UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
DEHRADUN

DISTRIBUTED SYSTEMS

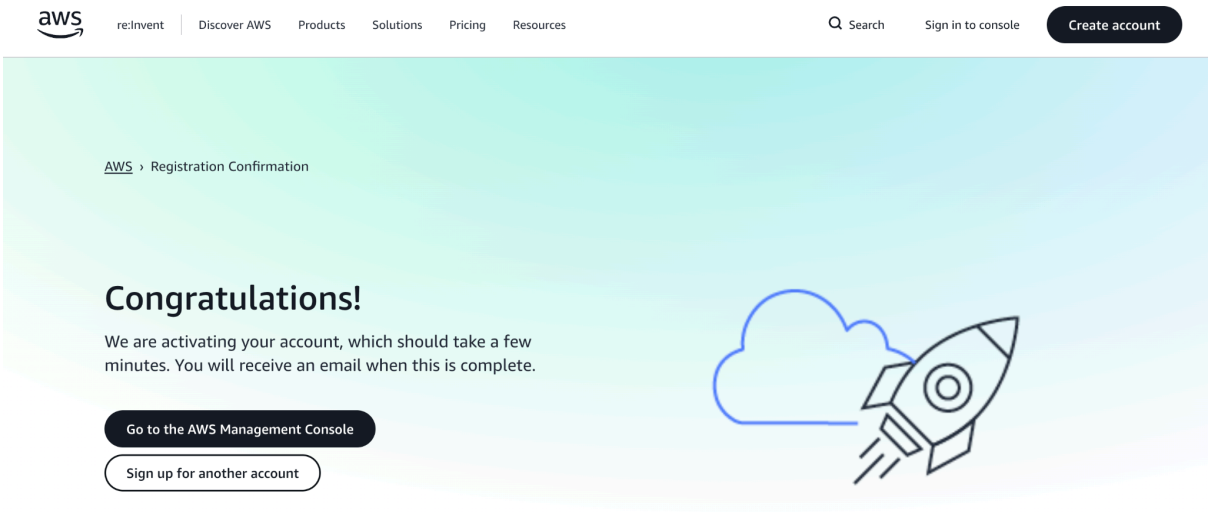
Serverless Computing with AWS Lambda

MTECH-COMPUTER SCIENCE ENGINEERING
CYBER SECURITY AND FORENSICS

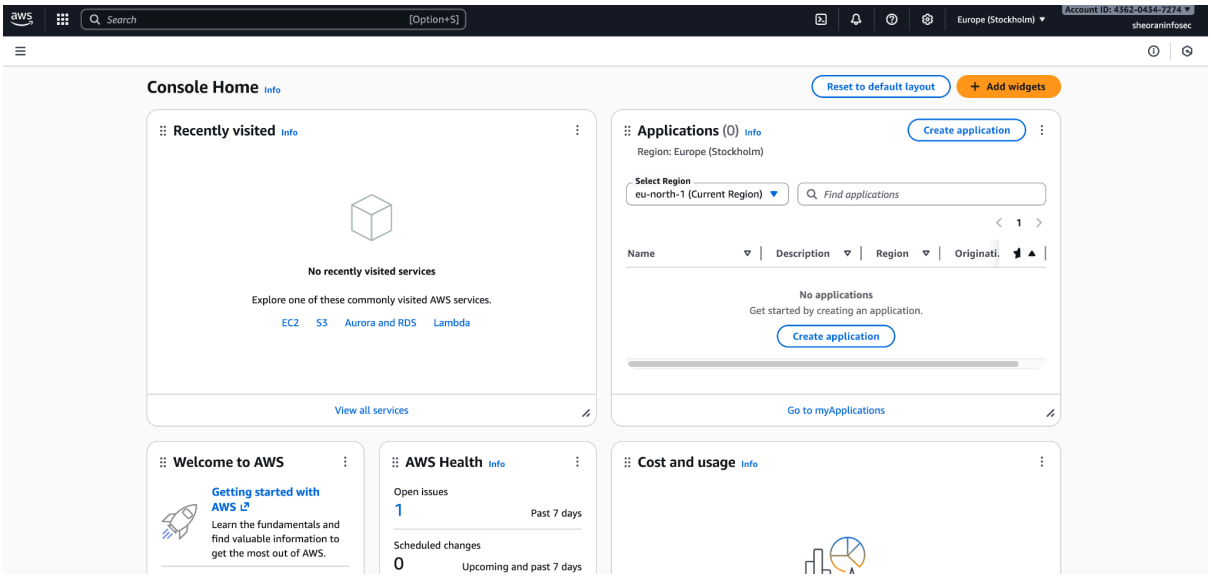
Name: Jigesh Sheoran
SAP ID: 590025428

Objective — Develop and deploy a serverless function using AWS Lambda. Write a simple function triggered by an event (e.g., HTTP request) to understand the principles of serverless computing and its use in cloud-based applications.

