Name: Rathna Sekhar Reddy V

Emp.id: ENC/16811

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.

STEP 1: Create DynamoDB Table

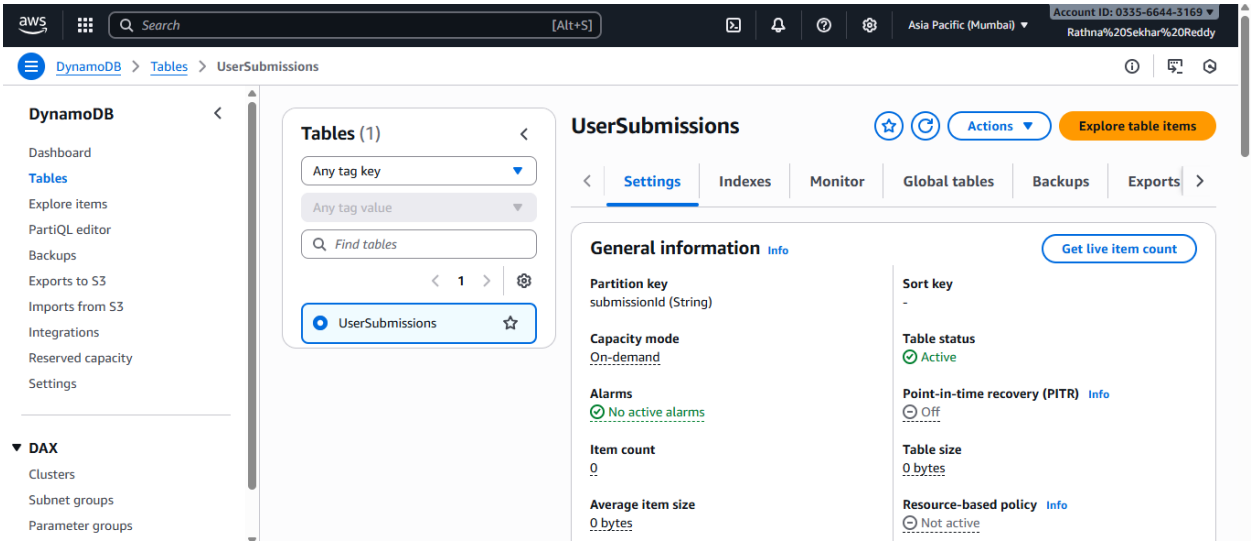
1. Open the DynamoDB console

- 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

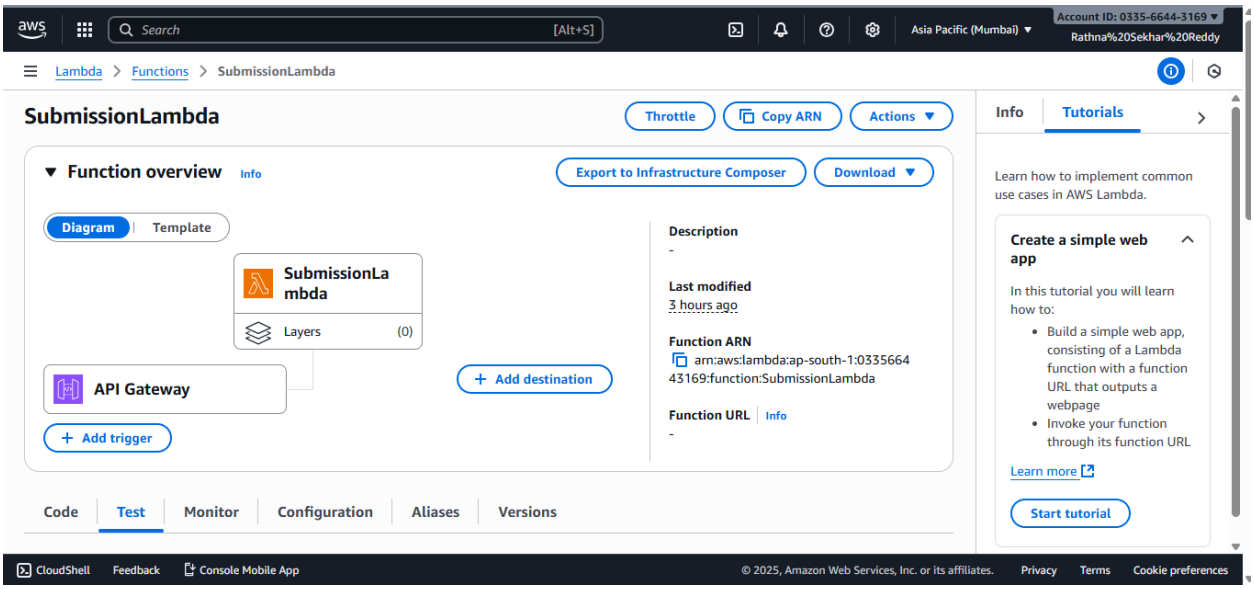
- `AmazonDynamoDBFullAccess` *(for simplicity — can later restrict to the table)*

