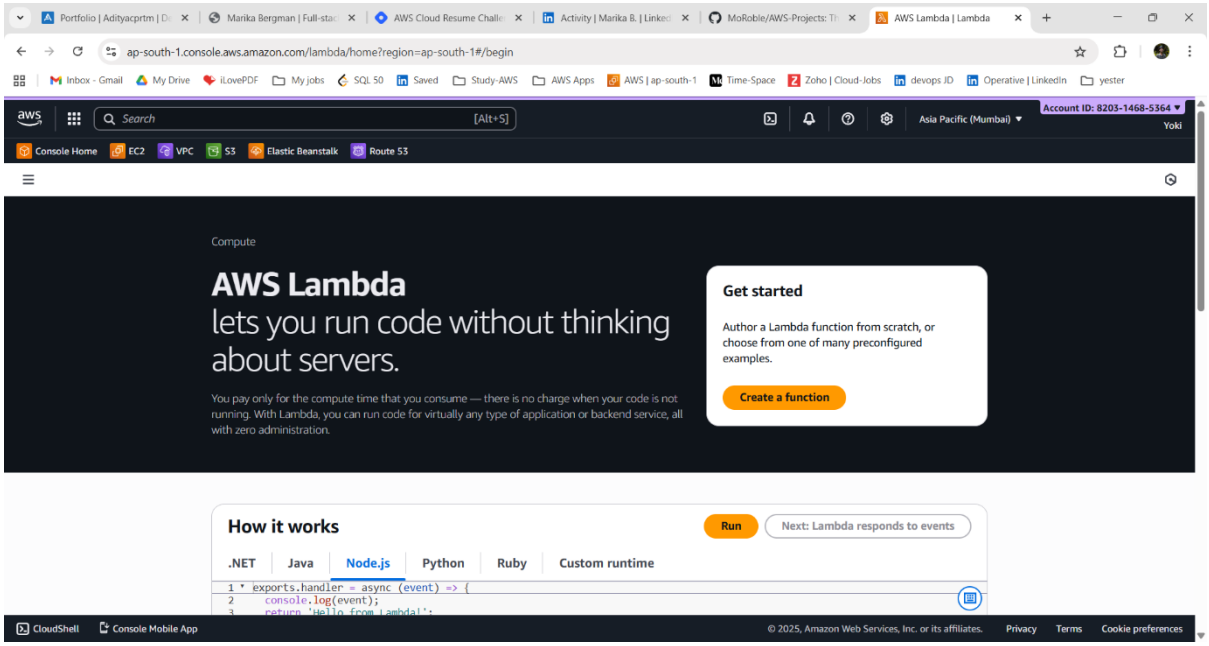
1 - Deploy Backend with AWS Lambda + API Gateway.

~ In this first stage, you will build a **Serverless API** that return cloud fun facts.



Step - 1: Create an AWS Lambda Function:

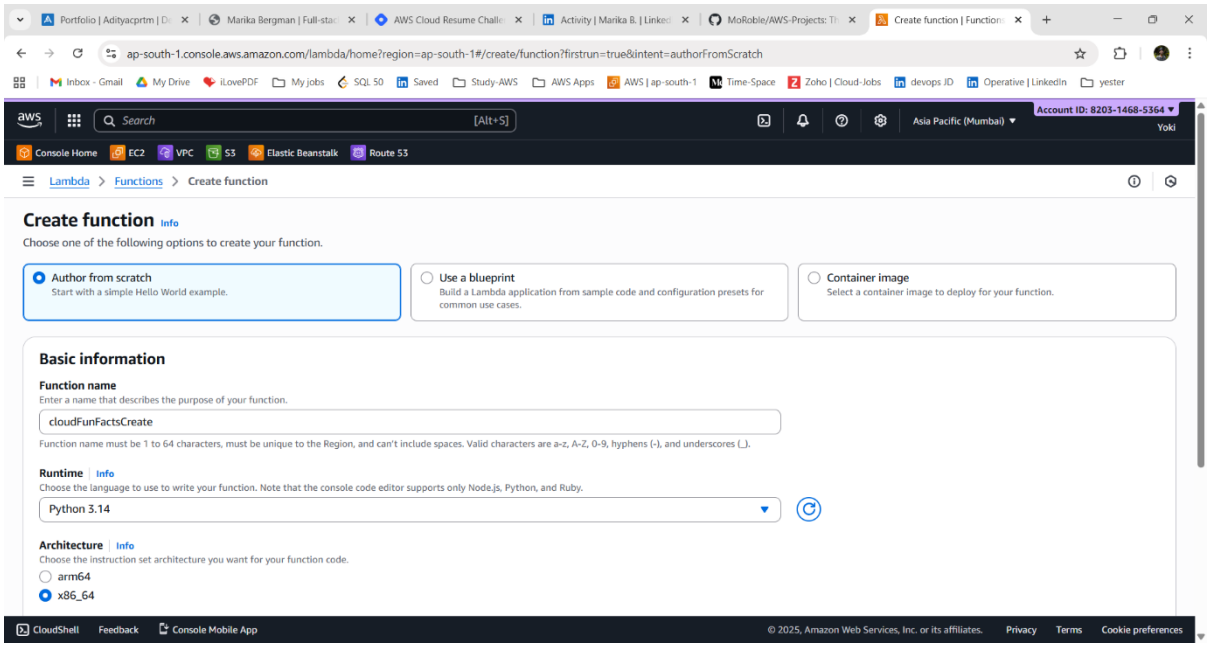
1. Go to AWS Management Console → Seach for **"Lambda".**

2. click "**Create Function"** in AWS Lambda.

