

# Deploying a website using AWS.

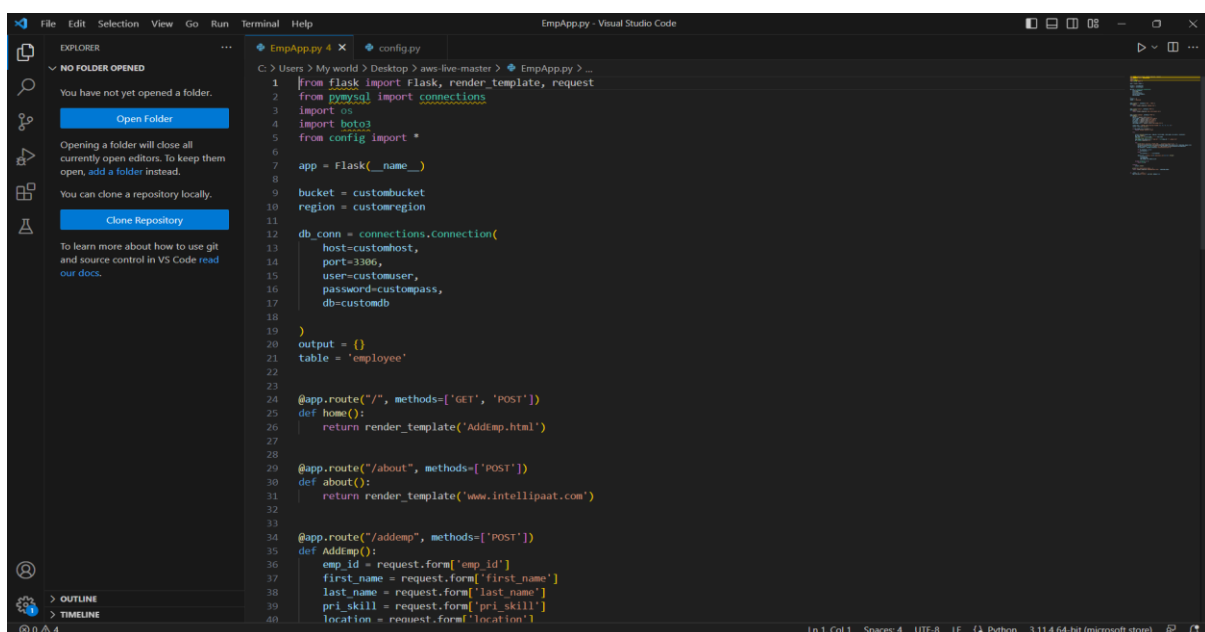
**AIM:** Deploying a simple website created using the python and then deploying it using the AWS console where the data entered by the end user in the website is collected and stored in the RDS and S3.

**RESOURCES USED:** RDS, S3, EC2, IAM.

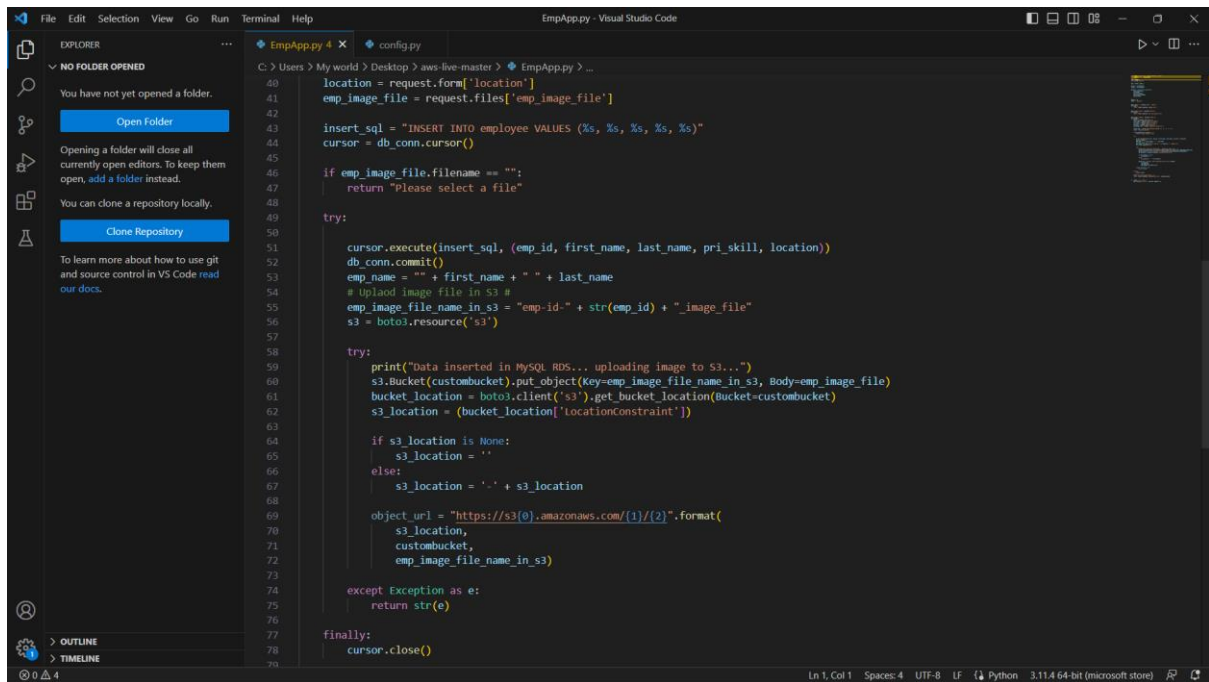
**STEP 1:** Create the RDS, S3, and EC2 with the required names and the configuration.

Where **Ubuntu** is used as the AMI for the Ec2 machine.

