**docker compose**

create the following files:

* requirements.txt - describes the Python dependencies, we’ll use this file to install Flask and the Redis Python interface to be used in the Python application.
* app.py - defines our Flask application
* Dockerfile - defines the Docker image of the Flask application

$ mkdir mycomp

$ cd mycomp

$ vi requirements.txt:

flask

redis<3.0.0

$

$ vi app.py

from flask import Flask, request, jsonify

from redis import Redis

app = Flask(\_\_name\_\_)

redis = Redis(host="redis", db=0, socket\_timeout=5, charset="utf-8", decode\_responses=True)

@app.route('/', methods=['POST', 'GET'])

def index():

if request.method == 'POST':

name = request.json['name']

redis.rpush('students', {'name': name})

return jsonify({'name': name})

if request.method == 'GET':

return jsonify(redis.lrange('students', 0, -1))

$

$

$ vi Dockerfile:

FROM python:3.7.0-alpine3.8

WORKDIR /usr/src/app

COPY requirements.txt ./

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

COPY . .

ENV FLASK\_APP=app.py

CMD flask run --host=0.0.0.0

$

$

$ vi docker-compose.yml

version: '3'

services:

app:

build: .

image: takacsmark/flask-redis:1.0

environment:

- FLASK\_ENV=development

ports:

- 5000:5000

redis:

image: redis:4.0.11-alpine

$

$ tree

$

$ docker-compose up --build

$

To test

$ curl --header "Content-Type: application/json" \

--request POST \

--data '{"name":"Kumar"}' \

localhost:5000

{

"name": "Kumar"

}

$

$ curl localhost:5000

$ docker-compose log

$ docker-compose top

$ docker-compose exec app ps

$ docker-compose exec redis redis-cli lrange students 0 -1

$ docker-compose run app tail -f > dev0

$ docker-compose ps

$ docker-compose down

$ docker-compose up -d