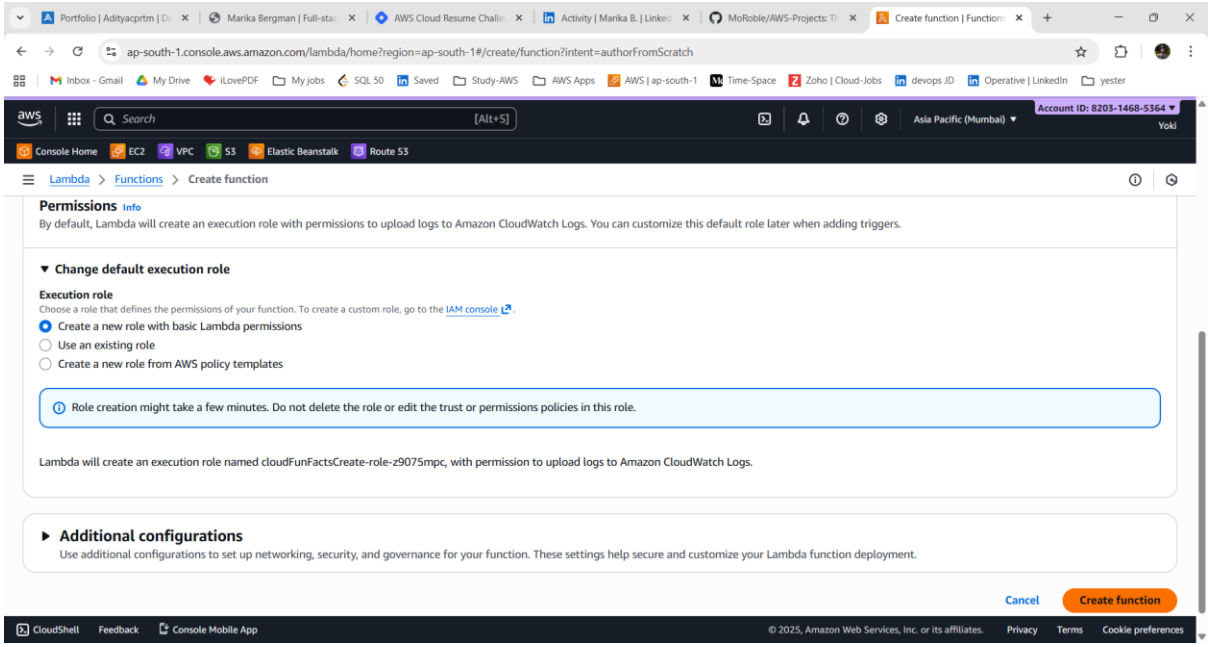
3. Select **Author from scratch**.

Function name: CloudFunFacts

Runtime: Python 3



4. Permissions: Create a new role with basic Lambda permissions.



5. Click “Create Function”.

👉 What’s happening here?

- Lambda is AWS’s serverless compute service — you don’t need to manage servers, scaling, or patching.

- By creating this function, you’re telling AWS: “Whenever this function is triggered, run my Python code and return the output.”

- The basic Lambda role gives it permission to write logs into CloudWatch so you can debug later.

Step - 2: Add code in AWS Lambda Function:

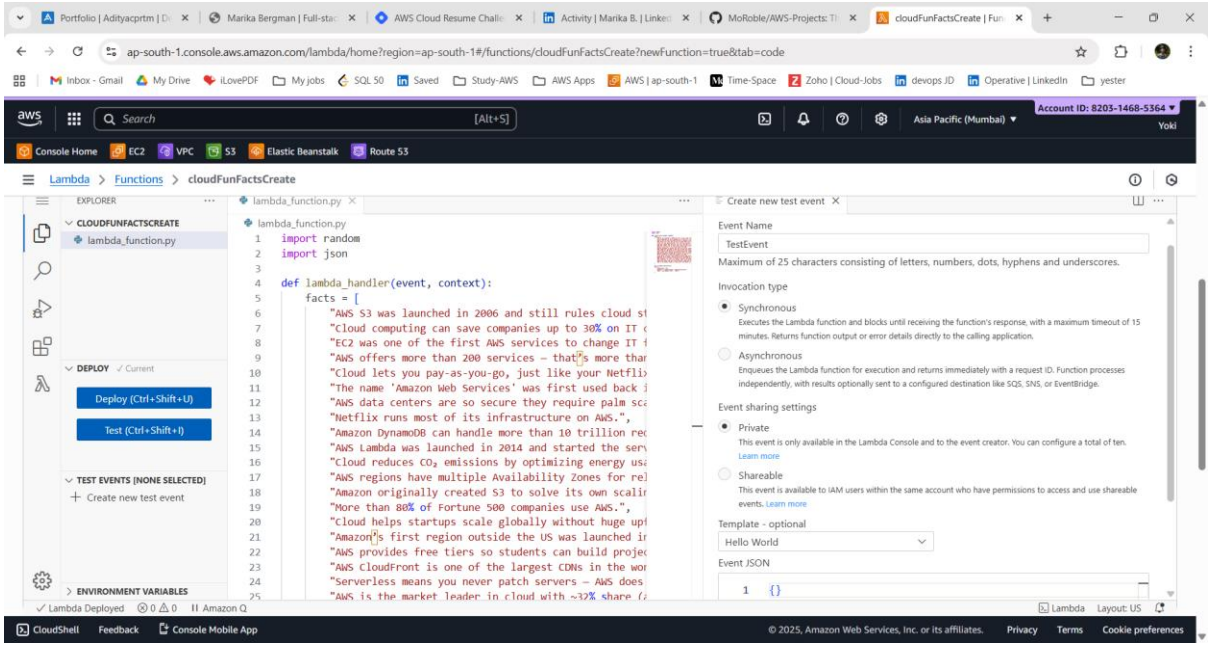
1. In the Lambda **"Code Source"** section.

