Name: Syam Krishna Reddy P | Date: 31st oct, 2025

Emp.id: ENC/16801 | Tra ck: Java

Question:

Create a serverless web application that does CRUD operation on data through an HTML form hosted on EC2, processes it via Lambda functions, and stores it in DynamoDB.

Architecture Components

1. **EC2 Instance**: Hosts a static HTML form
2. **API Gateway**: REST API endpoint
3. **Lambda Functions**: Process form data
4. **DynamoDB**: Stores submitted form data
5. **IAM Roles**: Secure permissions between services

STEP 1: Create DynamoDB Table

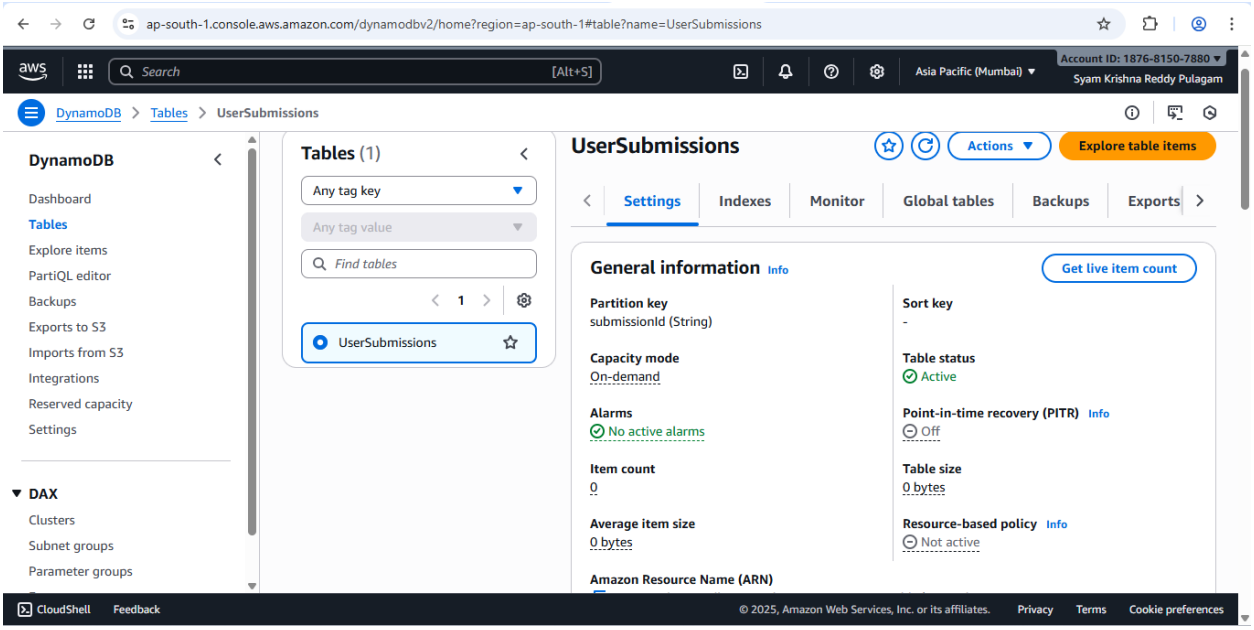
1. Open the DynamoDB console

- Go to AWS Console → DynamoDB → Tables → Create table

2. Table details

- Table name: `UserSubmissions`
- Partition key: `submissionId` (String)
- Leave Sort key empty
- Choose Provisioned or On-demand capacity (On-demand is fine for simplicity)

