CC REPORT ASSIGNMENT – 1

NAME: RAHUL VARMA

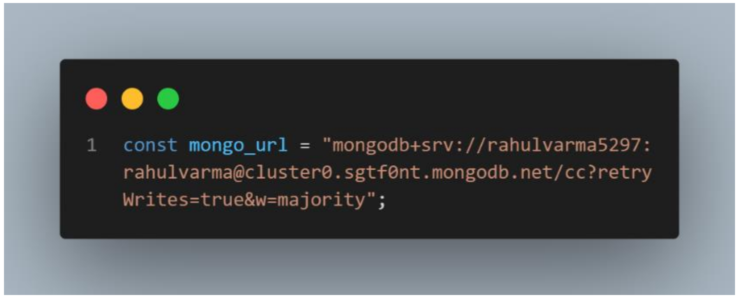
Roll No: S20200010212

1.Details of my setup:

(IP, port, GET parameter name etc.)

CREATED TWO SERVICES

- Named service1 and service2
- Using Express created a get request for each service for showing output on the browser page.
- Using mongoose (mongo DB) connected to a data base
- Link for my mongo DB database is show below.



```
const mongo_url = "mongodb+srv://rahulvarma5297:rahulvarma@cluster0.sgtf0nt.mongodb.net/cc?retryWrites=true&w=majority";
```

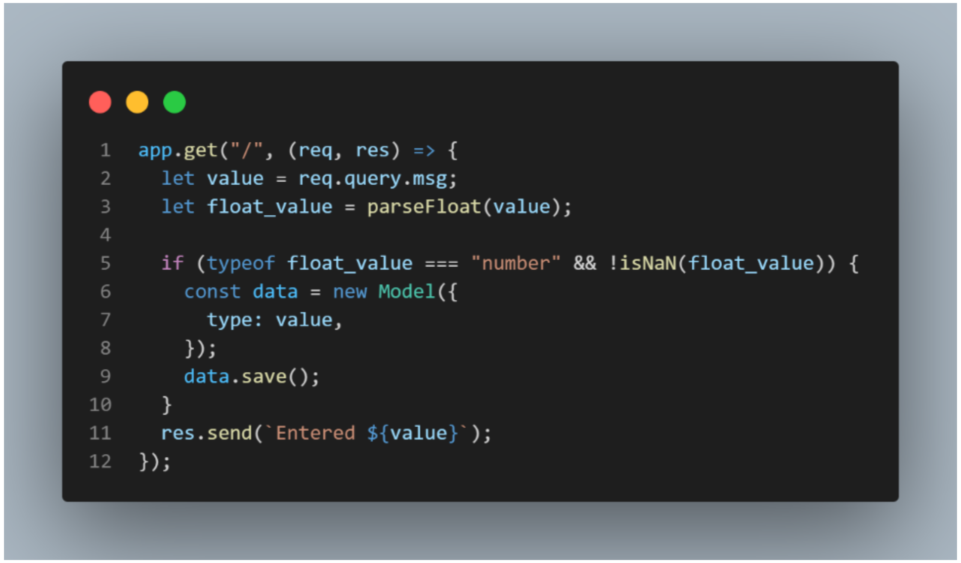
Link:
"mongodb+srv://rahulvarma5297:rahulvarma@cluster0.sgtf0nt.mongodb.net/cc?retryWrites=true&w=majority"

Created Schema and Model for inserting data into database and accessing the data from the database

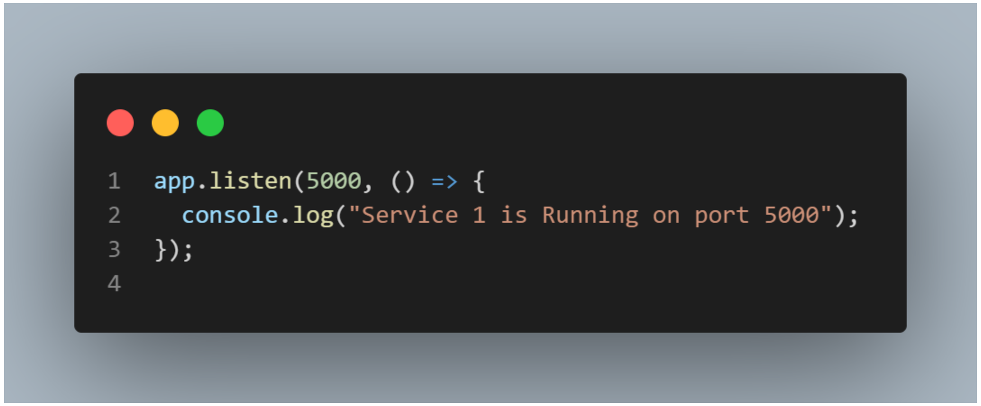
SERVICE 1:

PORT = 5000

GET parameter name: msg



```
app.get("/", (req, res) => {
  let value = req.query.msg;
  let float_value = parseFloat(value);

  if (typeof float_value === "number" && !isNaN(float_value)) {
    const data = new Model({
      type: value,
    });
    data.save();
  }
  res.send(`Entered ${value}`);
});
```

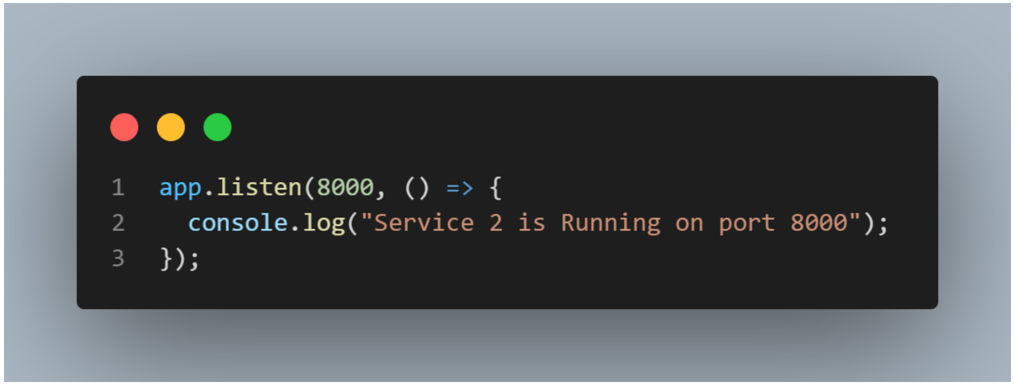


```
app.listen(5000, () => {
  console.log("Service 1 is Running on port 5000");
});

```

SERVICE 2:

PORT: 8000

No Parameter for service 2, just the output is average value.



```
app.listen(8000, () => {
  console.log("Service 2 is Running on port 8000");
});
```

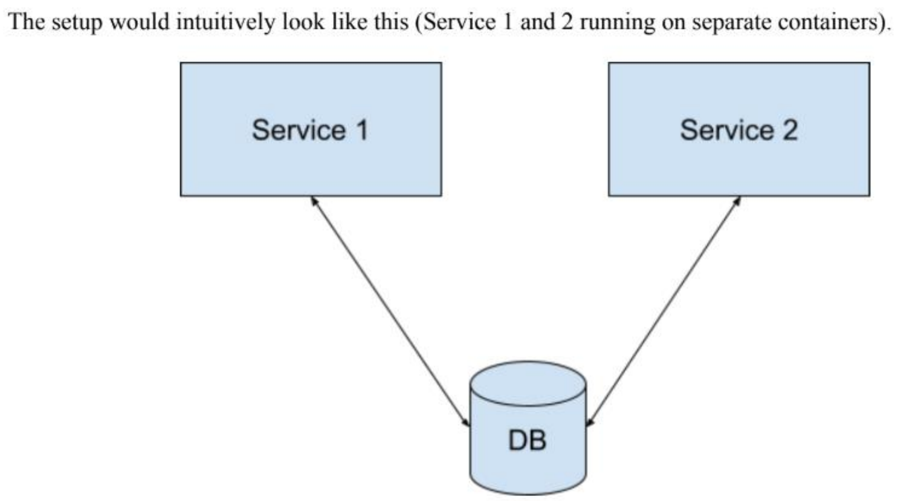
2.Web technology used, database used, how the setup was hosted online, which cloud provider was used etc. included the Docker files/YAML files and about the build setup.

- Web Technology used is Node.js (express).
- Database used is MongoDB.
- Cloud Provider used is Microsoft Azure.

