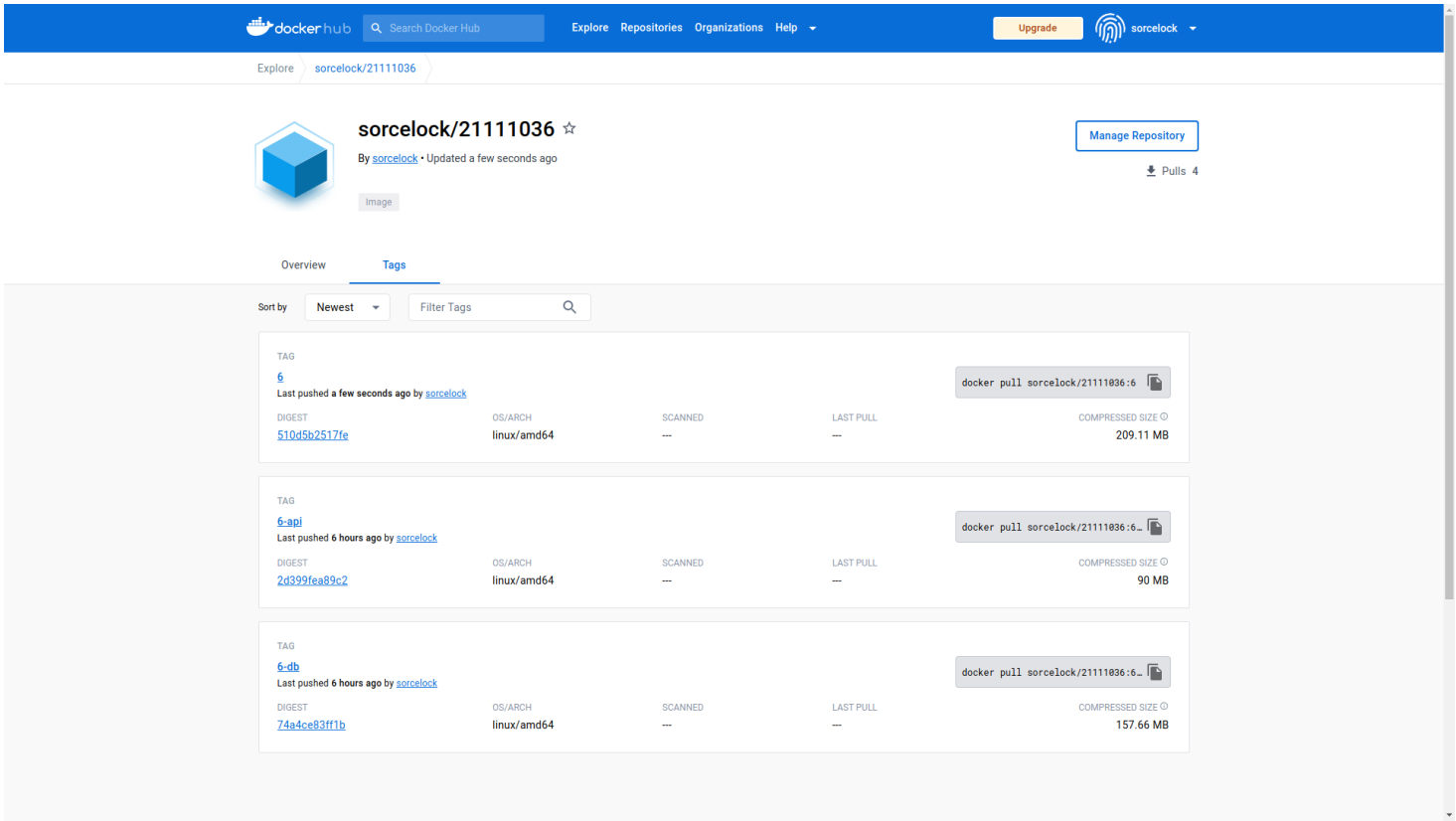
Devops - Assignment - 06

RollNo: 21111036
Class: MSc. C.S. B-21

Problem Statement:

1. Create a docker custom image using Dockerfile with all the available flags
2. Write RESTful APIs for student management in python language using flask.
3. User mysql as database
4. Copy the code in the docker image and give the image tag as syntax below: <YOUR_USERNAME>/<ROLL_NUMBER>:<Assignment_Number>
5. Push the docker image to https://hub.docker.com under your account
6. Paste the screenshot of uploaded image in the PDF file

Ans:



Single Image

- Use `docker pull sorcelock/21111036:6` to pull the image
- Use `docker run --network host -p 5000:5000 -p 3306:3306 sorcelock/21111036:6` to run the image.

