

# CBS 321 Cloud Computing & Security

## SHRAVASTI OHOL 2022BCY0012

### 1. Adding MFA to root user account

#### a. MFA Details

The screenshot shows the AWS IAM 'My security credentials' page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' selected. The main area displays account details (Account name: Shravasti, AWS account ID: 619071313125), multi-factor authentication (MFA) information (one device assigned), and access key details (no access keys currently). Navigation links at the bottom include CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

#### b. Signing in again

The left side of the image shows the AWS sign-in page. It has a 'Sign in' header, two radio button options ('Root user' and 'IAM user'), a 'Root user email address' input field containing 'pamhalpert1508@gmail.com', and a 'Next' button. Below the input field, there's a small note about agreeing to the AWS Customer Agreement and Privacy Notice. At the bottom, there's a link to 'Create a new AWS account'. The right side features a promotional banner for 'AWS re:Invent DEC. 2 - DEC. 6, 2024 | LAS VEGAS, NV'. The banner highlights 'Automate code reviews with Amazon Q Developer - get immediate feedback on your code so you can iterate faster' and includes a 'Learn more' button.

**aws**

### Multi-factor authentication

Your account is secured using multi-factor authentication (MFA). To finish signing in, turn on or view your MFA device and type the authentication code below.

Email address: pamhalper1508@gmail.com

MFA code

**Submit**

Troubleshoot MFA

**Cancel**



AWS re:Invent  
DEC. 2 – DEC. 6, 2024 | LAS VEGAS, NV

Automate code reviews with Amazon Q Developer - get immediate feedback on your code so you can iterate faster

**Learn more**

**IAM Dashboard**

**Security recommendations**

- Root user has MFA
- Root user has no active access keys

**IAM resources**

User groups	Users	Roles	Policies	Identity providers
0	0	2	0	0

**What's new**

- Introducing resource control policies (RCPs) to centrally restrict access to AWS resources. 2 months ago
- AWS IAM now supports PrivateLink in the AWS GovCloud (US) Regions. 2 months ago
- Streamline automation of policy management workflows with service reference information. 3 months ago
- Amazon S3 Access Grants introduce the ListCallerAccessGrants API. 5 months ago

**AWS Account**

Account ID: 619071313125  
Account Alias: Create  
Sign-in URL for IAM users in this account: <https://619071313125.sigin.aws.amazon.com/console>

**Quick Links**

**Tools**

Policy simulator: The simulator evaluates the policies that you choose and determines the effective permissions for each of the actions that you specify.

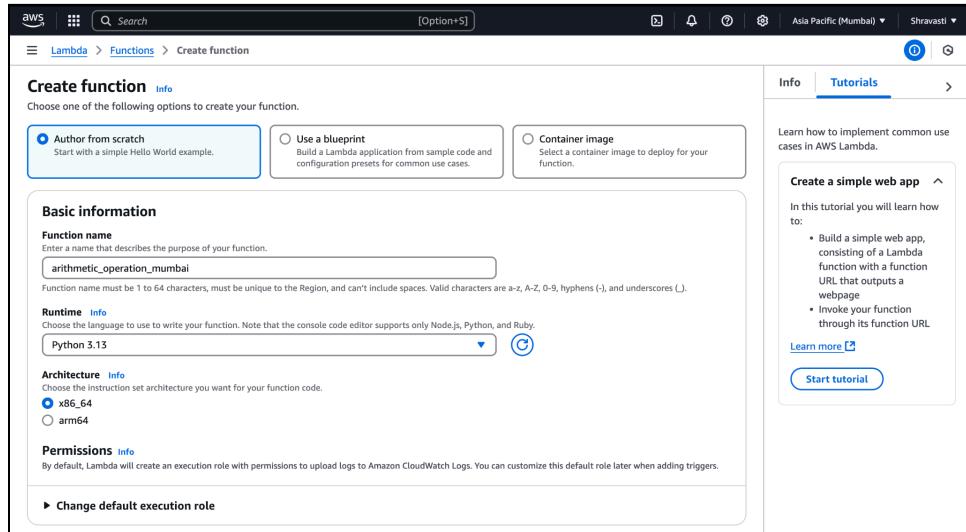
## 2. Lambda function

Allows you to run code without using a dedicated server.