The Process of the Setup:

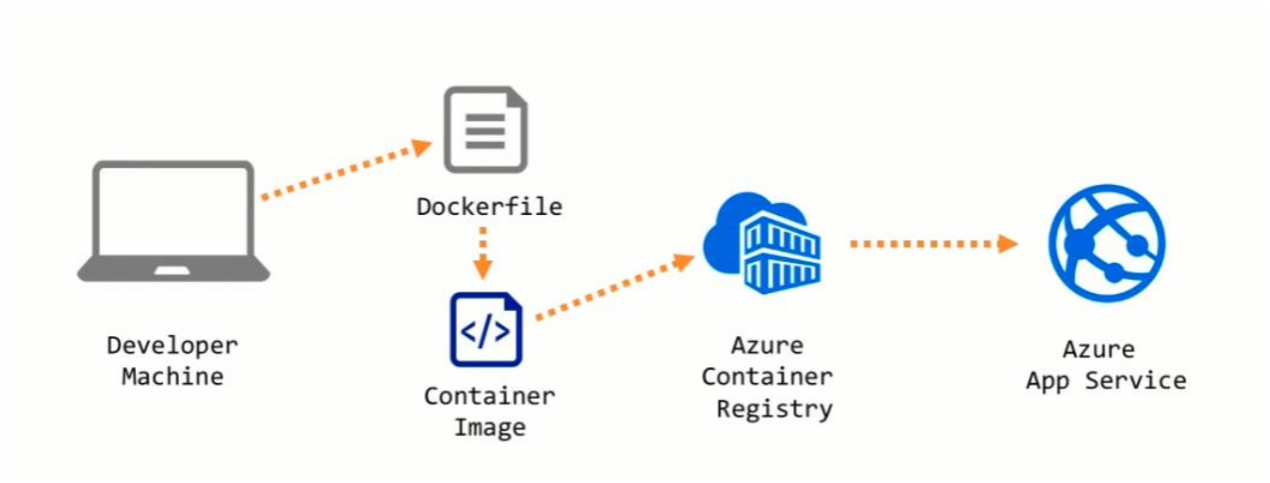
Taking a Developer Machine, a docker file and building our own container image and pushing that container image to a container registry by using the Azure container registry.

Push a container image to it and then run a container based off that image in azure app service as a web application.



The setup would intuitively look like this (Service 1 and 2 running on separate containers).

Process in one image:

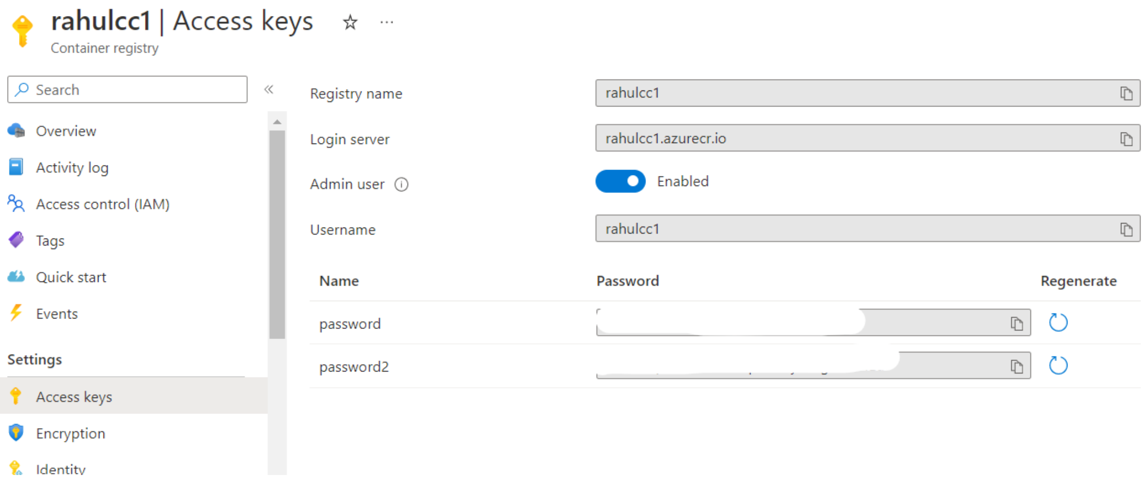


Created a New Resource using Container Registry for service 1 and service 2

Created a new container Registry with Registry Name rahulcc1 for service 1

And rahulcc2 for service 2.

