# Project Documentation

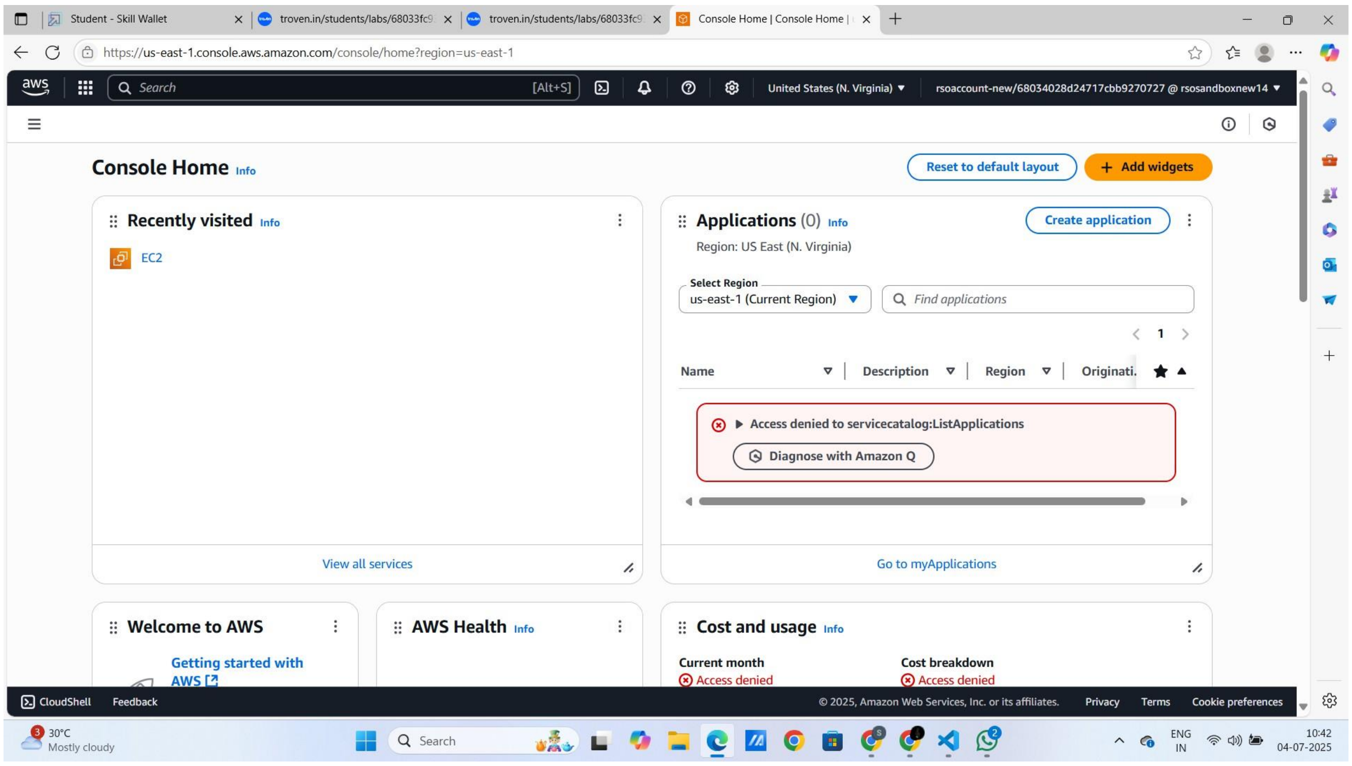
Title: MedTrack - Cloud-Enabled Healthcare Management System

Description: A web-based system built with Flask, hosted on AWS EC2, utilizing DynamoDB for backend storage and AWS SNS for real-time notifications.

Features include:

- User authentication and role-based access
- Appointment booking
- Diagnosis management by doctors
- DynamoDB integration for scalability
- EC2 instance deployment for hosting

## Screenshot 1



Access denied to servicecatalog:ListApplications

Diagnose with Amazon Q

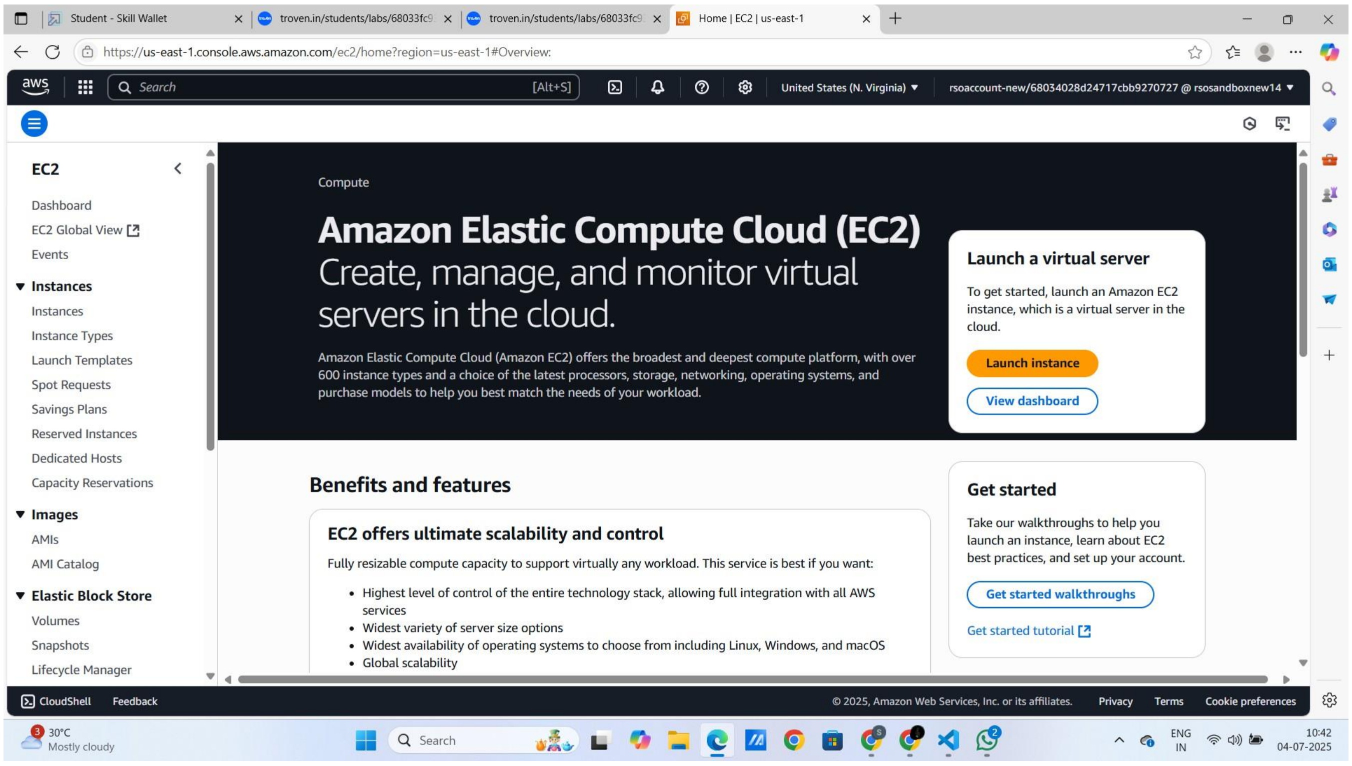
Current month
Access denied

Cost breakdown
Access denied

© 2025, Amazon Web Services, Inc. or its affiliates.