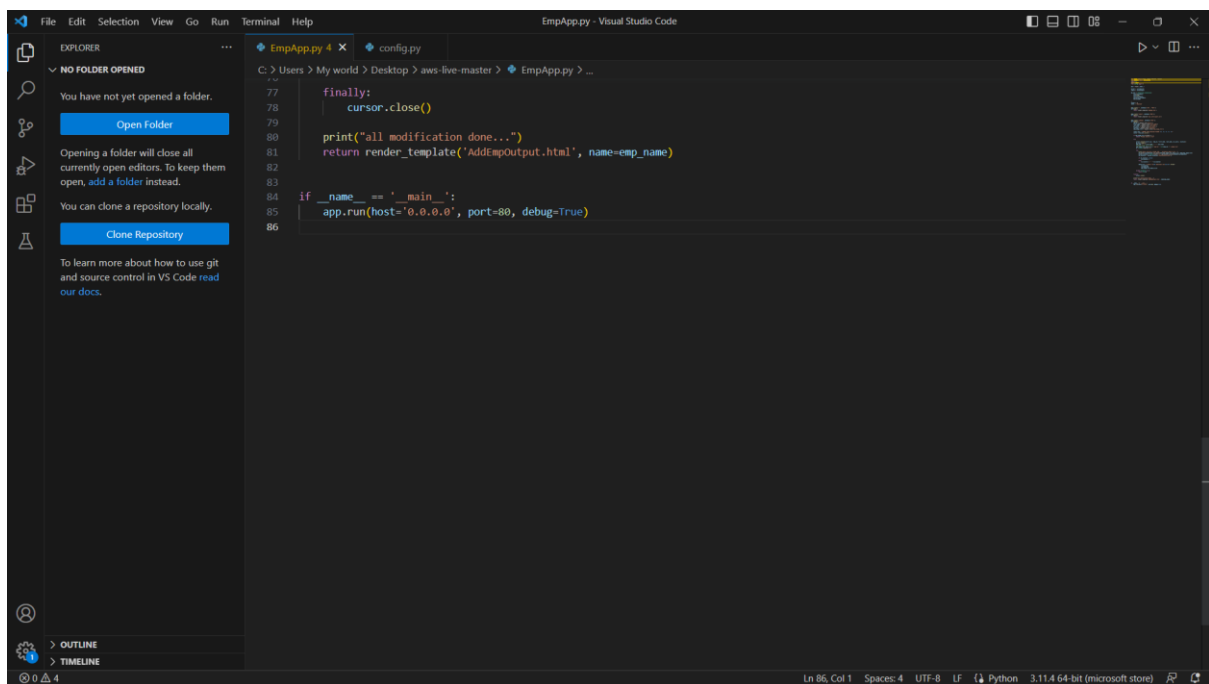
**STEP 2:** Write the code required for the website using the front end and back end using the flask module and name it as the **EmpApp.py** file as it is written using the python3.



```
1 from flask import Flask, render_template, request
2 from pymysql import connections
3 import os
4 import boto3
5 from config import *
6
7 app = Flask(__name__)
8
9 bucket = custombucket
10 region = customregion
11
12 db_conn = connections.Connection(
13     host=customhost,
14     port=3306,
15     user=customuser,
16     password=custompass,
17     db=customdb
18 )
19
20 output = {}
21 table = 'employee'
22
23
24 @app.route("/", methods=['GET', 'POST'])
25 def home():
26     return render_template("AddEmp.html")
27
28
29 @app.route("/about", methods=['POST'])
30 def about():
31     return render_template("www.intellipaat.com")
32
33
34 @app.route("/addemp", methods=['POST'])
35 def AddEmp():
36     emp_id = request.form['emp_id']
37     first_name = request.form['first_name']
38     last_name = request.form['last_name']
39     pri_skill = request.form['pri_skill']
40     location = request.form['location']
```



```
File Edit Selection View Go Run Terminal Help EmpApp.py - Visual Studio Code
EMPLOYER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
You can clone a repository locally.
Clone Repository
To learn more about how to use git and source control in VS Code read our docs.
C:\Users\My world\Desktop> aws-live-master > EmpApp.py > ...
40 location = request.form['location']
41 emp_image_file = request.files['emp_image_file']
42
43 insert_sql = "INSERT INTO employee VALUES (%s, %s, %s, %s, %s)"
44 cursor = db_conn.cursor()
45
46 if emp_image_file.filename == "":
47     return "Please select a file"
48
49 try:
50
51     cursor.execute(insert_sql, (emp_id, first_name, last_name, pri_skill, location))
52     db_conn.commit()
53     emp_name = "" + first_name + " " + last_name
54     # Upload image file in S3 #
55     emp_image_file_name_in_s3 = "emp-id-" + str(emp_id) + "_image_file"
56     s3 = boto3.resource('s3')
57
58     try:
59         print("Data inserted in MySQL RDS... uploading image to S3...")
60         s3.Bucket(custombucket).put_object(Key=emp_image_file_name_in_s3, Body=emp_image_file)
61         bucket_location = boto3.client('s3').get_bucket_location(bucket=custombucket)
62         s3_location = (bucket_location['locationConstraint'])
63
64         if s3_location is None:
65             s3_location = ''
66         else:
67             s3_location = '.' + s3_location
68
69         object_url = "https://s3[0].amazonaws.com/[1]/[2]".format(
70             s3_location,
71             custombucket,
72             emp_image_file_name_in_s3)
73
74     except Exception as e:
75         return str(e)
76
77 finally:
78     cursor.close()
79
```



```
File Edit Selection View Go Run Terminal Help EmpApp.py - Visual Studio Code
EMPLOYER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
You can clone a repository locally.
Clone Repository
To learn more about how to use git and source control in VS Code read our docs.
C:\Users\My world\Desktop> aws-live-master > EmpApp.py > ...
77 finally:
78     cursor.close()
79
80 print("all modification done...")
81 return render_template("AddEmpOutput.html", name=emp_name)
82
83
84 if __name__ == '__main__':
85     app.run(host="0.0.0.0", port=80, debug=True)
86
```