We get Access keys for both container Registry like shown in below image:



rahulcc1 | Access keys

Container registry

Registry name: rahulcc1

Login server: rahulcc1.azurecr.io

Admin user: Enabled

Username: rahulcc1

Name	Password	Regenerate
password	[redacted]	↻
password2	[redacted]	↻

We run the command in service1 and service2 terminal from our local system

```
docker login rahulcc1.azurecr.io
```

```
docker login rahulcc2.azurecr.io
```

Using the username and password shown in above image we can login for the container registry.

Now we should do build and push the code to the container registry.

Commands for building the code is:

```
docker build -t rahulcc1.azurecr.io/rahulcc1:latest .

docker build -t rahulcc2.azurecr.io/rahulcc2:latest .
```

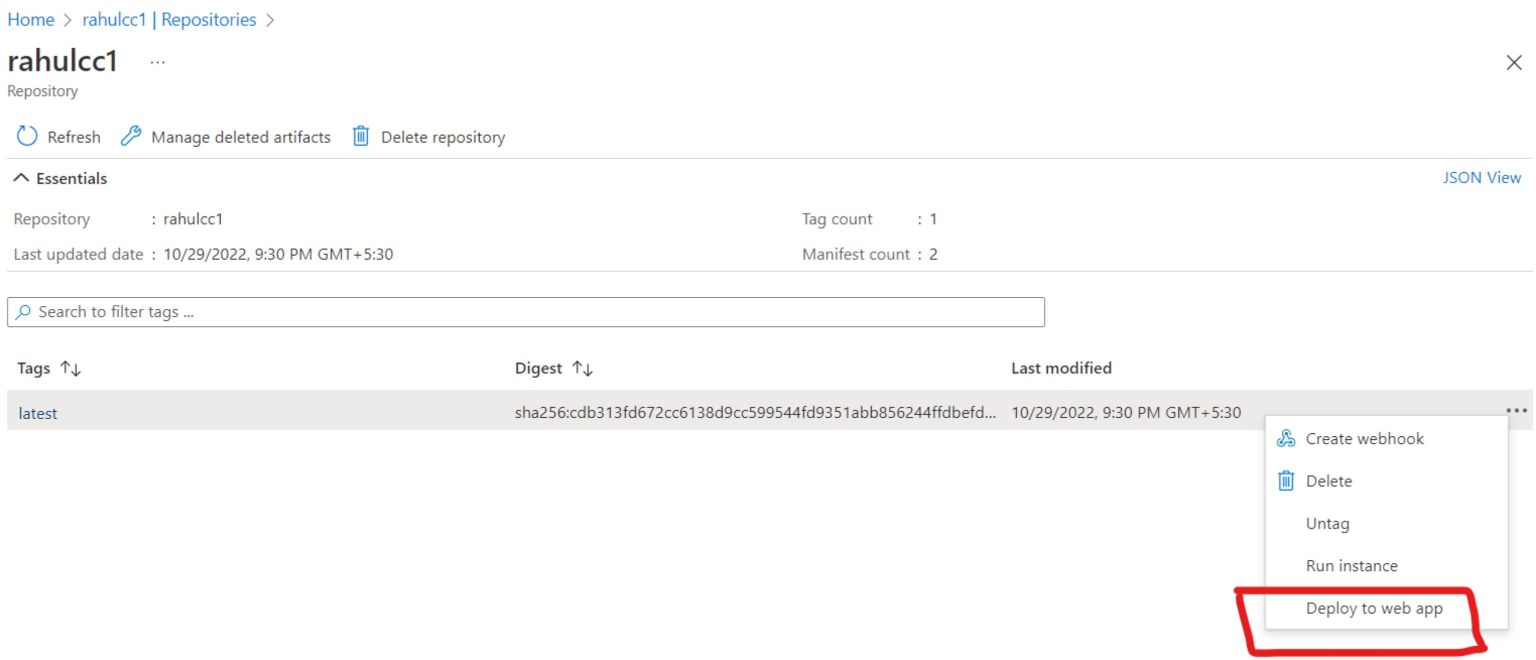
Commands for Pushing the code is:

```
docker push rahulcc1.azurecr.io/rahulcc1:latest

docker push rahulcc2.azurecr.io/rahulcc2:latest
```

Now the Code is Pushed into the Container Registries.