This screenshot shows the home page of aws console

## Screenshot 2



Student - Skill Wallet | troven.in/students/labs/68033fc9 | troven.in/students/labs/68033fc9 | Home | EC2 | us-east-1

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Overview:

United States (N. Virginia) ▼ rsoaccount-new/68034028d24717cbb9270727 @ rsosandboxnew14 ▼

EC2

Dashboard
EC2 Global View
Events

▼ Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations

▼ Images
AMIs
AMI Catalog

▼ Elastic Block Store
Volumes
Snapshots
Lifecycle Manager

Compute

**Amazon Elastic Compute Cloud (EC2)**
Create, manage, and monitor virtual servers in the cloud.

Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 600 instance types and a choice of the latest processors, storage, networking, operating systems, and purchase models to help you best match the needs of your workload.

**Launch a virtual server**

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance | View dashboard

**Benefits and features**

**EC2 offers ultimate scalability and control**

Fully resizable compute capacity to support virtually any workload. This service is best if you want:

- Highest level of control of the entire technology stack, allowing full integration with all AWS services
- Widest variety of server size options
- Widest availability of operating systems to choose from including Linux, Windows, and macOS
- Global scalability

**Get started**

Take our walkthroughs to help you launch an instance, learn about EC2 best practices, and set up your account.

Get started walkthroughs

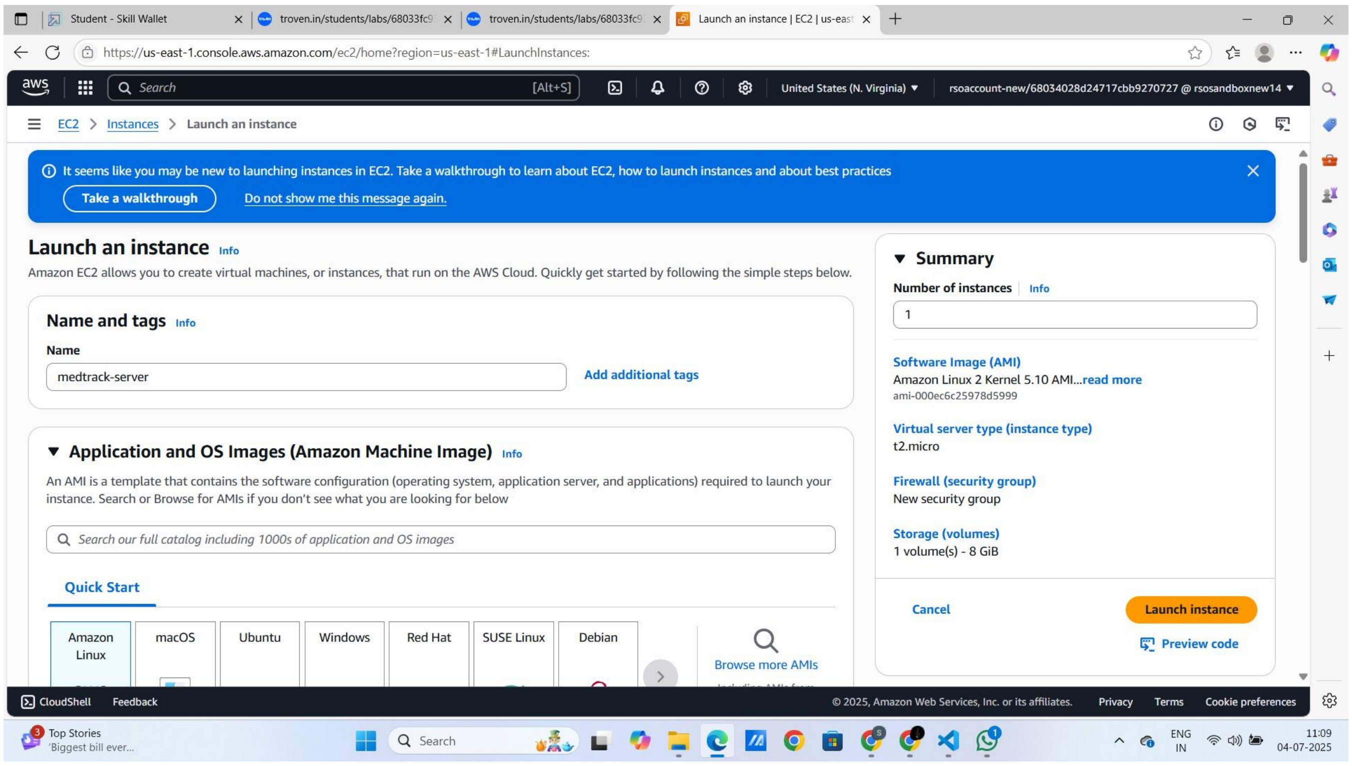
Get started tutorial

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

30°C Mostly cloudy | ENG IN | 10:42 04-07-2025

This  screenshot shows the home pages of EC2.

## Screenshot 3


It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: medtrack-server

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more ami-000ec6c25978d5999

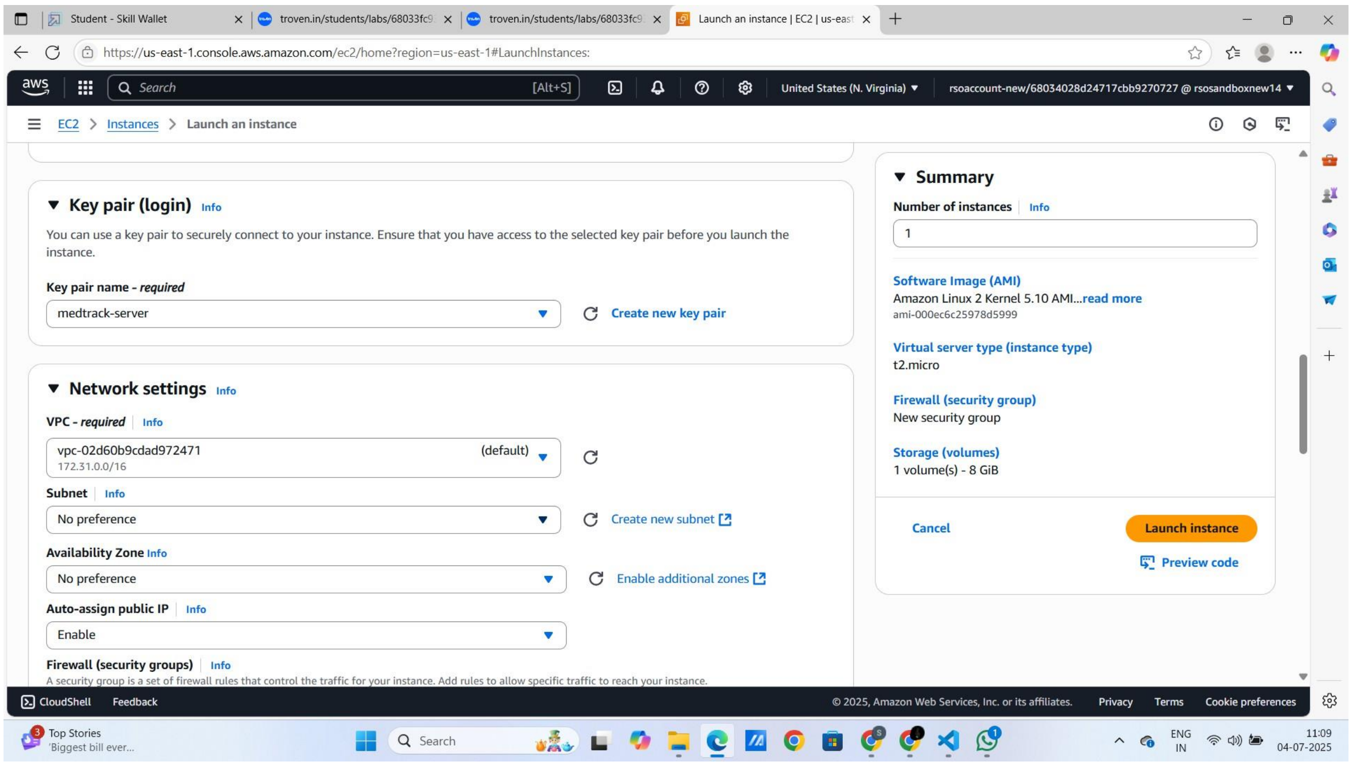
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