2. copy and paste the below code.

Lambda Code:

```
import random
import json
def lambda_handler(event, context):


    facts = [
        "AWS S3 was launched in 2006 and still rules cloud storage.",
        "Cloud computing can save companies up to 30% on IT costs.",
        "EC2 was one of the first AWS services to change IT forever.",
        "AWS offers more than 200 services — that's more than Starbucks drinks!",
        "Cloud lets you pay-as-you-go, just like your Netflix subscription.",
        "The name 'Amazon Web Services' was first used back in 2002.",
        "AWS data centers are so secure they require palm scanners.",
        "Netflix runs most of its infrastructure on AWS.",
        "Amazon DynamoDB can handle more than 10 trillion requests per day.",
        "AWS Lambda was launched in 2014 and started the serverless trend.",
        "Cloud reduces CO₂ emissions by optimizing energy usage.",
        "AWS regions have multiple Availability Zones for reliability.",
        "Amazon originally created S3 to solve its own scaling issues.",
        "More than 80% of Fortune 500 companies use AWS.",
        "Cloud helps startups scale globally without huge upfront costs.",
        "Amazon's first region outside the US was launched in Ireland (2007).",
        "AWS provides free tiers so students can build projects affordably.",
        "AWS CloudFront is one of the largest CDNs in the world.",
        "Serverless means you never patch servers — AWS does it for you!",
        "AWS is the market leader in cloud with ~32% share (as of 2025)."
    ]
        fact = random.choice(facts)
    return {
        "statusCode": 200,
        "headers": {"Content-Type": "application/json"},
        "body": json.dumps({"fact": fact})
    }
```



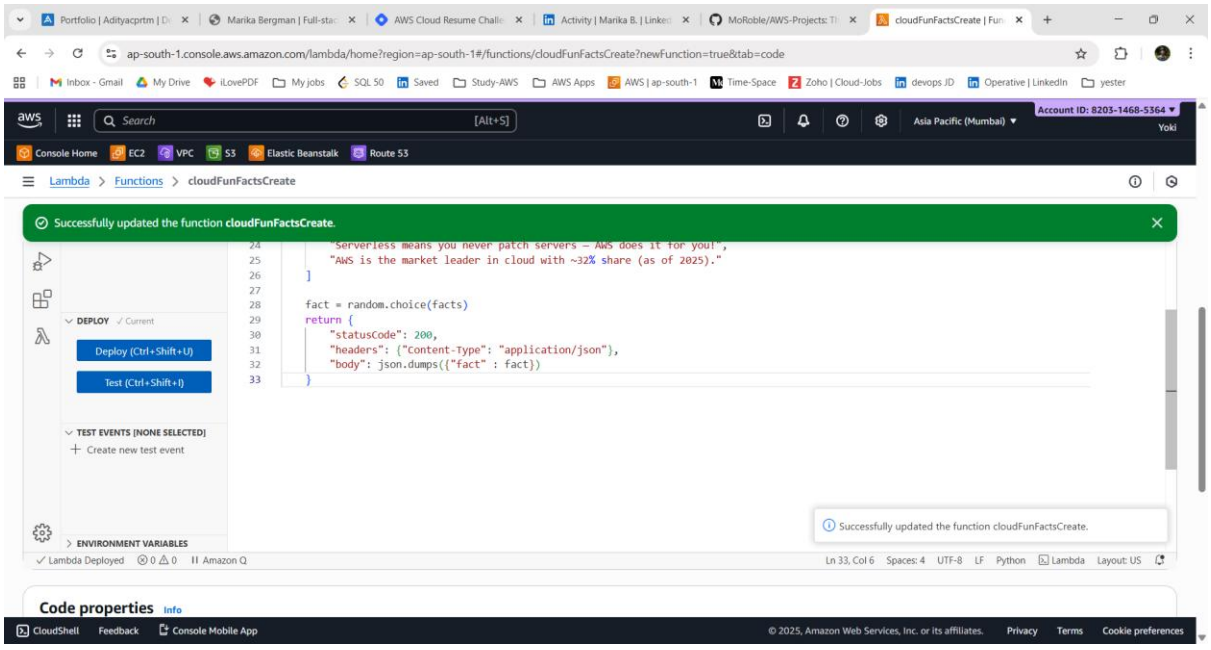
👉 **What this does:**

- Picks a random fun fact from our list.