Essentially, Platform-As-A-Service. Three ways, Author from scratch, Use a blueprint, Container image.

### a. Arithmetic Operations

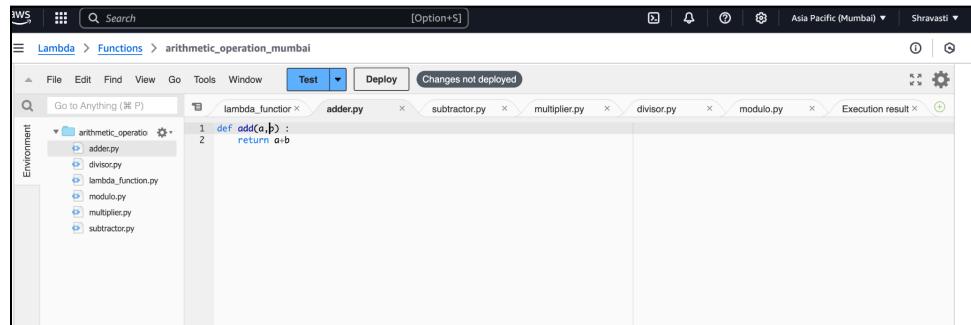
#### i. Create a function



#### ii. Write the code and save it

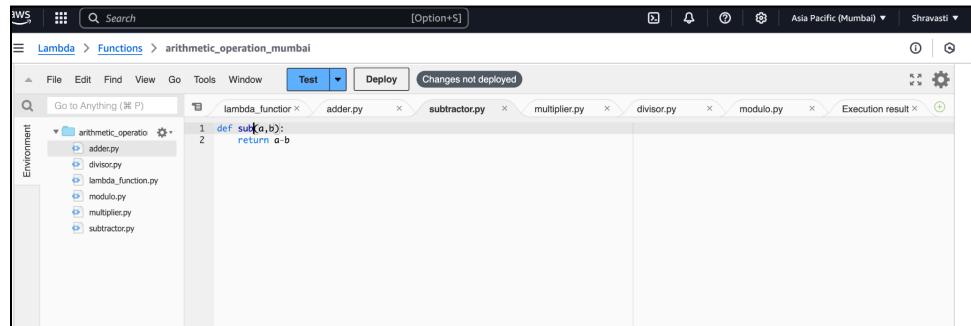
```
1 import json
2 from adder import add
3 from subtractor import sub
4 from multiplier import mul
5 from divisor import div
6 from modulo import mod
7 print("Loading Function")
8 print("-----")
9 print("Simple Calculator")
10 print("-----")
11 def lambda_handler(event, context):
12     print("Received Event: " + json.dumps(event, indent=2))
13     for item in event:
14         a = item['a']
15         b = item['b']
16         op = item['op']
17         if op == "add":
18             print("The sum of {a} and {b} is {add(a, b)}")
19         elif op == "sub":
20             print("The difference of {a} and {b} is {sub(a, b)}")
21         elif op == "mul":
22             print("The product of {a} and {b} is {mul(a, b)}")
23         elif op == "div":
24             print("The quotient of {a} and {b} is {div(a, b)}")
25         elif op == "mod":
26             print("The remainder of {a} divided by {b} is {mod(a, b)}")
27
print("-----")
```