Now for Each Container Registry we should deploy to the web.

For Deploying we get the option as shown in the below image:



rahulcc1

Repository

Repository : rahulcc1

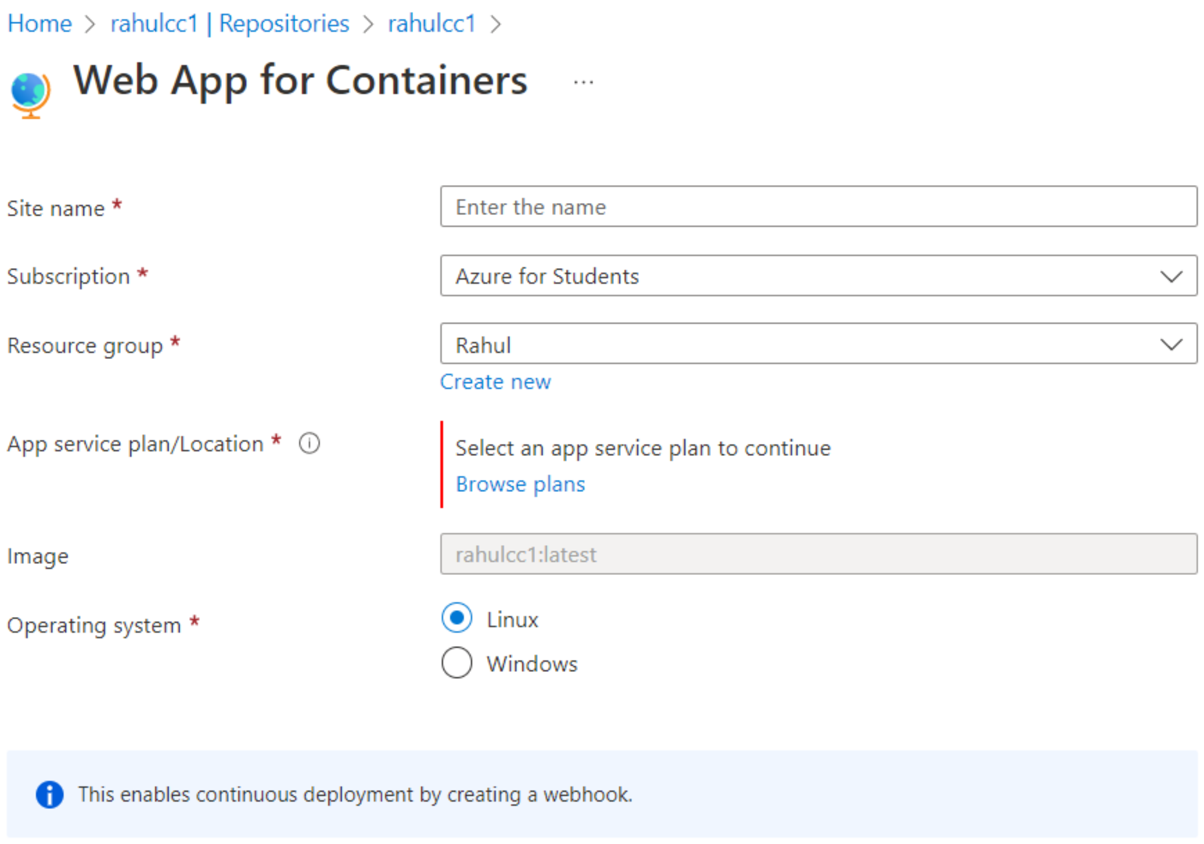
Last updated date : 10/29/2022, 9:30 PM GMT+5:30

Tag count : 1

Manifest count : 2

Tags ↑↓	Digest ↑↓	Last modified
latest	sha256:cdb313fd672cc6138d9cc599544fd9351abb856244ffdbefd...	10/29/2022, 9:30 PM GMT+5:30

We get this page for Deploying to web app



Home > rahulcc1 | Repositories > rahulcc1 >

Web App for Containers ...