This screenshot shows Launching an instance

## Screenshot 4



Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*: medtrack-server

Network settings Info

VPC - *required*: vpc-02d60b9cdad972471 (default) 172.31.0.0/16

Subnet: No preference

Availability Zone: No preference

Auto-assign public IP: Enable

Firewall (security groups)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more ami-000ec6c25978d5999

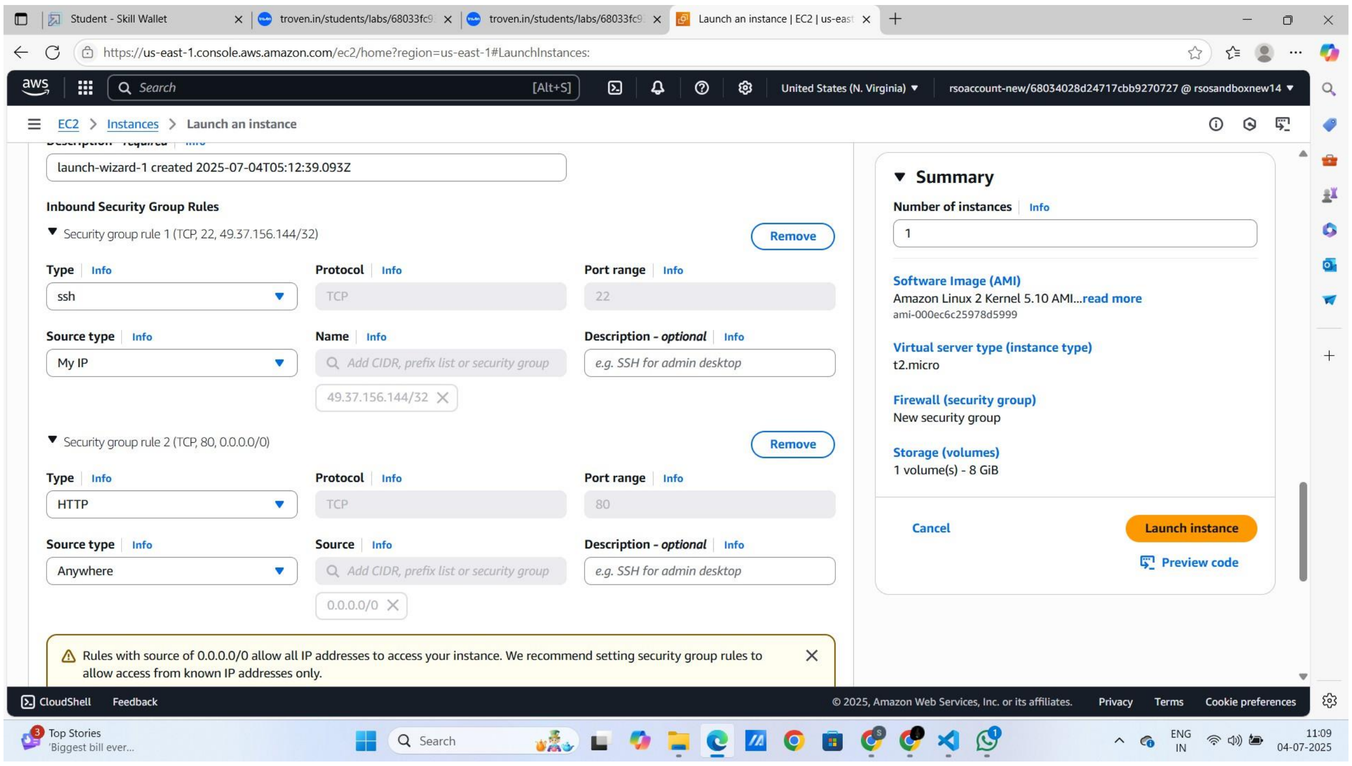
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

This screenshot shows the key pair, network settings in ec2 instance

## Screenshot 5



https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

EC2 > Instances > Launch an instance

launch-wizard-1 created 2025-07-04T05:12:39.093Z

**Inbound Security Group Rules**

▼ Security group rule 1 (TCP, 22, 49.37.156.144/32)

| Type | Protocol | Port range |
|---|---|---|
| ssh | TCP | 22 |

| Source type | Name | Description - optional |
|---|---|---|
| My IP | 49.37.156.144/32 | e.g. SSH for admin desktop |

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

| Type | Protocol | Port range |
|---|---|---|
| HTTP | TCP | 80 |

| Source type | Source | Description - optional |
|---|---|---|
| Anywhere | 0.0.0.0/0 | e.g. SSH for admin desktop |

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

**Summary**

Number of instances: 1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-000ec6c25978d5999