Code:

```
from flask import Flask, render_template, request
from pymysql import connections
import os
import boto3
from config import *

app = Flask(__name__)
```

```

bucket = custombucket
region = customregion

db_conn = connections.Connection(
    host=customhost,
    port=3306,
    user=customuser,
    password=custompass,
    db=customdb
)

output = {}
table = 'employee'

@app.route("/", methods=['GET', 'POST'])
def home():
    return render_template('AddEmp.html')

@app.route("/about", methods=['POST'])
def about():
    return render_template('www.intellipaat.com')

@app.route("/addemp", methods=['POST'])
def AddEmp():
    emp_id = request.form['emp_id']
    first_name = request.form['first_name']
    last_name = request.form['last_name']
    pri_skill = request.form['pri_skill']
    location = request.form['location']
    emp_image_file = request.files['emp_image_file']

    insert_sql = "INSERT INTO employee VALUES (%s, %s, %s, %s, %s)"
    cursor = db_conn.cursor()

    if emp_image_file.filename == "":
        return "Please select a file"

    try:
        cursor.execute(insert_sql, (emp_id, first_name, last_name, pri_skill,
location))
        db_conn.commit()
        emp_name = "" + first_name + " " + last_name
        # Uplaod image file in S3 #
        emp_image_file_name_in_s3 = "emp-id-" + str(emp_id) + "_image_file"

```

```

s3 = boto3.resource('s3')

try:
    print("Data inserted in MySQL RDS... uploading image to S3...")
    s3.Bucket(custombucket).put_object(Key=emp_image_file_name_in_s3,
Body=emp_image_file)
    bucket_location =
boto3.client('s3').get_bucket_location(Bucket=custombucket)
    s3_location = (bucket_location['LocationConstraint'])

    if s3_location is None:
        s3_location = ''
    else:
        s3_location = '-' + s3_location

    object_url = "https://s3{0}.amazonaws.com/{1}/{2}".format(
        s3_location,
        custombucket,
        emp_image_file_name_in_s3)

except Exception as e:
    return str(e)

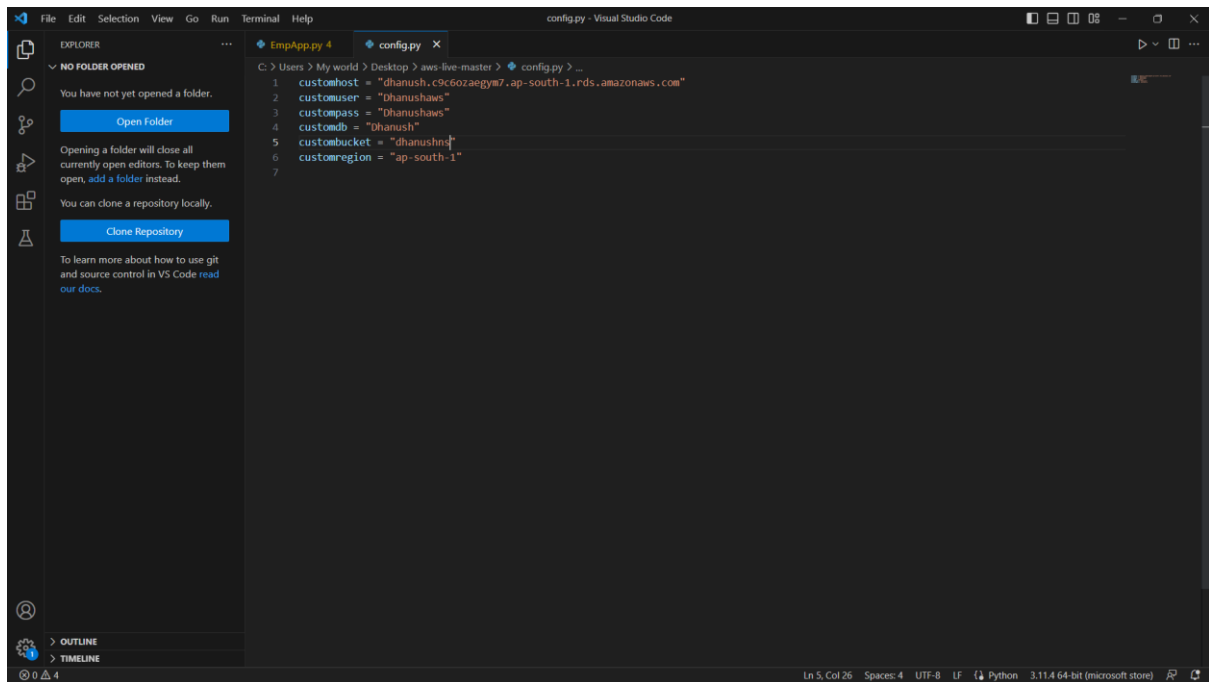
finally:
    cursor.close()

print("all modification done...")
return render_template('AddEmpOutput.html', name=emp_name)

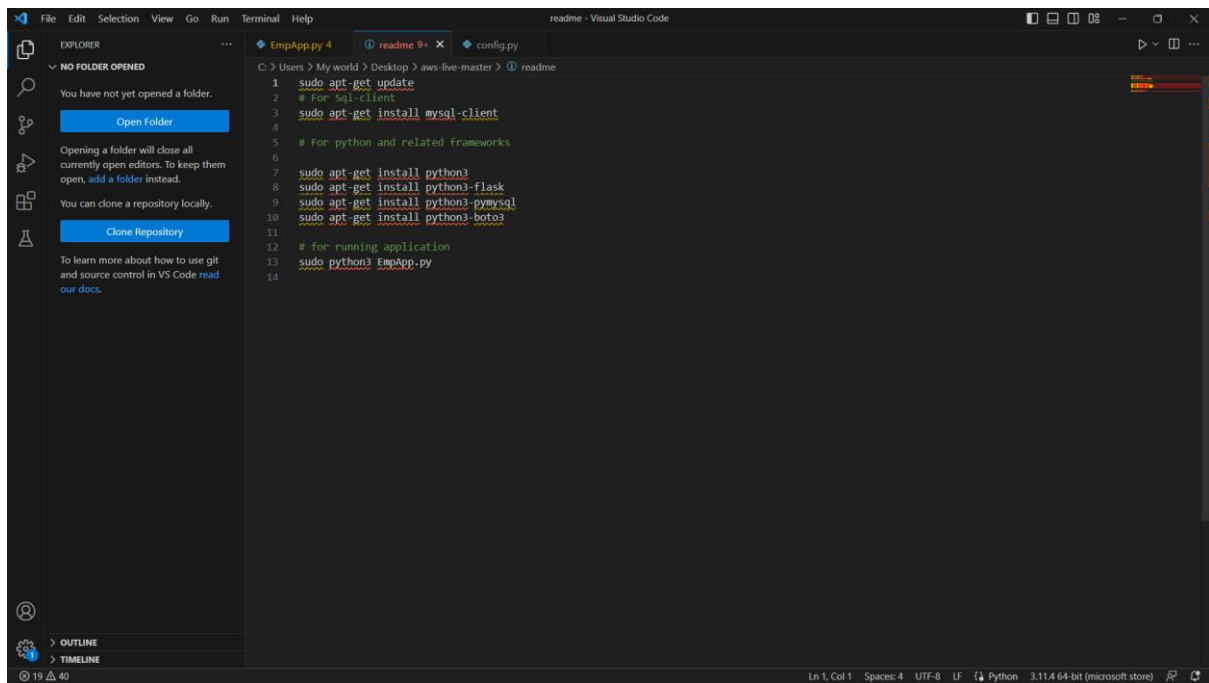
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=80, debug=True)

