

DISTRIBUTED SYSTEMS

PROJECT PHASE – I REPORT

Frameworks & Programming Language:

1. Flask
2. Python
3. MongoDB

Project Description:

Our application consists of 2 separate containers, out of which one has server on it, while on the other hand, database is stored in the other one. Both the containers are connected using network bridge for intercommunication. We have also mapped the database volume to our local machine volume so that data remains persistent, even after shutting down the containers.

Instructions Executed:

- **docker images**

```
C:\Users\Prashant Kalyani>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
myimage              latest              d74b208b71e1       5 minutes ago      932MB
<none>               <none>              473517108a0e       57 minutes ago     932MB
hello-world          latest              feb5d9fea6a5       3 days ago         13.3kB
mongo                latest              ccf4b4ee3bee       7 days ago         685MB

C:\Users\Prashant Kalyani>
```

Above instruction displays the list of all the images in our local machine after instruction execution

- **docker build -t myimage .**

```
0 D:\docker_files>docker build -t myimage .
go [+] Building 7.5s (9/9) FINISHED
    => [internal] load build definition from Dockerfile
    => => transferring dockerfile: 32B
    => [internal] load .dockerignore
    => => transferring context: 2B
    => [internal] load metadata for docker.io/library/python:3.10.0rc2-bullseye
    => [internal] load build context
    => => transferring context: 532B
    => CACHED [1/4] FROM docker.io/library/python:3.10.0rc2-bullseye@sha256:ffbf378f6e0cc7e73ae
    => [2/4] COPY . /src
    => [3/4] RUN pip install Flask
    => [4/4] RUN pip install pymongo
    => exporting to image
    => => exporting layers
    => => writing image sha256:121f7ef5b424f52f1abbc71f8a6aee06e2b57cb83f6b3f888b7498d3aac1e7a1
```

We created a new image with the above instructions.

The following docker file contained installation of Flask and pymongo libraries command and configuration for the image.

- **docker run -it -p 80:5000 --network mode-webapp-network --name my_app --rm --env-file .env myimage**

```
D:\docker_files>docker run -it -p 80:5000 --network mode-webapp-network --name my_app --rm --env-file .env myimage
MongoClient(host=['mongodb1:27017'], document_class=dict, tz_aware=False, connect=True)
* Serving Flask app 'first_flask_app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://172.18.0.3:5000/ (Press CTRL+C to quit)
```

When we executed the above the instruction, the container was mapped to the localhost, port no 80 and was connected to the network bridge “mode-network-webapp-network”

- **localhost:80**

HTML Forms

Name:

UB Number:

Above image is the flask app running on the localhost at port 80

- **After submitting data**



After submitting the data, successful image is popped up, indicating that the data has been stored in the database

Database Output:

Database output

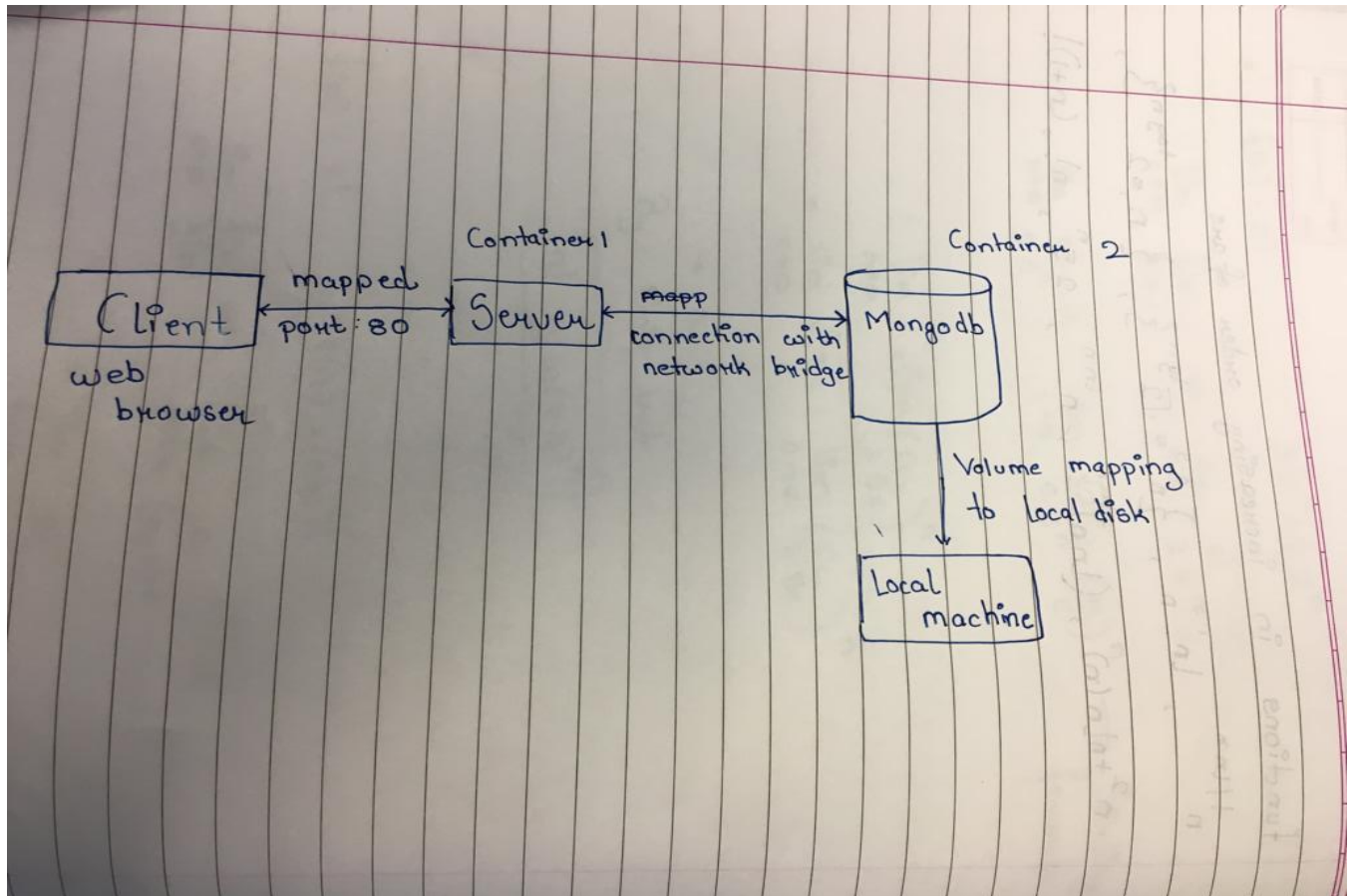
User Name:sid Ub Person:50415023

User Name:sid Ub Person:50415033

User Name:Sidney Ub Person:50475034

In the above image, the last entry was the most recently inserted through our flask application.

Block Diagram:



The image shows a Windows 10 desktop environment. Two windows are open. The left window is a terminal titled 'C:\Windows\System32\cmd.exe' with the command 'docker build -t myimage' entered. The output shows the Docker build process, including the installation of Flask and pymongo. The right window is a code editor titled 'C:\Windows\System32\cmd.exe' showing a Python script for a Flask application. The script defines a Flask app, connects to a MongoDB database, and has a route for a POST request. The desktop background is a Windows 10 wallpaper, and the taskbar at the bottom shows various application icons and the system clock.