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

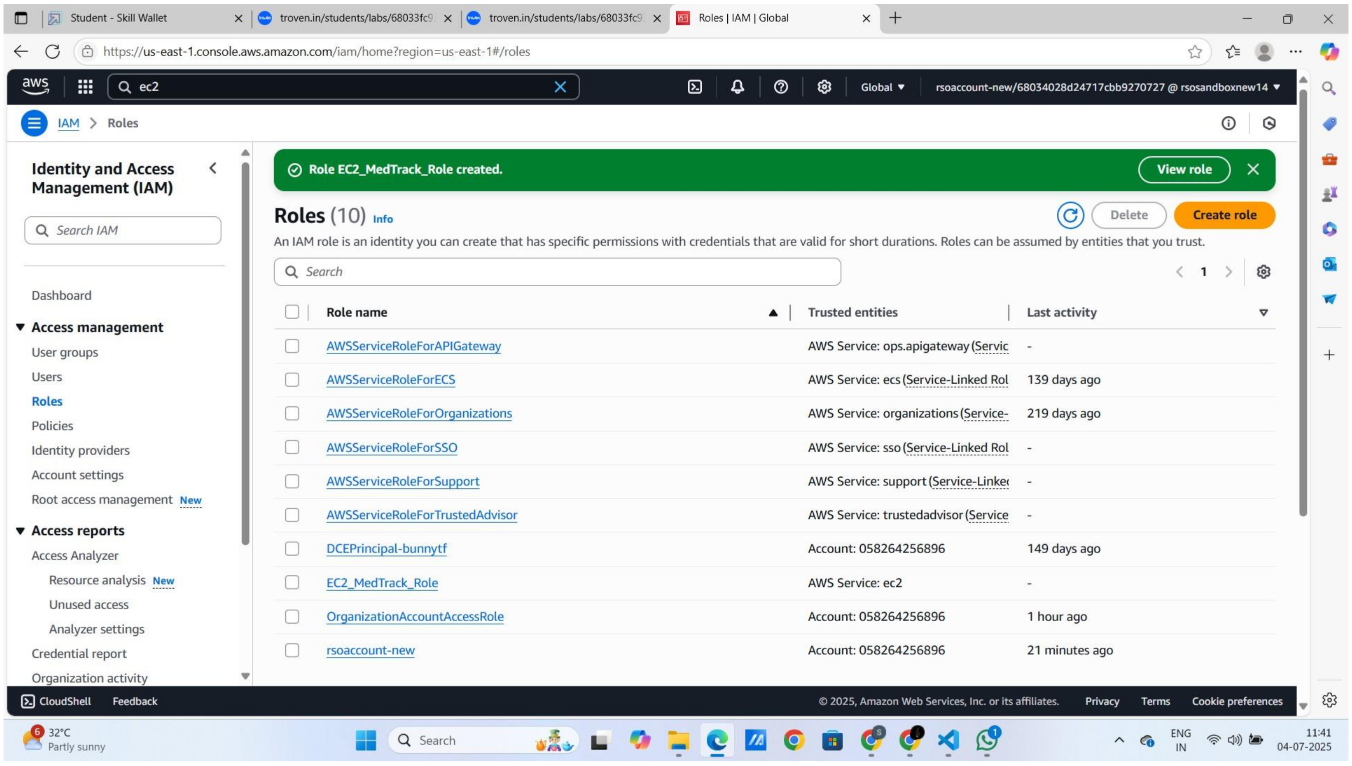
Storage (volumes)
1 volume(s) - 8 GiB

© 2025, Amazon Web Services, Inc. or its affiliates.

This screenshot shows the inbound security group rules in ec2 instance

.

## Screenshot 6



Role EC2_MedTrack_Role created.

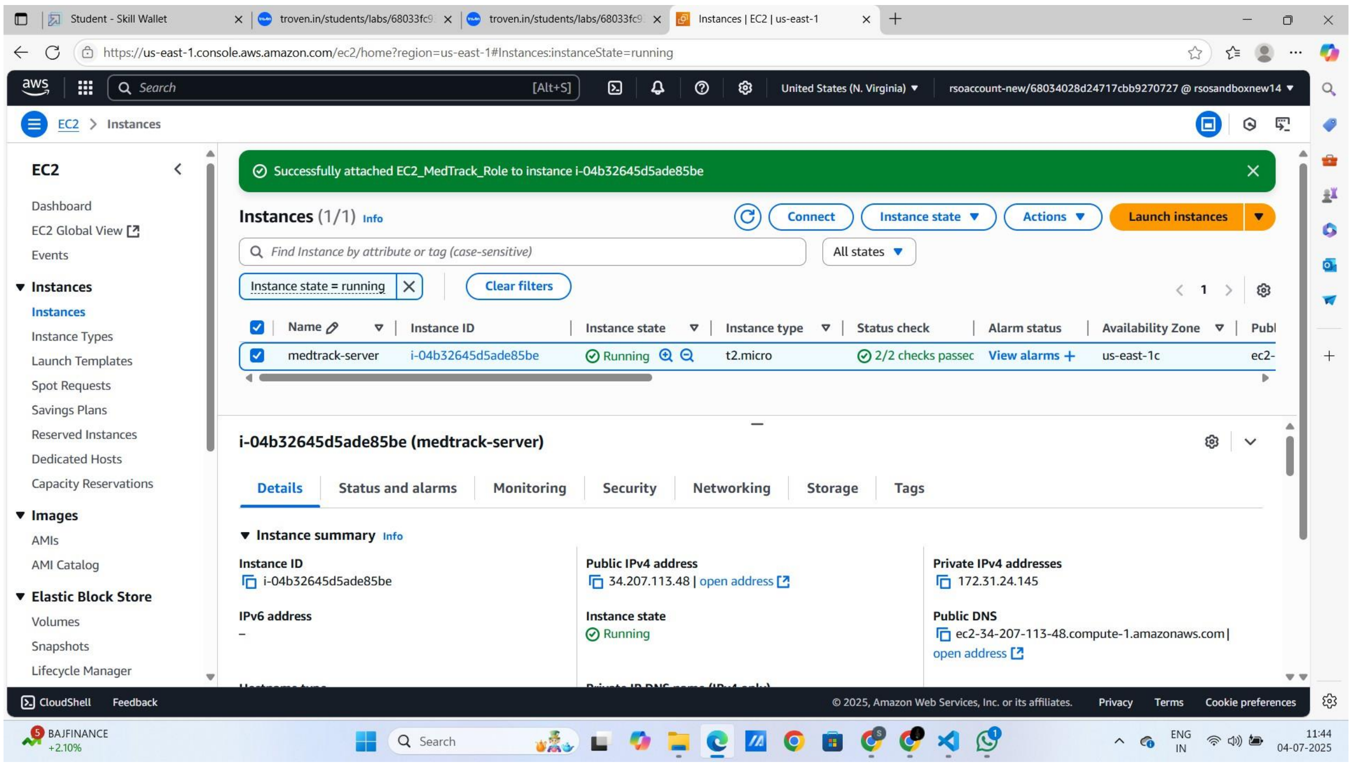
Roles (10)

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

| Role name | Trusted entities | Last activity |
|---|---|---|
| AWSServiceRoleForAPIGateway | AWS Service: ops.apigateway (Servic | - |
| AWSServiceRoleForECS | AWS Service: ecs (Service-Linked Rol | 139 days ago |
| AWSServiceRoleForOrganizations | AWS Service: organizations (Service- | 219 days ago |
| AWSServiceRoleForSSO | AWS Service: sso (Service-Linked Rol | - |
| AWSServiceRoleForSupport | AWS Service: support (Service-Linke | - |
| AWSServiceRoleForTrustedAdvisor | AWS Service: trustedadvisor (Service | - |
| DCEPrincipal-bunnytf | Account: 058264256896 | 149 days ago |
| EC2_MedTrack_Role | AWS Service: ec2 | - |
| OrganizationAccountAccessRole | Account: 058264256896 | 1 hour ago |
| rsoaccount-new | Account: 058264256896 | 21 minutes ago |