3. Click Create table



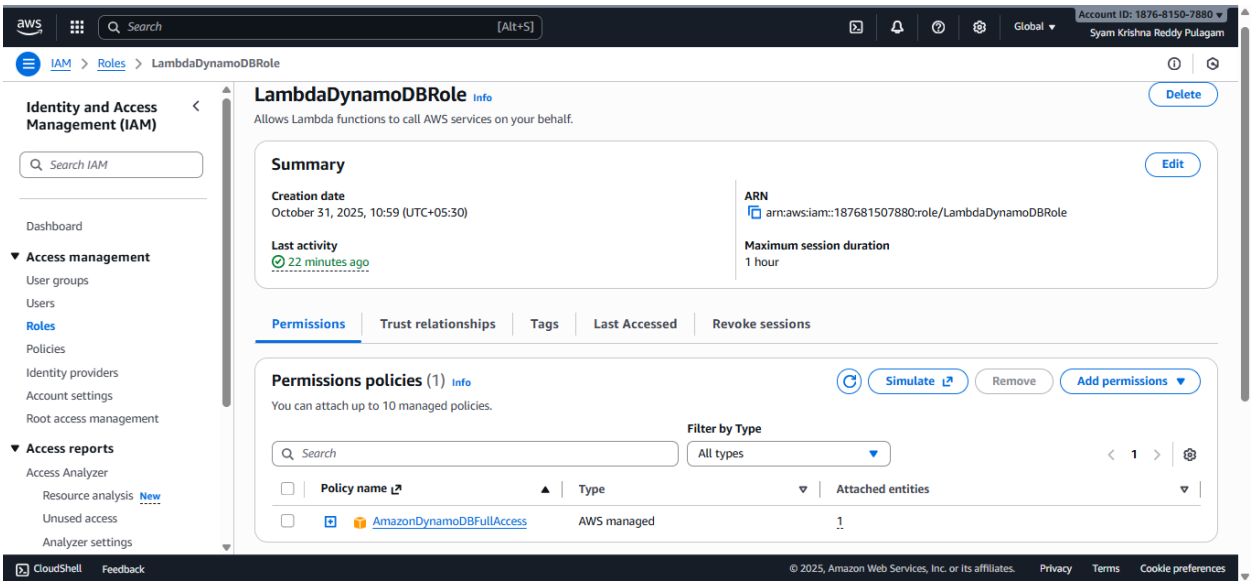
STEP 2: Create IAM Role for Lambda

1. Go to IAM → Roles → Create role

- Select AWS Service
- Use case: Lambda
- Click Next

2. Attach permissions

- `AmazonDynamoDBFullAccess` *(for simplicity — can later restrict to the table)*

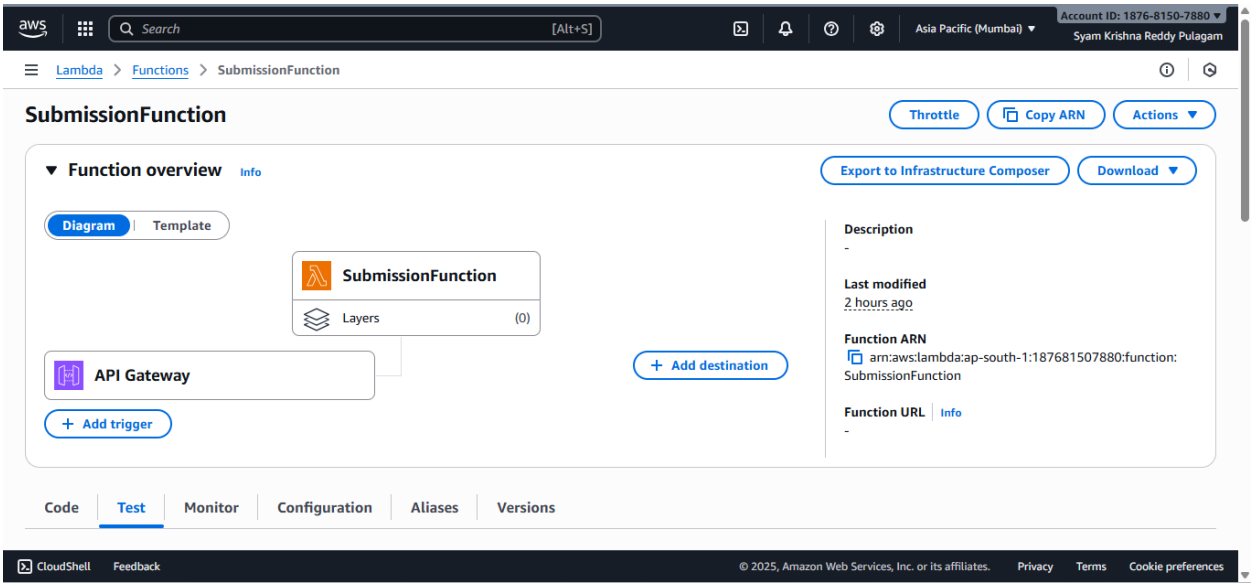


STEP 3: Create Lambda Functions

You'll create **two** functions:

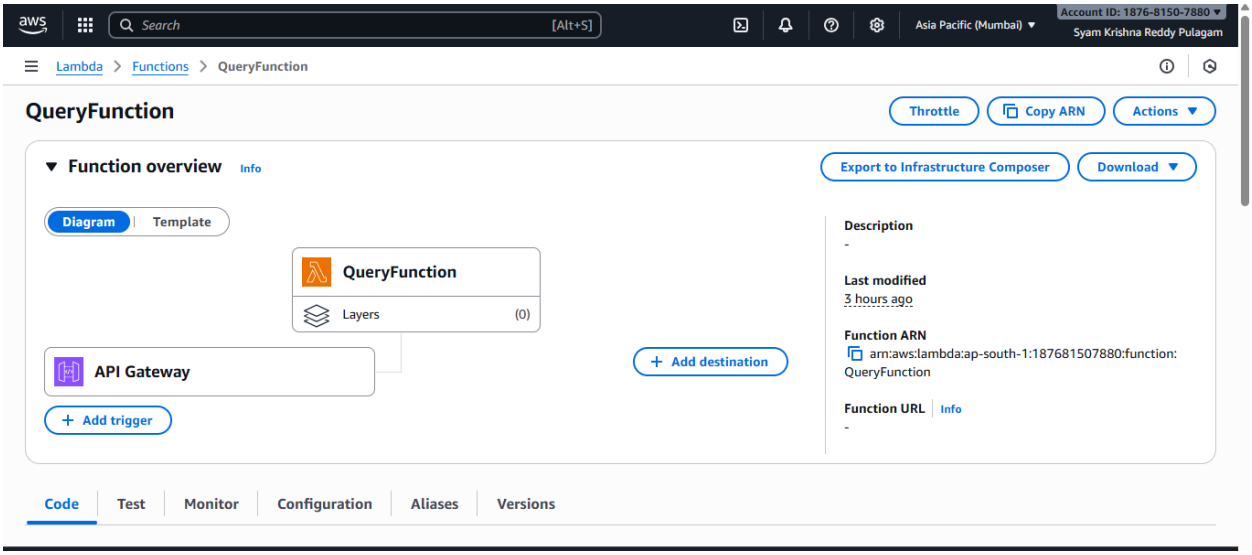
- A. Submission Lambda (POST /submit)

1. Go to AWS Lambda → Create function
    a. Name: `SubmissionFunction`
    b. Runtime: Python 3.9
    c. Execution role: Use existing role → LambdaDynamoDBRole
2. Click Create function



