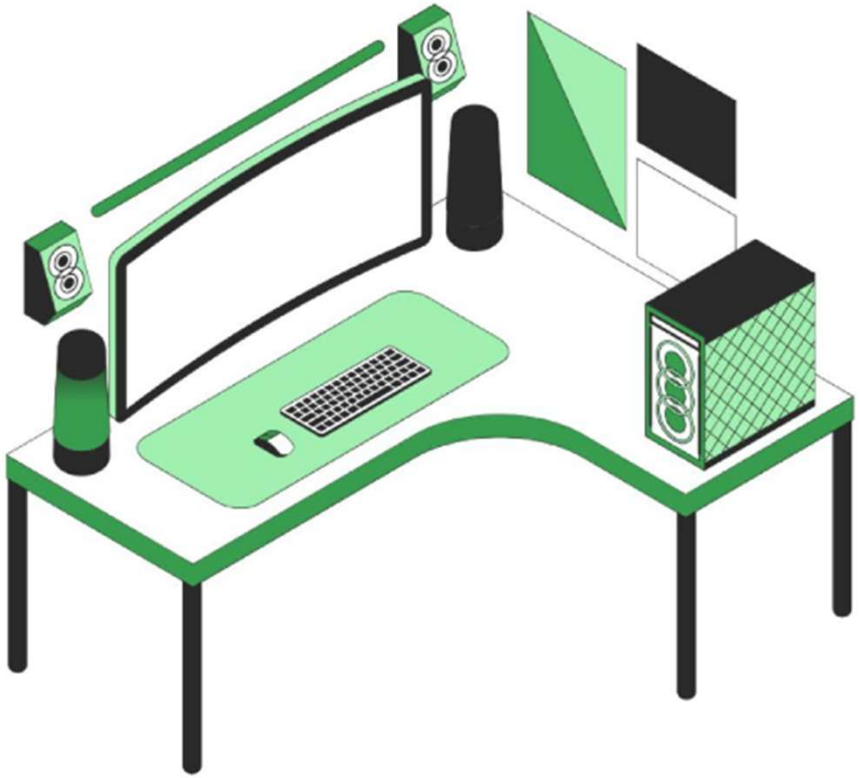
Registry URL — Repository — Tag

Docker Image Naming



docker image Commands

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



Do you have any questions?

Send it to us! We hope you learned something new.

TECHPRO EDUCATION