This screenshot show IAM roles  created

## Screenshot 7



Successfully attached EC2_MedTrack_Role to instance i-04b32645d5ade85be

Instances (1/1)

Instance state = running

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Publ |
|---|---|---|---|---|---|---|---|
| medtrack-server | i-04b32645d5ade85be | Running | t2.micro | 2/2 checks passec | View alarms + | us-east-1c | ec2- |

i-04b32645d5ade85be (medtrack-server)

Instance summary

Instance ID: i-04b32645d5ade85be

Public IPv4 address: 34.207.113.48 | open address

Private IPv4 addresses: 172.31.24.145

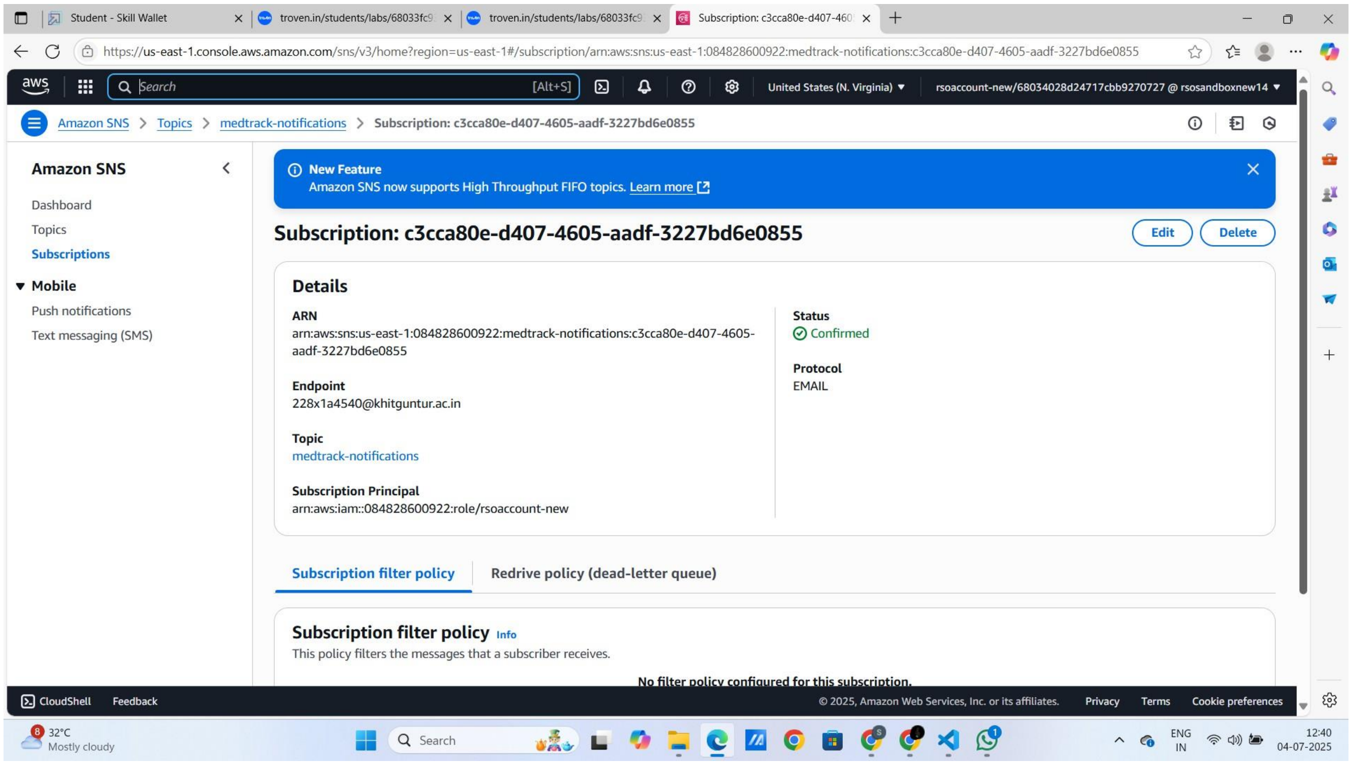
IPv6 address: –

Instance state: Running

Public DNS: ec2-34-207-113-48.compute-1.amazonaws.com | open address

This screenshot shows  the status of instance

## Screenshot 8


Amazon SNS > Topics > medtrack-notifications > Subscription: c3cca80e-d407-4605-aadf-3227bd6e0855

New Feature
Amazon SNS now supports High Throughput FIFO topics. Learn more

Subscription: c3cca80e-d407-4605-aadf-3227bd6e0855

Details

ARN
arn:aws:sns:us-east-1:084828600922:medtrack-notifications:c3cca80e-d407-4605-aadf-3227bd6e0855

Endpoint
228x1a4540@khitguntur.ac.in

Topic
medtrack-notifications

Subscription Principal
arn:aws:iam::084828600922:role/rsoaccount-new

Status
Confirmed

Protocol
EMAIL

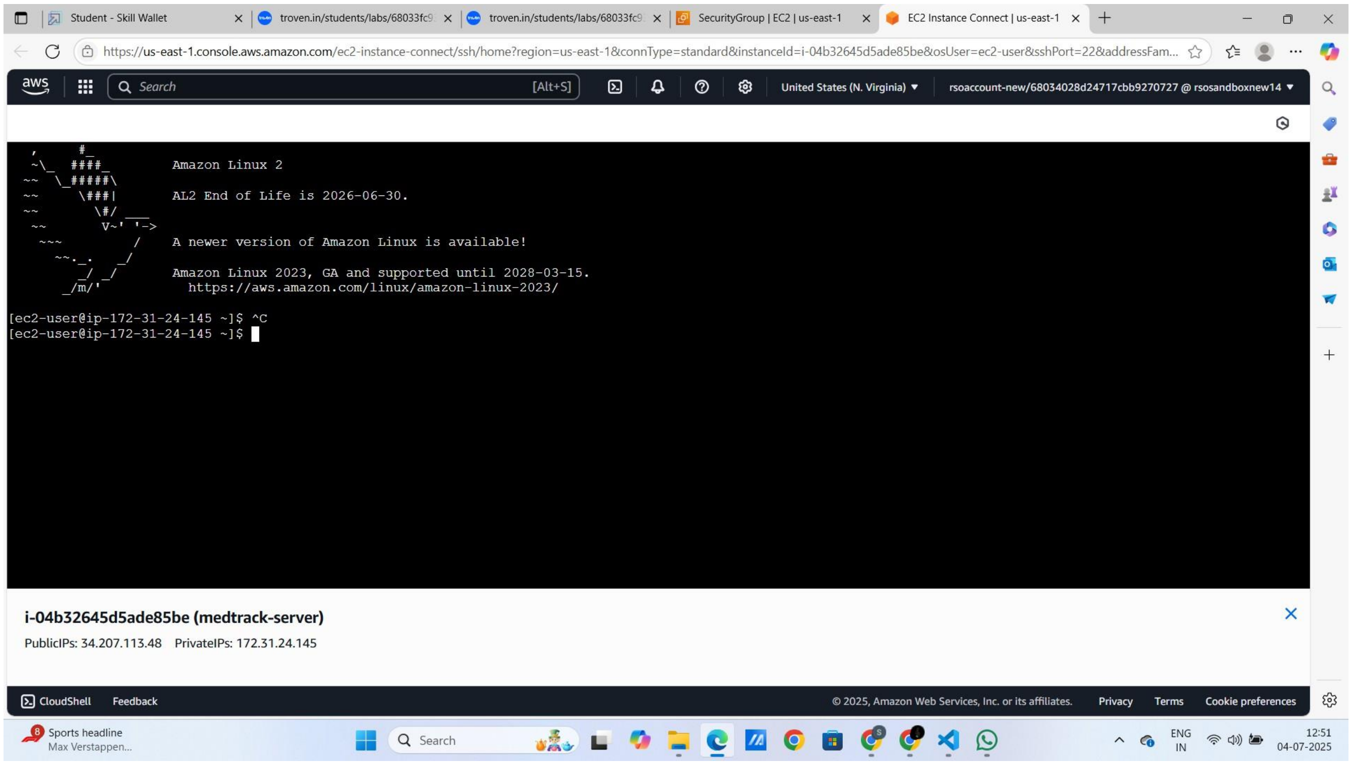
Subscription filter policy
This policy filters the messages that a subscriber receives.