```

**STEP 4:** Add the Required Configuration file and specify the contents of the AWS usernames of the Databases, S3, region used ,and the password for the database.



**STEP 5:** Add a read me file which shows the required commands needs to be installed in the EC2 server.



**STEP 6:** Now using the **Putty** connect to the EC2 server and then run the following linux commands.

- **sudo apt-get update:** updates the machine
- **sudo apt-get install mysql-client:** installs mysql client in EC2

Connect to mysql Client:

- **mysql -h <midpoint of the RDS> -u <username while creating the RDS> -p <password of RDS>**
- Here: **mysql -h dhanush.c9c60zaegym7.ap-south-1.rds.amazonaws.com -u Dhanushaws -p Dhanushaws**

Now create the database and the table in the mysql using the following commands:

- **Create database Dhanush;** -> creates Dhanush database
- **Create table employee(<contents>);** -> creates table employee

```
ubuntu@ip-172-31-5-205: ~$ mysql
Copyright (c) 2000, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| employee |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> clear
mysql>
mysql>
mysql> create database Dhanush;
Query OK, 1 row affected (0.00 sec)

mysql> use Dhanush;
Database changed
mysql> show databases;
+-----+
| Database |
+-----+
| Dhanush |
| employee |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
6 rows in set (0.00 sec)

mysql> use Dhanush;
Database changed
mysql> create table employee(
    -> empid varchar(20),
    -> fname varchar(20),
    -> lname varchar(20),
    -> pri_skill varchar(20),
    -> location varchar(20));
Query OK, 0 rows affected (0.02 sec)

mysql>
```

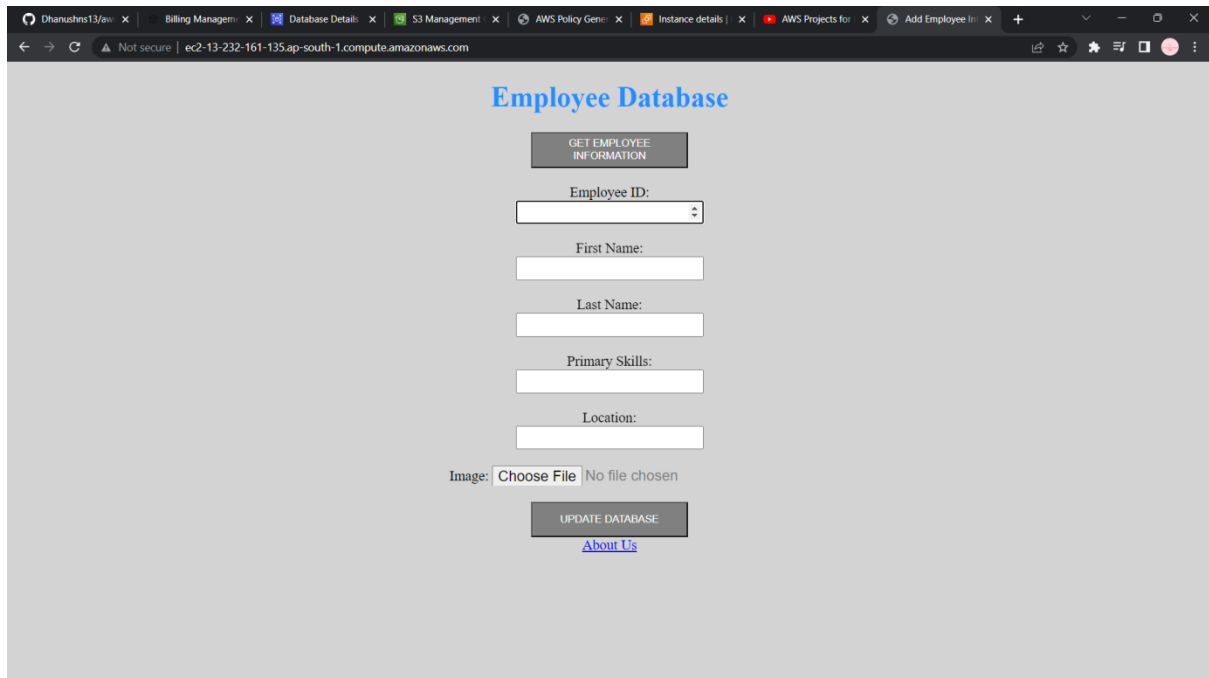
**STEP 7:** Now install the pre requisite commands(from Readme.file) for the code to run in the EC2 machine:

- `sudo apt-get install python3`
- `sudo apt-get install python3-pip`:used to install req packages
- `sudo apt-get install python3-flask`
- `suod apt-get install python3-boto3`:provides SDK for python

## **OUTPUT:**

**STEP 8:** Now run the python Command to run the Website in the Live page:

- `sudo python3 EmpApp.py`
- Use the EC2 Public Ipv4 address in the URL to open the website.



