user

→ App.py
  ↳ uses Flask, Flask_restful for creating repository of resources.
    ↳ That are end point accessed

→ Requirement.txt → Flask
  R. - - - dependencies

→ Dockerfile.
  ↳ How to build the machine from scratch
    ↳ will perform → creating VM, down OS image + Python image, depend + RUN app.
  ↳ From Python3
    ⇒ This downloads the python 3 application image
      ↳ this image itself uses dockerfile that downloads various other thing
        ↳ OS ubuntu
        ↳ other dependencies.
      ↳ we can access this image (Python3) from docker-hub the same way we use modules from git-hub

Like u open Python.

↳ Workdir /usr/src/app
  ⇒ Make dir + ~~Access~~ Remember its access point for future use.

↳ Copy requirements.txt
  ↳ Copy the txt file present in the folder (where dockerfile is) to the machine folder

machine folder is built by at usr/src/app, the virtual location

↳ RUN pip install --no-cache-dir -r requirements.txt

Execute this command ⇒ install dependencies

↳ assert no cached dir from before

-r → Recursively install all sub-dependencies

↳ COPY . .

your machine ↳ Virtual Machine

(All files / neighbour files / siblings of Dockerfile)

↳ CMD ["python" "app.py"]

We write python3 app.py.

** But we run app.py using Flask.

↳ ∴ some more instructions need to be passed.

web { App.py , requirements.txt and Dockerfile

↳ Folder

↳ Cmds for creating VM + running app

text.

↳ We create docker-compose.yml YAML file.

⇒ For each application we create a docker container → To run that module

→ We sync these container using YAML file.

* ⇒ version: '3'

↳ Tells the version to use

** ⇒ services:

web:

build: ./web → Location on machine.

Create and run web application

↳ Achieved using dockerfile in app.

ports:

- "5000:5000"

db:

build: ./db

↳ Request to build things up

sudo docker-compose build

sudo docker-compose up