### iii. Create modules for all the operations



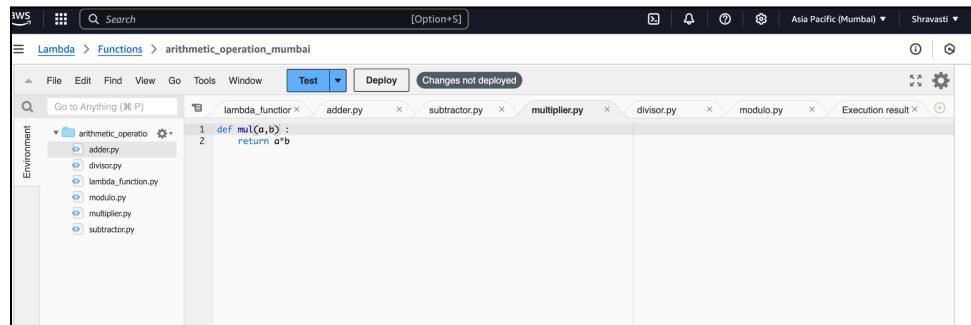
The screenshot shows the AWS Lambda Functions interface. The left sidebar shows a project named "arithmetic\_operation\_mumbai" with several files: adder.py, divisor.py, lambda\_function.py, modulo.py, multiplier.py, and subtractor.py. The main editor window displays the contents of adder.py:

```
1 def add(a,b):
2     return a+b
```



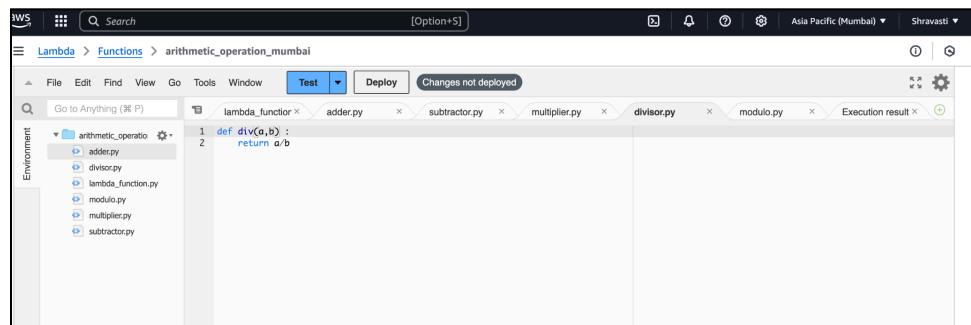
The screenshot shows the AWS Lambda Functions interface. The left sidebar shows the same project structure. The main editor window displays the contents of subtractor.py:

```
1 def sub(a,b):
2     return a-b
```



The screenshot shows the AWS Lambda Functions interface. The left sidebar shows the same project structure. The main editor window displays the contents of multiplier.py:

```
1 def mul(a,b):
2     return a*b
```



The screenshot shows the AWS Lambda Functions interface. The left sidebar shows the same project structure. The main editor window displays the contents of divisor.py:

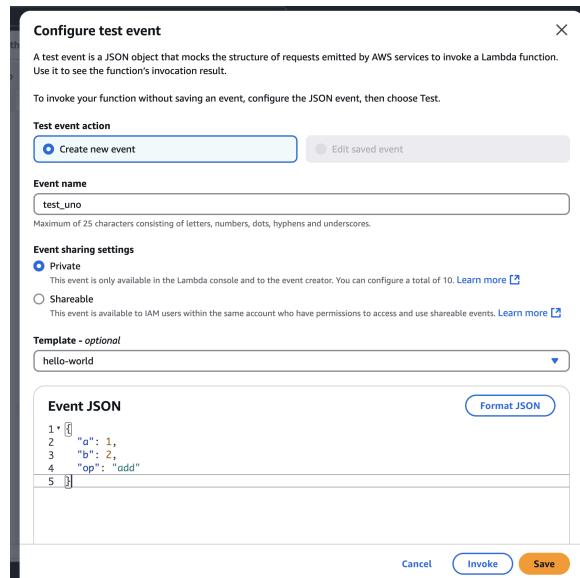
```
1 def div(a,b):
2     return a/b
```

```

1 def mod(a,b):
2     return a%b

```

#### iv. Write test cases for each operation



#### v. Deploy changes and test the code

**Code source** Info

**EXPLORER** ARITHMETIC\_OPERATION... lambda\_function.py | modulo.py

**DEPLOY** Deploy (OK) Test (OK)

**TEST EVENTS [SELECT...]**

- + Create new test...
- ▽ Private saved e...
  - test\_1no

**lambda\_function.py**

```

1 import json
2 from adder import add

```

**PROBLEMS** OUTPUT CODE REFERENCE LOG TERMINAL

Status: Succeeded  
Test Event Name: test\_1no

Response: null

Function Logs: Loading Function

Simple Calculator

```

    The sum of 1 and 2 is 3
    The difference of 10 and 2 is 8
    The product of 10 and 2 is 20
    The remainder of 10 divided by 3 is 1
    The remainder of 10 divided by 5 is 0