- Returns it as a JSON response with status code 200 (success).

- The event and context arguments let Lambda handle inputs (like API calls) — even if we're not using them yet.

3. Click **Deploy** to save the changes.



👉 **What's happening here?**

- Picks a random fun fact from our list and returns it as a JSON response with status code 200 (success).

- The event and context arguments let Lambda handle inputs (like API calls) — even if we're not using them yet.

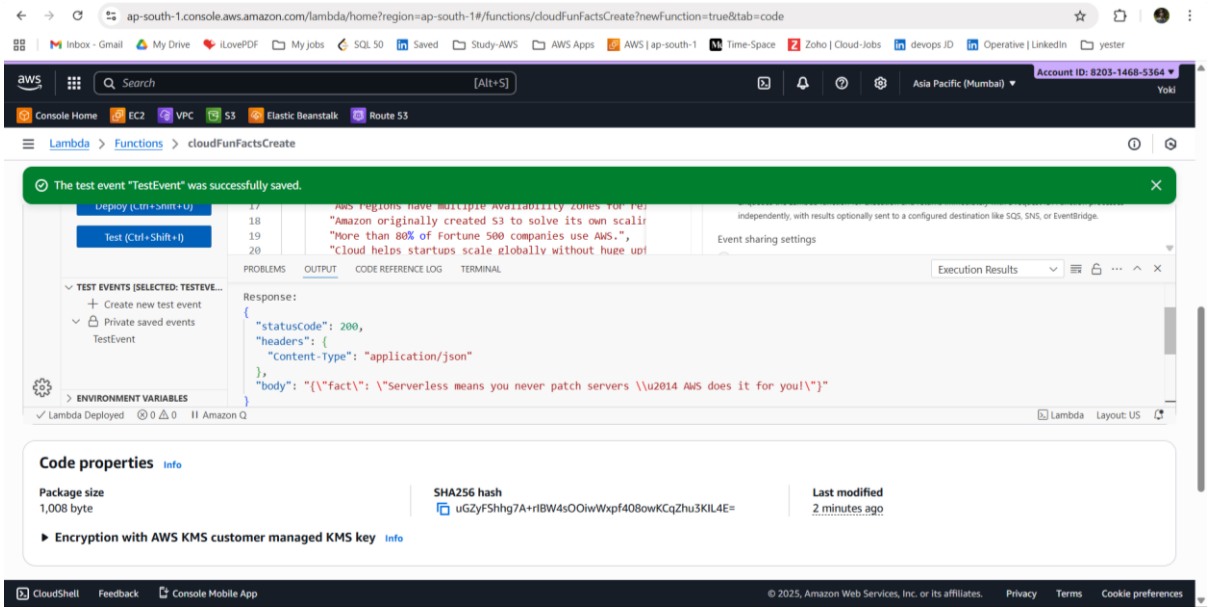
Step - 3: Test Lambda:

1. Click **“Test”** → Create a new Test event → Name: **TestEvent1**

2. Leave the default JSON ({})

3. Run the test > Save and Click Invoke.

→ You should see a random fun fact returned in the response.

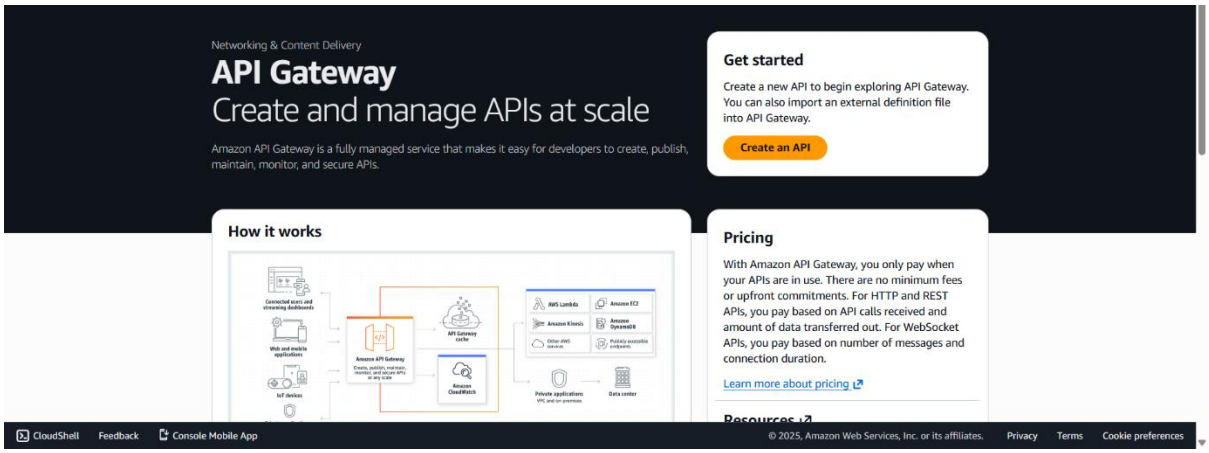


👉 What’s happening here?

