

MIT WORLD PEACE UNIVERSITY

**Cloud Infrastructure and Security
Third Year B. Tech, Semester 6**

**MANAGING S3 OPERATIONS ON A SPECIFIC
ACCOUNT USING BOTO3 ON PYTHON**

ASSIGNMENT 4

Prepared By

**Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 10**

March 28, 2024

Contents

1 Aim	1
2 Objectives	1
3 Theory	1
3.1 Boto3	1
3.2 Amazon S3	2
4 Python Script	2
5 Running the Script	4
6 Platform	5
7 FAQs	5
8 Conclusion	5
References	6

1 Aim

Write a program to Manage and monitor S3 operations to a specific Account using BOTO3 or equivalent libraries

2 Objectives

1. To learn about BOTO3 library in Python

3 Theory

3.1 Boto3

Boto3 is the Amazon Web Services (AWS) SDK for Python. It allows Python developers to interact with AWS services such as Amazon S3, EC2, DynamoDB, and many others programmatically. Boto3 provides an easy-to-use interface to make API calls to AWS services, manage AWS resources, and automate various tasks in your applications or scripts. It handles authentication, request signing, and error handling, making it simple to integrate AWS functionality into your Python applications.

Some key features of Boto3 include:

- **Comprehensive AWS Service Coverage:** Boto3 supports a wide range of AWS services, allowing developers to interact with almost every aspect of the AWS ecosystem.
- **Resource-Oriented Interface:** Boto3 provides a resource-oriented interface that allows developers to work with AWS resources in a natural and intuitive way. Resources represent entities such as EC2 instances, S3 buckets, DynamoDB tables, and more.
- **Low-Level Client Interface:** In addition to the resource-oriented interface, Boto3 also offers a low-level client interface that allows direct interaction with AWS service APIs. This interface provides more control and flexibility but requires a deeper understanding of the underlying service APIs.
- **Easy Configuration and Authentication:** Boto3 simplifies the process of configuring AWS credentials and authentication, supporting various authentication methods such as IAM roles, access keys, and temporary security credentials.
- **Integration with AWS Services:** Boto3 integrates seamlessly with other AWS services and features, such as AWS Lambda, Amazon SQS, Amazon SNS, and Amazon CloudWatch, enabling developers to build sophisticated applications and workflows.
- **Active Community and Support:** Boto3 benefits from an active community of developers and contributors, ensuring continuous improvement, updates, and support for new AWS features and services.

For more information and detailed documentation on Boto3, refer to the official AWS documentation: <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>.

3.2 Amazon S3

Amazon Simple Storage Service (Amazon S3) is a scalable object storage service offered by Amazon Web Services (AWS). It provides developers and businesses with a secure, durable, and highly available storage infrastructure for storing and retrieving any amount of data at any time. Amazon S3 is designed to offer 99.999999999 Percent (11 nines) durability and 99.99 Percentage availability of objects over a given year, making it suitable for a wide range of use cases, from simple file storage to complex data analytics and archival solutions.

Key features of Amazon S3 include:

- **Scalability and Durability:** Amazon S3 is designed to scale seamlessly to accommodate growing amounts of data and offers high durability for stored objects through data replication and redundancy across multiple geographically distributed data centers.
- **Data Protection and Security:** Amazon S3 provides multiple layers of data protection and security features, including encryption at rest and in transit, access control using AWS Identity and Access Management (IAM) policies and bucket policies, and versioning to protect against accidental deletions or overwrites.
- **Flexible Storage Classes:** Amazon S3 offers multiple storage classes with different levels of durability, availability, and performance characteristics to optimize storage costs and meet specific use case requirements, including Standard, Intelligent-Tiering, Glacier, and more.
- **Integration and Compatibility:** Amazon S3 seamlessly integrates with other AWS services such as AWS Lambda, Amazon CloudFront, Amazon Athena, and Amazon Redshift, enabling developers to build powerful applications and workflows that leverage the capabilities of S3 for data storage and retrieval.
- **Lifecycle Management:** Amazon S3 provides lifecycle policies that allow you to automatically transition objects between storage classes or expire objects based on predefined rules, helping you optimize storage costs and compliance requirements over time.
- **Cross-Region Replication:** Amazon S3 supports cross-region replication, allowing you to replicate objects across different AWS regions for disaster recovery, data locality, and compliance reasons, ensuring data resilience and availability in multiple geographic locations.