```

END RequestID: 8c38f7c8-3466-4ba1-838c-7db4e43d622b5 Version: \$LATEST

REPORT RequestID: 8c38f7c8-3466-4ba1-838c-7db4e43d622b5 Duration: 3.82 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 31 MB Init Duration: 93.35 ms

Request ID: 8c38f7c8-3466-4ba1-838c-7db4e43d622b5

Ln 36, Col 72 (55 selected) Tab Size: 4 UTF-8 LF Python Lambda Layout: U.S. ▾

## b. Palindrome checker

### i. Create function and write the code

The screenshot shows the AWS Lambda Function Editor interface. The code editor contains the following Python script:

```
lambda_function.py
1 import json
2
3 print("Loading Function")
4 print("-----")
5 print("Palindrome Checker")
6 print("-----")
7
8 def is_palindrome(s):
9
10    s = s.lower().replace(" ", "").replace("#", "").replace("@", "") # Normalize the string
11    return s == s[::-1]
12
13 def lambda_handler(event, context):
14    for item in event:
15        input_string = item['string']
16        result = is_palindrome(input_string)
17
18        if result:
19            print(f"The string '{input_string}' is a palindrome.")
20        else:
21            print(f"The string '{input_string}' is not a palindrome.")
22
23    return {
24        'statusCode': 200,
25        'body': json.dumps({'result': result, 'string': input_string})
26    }
27
```

The sidebar on the left shows the project structure under 'PALINDROME' with files 'lambda\_function.py' and 'lambda\_function.pyc'. It also displays 'DEPLOY [UNDEPLOYED CHANGES]' with a 'Test (OK!)' button highlighted.

### ii. Create test cases for both the possibilities i.e. palindrome and not a palindrome

The screenshot shows the 'Test event' configuration screen. The 'Event name' dropdown is set to 'test'. The 'Event JSON' field contains the following JSON array:

```
[{"string": "racecar", "output": "racecar is not a palindrome"}, {"string": "mom", "output": "mom is a palindrome"}, {"string": "depends", "output": "depends is not a palindrome"}, {"string": "hey", "output": "hey is not a palindrome"}]
```

### iii. Deploy the code and test it

The screenshot shows the AWS Lambda Function Editor interface. The function name is "palindrome".

**Code source:**

```
lambda_function.py
13 def lambda_handler(event, context):
14     if isinstance(event, str):
15         event = json.loads(event)
16     if not isinstance(event, dict):
17         return {"isPalindrome": False}
18
19     response = {
20         "statusCode": 200,
21         "body": f'{{"string": "{event}", "isPalindrome": {str(isPalindrome)} }}'
22     }
23
24     return response
```

**PROBLEMS:** None

**OUTPUT:** Status: Succeeded  
Test Event Name: test

**Response:**

```
{"statusCode": 200,
"body": "{\"string\": \"racecar\", \"isPalindrome\": true}, {\"string\": \"mom\", \"isPalindrome\": true}, {\"string\": \"depends\", \"isPalindrome\": false}, {\"string\": \"hey\", \"isPalindrome\": false}"}
```

**FUNCTION LOGS:**

```
START RequestId: e59c9cb1-47e5-4498-9f6d-b737fac324bb Version: $LATEST
The string "racecar" is a palindrome.
The string "mom" is a palindrome.
The string "depends" is not a palindrome.
The string "hey" is not a palindrome.

END RequestId: e59c9cb1-47e5-4498-9f6d-b737fac324bb
REPORT RequestId: e59c9cb1-47e5-4498-9f6d-b737fac324bb Duration: 1.42 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 31 MB
Request ID: e59c9cb1-47e5-4498-9f6d-b737fac324bb
```

**TEST EVENTS [SELECTED: TEST]**

- + Create new test event
- ▽ Private saved events

test

**ENVIRONMENT VARIABLES**