Employee Database

GET EMPLOYEE INFORMATION

Employee ID:

First Name:

Last Name:

Primary Skills:

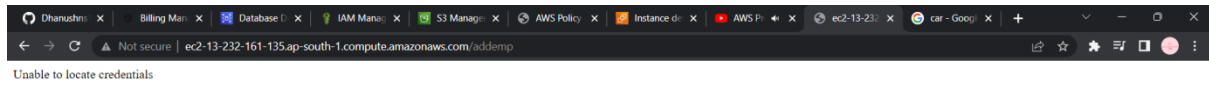
Location:

Image: Choose File No file chosen

UPDATE DATABASE

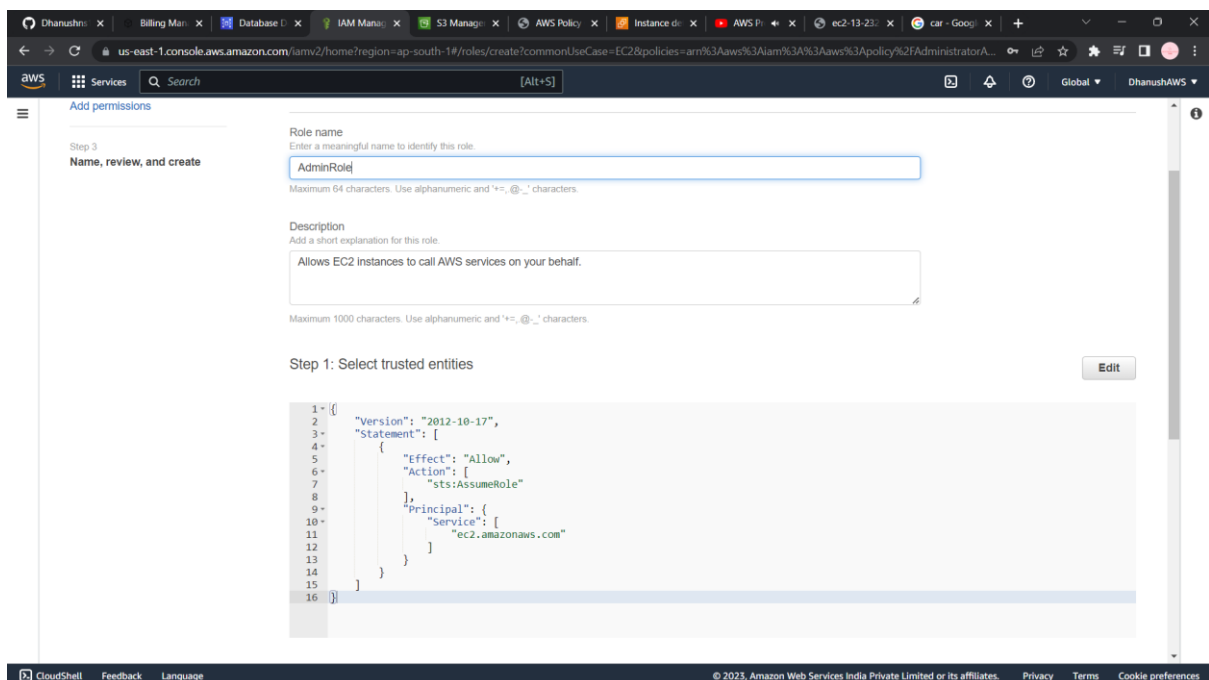
[About Us](#)

Here when u click the update database u will have an error as the EC2 needs to have access to S3



This can be overcome with the help of **IAM Roles**

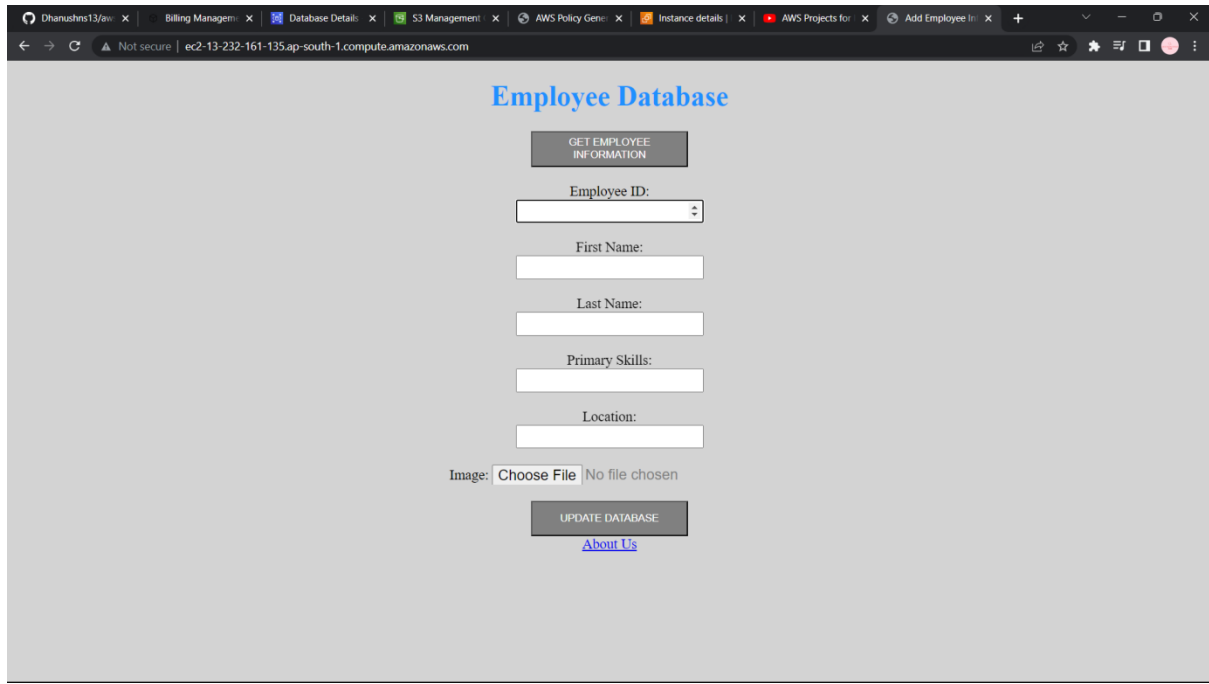
**STEP 9:** Create the IAM role using the IAM service where the EC2 can interact with the S3 Service