Site name * Enter the name

Subscription * Azure for Students

Resource group * Rahul

Create new

App service plan/Location * ⓘ Select an app service plan to continue

Browse plans

Image rahulcc1:latest

Operating system * ● Linux ○ Windows

ⓘ This enables continuous deployment by creating a webhook.

We can give the site name and the link is generated

Site Name given are as follow:

s20200010212service1 for service1

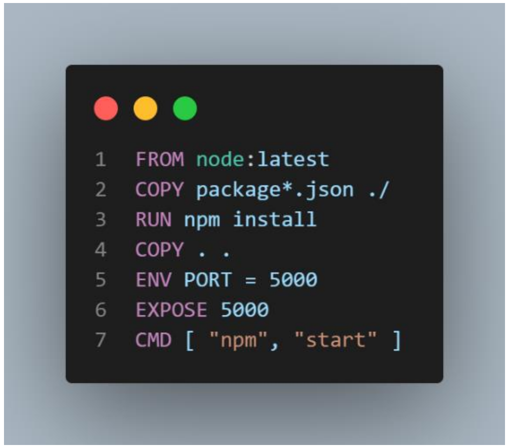
s20200010212service2 for service2

Then the Link Generated are as follows:

SERVICE 1: https://s20200010212service1.azurewebsites.net

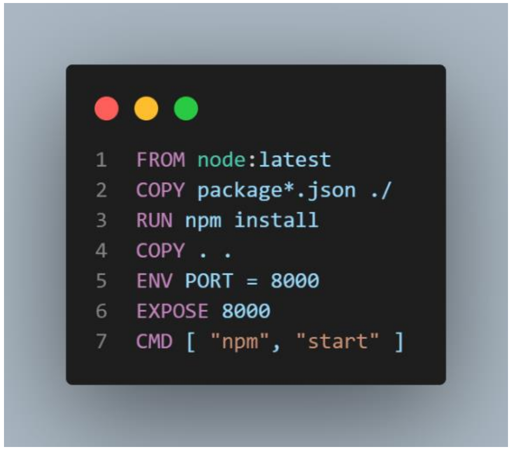
SERVICE 2: https://s20200010212service2.azurewebsites.net

Docker File Used for service 1 is:



```
FROM node:latest
COPY package*.json ./
RUN npm install
COPY . .
ENV PORT = 5000
EXPOSE 5000
CMD [ "npm", "start" ]
```