4 Python Script

```

1 import boto3
2 s3 = boto3.client('s3')
3
4 import json
5
6 def put(key, value):
7     # store something
8     s3.put_object(
9         Body=json.dumps({key: value}),
10        Bucket="cyclic-tame-red-seagull-shoe-us-east-1",
11        Key=f"some_files/{key}.json"
12    )
13    print("Stored successfully")
14

```

```

15 def get(key):
16     # get it back
17     my_file = s3.get_object(
18         Bucket="cyclic-tame-red-seagull-shoe-us-east-1",
19         Key=f"some_files/{key}.json"
20     )
21     return json.loads(my_file['Body'].read())
22
23 def delete(key):
24     # delete the key
25     s3.delete_object(
26         Bucket="cyclic-tame-red-seagull-shoe-us-east-1",
27         Key=f"some_files/{key}.json"
28     )
29
30 def listbucket():
31     # list all keys
32     my_bucket = s3.list_objects(
33         Bucket="cyclic-tame-red-seagull-shoe-us-east-1"
34     )
35     return my_bucket
36
37
38 if __name__ == "__main__":
39     print("Welcome to CIS assignment 4")
40     print("What do you want to do: ")
41     print("1. Store a key value pair")
42     print("2. Retrieve a value for a key")
43     print("3. Delete a key value pair")
44     print("4. List all keys in the bucket")
45     print("5. Exit")
46
47 choice = input("Enter your choice: ")
48
49 if choice == "1":
50     key = input("Enter the key: ")
51     value = input("Enter the value: ")
52     put(key, value)
53
54 elif choice == "2":
55     key = input("Enter the key: ")
56     print(get(key))
57
58 elif choice == "3":
59     key = input("Enter the key: ")
60     delete(key)
61
62 elif choice == "4":
63     print(listbucket())
64
65 else:
66     print("Invalid choice. Exiting...")

```

5 Running the Script

```
(awsenv) PS P:\Classes\Univeristy\Third Year\Second Semester\Cloud Infrastructure Systems\Lab\Assignment
_4> $Env:AWS_REGION="us-east-1"
(awsenv) PS P:\Classes\Univeristy\Third Year\Second Semester\Cloud Infrastructure Systems\Lab\Assignment
_4> $Env:AWS_ACCESS_KEY_ID="ASIAUYFMFX06WBKA5K4"
(awsenv) PS P:\Classes\Univeristy\Third Year\Second Semester\Cloud Infrastructure Systems\Lab\Assignment
_4> $Env:AWS_SECRET_ACCESS_KEY="ZPOyy5FFCHxD6HS3DczvI1sy3Pf+bPTTURlt2D4p"
(awsenv) PS P:\Classes\Univeristy\Third Year\Second Semester\Cloud Infrastructure Systems\Lab\Assignment
_4> $Env:AWS_SESSION_TOKEN="IQoJb3JpZ2luX2VjEK3//////////wEaCmFwLXNvdXR0LTEiRzBFAiEA4EbV3VB4K/sKLcqn5d
yWHphKETt8kKomLbHlMFC1ICICI5i7tD+bBqVEcQ+dNoFPOQIY11qaMkaI9fYPHnls/DKrgCCMb//////////wEQABoMMzI2Nzc4NjAw
MzgxIgyXjQPEjLJweP77ngojAJV9kg6tJdC4b8TtaxXnqUuxPIa5iKGAIUN3tWaFLBTxdsJfjEDEdcfSLoFBFL+/j171/kUOT7sq14c
AclPc4mN2HjAHq9t08IEIvPfp8b1+w0Qn1Vlm50wlPgcOP0ohhCgRjDyfDNfSij6QvYRhU4JUbiCxI4u4rpy5XWvPjdYvUNrrY1hJ8gg
RALxMDNB4341CjepOYuIC604rLTHaJqVqkzSJET7cfyz9pseaDdp20x+R7WN09pGIRFZfim+SuVHK9V1pRikrEaxExcJdFPtqUD2WZ6
k+G+NH5XnOX3nTghIS4BbGhg1hdq2WFLWibwbVwFjQ70QYzbWuCEDkv0zPWTgo1gyu0YC2URMNWJkrAGOp0BQa87X9G0GwQ3K4FcDhYG
ic9gYDXoV/IztKEvwHMNyLyRAPVn9iww+k+HKdRJZtaKajaXrIcKYe9Un2UjcXGs6LBg6bf2+cJgk0YYeR7zBjq+bhmj+rYlaAoq82xl
14iQWQCGgoCqkHLz/wtcniUUx6TKwnQnk7r3R7SKLuzRyyHMXHnkNSM/Bt8QmpyEiMu6ylAfX2Mzf0/Zd8zsrg=="
(awsenv) PS P:\Classes\Univeristy\Third Year\Second Semester\Cloud Infrastructure Systems\Lab\Assignment
_4> python .\krishnaraj_cis_4.py
Welcome to CIS assignment 4
What do you want to do:
1. Store a key value pair
2. Retrieve a value for a key
3. Delete a key value pair
4. List all keys in the bucket
5. Exit
Enter your choice: 1
```