Step 1: Set up a free tier AWS Account.
Step 2: Go to: https://aws.amazon.com / Select Create AWS Account / Choose Personal Account / Select Free Tier



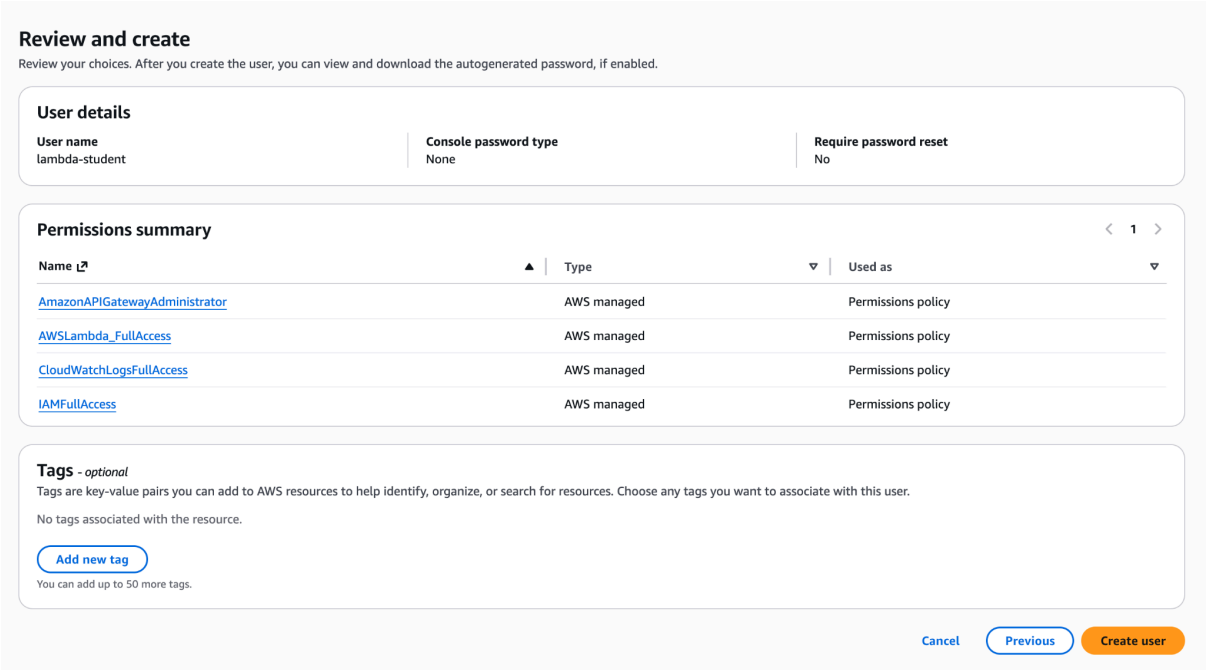
Step 3: Login in to your AWS Console once you receive confirmation email.



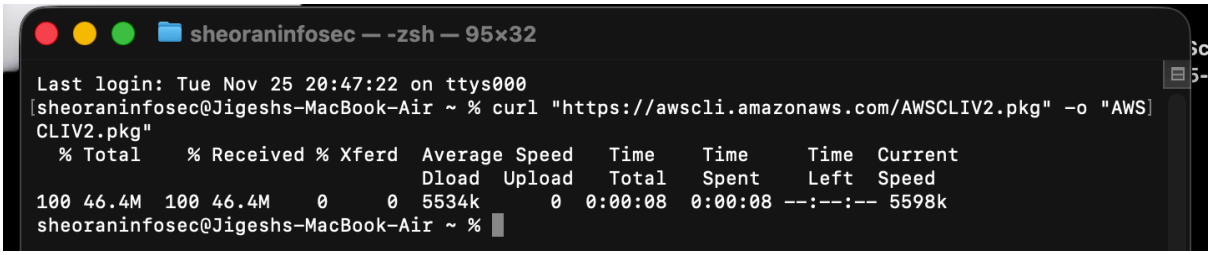
Step 3: Create a IAM User (to gain CLI Access)
IAM → Users → Create User (Username: lambda-student)

Step 4: Then generate Access Keys under
(IAM → Users → lambda-student → Security Credentials → Create Access Key)