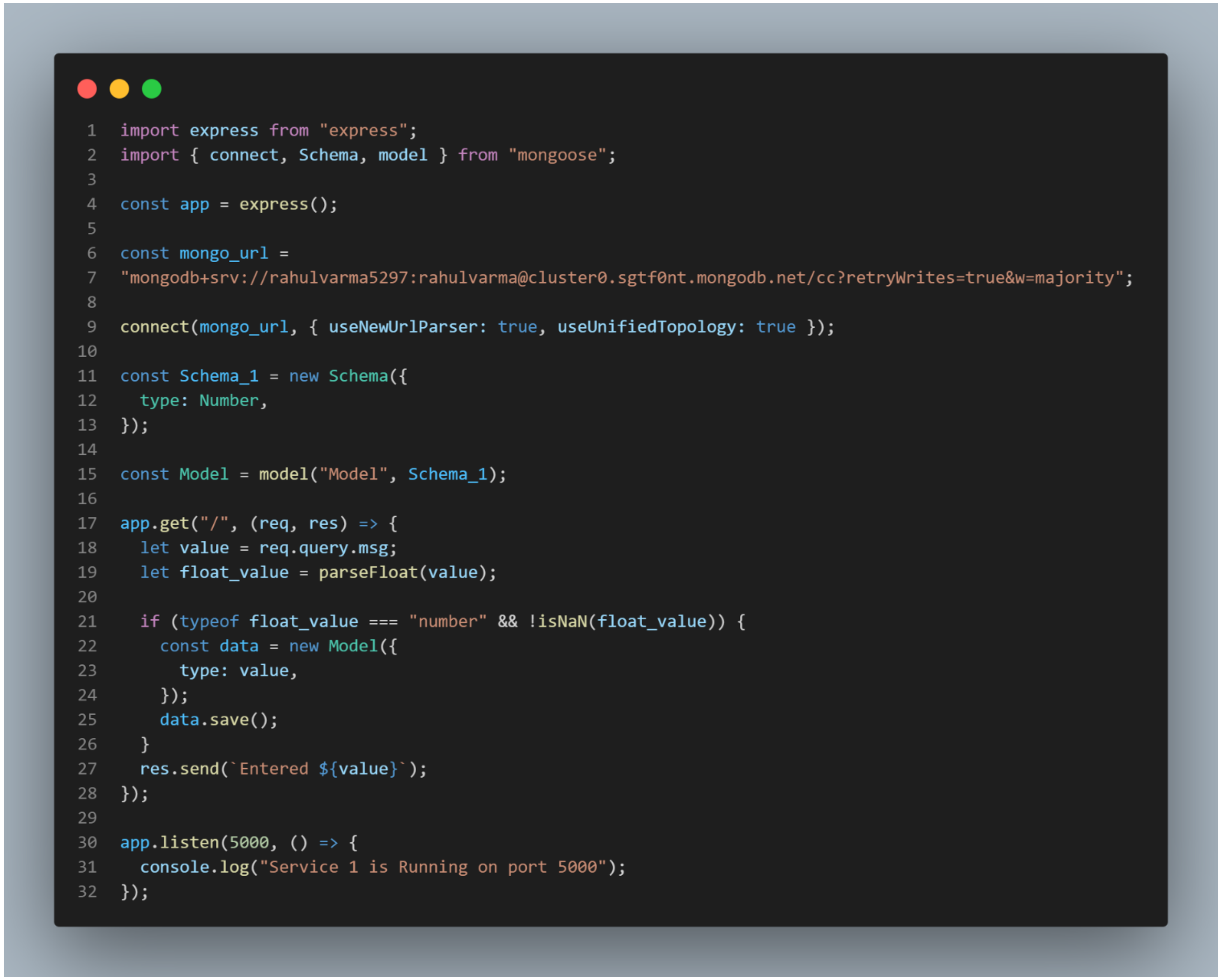
Docker File Used for service 2 is:



```
FROM node:latest
COPY package*.json ./
RUN npm install
COPY . .
ENV PORT = 8000
EXPOSE 8000
CMD [ "npm", "start" ]
```