STEP 3: Create Lambda Functions

You'll create **two** functions:

- A. Submission Lambda (POST /submit)

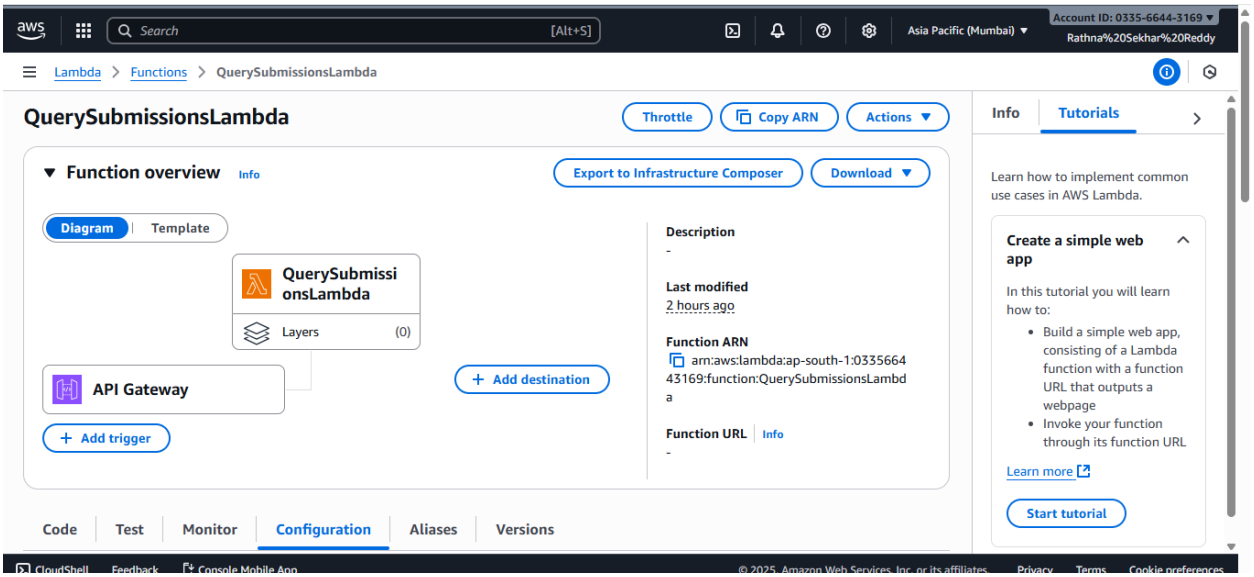
1. Go to AWS Lambda → Create function
    a. Name: `SubmissionFunction`
    b. Runtime: Python 3.9
    c. Execution role: Use existing role → LambdaDynamoDBRole
2. Click Create function



