

Program 5: Create a Lambda Function to Add , Update and Delete a File in S3 Bucket.

Step 1: Create an S3 Bucket First, you need to create an S3 bucket in the AWS Management Console.

- Open the Amazon S3 console: You can find this by typing 'S3' into the search bar at the top of the AWS Management Console.
- Create a new bucket: Click on the 'Create bucket' button. This will open a dialog where you can configure your new bucket.
- Configure your bucket:
- Bucket name: Enter a unique DNS-compliant name for your new bucket. Remember that the bucket name must be unique across all existing bucket names in Amazon S3.
- Region: Choose the region where you want the bucket to reside.
- Copy settings: If you want to copy settings from an existing bucket, you can select the bucket from the list.
- Set permissions: Be careful with the permissions you grant. By default, all buckets and objects are private in Amazon S3. You can manage access to your bucket by creating bucket policies or bucket ACLs.
- Review: Review your settings and click 'Create bucket'.

Step 2: Create an IAM Role Next, create an IAM role that has permissions to access S3 and Lambda services.

- Sign in to the AWS Management Console: If you don't have an account, you'll need to create one.
- Open the IAM console: You can find this by typing 'IAM' into the search bar at the top of the AWS Management Console.
- Create a new role: Navigate to the "Roles" section and click on the 'Create role' button.
- Select type of trusted entity: Choose 'AWS service', then in the service list, choose 'Lambda'. Click 'Next: Permissions' to proceed.
- Attach permissions policies: Here, you'll need to attach the policies that grant the necessary permissions for S3 and Lambda services. For S3, you can attach the 'AmazonS3FullAccess' policy. For Lambda, you can attach the 'AWSLambdaExecute' policy.
- Review and create the role: Give your role a name, review the settings, and then click 'Create role'.

Step 3: Create a Lambda Function Now, create a new Lambda function and assign a policy to the IAM role.

1. **Sign in to the AWS Management Console:** If you don't have an account, you'll need to create one.
2. **Open the Lambda console:** You can find this by typing 'Lambda' into the search bar at the top of the AWS Management Console.
3. **Create a new function:** Click on the 'Create function' button. This will open a new page where you can configure your Lambda function.

4. Configure your function:

- **Function name:** Give your function a name.
- **Runtime:** Select the runtime that matches the language of your code. In your case, you should select a Python runtime.
- **Role:** Choose 'Use an existing role', and then select the IAM role you created in the previous step from the 'Existing role' dropdown.

5. **Function code:** In the 'Function code' section, you can write or paste your code. Paste the code you've provided.

```
import json
import boto3

s3 = boto3.client('s3')

def lambda_handler(event, context):
    operation = event['operation']

    # Perform operation
    if operation == 'list':
        return list_files(event['bucket'])
    elif operation == 'add':
        return add_file(event['bucket'], event['file_name'], event['file_content'])
    elif operation == 'update':
        return update_file(event['bucket'], event['file_name'], event['file_content'])
    elif operation == 'delete':
        return delete_file(event['bucket'], event['file_name'])
    else:
        return {
            'statusCode': 400,
            'body': json.dumps('Invalid operation')
        }

def list_files(bucket):
    response = s3.list_objects(Bucket=bucket)

    # Create a new response dictionary
    new_response = {
        'IsTruncated': response['IsTruncated'],
        'Marker': response['Marker'],
        'Name': response['Name'],
        'Prefix': response['Prefix'],
        'MaxKeys': response['MaxKeys'],
        'EncodingType': response['EncodingType'],
        'Contents': [{'Key': obj['Key'], 'Size': obj['Size']} for obj in response.get('Contents', [])]
```

```
}
```

```
return new_response
```

```
def add_file(bucket, file_name, file_content):  
    s3.put_object(Bucket=bucket, Key=file_name, Body=file_content)  
    return {'statusCode': 200}
```

```
def update_file(bucket, file_name, file_content):  
    s3.put_object(Bucket=bucket, Key=file_name, Body=file_content)  
    return {'statusCode': 200}
```

```
def delete_file(bucket, file_name):  
    s3.delete_object(Bucket=bucket, Key=file_name)  
    return {'statusCode': 200}
```

6. **Save and test the function:** Click 'Save' at the top right of the page. To test your function, click 'Test' at the top right of the page. You'll be prompted to configure a test event.

Test event JSON for your AWS Lambda function:

To list files:

```
{  
  "operation": "list",  
  "bucket": "your-bucket-name"  
}
```

To add a file:

```
{  
  "operation": "add",  
  "bucket": "your-bucket-name",  
  "file_name": "your-file-name",  
  "file_content": "your-file-content"  
}
```

To update a file:

```
{  
  "operation": "update",  
  "bucket": "your-bucket-name",  
  "file_name": "your-file-name",  
  "file_content": "your-new-file-content"  
}
```

To delete a file:

```
{  
  "operation": "delete",  
  "bucket": "your-bucket-name",  
  "file_name": "your-file-name"  
}
```