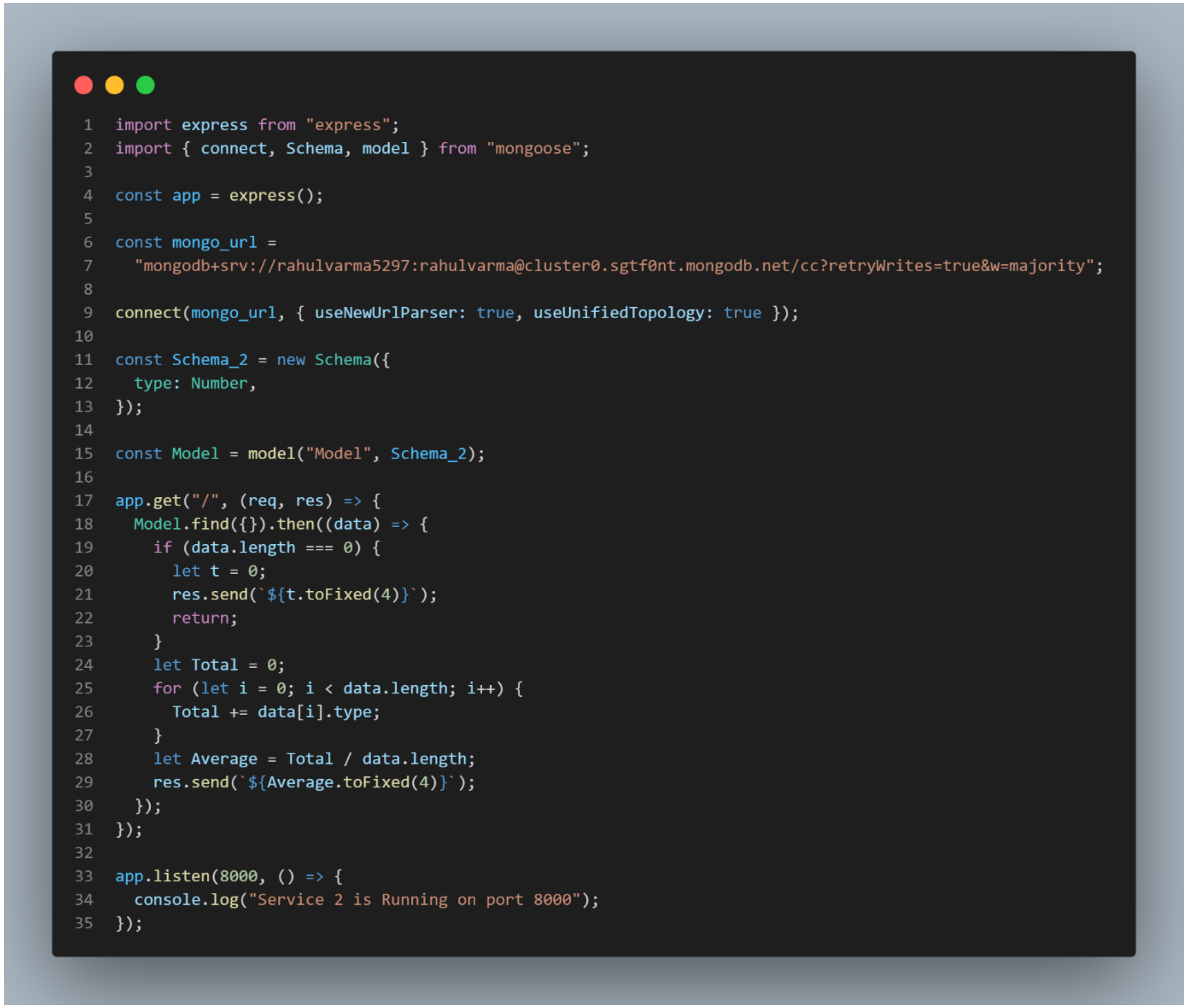
Full Code for service1 and service 2 is shown in the next two pages using node, express, mongoDB.

Index.js for service1:



```
import express from "express";
import { connect, Schema, model } from "mongoose";

const app = express();

const mongo_url =
"mongodb+srv://rahulvarma5297:rahulvarma@cluster0.sgtf0nt.mongodb.net/cc?retryWrites=true&w=majority";

connect(mongo_url, { useNewUrlParser: true, useUnifiedTopology: true });

const Schema_1 = new Schema({
  type: Number,
});

const Model = model("Model", Schema_1);

app.get("/", (req, res) => {
  let value = req.query.msg;
  let float_value = parseFloat(value);

  if (typeof float_value === "number" && !isNaN(float_value)) {
    const data = new Model({
      type: value,
    });
    data.save();
  }
  res.send(`Entered ${value}`);
});

app.listen(5000, () => {
  console.log("Service 1 is Running on port 5000");
});
```

Index.js for service 2:



```
import express from "express";
import { connect, Schema, model } from "mongoose";

const app = express();

const mongo_url =
  "mongodb+srv://rahulvarma5297:rahulvarma@cluster0.sgtf0nt.mongodb.net/cc?retryWrites=true&w=majority";

connect(mongo_url, { useNewUrlParser: true, useUnifiedTopology: true });

const Schema_2 = new Schema({
  type: Number,
});

const Model = model("Model", Schema_2);

app.get("/", (req, res) => {
  Model.find({}).then((data) => {
    if (data.length === 0) {
      let t = 0;
      res.send(`${t.toFixed(4)}`);
      return;
    }
    let Total = 0;
    for (let i = 0; i < data.length; i++) {
      Total += data[i].type;
    }
    let Average = Total / data.length;
    res.send(`${Average.toFixed(4)}`);
  });
});

app.listen(8000, () => {
  console.log("Service 2 is Running on port 8000");
});
```

CREATED TWO SEPARATE SERVICES, CONNECTED TO THE SAME BACKEND DATABASE.

For Evaluation:

DATABASE IS CLEARED.

Enter Data Using GET Parameter

Example: https://s20200010212service1.azurewebsites.net/?msg=10

And add another as msg=20

Check in the service 2 link (ie. https://s20200010212service2.azurewebsites.net)

Average value will be 15.

← THANK YOU →