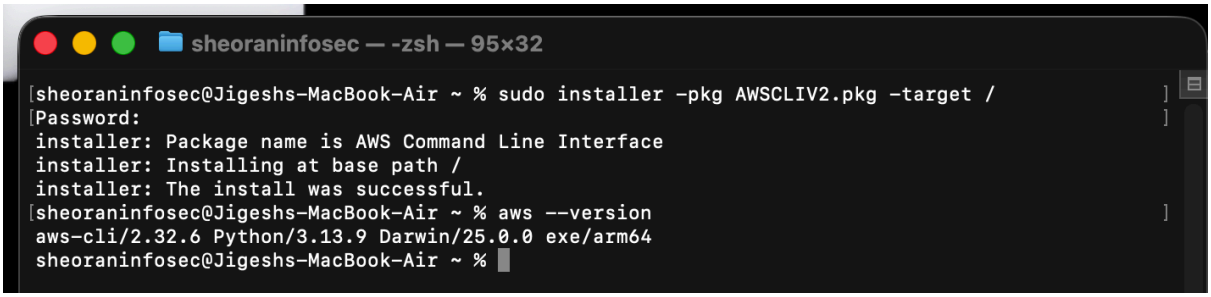
Step 5: Save the access keys to somewhere safe. Will need them later to access your aws account through Command Line Interface.



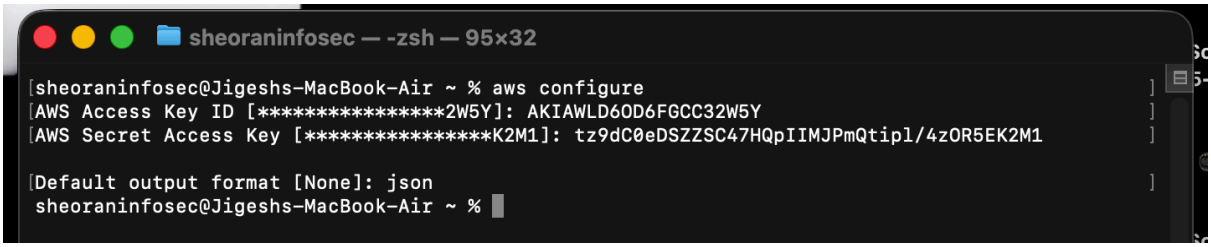
Step 6: Install AWS CLI on macOS.



Step 7: Verify if it installed correctly by checking the version.



Step 8: Configure AWS CLI with your IAM credentials.

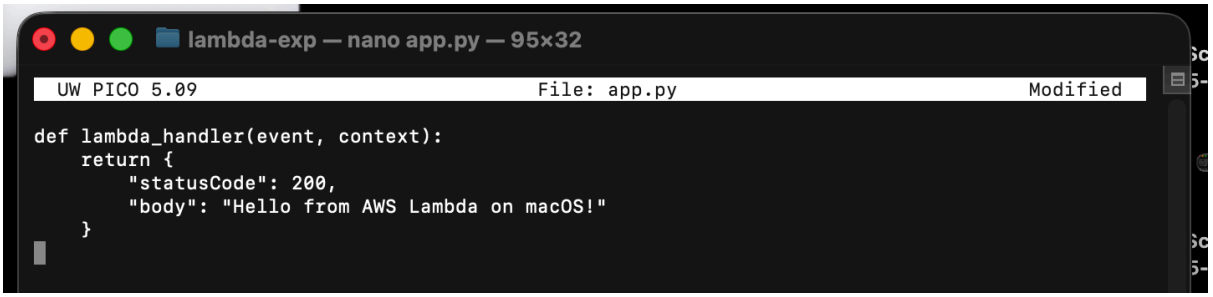


Step 9: Set the default Region to : ap-south-1 (Mumbai) & output format : json.

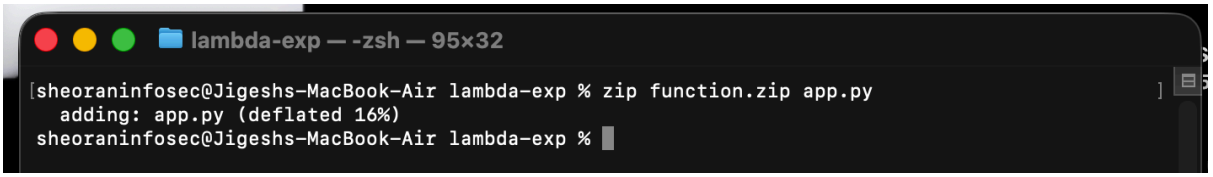
Step 10: Create Lambda Function Folder

```
[Default output format [None]: json
[sheoraninfosec@Jigeshs-MacBook-Air ~ % mkdir lambda-exp
[sheoraninfosec@Jigeshs-MacBook-Air ~ % cd lambda-exp
[sheoraninfosec@Jigeshs-MacBook-Air lambda-exp % pwd
 /Users/sheoraninfosec/lambda-exp
[sheoraninfosec@Jigeshs-MacBook-Air lambda-exp % nano app.py
 sheoraninfosec@Jigeshs-MacBook-Air lambda-exp %
```

Step 11: Create lambda_handler file.