Dockerfile

```
FROM mysql:debian

COPY requirements.txt /srv
COPY app-run.sh /bin

ENV MYSQL_DATABASE="students"
ENV MYSQL_USER="admin"
ENV MYSQL_PASSWORD="admin@123"
ENV MYSQL_ROOT_PASSWORD="secret"
ENV MYSQL_PORT="3306"
ENV MYSQL_HOST="localhost"

ENV FLASK_APP=/srv/api.py
ENV FLASK_DEBUG=0

EXPOSE 3306
EXPOSE 5000

RUN apt-get update && \
    apt-get install -y python3 python3-pip default-libmysqlclient-dev gcc && \
    rm -rf /var/lib/apt/lists/* && \
        pip install --trusted-host pypi.python.org -r /srv/requirements.txt && \
        apt-get autoremove --purge -y gcc && \
        apt-get clean

COPY schema.sql /docker-entrypoint-initdb.d
COPY api.py /srv/api.py
ENTRYPOINT app-run.sh
```

app-run.sh

```
#!/bin/sh

docker-entrypoint.sh mysqld &
flask -A /srv/api.py run
```

requirements

```
Flask==2.2.3
Flask-MySQL==1.5.2
Flask-MySQLdb==1.0.1
Flask-RESTful==0.3.9
```

Compose

Image has been uploaded as 2 separate units one database image for `mysql` as `db` and other for the `api`.

Use the following `docker-compose.yml` file to run the project

```yaml
version: "3.8"

services:
  db:
    image: sorcelock/21111036:6-db
    restart: always
    networks:
      - net-6
    environment:
      MYSQL_DATABASE: students
      MYSQL_USER: admin
      MYSQL_PASSWORD: admin@123
      MYSQL_ROOT_PASSWORD: secret
    ports:
      - "3306:3306"
    volumes:
      - data-6:/var/lib/mysql

  api:
    image: sorcelock/21111036:6-api
    restart: always
    networks:
      - net-6
    environment:
      MYSQL_HOST: db
      MYSQL_PORT: 3306
      MYSQL_USER: admin
      MYSQL_PASSWORD: admin@123
      MYSQL_DATABASE: students
    ports:
      - "5000:5000"
    depends_on:
      - db

networks:
  net-6:

volumes:
  data-6:
```

`Dockerfile` for `db`

```
FROM mysql:latest

COPY schema.sql /docker-entrypoint-initdb.d
```

`Dockerfile` for `api`

```
FROM ubuntu:latest

WORKDIR /app

COPY requirements.txt .

RUN apt-get update && \
    apt-get install -y python3 python3-pip default-mysql-client default-
libmysqlclient-dev gcc && \
    rm -rf /var/lib/apt/lists/* && \
        pip install --trusted-host pypi.python.org -r requirements.txt && \
        apt-get autoremove --purge -y gcc && \
        apt-get clean

COPY . /app

ENV FLASK_APP=app.py
ENV FLASK_DEBUG=0

EXPOSE 5000


ENTRYPOINT flask run --host=0.0.0.0
```

Schema for the database (you don't have to execute it, included in the `db` image):

```
CREATE TABLE `students` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `phone` varchar(15) NOT NULL,
  `address` varchar(255) DEFAULT NULL,
  `date_of_birth` date DEFAULT NULL,
  `course` varchar(255) DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp(),
  PRIMARY KEY (`id`),
  UNIQUE KEY `email` (`email`)
);
```

Use the following commands from terminal to use the API
(or you can use tools like POSTMAN)

```
# To get a student's details
$ curl -X GET localhost:5000/students/<student_id_from_table>

# To get all students details
$ curl -X GET localhost:5000/students/0

# To Enter a record in student table
$ curl -H 'Content-Type: application/json' -X POST -d '{"name": "Name", "email":
"Email", "phone": "9876543210", "address": "Address", "date_of_birth": "1970-01-
01", "course": "LLAP"}' localhost:5000/students

# To update a record
$ curl -H 'Content-Type: application/json' -X PUT -d '{"name": "Name", "email":
"Email", "phone": "9876543201", "address": "Address", "date_of_birth": "1970-01-
01", "course": "DEVOPS"}' localhost:5000/students/<student_id_from_table>

# To Delete a record
$ curl -X DELETE localhost:5000/students/<student_id_from_table>
```