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

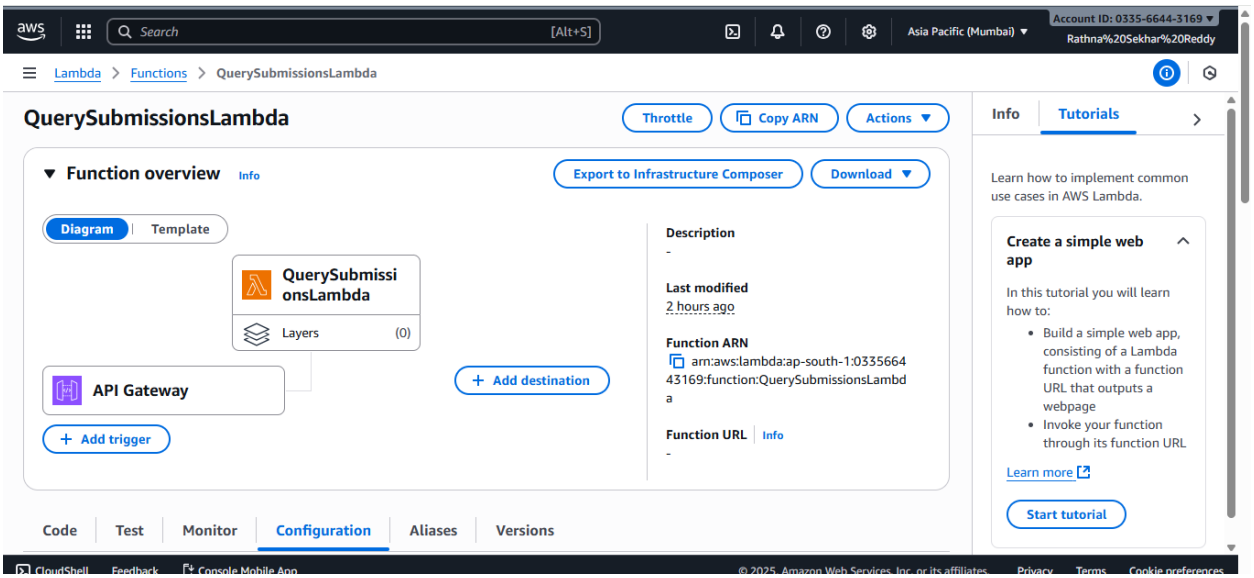
1. Create another Lambda:

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



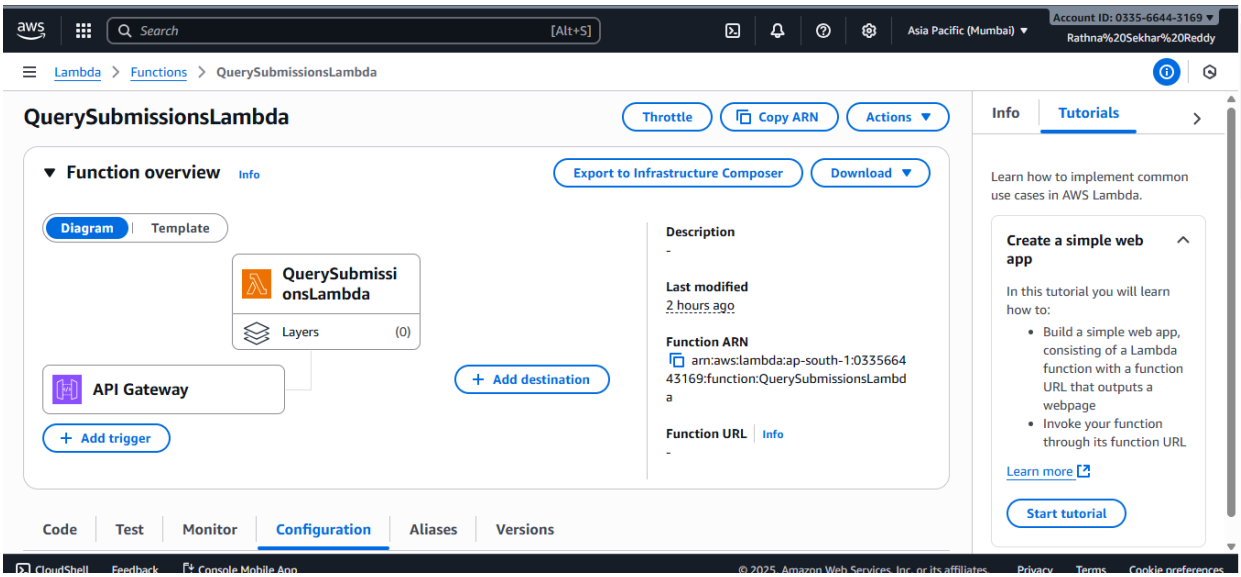
STEP 4: Create API Gateway

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