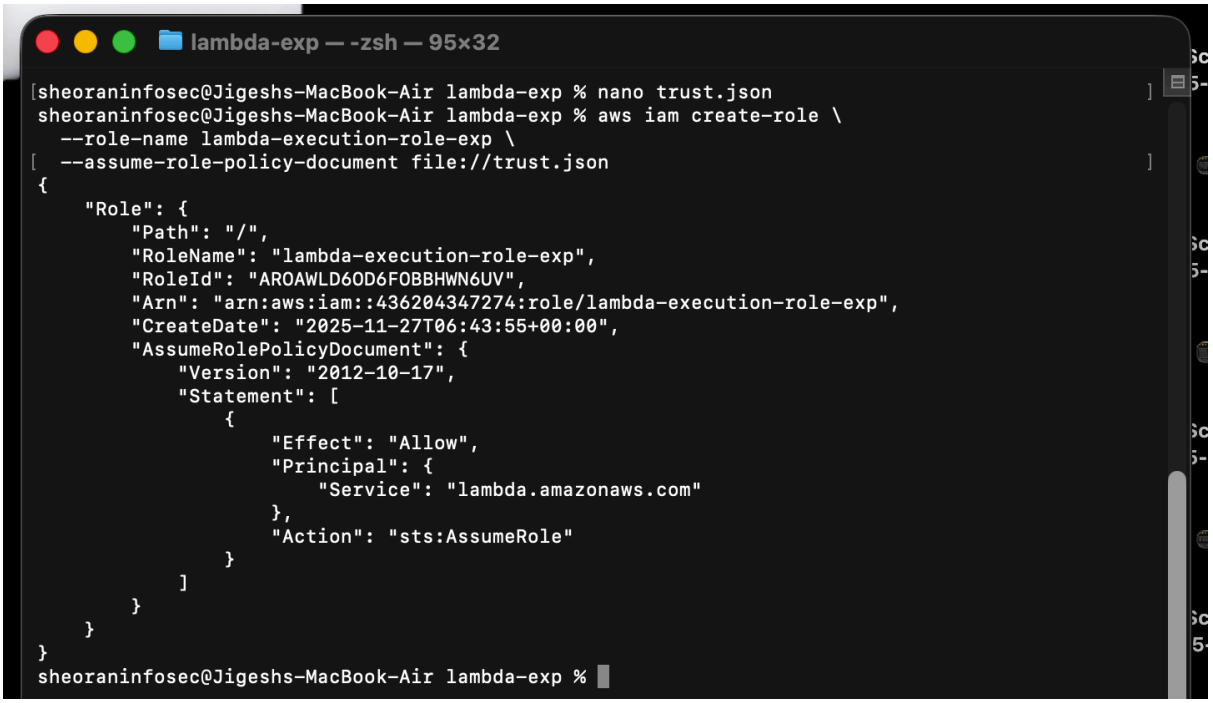


Step 12: Zip the Lambda Code



Step 13: Write tryst policy, recommended but not necessary.

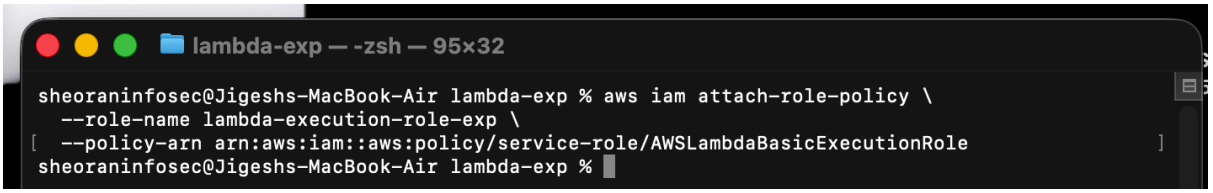
Step 14: Create IAM Execution Role (for Lambda)



```
lambda-exp — -zsh — 95×32

sheoraninfosec@Jigeshs-MacBook-Air lambda-exp % nano trust.json
sheoraninfosec@Jigeshs-MacBook-Air lambda-exp % aws iam create-role \
  --role-name lambda-execution-role-exp \
  --assume-role-policy-document file://trust.json
{
    "Role": {
        "Path": "/",
        "RoleName": "lambda-execution-role-exp",
        "RoleId": "AROAWLD6OD6FOBBHWN6UV",
        "Arn": "arn:aws:iam::436204347274:role/lambda-execution-role-exp",
        "CreateDate": "2025-11-27T06:43:55+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "lambda.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
sheoraninfosec@Jigeshs-MacBook-Air lambda-exp %
```