Figure 1: Pasting Credentials in the Powershell

```
What do you want to do:
1. Store a key value pair
2. Retrieve a value for a key
3. Delete a key value pair
4. List all keys in the bucket
5. Exit
Enter your choice: 1
Enter the key: test
Enter the value: 123
Stored successfully
(awsenv) PS P:\Classes\Univeristy\Third Year\Second Semester\Cloud Infrastructure Systems\Lab\Assignment
_4> python .\krishnaraj_cis_4.py
Welcome to CIS assignment 4
What do you want to do:
1. Store a key value pair
2. Retrieve a value for a key
3. Delete a key value pair
4. List all keys in the bucket
5. Exit
Enter your choice: 2
Enter the key: test
{'test': '123'}
(awsenv) PS P:\Classes\Univeristy\Third Year\Second Semester\Cloud Infrastructure Systems\Lab\Assignment
_4> python .\krishnaraj_cis_4.py
Welcome to CIS assignment 4
```

Figure 2: Storing and Getting Values

```
(awsenv) PS P:\Classes\Univeristy\Third Year\Second Semester\Cloud Infrastructure Systems\Lab\Assignment
_4> python .\krishnaraj_cis_4.py
Welcome to CIS assignment 4
What do you want to do:
1. Store a key value pair
2. Retrieve a value for a key
3. Delete a key value pair
4. List all keys in the bucket
5. Exit
Enter your choice: 4
{'ResponseMetadata': {'RequestId': 'RKMPMJBSCMCN2KY8', 'HostId': 'ZJidebcShfr1FzDjxbF8xRbYWo3dmtESM9m267W/P5oIk10j1xaxNT2voe4qHxrA9TjmTm12Ts=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2': 'ZJidebcShfr1FzDjxbF8xRbYWo3dmtESM9m267W/P5oIk10j1xaxNT2voe4qHxrA9TjmTm12Ts=', 'x-amz-request-id': 'RKMPMJBSCMCN2KY8', 'date': 'Wed, 27 Mar 2024 20:58:35 GMT', 'x-amz-bucket-region': 'us-east-1', 'content-type': 'application/xml', 'transfer-encoding': 'chunked', 'server': 'AmazonS3'}, 'RetryAttempts': 0}, 'IsTruncated': False, 'Marker': '', 'Contents': [{"Key": "cyclic-db/ac81e671/stream_lambda.zip", "LastModified": datetime.datetime(2023, 10, 3, 1, 34, 36, tzinfo=tzutc()), "ETag": '"2eaef020e08576e3234c5f8b965cfb20"', 'Size': 13051511, 'StorageClass': 'STANDARD', 'Owner': {'DisplayName': 'aws-root+prod-barred-owl', 'ID': '893fdb8c7945d41c9ca84c9640ca317971c865c01785949e66d01364ad1bf1f6'}}, {"Key": "cyclic/dist/tame-red-seagull-lambda.zip", "LastModified": datetime.datetime(2023, 10, 24, 21, 19, 3, tzinfo=tzutc()), "ETag": '"4e43f9cd48dd1e65b373f947f8a7327e-9"', 'Size': 42045078, 'StorageClass': 'STANDARD', 'Owner': {'DisplayName': 'aws-root+prod-barred-owl', 'ID': '893fdb8c7945d41c9ca84c9640ca317971c865c01785949e66d01364ad1bf1f6'}}, {"Key": "some_files/cis_a4.json", "LastModified": datetime.datetime(2024, 3, 27, 20, 53, 54, tzinfo=tzutc()), "ETag": '"b0b870266d92c6cbe61cc97d6ccf4118"', 'Size': 15, 'StorageClass': 'STANDARD', 'Owner': {'DisplayName': 'aws-root+prod-barred-owl', 'ID': '893fdb8c7945d41c9ca84c9640ca317971c865c01785949e66d01364ad1bf1f6'}}]}
```

Figure 3: Listing all files.

```
(awsenv) PS P:\Classes\Univeristy\Third Year\Second Semester\Cloud Infrastructure Systems\Lab\Assignment
_4> python .\krishnaraj_cis_4.py
Welcome to CIS assignment 4
What do you want to do:
1. Store a key value pair
2. Retrieve a value for a key
3. Delete a key value pair
4. List all keys in the bucket
5. Exit
Enter your choice: 3
Enter the key: test
(awsenv) PS P:\Classes\Univeristy\Third Year\Second Semester\Cloud Infrastructure Systems\Lab\Assignment
_4> |
```

Figure 4: Deleting a File

6 Platform

Operating System: Windows 11

IDEs or Text Editors Used: Visual Studio Code

Compilers or Interpreters: Python 3.10.1

7 FAQs

8 Conclusion

In this assignment, we learned how to manage and monitor S3 operations to a specific account using BOTO3 in python. We inserted and changed data in a project that was deployed on Amazon s3 using a PaaS called Cyclic.

References