- You’re manually invoking the Lambda in the console
- AWS runs your code in a temporary environment, then destroys it (that’s the beauty of serverless).

Step - 4: Create API Gateway:

→ Now we’ll make this Lambda accessible through a public API endpoint.

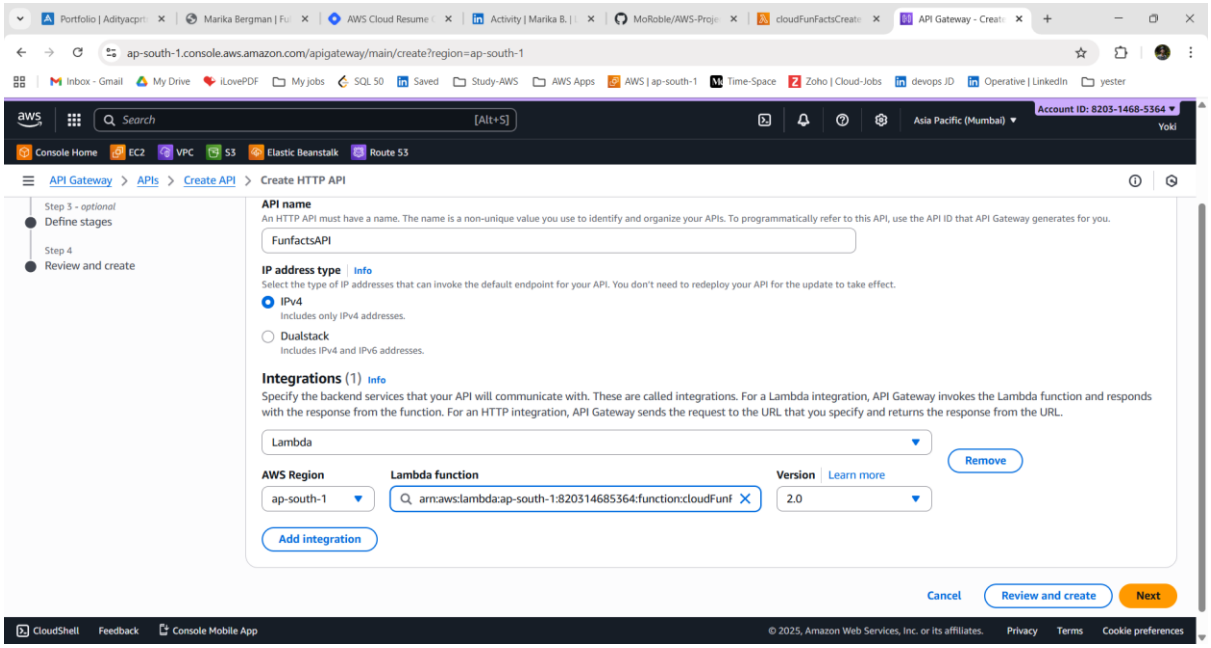


1. Go to API Gateway → Click Create API.

2. Select HTTP API (simpler and cheaper than REST API).

API Name → FunfactsAPI

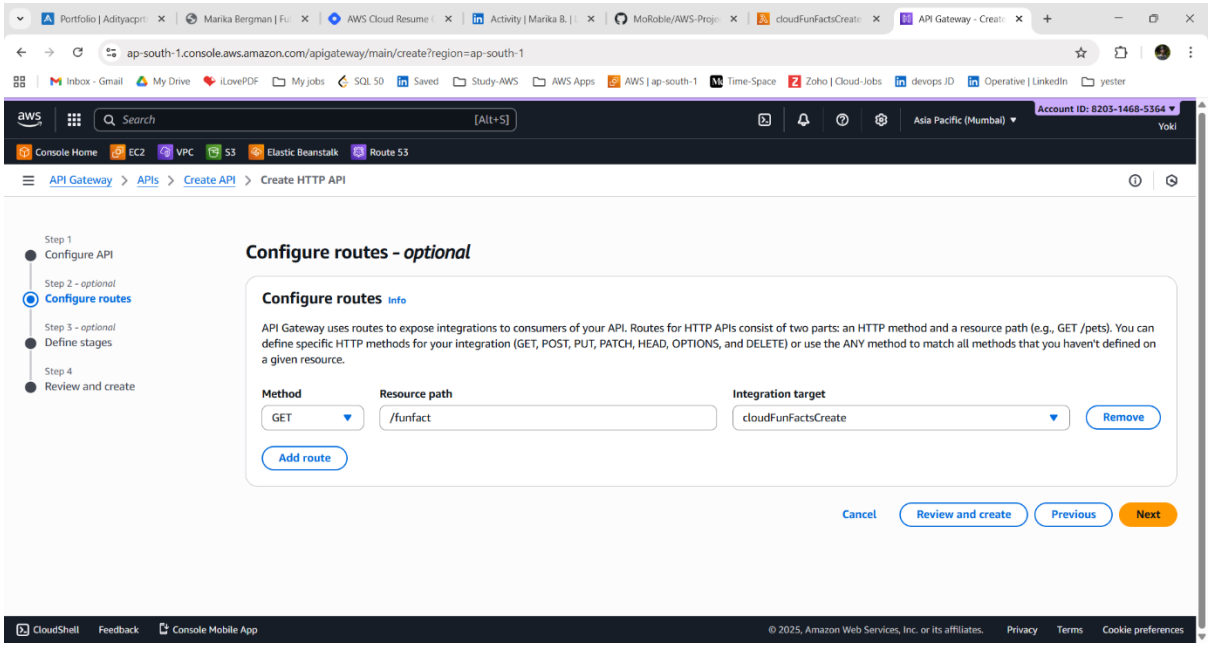
Integration → Select your Lambda function CloudFunFacts > Click Next.



