# CloudChatter - AWS Deployment Guide

This document provides a step-by-step guide to building and deploying the CloudChatter Flask chat application on AWS using EC2, RDS (MySQL), and S3.

## Step 1: Create the Flask Project in VS Code

1. Open VS Code and create a folder named 'chatapp'.  
2. Inside 'chatapp', create the following files:  
 - app.py (main Flask app)

from flask import Flask, render\_template, request, redirect

import mysql.connector

from datetime import datetime

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

# MySQL Database Connection (AWS RDS)

db = mysql.connector.connect(

    host="chatdb-instance.cpoguyq0syid.eu-north-1.rds.amazonaws.com",

    user="admin",

    password="venkatasrisatyavarshini",

    database="chatdb"

)

cursor = db.cursor()

# Home route - show messages

@app.route('/')

def home():

    cursor.execute("SELECT sender, message, timestamp FROM messages ORDER BY timestamp DESC")

    messages = cursor.fetchall()

    return render\_template("index.html", messages=messages)

# Route to send a message

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

def send():

    sender = request.form['sender']

    message = request.form['message']

    timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

    cursor.execute(

        "INSERT INTO messages (sender, message, timestamp) VALUES (%s, %s, %s)",

        (sender, message, timestamp)

    )

    db.commit()

    return redirect('/')

if \_\_name\_\_ == '\_\_main\_\_':

    # Run on all interfaces so EC2 can serve it

    app.run(host='0.0.0.0', port=5000, debug=True)

- templates/layout.html (HTML layout template)

<!DOCTYPE html>

<html>

<head>

    <title>Simple Chat App</title>

    <style>

        body { font-family: Arial, sans-serif; background-color: #f5f5f5; padding: 20px; }

        .chat-box { background: white; padding: 20px; border-radius: 10px; width: 500px; margin: auto; }

        .message { border-bottom: 1px solid #ddd; padding: 5px; }

        .timestamp { font-size: 0.8em; color: gray; }

        input, textarea { width: 100%; padding: 8px; margin: 5px 0; }

        button { padding: 10px 15px; background-color: #4CAF50; color: white; border: none; cursor: pointer; }

        button:hover { background-color: #45a049; }

    </style>

</head>

<body>

    <div class="chat-box">

        {% block content %}{% endblock %}

    </div>

</body>

</html>

- templates/index.html (chat UI template)

{% extends "layout.html" %}

{% block content %}

<h2>Chat Room</h2>

{% for sender, message, timestamp in messages %}

<div class="message">

    <strong>{{ sender }}</strong>: {{ message }}<br>

    <span class="timestamp">{{ timestamp }}</span>

</div>

{% endfor %}

<h3>Send a Message</h3>

<form method="POST" action="/send">

    <input type="text" name="sender" placeholder="Your name" required>

    <textarea name="message" placeholder="Type your message here..." required></textarea>