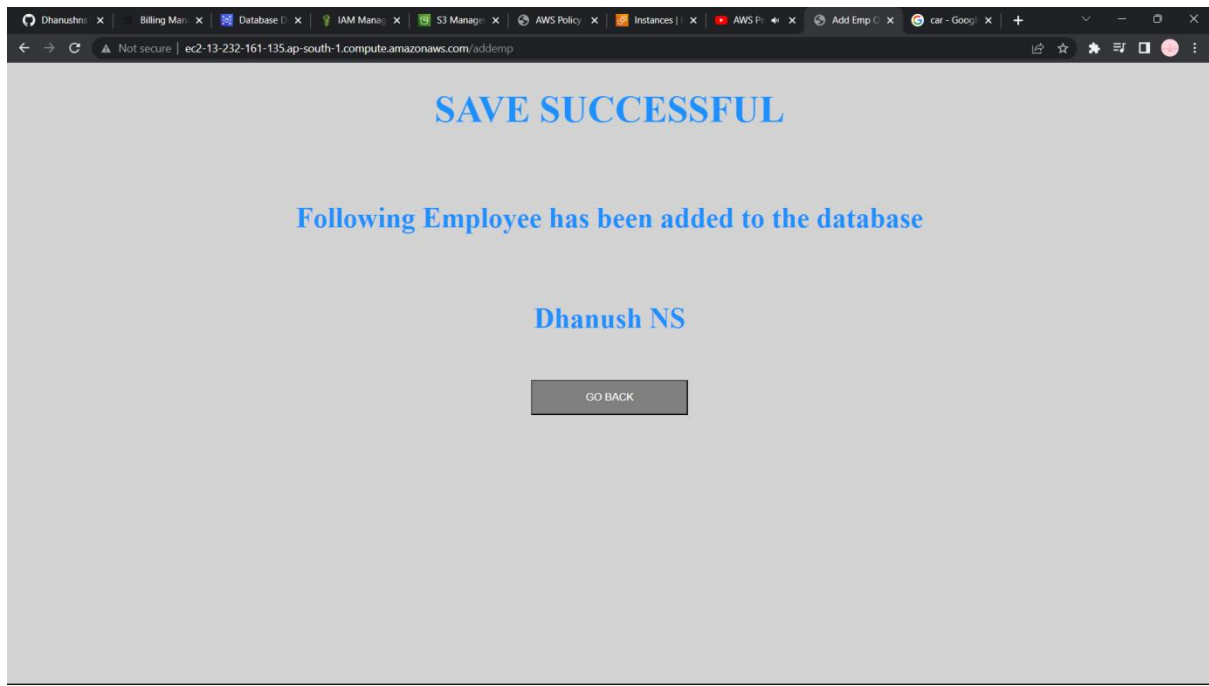
**Note:** We should also disable the **Block Public Access** and assign the **S3 policy** to **Get Object** to the S3 bucket so that we can view the file uploaded by the end user using the website.

## FINAL OUTPUT:



The screenshot shows a web browser window with the URL `ec2-13-232-161-135.ap-south-1.compute.amazonaws.com`. The page title is "Employee Database". It features a form with the following fields and buttons:

- GET EMPLOYEE INFORMATION** (button)
- Employee ID:** (text input field)
- First Name:** (text input field)
- Last Name:** (text input field)
- Primary Skills:** (text input field)
- Location:** (text input field)
- Image:** (file upload area with "Choose File" button and "No file chosen" text)
- UPDATE DATABASE** (button)
- [About Us](#) (link)



The screenshot shows the same web browser window after a successful save operation. The page displays a confirmation message:

- SAVE SUCCESSFUL** (large blue text)
- Following Employee has been added to the database** (blue text)
- Dhanush NS** (blue text)
- GO BACK** (button)

## DATA COLLECTED USING RDS AND S3:

- The data is collected and stored in the RDS and S3 and can be viewed in the database using:
- **Use Dhanush;** -> changes to Dhanush database
- **Select \* from employee;** -> shows data from the table created and data uploaded by the end user