No filter policy configured for this subscription.

Student - Skill Wallet
troven.in/students/labs/68033fc9
https://us-east-1.console.aws.amazon.com/sns/v3/home?region=us-east-1#/subscription/arn:aws:sns:us-east-1:084828600922:medtrack-notifications:c3cca80e-d407-4605-aadf-3227bd6e0855
United States (N. Virginia)
rsoaccount-new/68034028d24717cbb9270727 @ rsosandboxnew14
Amazon SNS
Dashboard
Topics
Subscriptions
Mobile
Push notifications
Text messaging (SMS)
Edit
Delete
Subscription filter policy
Redrive policy (dead-letter queue)
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
32°C Mostly cloudy
12:40
04-07-2025

This screenshot shows the Amazon SNS

## Screenshot 9



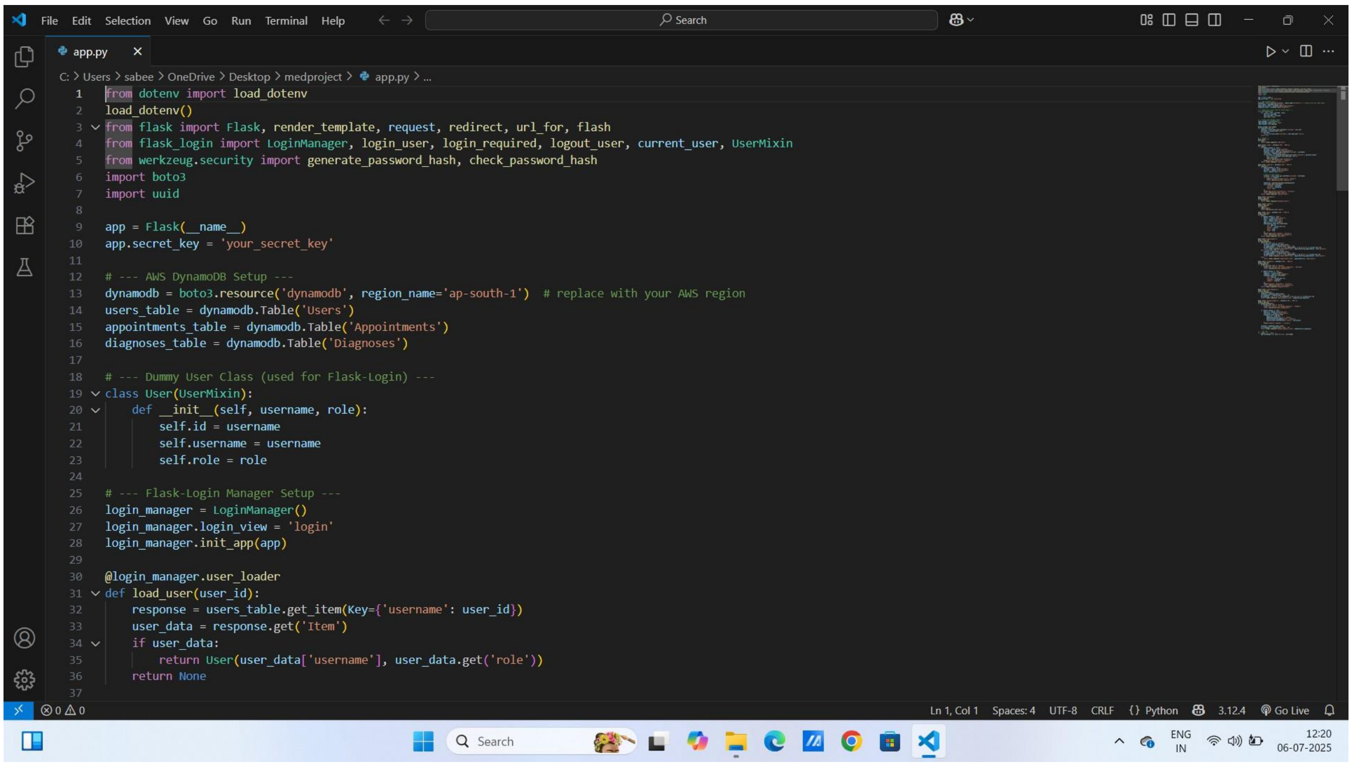
```
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-24-145 ~]$ ^C
[ec2-user@ip-172-31-24-145 ~]$ 
```