    <button type="submit">Send</button>

</form>

{% endblock %}

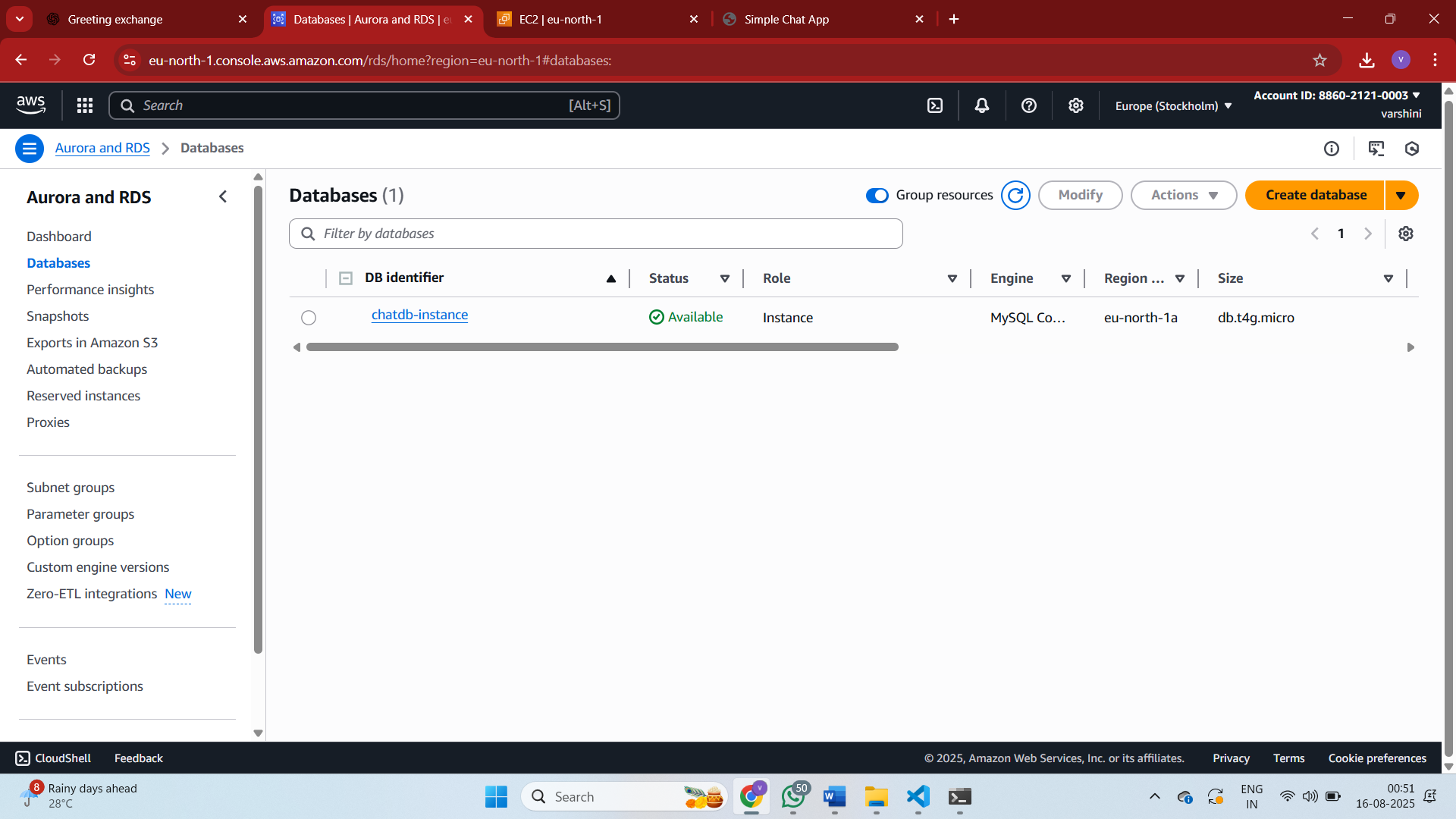
3. Install Flask locally for testing: pip install flask  
4. Run locally: python app.py

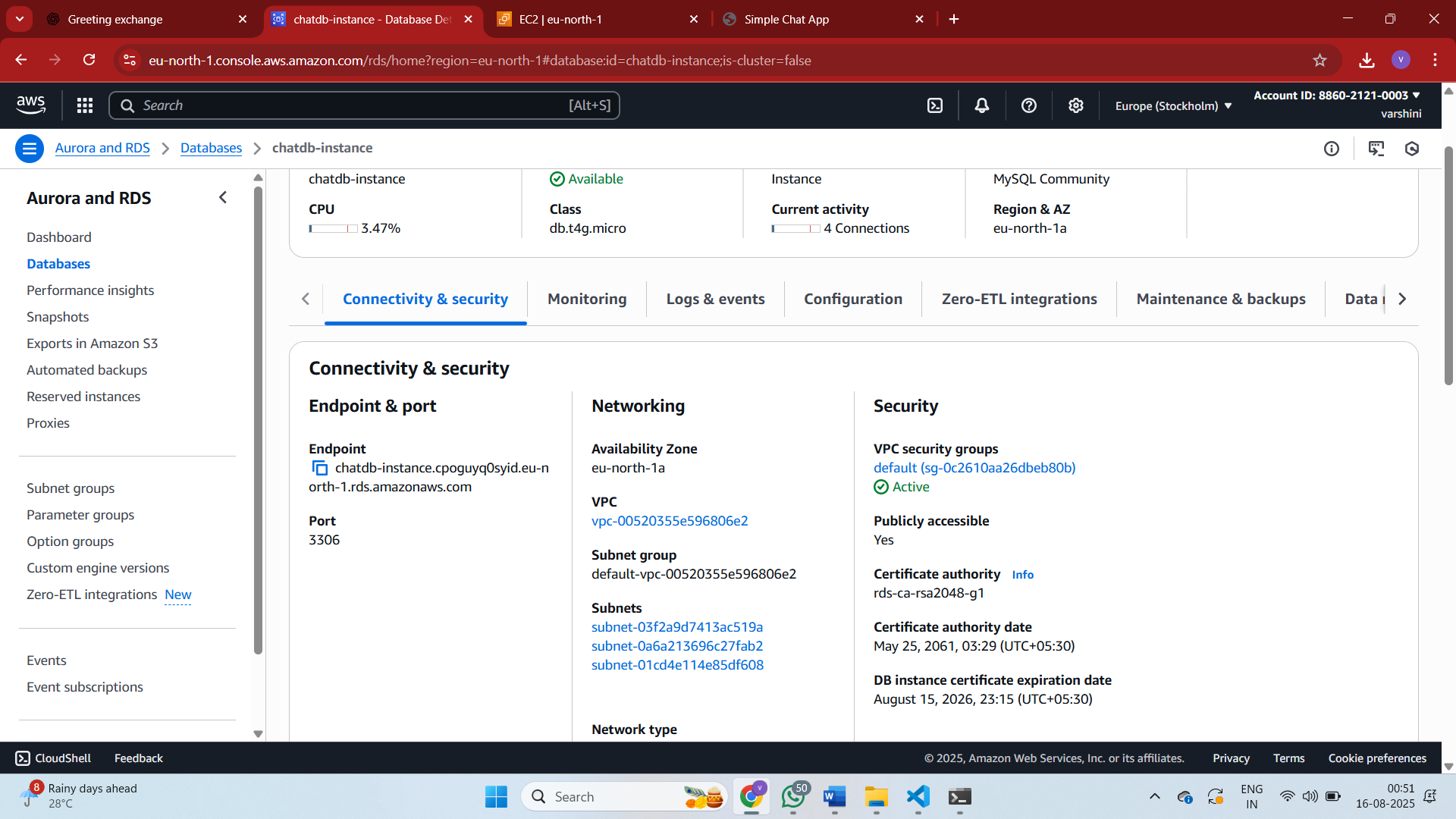
## Step 2: Create the Database Schema in MySQL Workbench