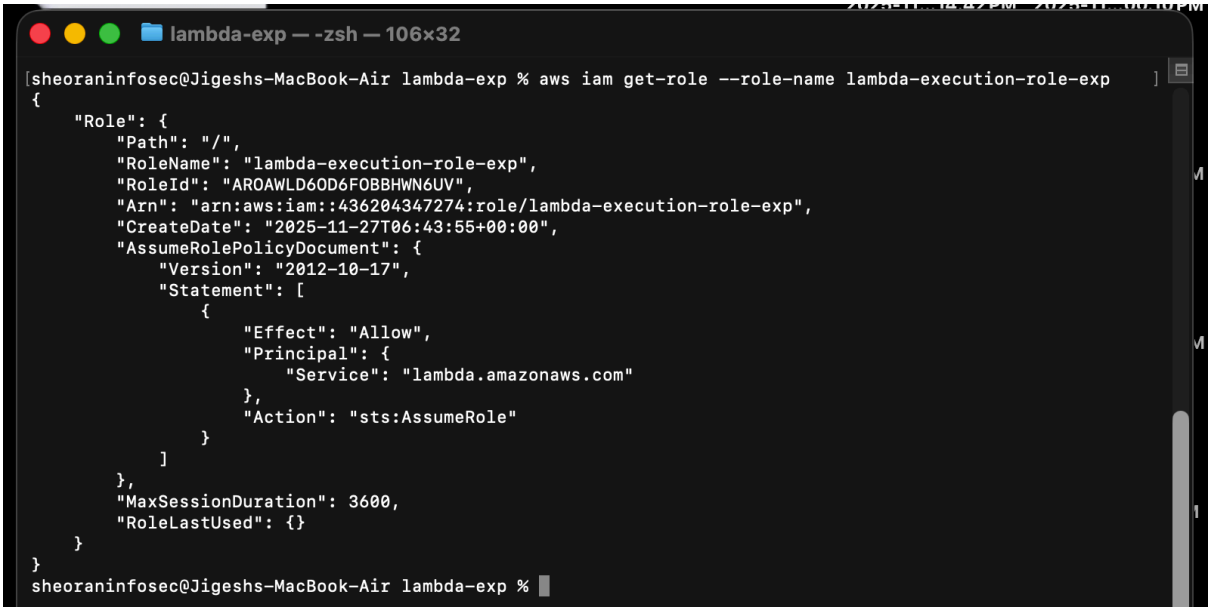
Step 15: Also attach the basic execution policy.



```
lambda-exp — -zsh — 95×32

sheoraninfosec@Jigeshs-MacBook-Air lambda-exp % aws iam attach-role-policy \
  --role-name lambda-execution-role-exp \
  --policy-arn arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
sheoraninfosec@Jigeshs-MacBook-Air lambda-exp %
```