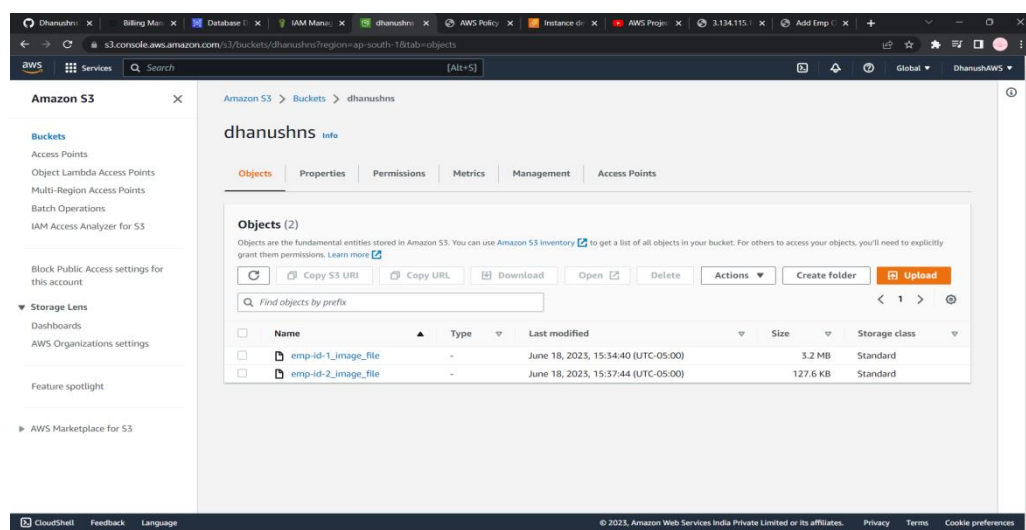
```
mysql> use Dhanush;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

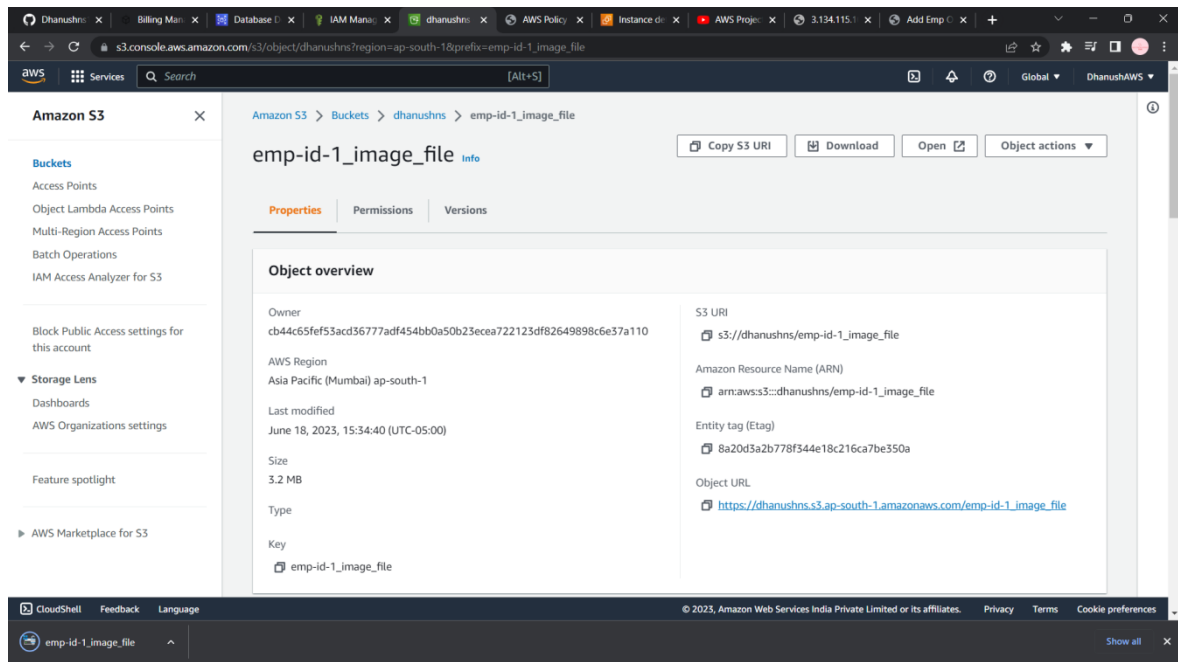
Database changed
mysql> select * from employee;
+-----+-----+-----+-----+-----+
| empid | fname | lname | pri_skill | location |
+-----+-----+-----+-----+-----+
| 1     | Dhanush | NS    | AWS      | Chennai  |
| 1     | Dhanush | NS    | AWS      | Chennai  |
| 2     | Spandana | Nalla | Web-Dev  | Andhra   |
| 2     | Spandana | Nalla | Web-Dev  | Andhra   |
| 2     | Spandana | Nalla | Web-Dev  | Andhra   |
| 2     | Spandana | Nalla | Web-Dev  | Andhra   |
| 1     | Dhanush | NS    | AWS      | Chennai  |
| 3     | Gopi   | K     | AWS      | Pune     |
| 4     | Ajay   | D     | Python   | Chennai  |
| 5     | Shehanaz | Sk    | Dev      | Andhra   |
+-----+-----+-----+-----+-----+
10 rows in set (0.01 sec)

mysql>
```

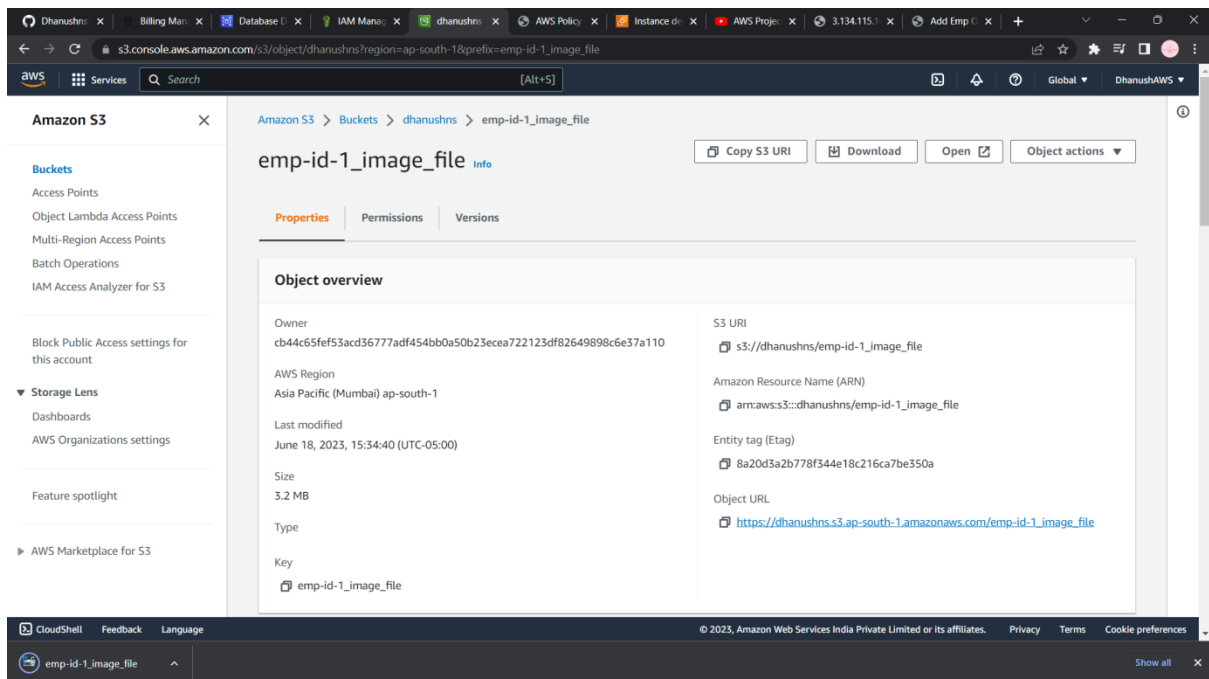
Uploading multiple times leads to versioning of the data in RDS.