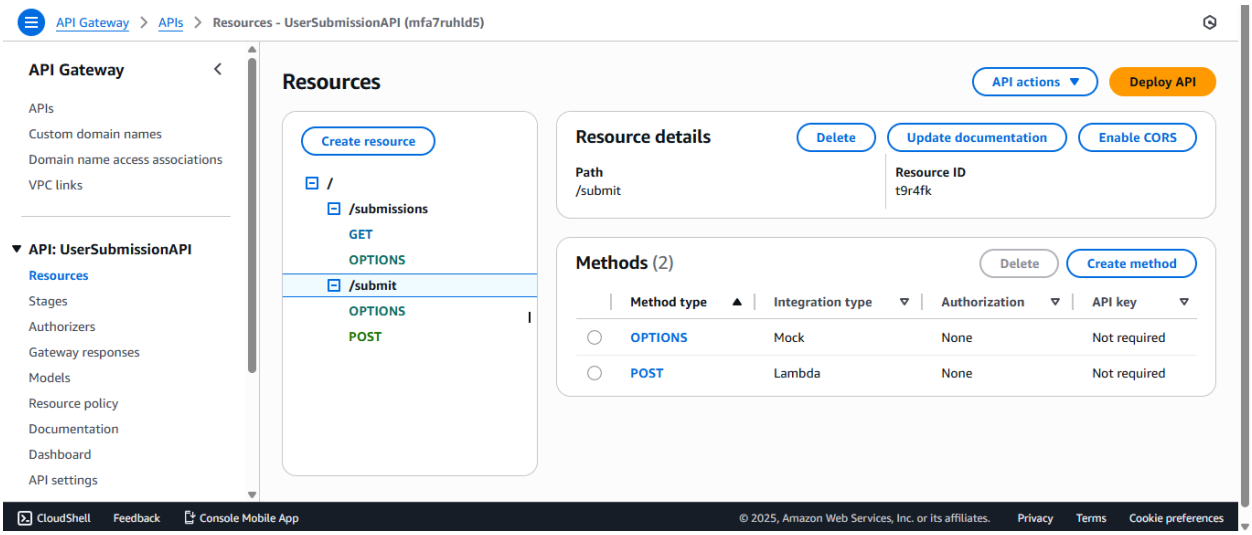
4.1 Create Resource `/submit`

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

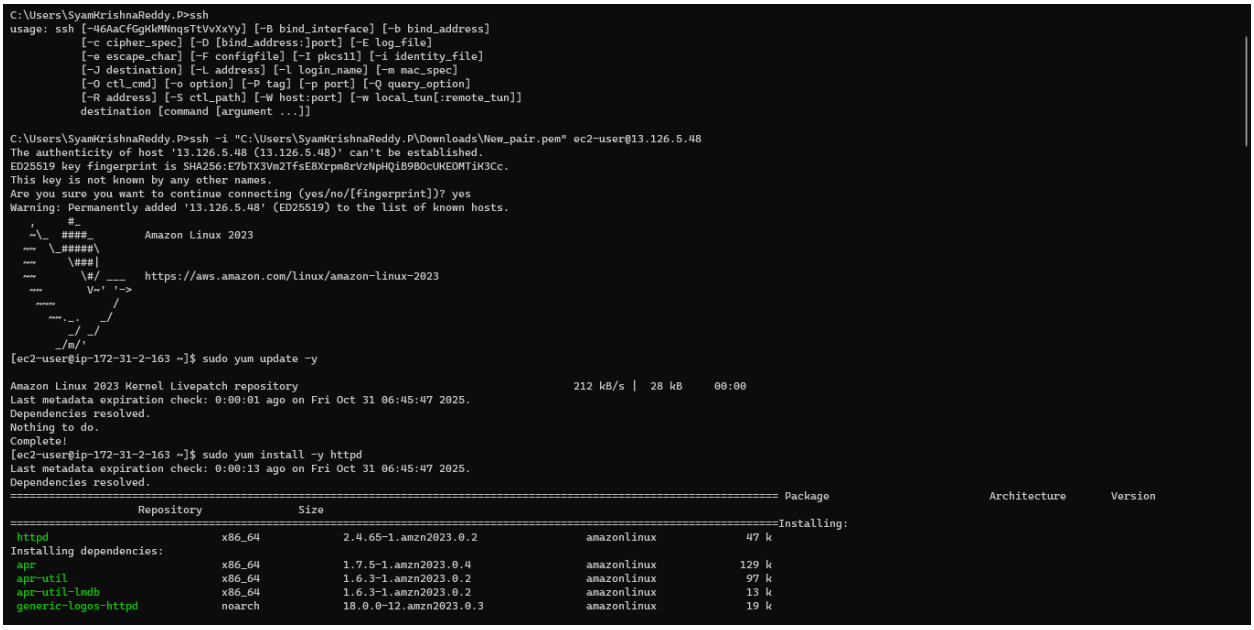
STEP 5: Launch EC2 Instance and Host HTML