**i-04b32645d5ade85be (medtrack-server)**

PublicIPs: 34.207.113.48 PrivateIPs: 172.31.24.145

This screenshot shows the amazon Linux 2 is available

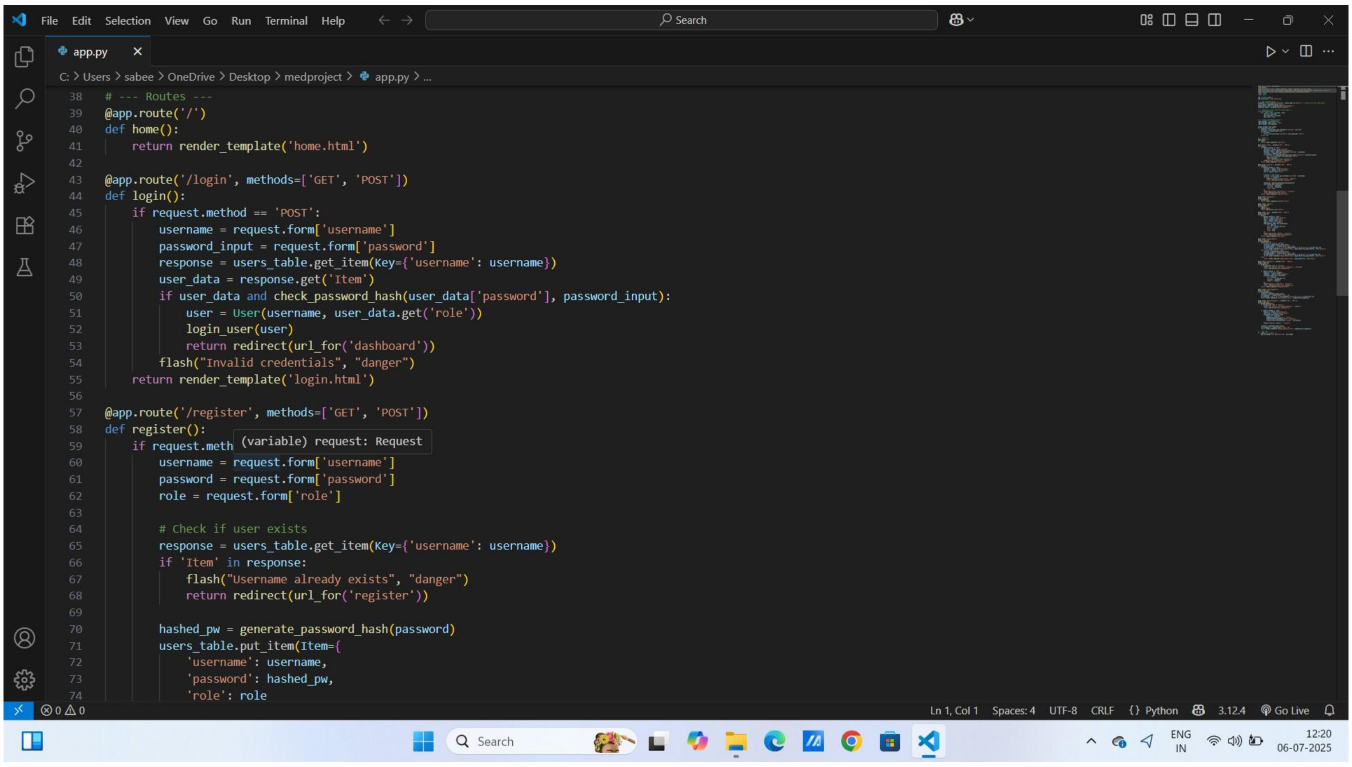
## Screenshot 10


File Edit Selection View Go Run Terminal Help
app.py
C: > Users > sabee > OneDrive > Desktop > medproject > app.py > ...
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.4 Go Live

```python
from dotenv import load_dotenv
load_dotenv()
from flask import Flask, render_template, request, redirect, url_for, flash
from flask_login import LoginManager, login_user, login_required, logout_user, current_user, UserMixin
from werkzeug.security import generate_password_hash, check_password_hash
import boto3
import uuid

app = Flask(__name__)
app.secret_key = 'your_secret_key'

# --- AWS DynamoDB Setup ---
dynamodb = boto3.resource('dynamodb', region_name='ap-south-1')  # replace with your AWS region
users_table = dynamodb.Table('Users')
appointments_table = dynamodb.Table('Appointments')
diagnoses_table = dynamodb.Table('Diagnoses')

# --- Dummy User Class (used for Flask-Login) ---
class User(UserMixin):
    def __init__(self, username, role):
        self.id = username
        self.username = username
        self.role = role

# --- Flask-Login Manager Setup ---
login_manager = LoginManager()
login_manager.login_view = 'login'
login_manager.init_app(app)

@login_manager.user_loader
def load_user(user_id):
    response = users_table.get_item(Key={'username': user_id})
    user_data = response.get('Item')
    if user_data:
        return User(user_data['username'], user_data.get('role'))
    return None

```

This screenshot the code used for app.py

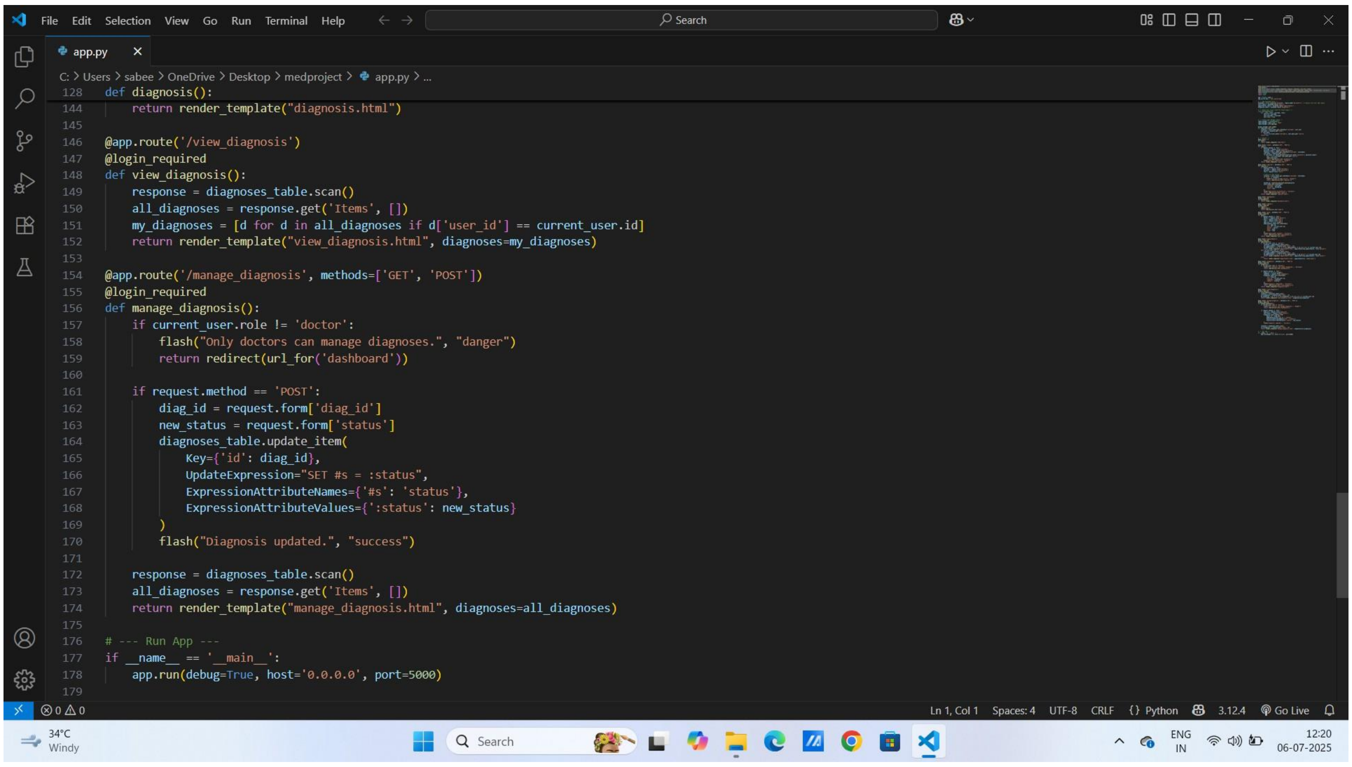
## Screenshot 11


File Edit Selection View Go Run Terminal Help
app.py
C: > Users > sabee > OneDrive > Desktop > medproject > app.py > ...
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.4 Go Live
12:20
06-07-2025

```python
# --- Routes ---
@app.route('/')
def home():
    return render_template('home.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password_input = request.form['password']
        response = users_table.get_item(Key={'username': username})
        user_data = response.get('Item')
        if user_data and check_password_hash(user_data['password'], password_input):
            user = User(username, user_data.get('role'))
            login_user(user)
            return redirect(url_for('dashboard'))
        flash("Invalid credentials", "danger")
    return render_template('login.html')

@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.meth (variable) request: Request
        username = request.form['username']
        password = request.form['password']
        role = request.form['role']

        # Check if user exists
        response = users_table.get_item(Key={'username': username})
        if 'Item' in response:
            flash("Username already exists", "danger")
            return redirect(url_for('register'))

        hashed_pw = generate_password_hash(password)
        users_table.put_item(Item={
            'username': username,
            'password': hashed_pw,
            'role': role
```