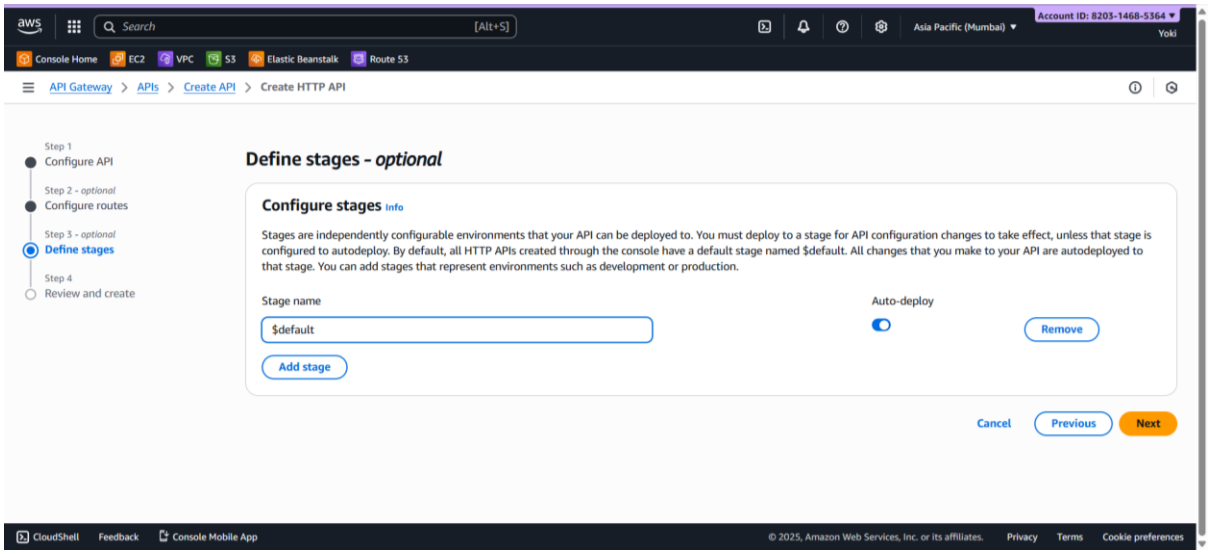
3. Add a route:

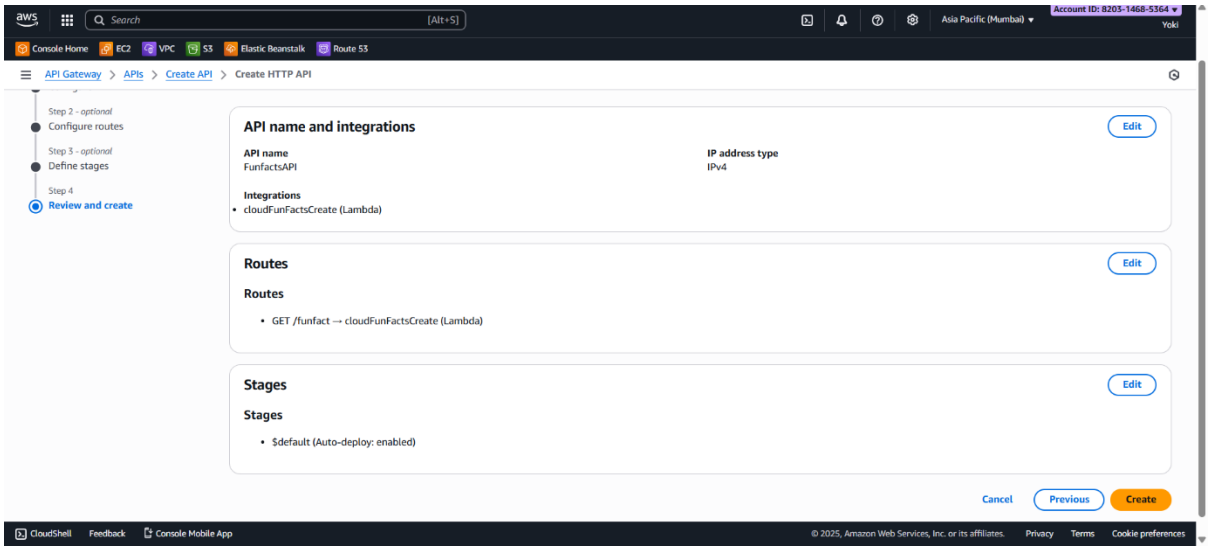
Method: GET

Path: /funfact

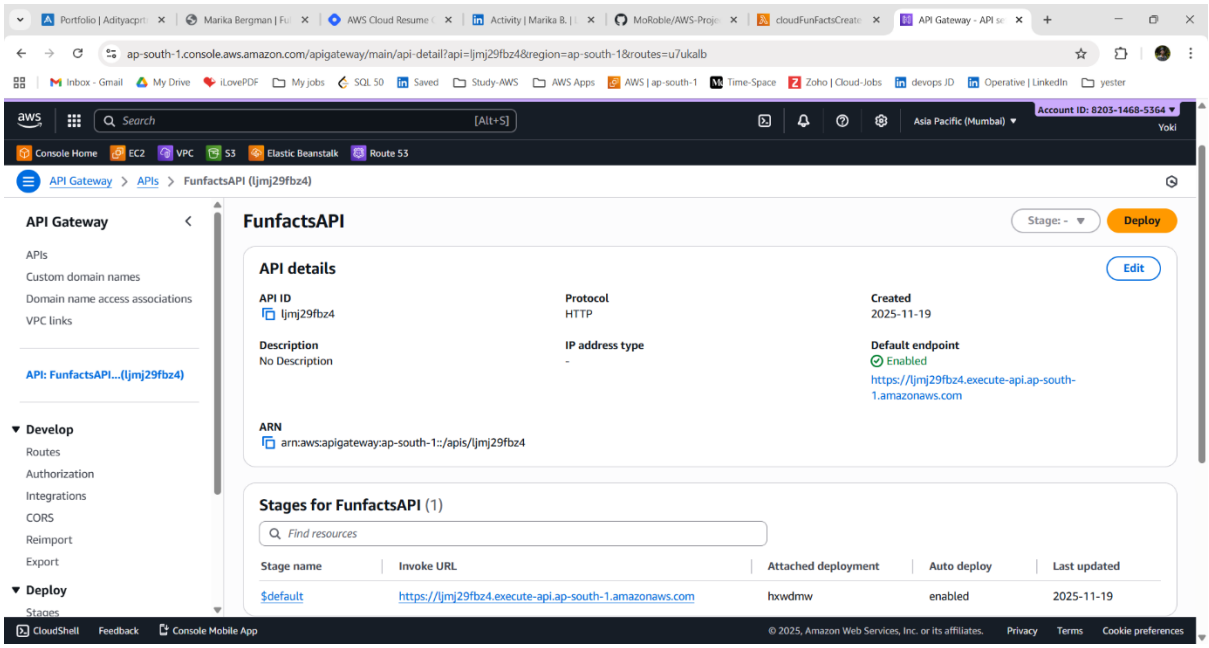


~ Stages = Default.





4. Click **"Create" to** create API GATEWAY.



👉 What's happening here?

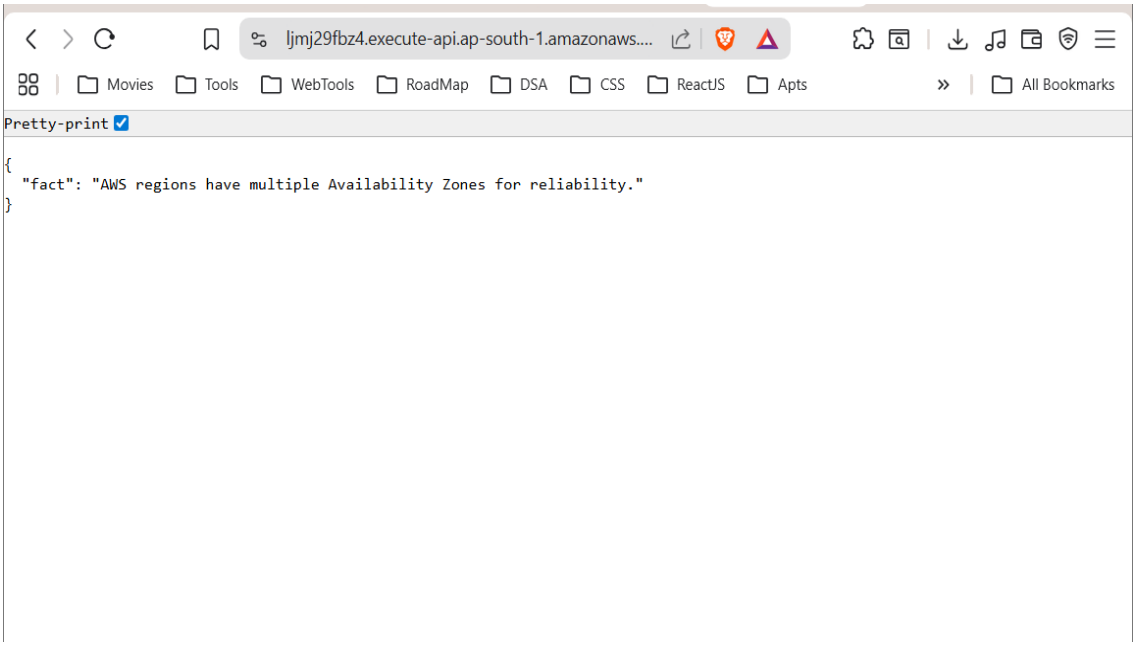
- API Gateway acts as the front door to your Lambda.