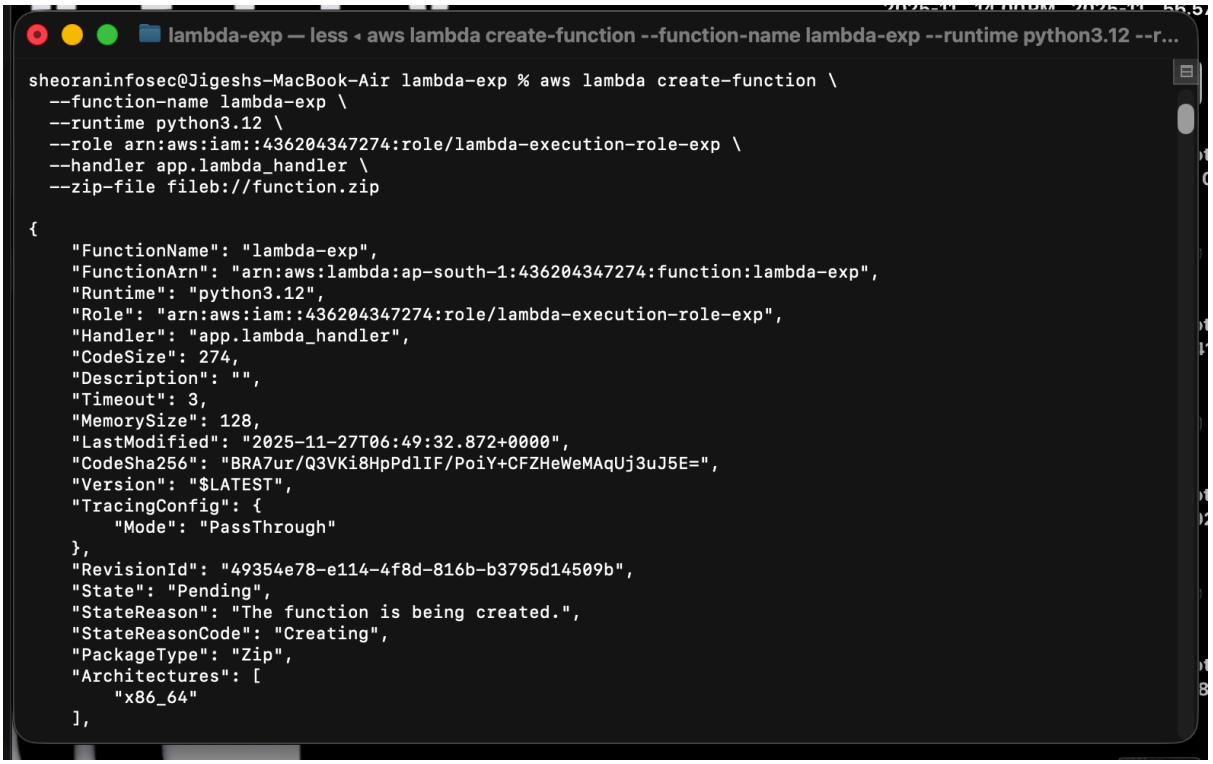
Step 16: Create Lambda Function (from CLI)



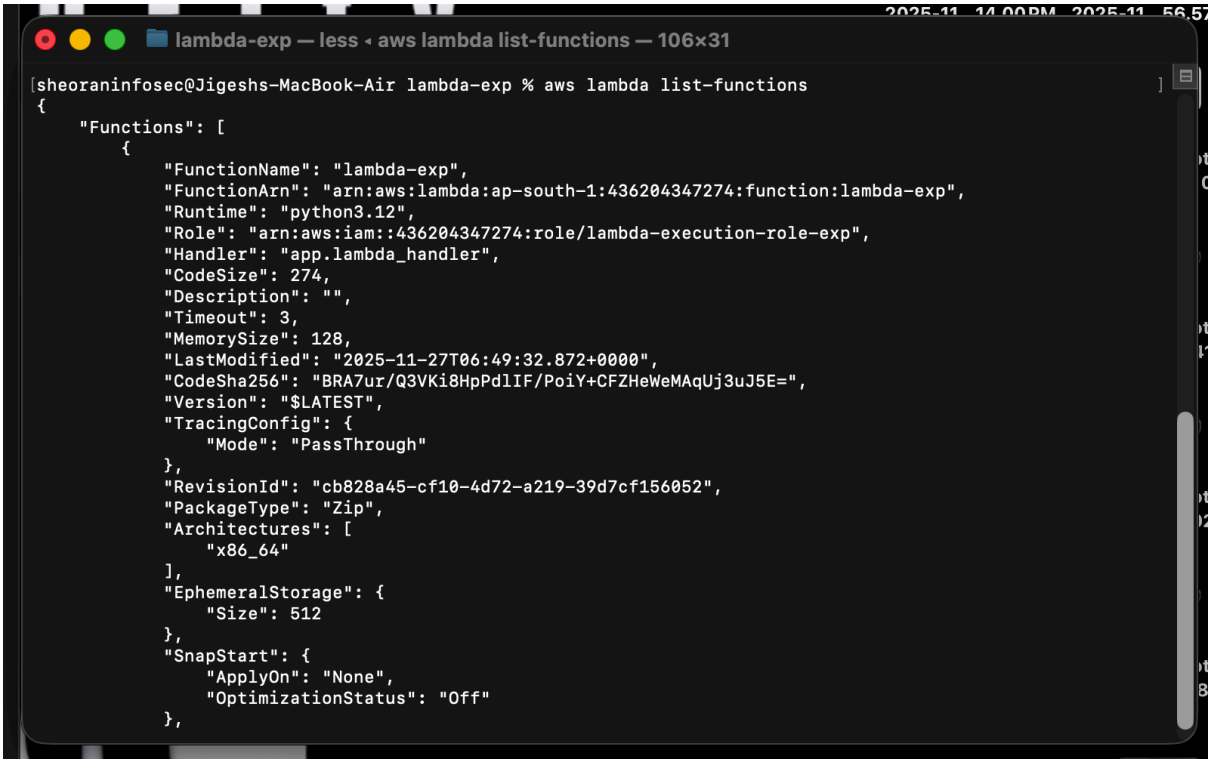
```
lambda-exp — -zsh — 106×32

sheoraninfosec@Jigeshs-MacBook-Air lambda-exp % aws iam get-role --role-name lambda-execution-role-exp
{
    "Role": {
        "Path": "/",
        "RoleName": "lambda-execution-role-exp",
        "RoleId": "AROAWLD6OD6FOBBHWN6UV",
        "Arn": "arn:aws:iam::436204347274:role/lambda-execution-role-exp",
        "CreateDate": "2025-11-27T06:43:55+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "lambda.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "MaxSessionDuration": 3600,
        "RoleLastUsed": {}
    }
}
sheoraninfosec@Jigeshs-MacBook-Air lambda-exp %
```