1. Open MySQL Workbench and connect to your local MySQL or AWS RDS instance.  
2. Run the following SQL commands:  
  
CREATE DATABASE IF NOT EXISTS chatdb;  
USE chatdb;  
DROP TABLE IF EXISTS messages;  
CREATE TABLE messages (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 sender VARCHAR(50) NOT NULL,  
 message TEXT NOT NULL,  
 timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP  
);

## Step 3: Create an AWS RDS MySQL Instance

1. Go to AWS Console → Search RDS → Create database.  
2. Choose 'Standard Create', MySQL engine, latest version.  
3. Template: Free tier.  
4. DB instance identifier: chatdb-instance, Master username: admin, enter password.  
5. DB instance class: db.t3.micro, storage: 20 GB.  
6. Public access: Yes, VPC: default.  
7. Initial database name: chatdb.  
8. Create database and wait until status is 'Available'.  
9. Edit the security group for the RDS to allow inbound MySQL/Aurora (3306) from:  
 - Your local IP (for Workbench)  
 - EC2 security group (for app connection)  
10. Copy the RDS endpoint from Connectivity & security tab.

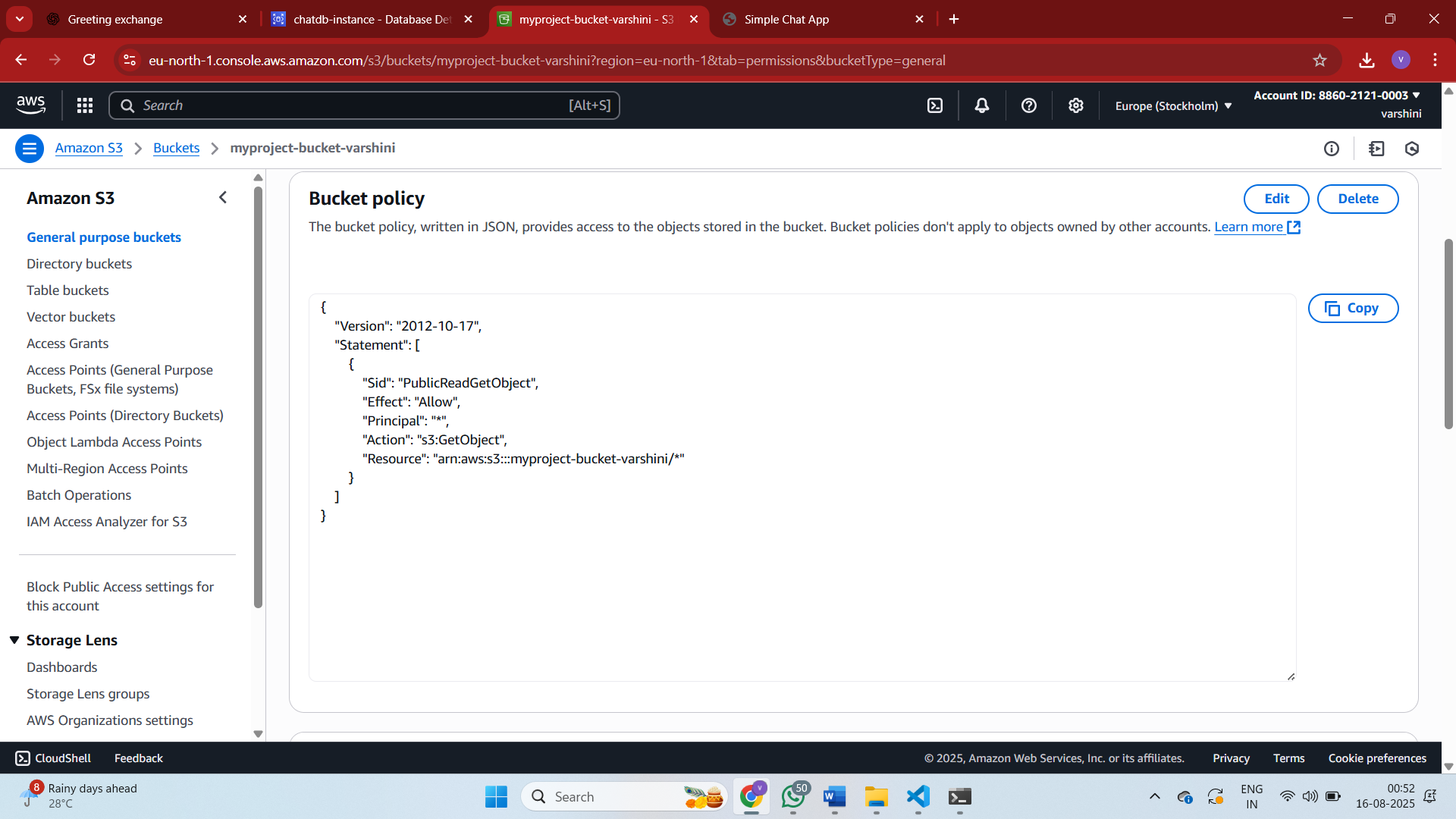


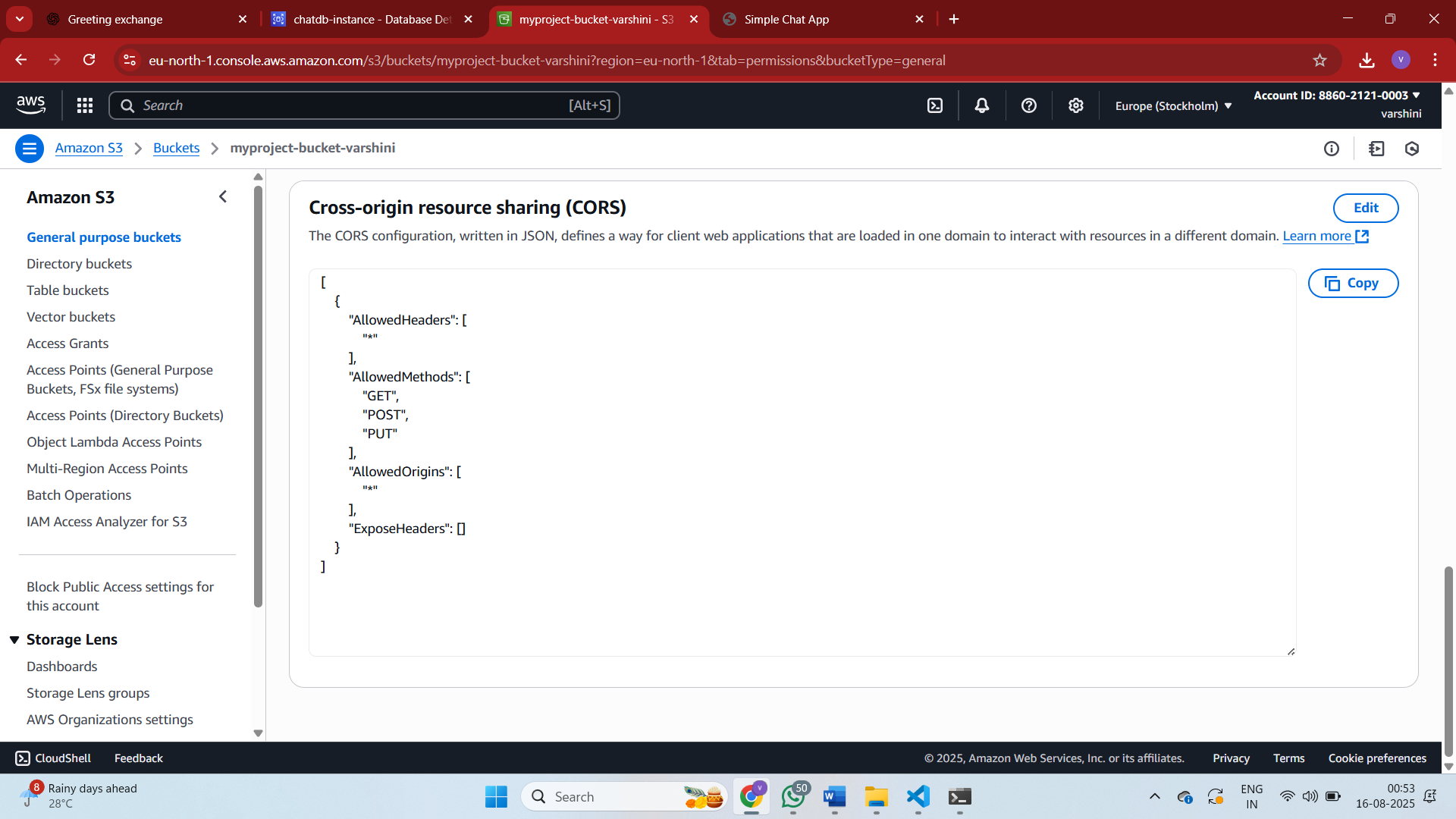


## Step 4: Create an S3 Bucket

1. Go to AWS Console → Search S3 → Create bucket.  
2. Bucket name: myproject-bucket-varshini (must be unique globally).  
3. Region: same as EC2 & RDS.

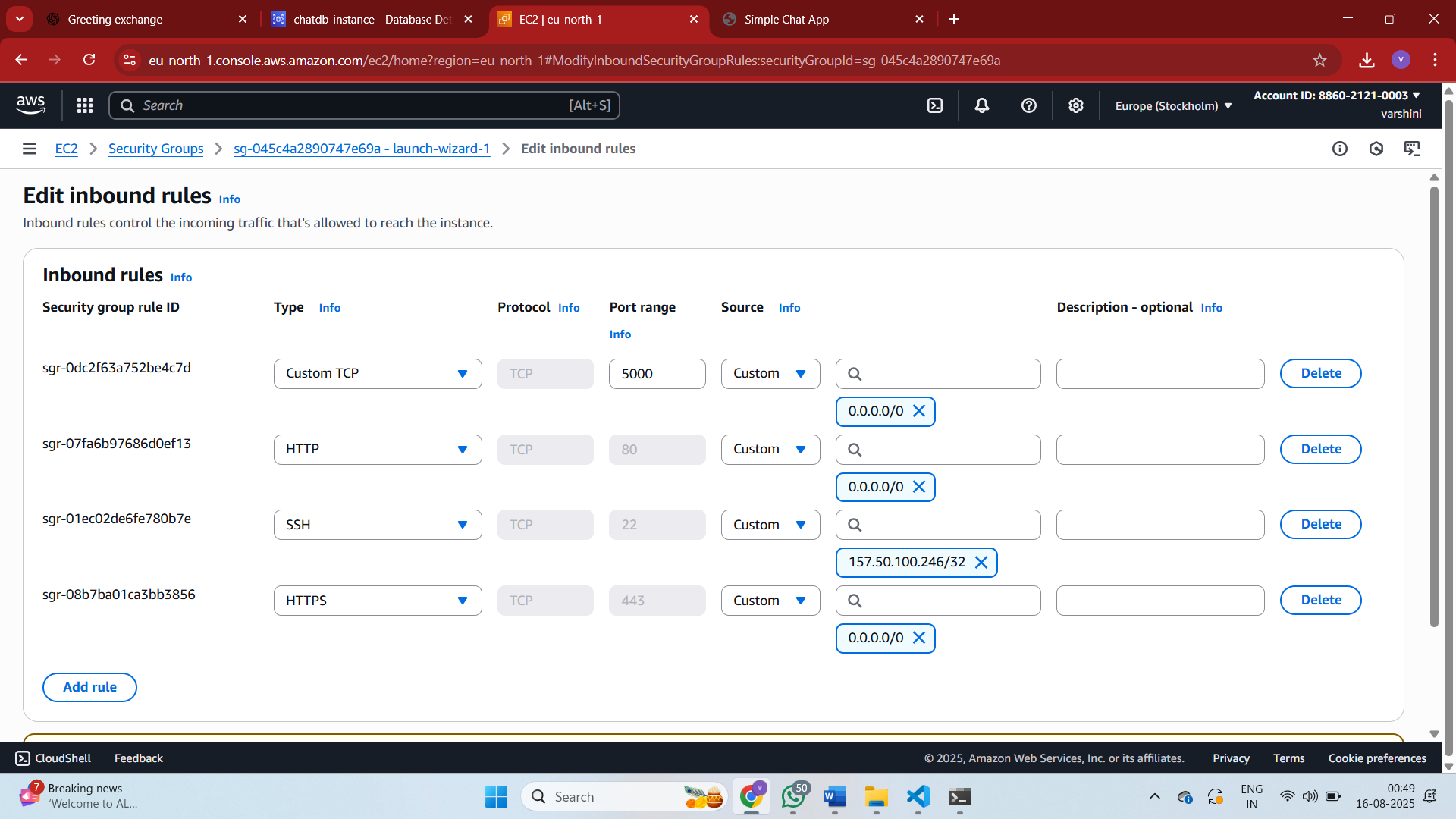
4. Block Public Access: keep ON if private.  
5. Create bucket.