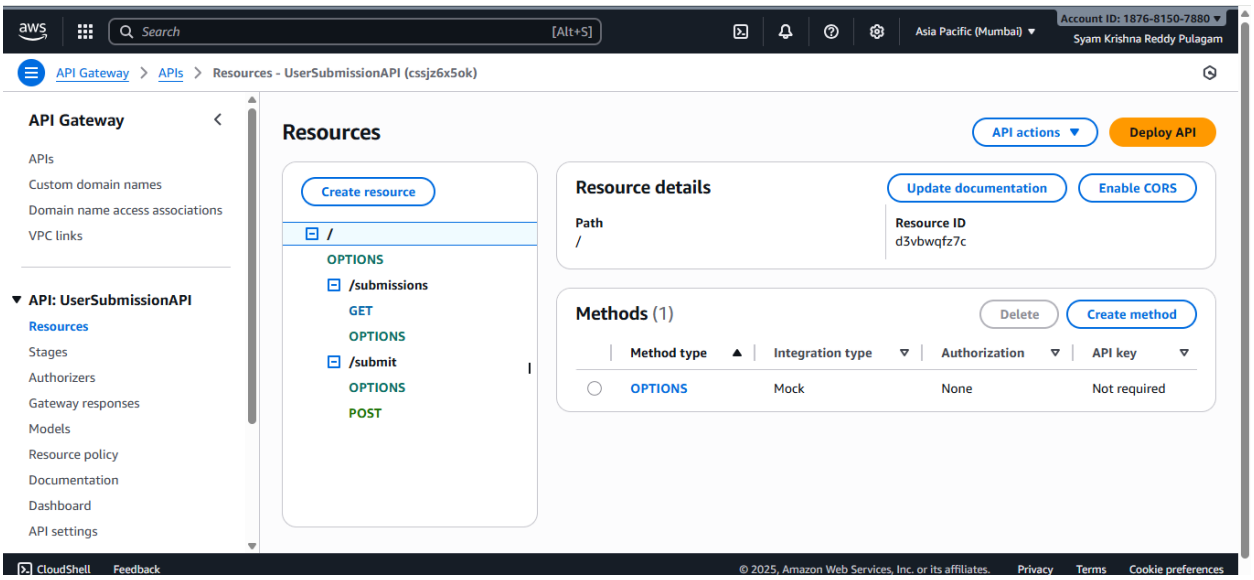
- **B. Query Lambda (GET /submissions)**

1. Create another Lambda:
    a. Name: `QueryFunction`
    b. Runtime: Python 3.9
    c. Role: Use existing role → LambdaDynamoDBRole



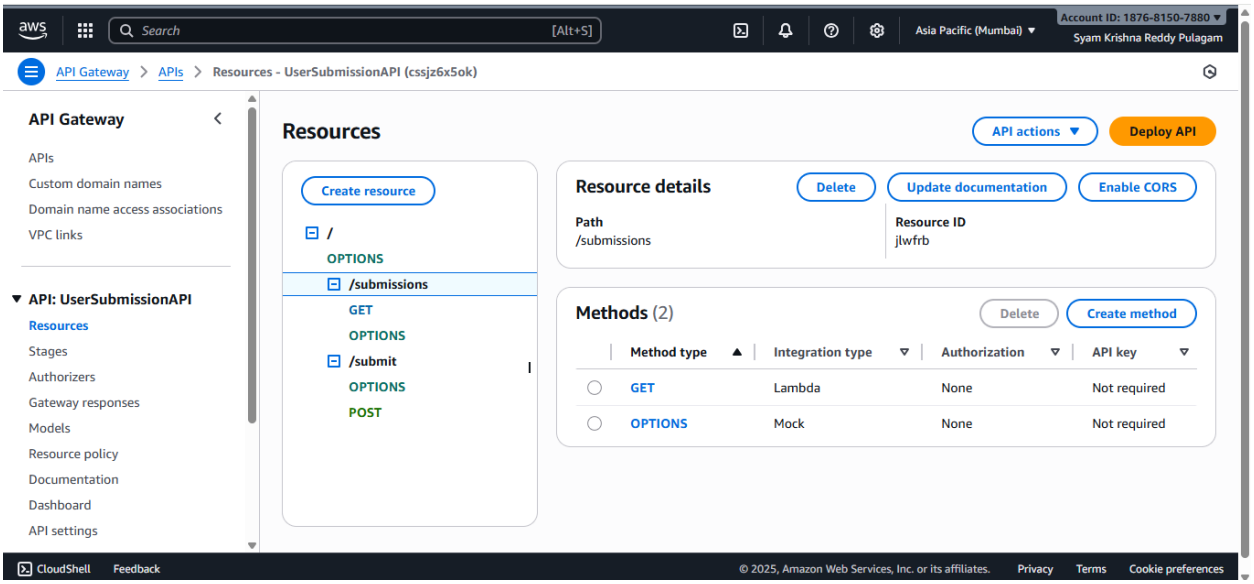
STEP 4: Create API Gateway

1. Go to Amazon API Gateway → Create API
    a. Choose REST API → Build
    b. Name: `UserSubmissionAPI`
    c. Endpoint Type: Regional
    d. Click Create API