## VIEWING OF S3 OBJECTS THAT ARE UPLOADED:

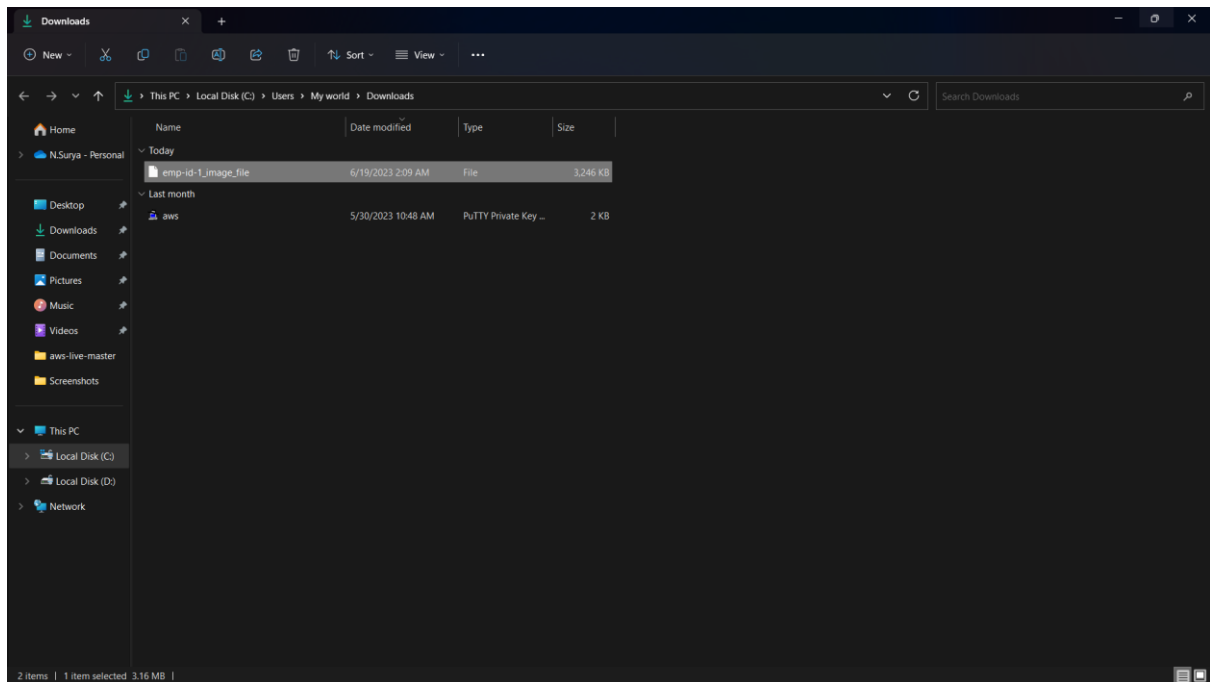




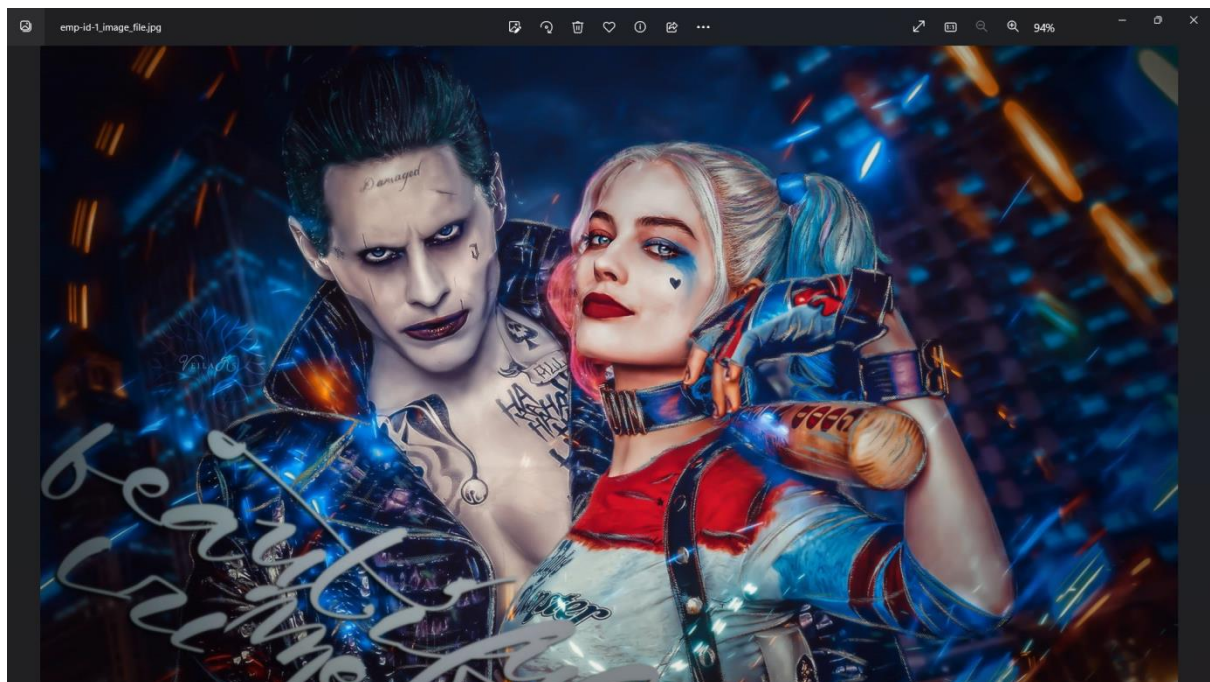
Here the **Object URL** is used to View the Object Uploaded by end user



Here it gets downloaded with the emp\_id of the user.



We can change the **format to jpg** and view the image



THIS IS THE IMAGE UPLOADED BY THE USER.

**RESULT:** The code and commands are successfully executed and the Data and images have been successfully stored in the database and S3.