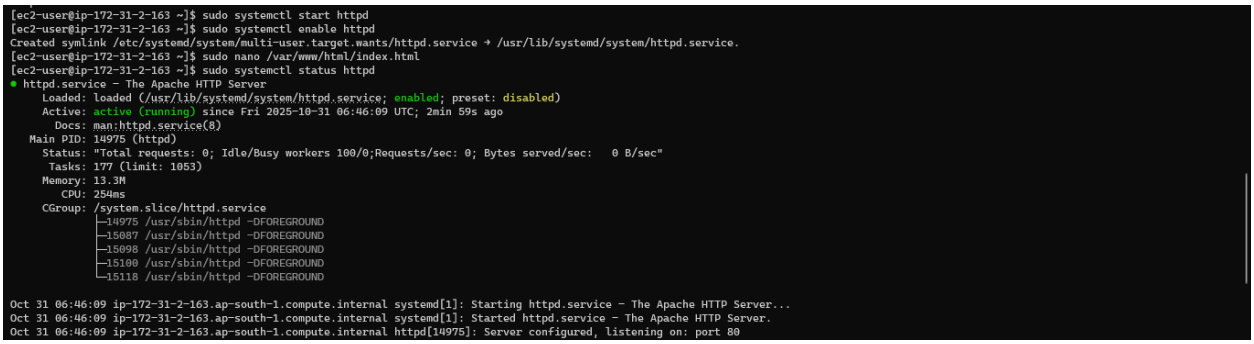
5.1 Launch EC2

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**

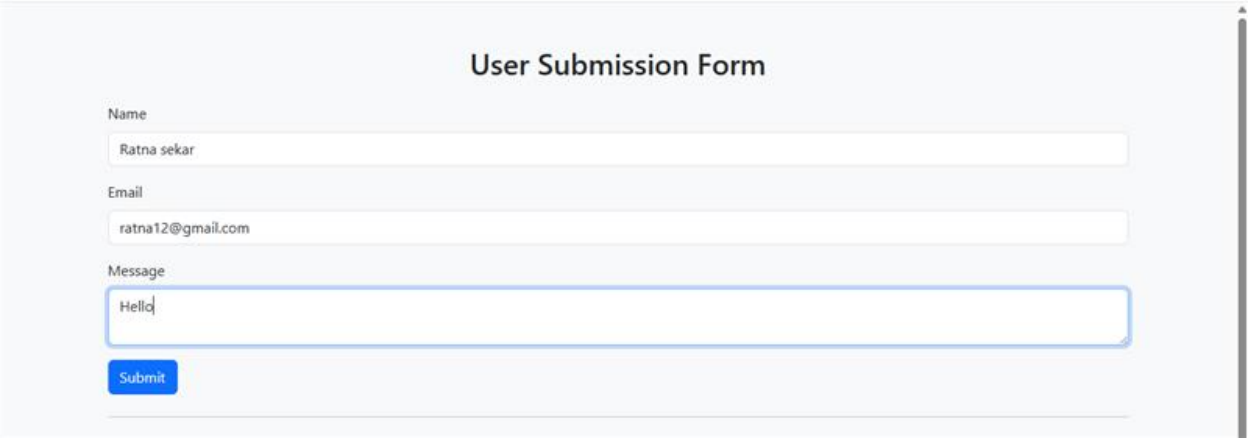
5.2 Connect via SSH and install apache and create html form





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!


User Submission Form
Name
Ratna sekar
Email
ratna12@gmail.com
Message
Hello
Submit

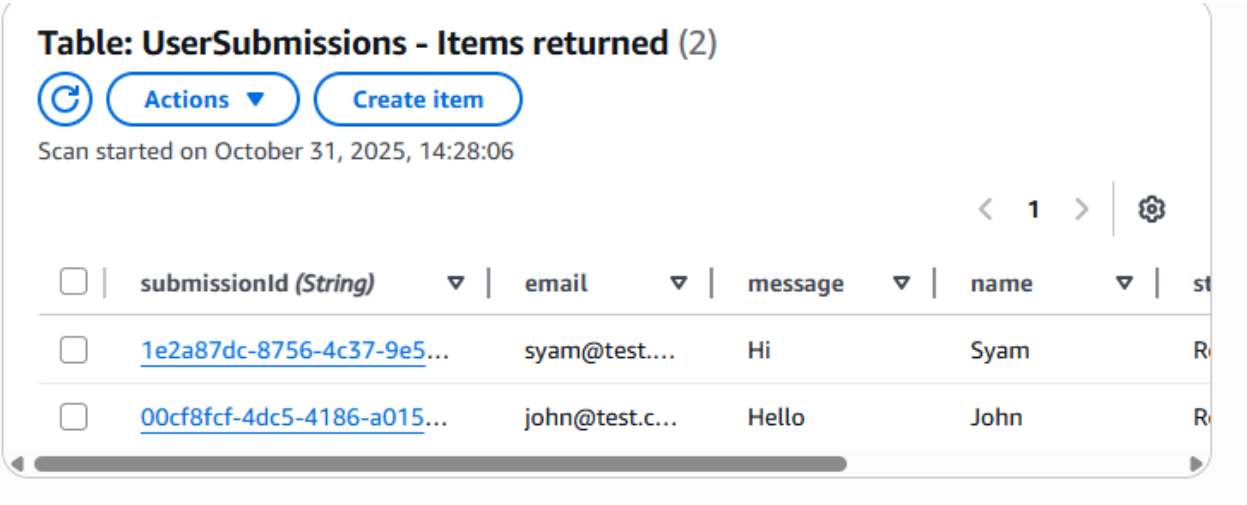


Table: UserSubmissions - Items returned (2)
Actions
Create item
Scan started on October 31, 2025, 14:28:06
1
submissionId (String)
email
message
name
1e2a87dc-8756-4c37-9e5...
syam@test....
Hi
Syam
00cf8fcf-4dc5-4186-a015...
john@test.c...
Hello
John