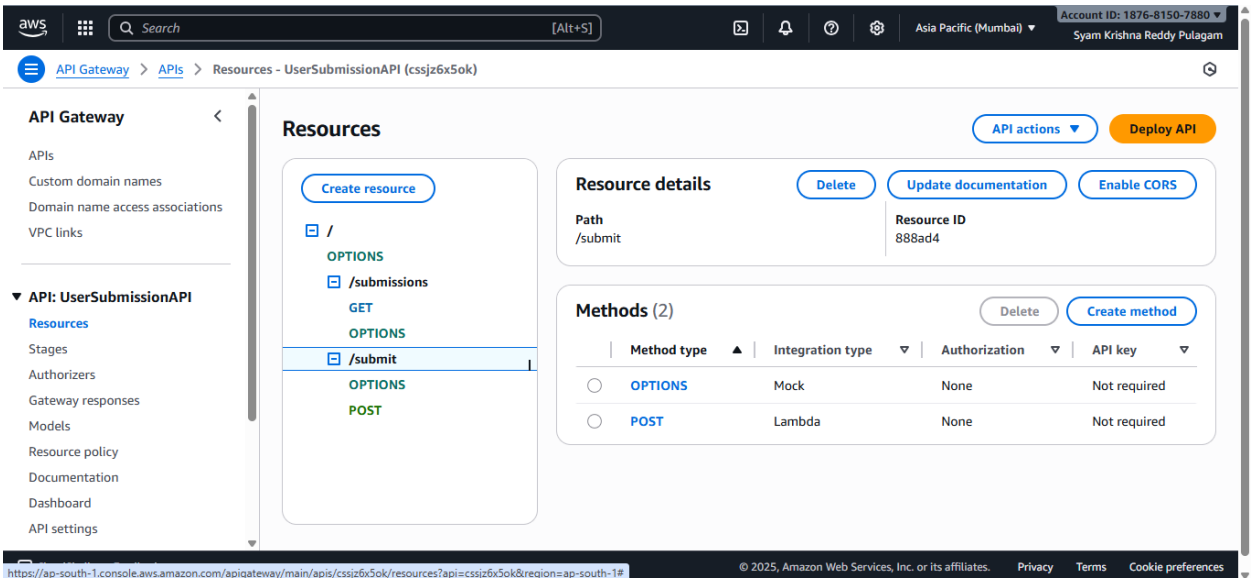
4.1 Create Resource `/submit`

1. Click **Actions → Create Resource**
    a. Resource name: `submit`
    b. Resource path: `/submit`
    c. Click **Create Resource**
2. With `/submit` selected → Click **Create Method → POST**
3. Integration type: **Lambda Function**
    a. Select your region
    b. Lambda Function: `SubmissionFunction`
    c. Save → Allow permission prompt



4.2 Create Resource `/submissions`

1. Create another resource `/submissions`
2. Add a **GET** method
    a. Integration: **Lambda Function**
    b. Lambda: `QueryFunction`
    c. Save → Allow permission prompt



4.3 Enable CORS

1. For each resource (`/submit` and `/submissions`):
    a. Click **Actions → Enable CORS**
    b. Accept defaults and confirm