## Step 5: Create an EC2 Instance

Go to AWS Console → Search EC2 → Launch instance.  
2. Name: cloudchatter-server.  
3. AMI: Ubuntu 22.04 LTS.  
4. Instance type: t2.micro.  
5. Key pair: Create new/download (e.g., myprojectserver-key.pem).  
6. Security group: allow HTTP(80), SSH(22), and Custom TCP port 5000 from 0.0.0.0/0.  
7. Launch instance.



## Step 6: Deploy Flask App to EC2

1. Open PowerShell on your local machine.  
2. Upload the project:  
scp -i "C:\Users\vvars\Downloads\myprojectserver-key.pem" -r "C:\Users\vvars\OneDrive\Desktop\chatapp" ubuntu@<EC2\_PUBLIC\_DNS>:~/  
3. Connect to EC2:  
ssh -i "C:\Users\vvars\Downloads\myprojectserver-key.pem" ubuntu@<EC2\_PUBLIC\_DNS>  
4. Inside EC2:  
 cd ~/chatapp  
 python3 -m venv venv  
 source venv/bin/activate  
 pip install flask boto3 mysql-connector-python  
5. Run the app:  
python app.py

## Step 7: Access the App in Browser

Open your browser and go to:  
http://<EC2\_PUBLIC\_DNS>:5000/  
Example:  
<http://ec2-13-53-193-23.eu-north-1.compute.amazonaws.com:5000/>

Link: <http://ec2-13-53-193-23.eu-north-1.compute.amazonaws.com:5000/>

1.Deploy to EC2 (PowerShell on your laptop → Ubuntu on server):

scp -i "C:\Users\vvars\Downloads\myprojectserver-key.pem" -r "C:\Users\vvars\OneDrive\Desktop\chatapp" [ubuntu@ec2-13-53-193-23.eu-north-1.compute.amazonaws.com:~/](mailto:ubuntu@ec2-13-53-193-23.eu-north-1.compute.amazonaws.com:~/)

2. **SSH into EC2** (PowerShell on your laptop):

ssh -i "C:\Users\vvars\Downloads\myprojectserver-key.pem" [ubuntu@ec2-13-53-193-23.eu-north-1.compute.amazonaws.com](mailto:ubuntu@ec2-13-53-193-23.eu-north-1.compute.amazonaws.com)

3. **Create venv & install deps** (inside EC2 Ubuntu terminal)

cd ~/chatapp

python3 -m venv venv

source venv/bin/activate

pip install flask mysql-connector-python boto3

4. **Run the app (inside EC2 Ubuntu terminal)**

python app.py

you will see links don’t press,becz it donot open on our laptop browser.

5. <http://ec2-13-53-193-23.eu-north-1.compute.amazonaws.com:5000/>

Public Dns of our EC2: ec2-13-53-193-23.eu-north-1.compute.amazonaws.com