This screenshot shows the continuous code of app.py

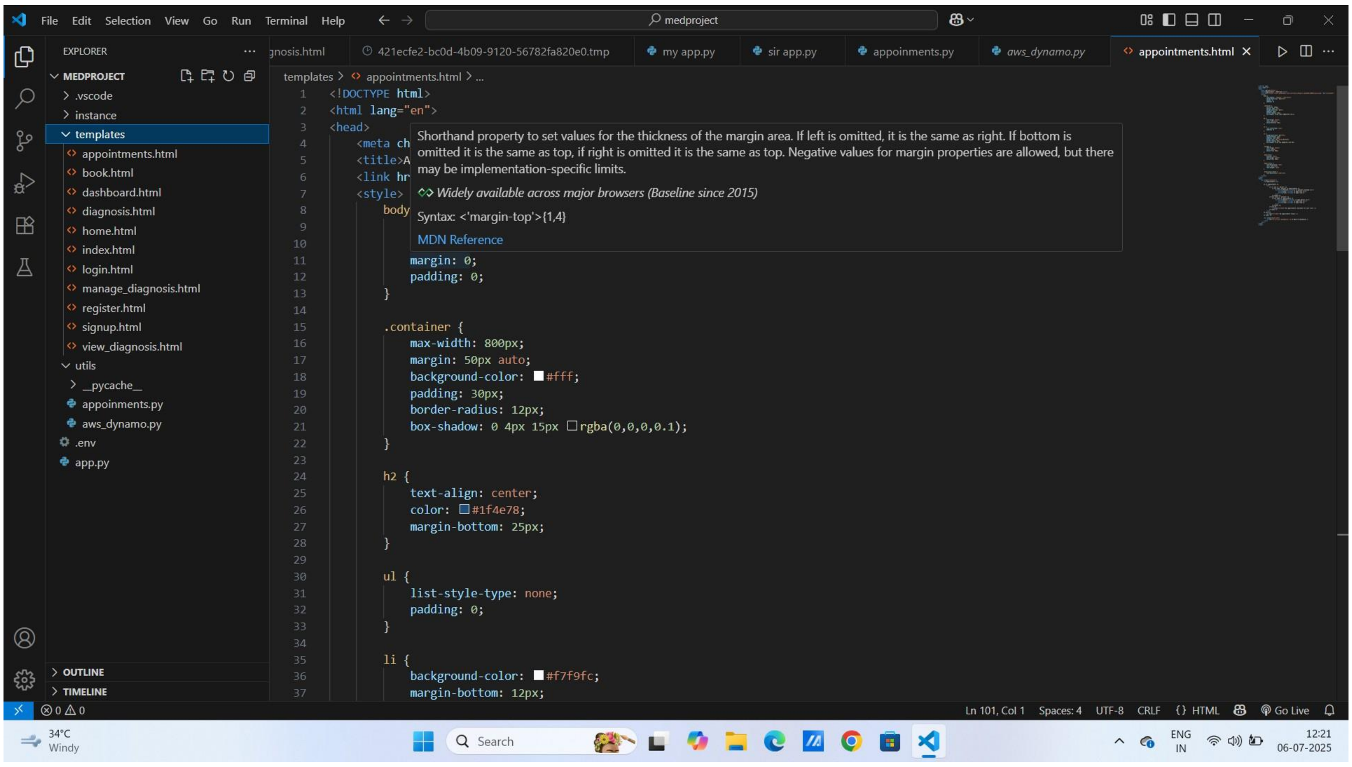
## Screenshot 12



```python
def diagnosis():
    return render_template("diagnosis.html")

@app.route('/view_diagnosis')
@login_required
def view_diagnosis():
    response = diagnoses_table.scan()
    all_diagnoses = response.get('Items', [])
    my_diagnoses = [d for d in all_diagnoses if d['user_id'] == current_user.id]
    return render_template("view_diagnosis.html", diagnoses=my_diagnoses)

@app.route('/manage_diagnosis', methods=['GET', 'POST'])
@login_required
def manage_diagnosis():
    if current_user.role != 'doctor':
        flash("Only doctors can manage diagnoses.", "danger")
        return redirect(url_for('dashboard'))

    if request.method == 'POST':
        diag_id = request.form['diag_id']
        new_status = request.form['status']
        diagnoses_table.update_item(
            Key={'id': diag_id},
            UpdateExpression="SET #s = :status",
            ExpressionAttributeNames={'#s': 'status'},
            ExpressionAttributeValues={':status': new_status}
        )
        flash("Diagnosis updated.", "success")

    response = diagnoses_table.scan()
    all_diagnoses = response.get('Items', [])
    return render_template("manage_diagnosis.html", diagnoses=all_diagnoses)

# --- Run App ---
if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```

This screenshot shows the app.py code ends

## Screenshot 13



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta ch
    <title>A
    <link hr
    <style>
        body


            margin: 0;
            padding: 0;
        }

        .container {
            max-width: 800px;
            margin: 50px auto;
            background-color: #fff;
            padding: 30px;
            border-radius: 12px;
            box-shadow: 0 4px 15px rgba(0,0,0,0.1);
        }

        h2 {
            text-align: center;
            color: #1f4e78;
            margin-bottom: 25px;
        }

        ul {
            list-style-type: none;
            padding: 0;
        }

        li {
            background-color: #f7f9fc;
            margin-bottom: 12px;
```

Shorthand property to set values for the thickness of the margin area. If left is omitted, it is the same as right. If bottom is omitted it is the same as top, if right is omitted it is the same as top. Negative values for margin properties are allowed, but there may be implementation-specific limits.

*Widely available across major browsers (Baseline since 2015)*

Syntax: <'margin-top'>{1,4}

MDN Reference

This screenshot shows the templates used in project

## Conclusion

This project, MedTrack, showcases the integration of cloud technologies in healthcare systems. By leveraging AWS services like EC2 and DynamoDB, it ensures scalability, reliability, and performance. The use of Flask makes the backend lightweight and efficient, while AWS SNS enhances the user experience through real-time communication. This system is a step forward in modern healthcare management solutions.