4.4 Deploy the API

1. Click **Actions → Deploy API**
    a. Deployment stage: **[New Stage]**
    b. Stage name: `prod`
2. Click **Deploy**

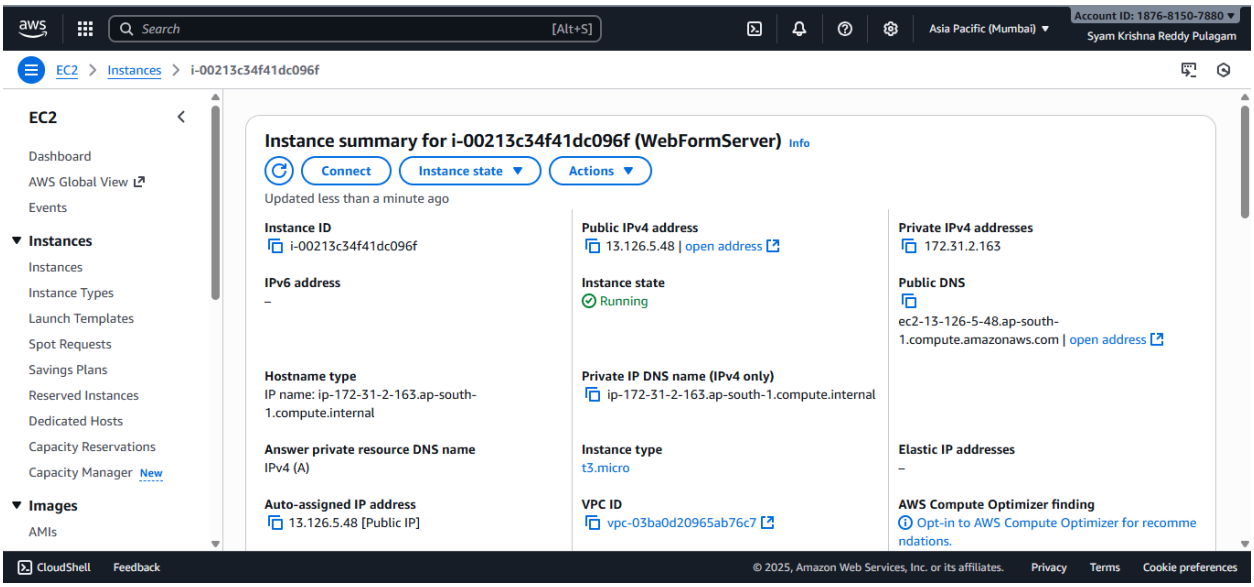
Now you'll get URLs like:

`POST` https://cssjz6x5ok.execute-api.ap-south-1.amazonaws.com/prod/submit
`GET` https://cssjz6x5ok.execute-api.ap-south-1.amazonaws.com/prod/submissions

STEP 5: Launch EC2 Instance and Host HTML

5.1 Launch EC2

1. Go to **EC2 → Launch Instance**
    a. Name: `WebFormServer`
    b. AMI: **Amazon Linux 2**
    c. Instance type: `t2.micro`
    d. Key pair: (Create one if needed)
    e. Security Group: Allow **HTTP (port 80)** and **SSH (port 22)**
2. Click **Launch Instance**