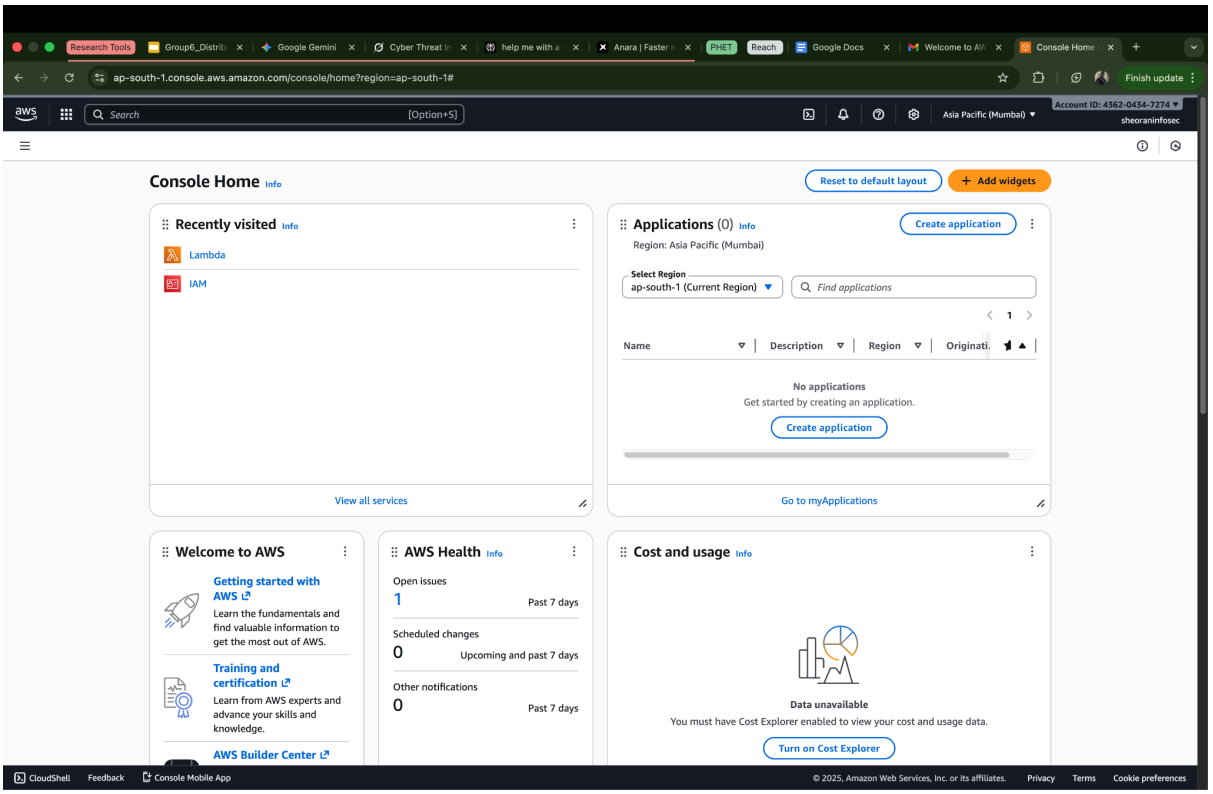
Step 17: Copy the Arn from the output of this created lambda function and paste it to " aws lambda create-function ".