- When someone hits **/funfact with a GET request**, API Gateway forwards it to Lambda, then returns the response.

Step - 5: Copy Endpoint:

1. In your API → go to Stages → Select **default**

2. Copy the Invoke URL (e.g., https://abcdedf.execute-api.ap-south -1.amazonaws.com).



Open your browser/Postman and test:

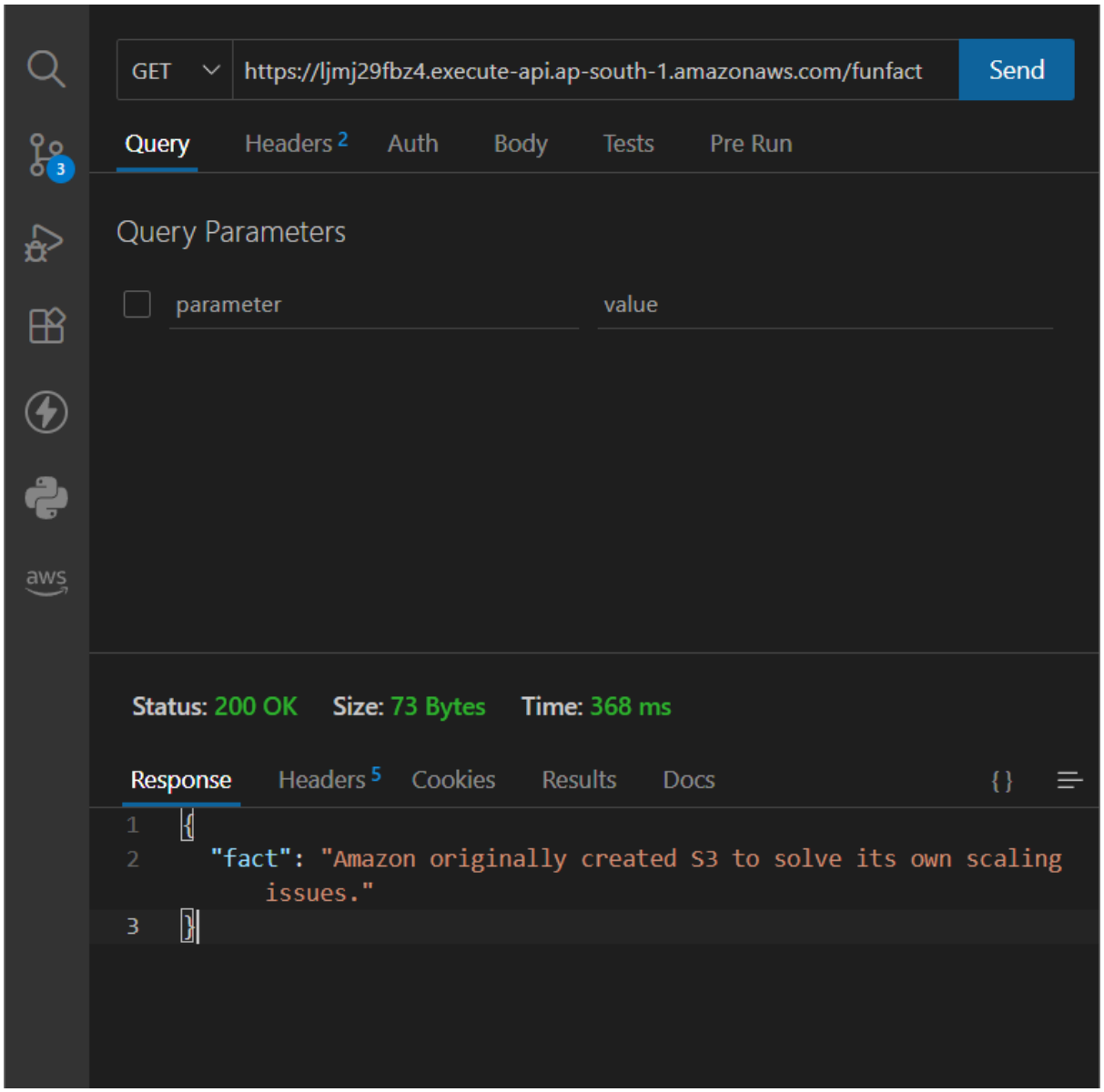
https://abcdedf.execute-api.ap-south -1.amazonaws.com

→ You should see something like:

{

"fact": "The name 'Amazon Web Services' was first used back in 2002."

}



Outcome of Stage - 1:

At this point, you've built a fully functional serverless API:

~ A **Python Lambda** that returns random cloud facts.

~ An **API Gateway endpoint** that makes it accessible over the internet.