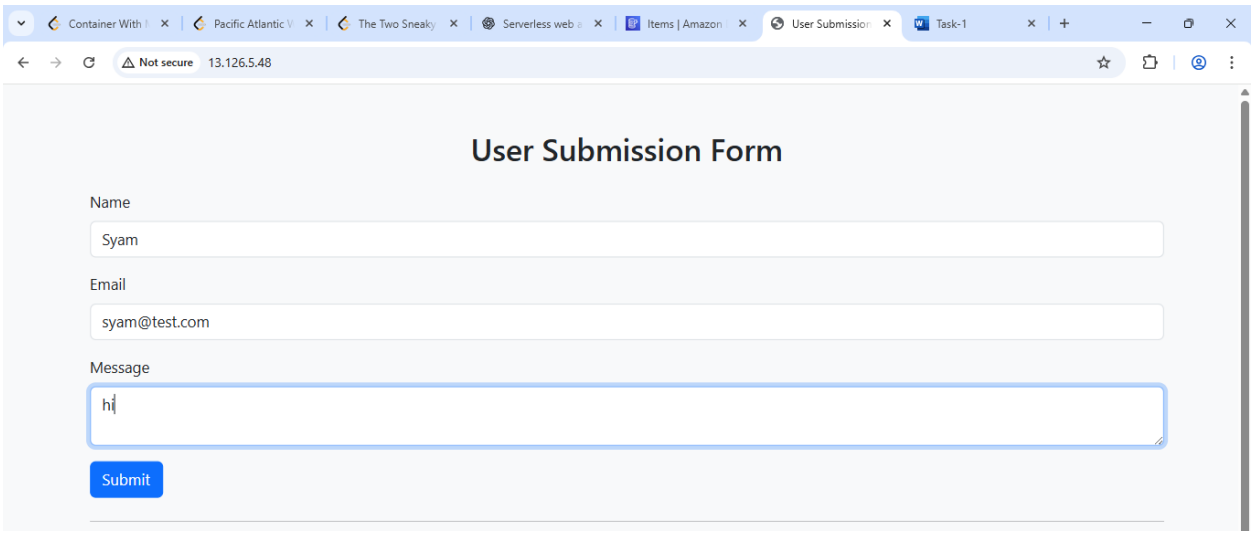
5.2 Connect via SSH and install apache and create html form

```
C:\Users\SyamKrishnaReddy.P>ssh
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface] [-b bind_address]
           [-c cipher_spec] [-D [bind_address:]port] [-E log_file]
           [-e escape_char] [-F configfile] [-I pkcs11] [-i identity_file]
           [-J destination] [-L address] [-l login_name] [-m mac_spec]
           [-O ctl_cmd] [-o option] [-P tag] [-p port] [-Q query_option]
           [-R address] [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]]
           destination [command [argument ...]]

C:\Users\SyamKrishnaReddy.P>ssh -i "C:\Users\SyamKrishnaReddy.P\Downloads\New_pair.pem" ec2-user@13.126.5.48
The authenticity of host '13.126.5.48 (13.126.5.48)' can't be established.
ED25519 key fingerprint is SHA256:E7bTX3Vm2TfsE8Xrpm8rVzNpHQiB9BOcUKEOMTiK3Cc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.126.5.48' (ED25519) to the list of known hosts.
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-2-163 ~]$ sudo yum update -y

Amazon Linux 2023 Kernel Livepatch repository                                                    212 kB/s |  28 kB     00:00
Last metadata expiration check: 0:00:01 ago on Fri Oct 31 06:45:47 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-2-163 ~]$ sudo yum install -y httpd
Last metadata expiration check: 0:00:13 ago on Fri Oct 31 06:45:47 2025.
Dependencies resolved.
================================================================================ Package                         Architecture       Version
                    Repository               Size
================================================================================Installing:
 httpd                          x86_64             2.4.65-1.amzn2023.0.2                amazonlinux              47 k
Installing dependencies:
 apr                            x86_64             1.7.5-1.amzn2023.0.4                 amazonlinux             129 k
 apr-util                       x86_64             1.6.3-1.amzn2023.0.2                 amazonlinux              97 k
 apr-util-lmdb                  x86_64             1.6.3-1.amzn2023.0.2                 amazonlinux              13 k
 generic-logos-httpd            noarch             18.0.0-12.amzn2023.0.3               amazonlinux              19 k
```

```
[ec2-user@ip-172-31-2-163 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-2-163 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-2-163 ~]$ sudo nano /var/www/html/index.html
[ec2-user@ip-172-31-2-163 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-10-31 06:46:09 UTC; 2min 59s ago
       Docs: man:httpd.service(8)
   Main PID: 14975 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 1053)
     Memory: 13.3M
        CPU: 254ms
     CGroup: /system.slice/httpd.service
             ├─14975 /usr/sbin/httpd -DFOREGROUND
             ├─15087 /usr/sbin/httpd -DFOREGROUND
             ├─15098 /usr/sbin/httpd -DFOREGROUND
             ├─15100 /usr/sbin/httpd -DFOREGROUND
             └─15118 /usr/sbin/httpd -DFOREGROUND

Oct 31 06:46:09 ip-172-31-2-163.ap-south-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Oct 31 06:46:09 ip-172-31-2-163.ap-south-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Oct 31 06:46:09 ip-172-31-2-163.ap-south-1.compute.internal httpd[14975]: Server configured, listening on: port 80
```

5.5 Test it!

1. Open browser → enter **EC2 Public IP**
2. Fill out the form → Submit
3. Go to **DynamoDB → Tables → UserSubmissions → Explore table items**
    a. You'll see your submission!



All Submissions

Refresh

Submission ID	Name	Email	Message	Submission Date	Status
1e2a87dc-8756-4c37-9e59-659514811b53	Syam	syam@test.com	Hi	2025-10-31T08:24:30.722610	Received
00cf8fcf-4dc5-4186-a015-73bdfb5d44f9	John	john@test.com	Hello	2025-10-31T07:18:29.634124	Received