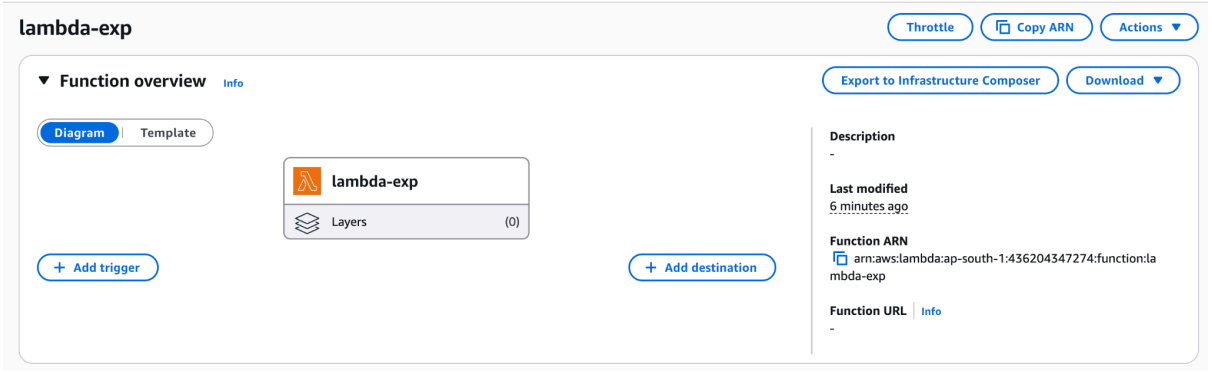
Step 18: Check if the lambda function is deployed correctly or not.



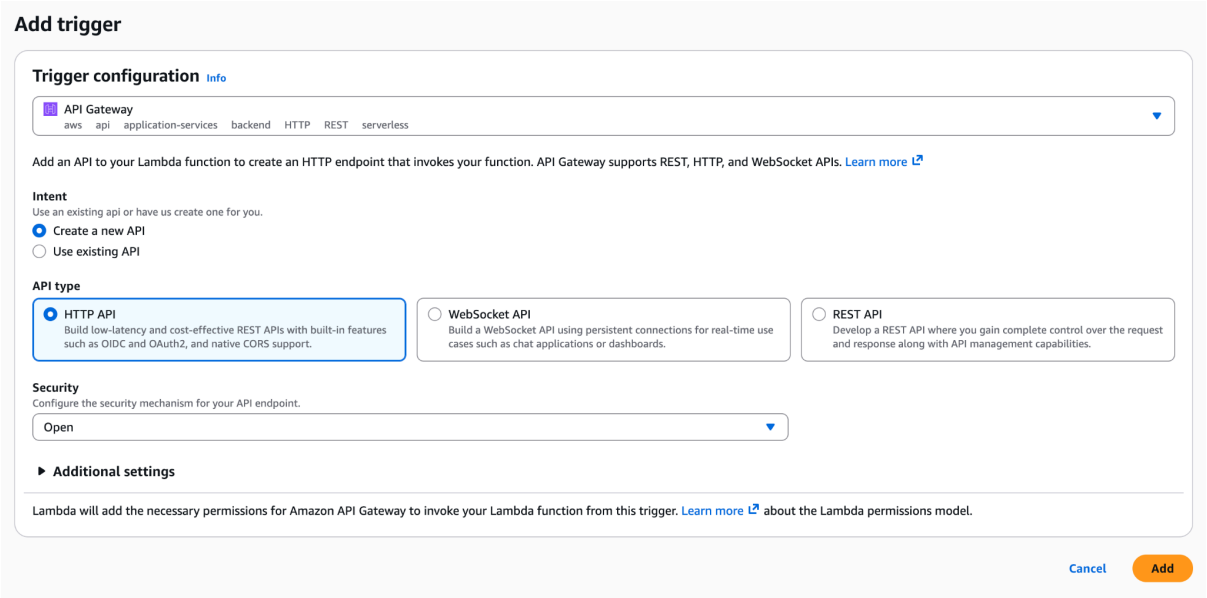
Step 19: Create API Gateway Trigger (via AWS Console)



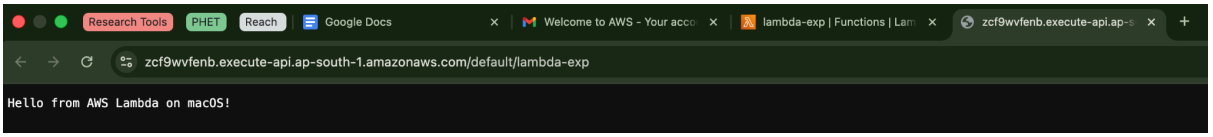
AWS Console → Lambda → Open lambda-exp →Click Add Trigger → Select API Gateway



Step 20: Choose -
Create a new API
API Type: HTTP API
Security: Open
Click Add



Step 21: You'll get a public HTTPS URL. Paste it into a new tab and you will see that you AWS Lambda is running perfectly.



Step 22: Not necessary / but recommended.

API Gateway cleanup:
AWS Console → API Gateway → Your API → Delete

If you don't need CLI access after performing this experiment it is advised to delete API Keys to avoid unnecessary free access.

—----------END OF DOCUMENT —----------