Cycle 08 AWS Homework

1. Create an S3 Static Website

Purpose: Host static HTML/CSS/JS content directly from S3.

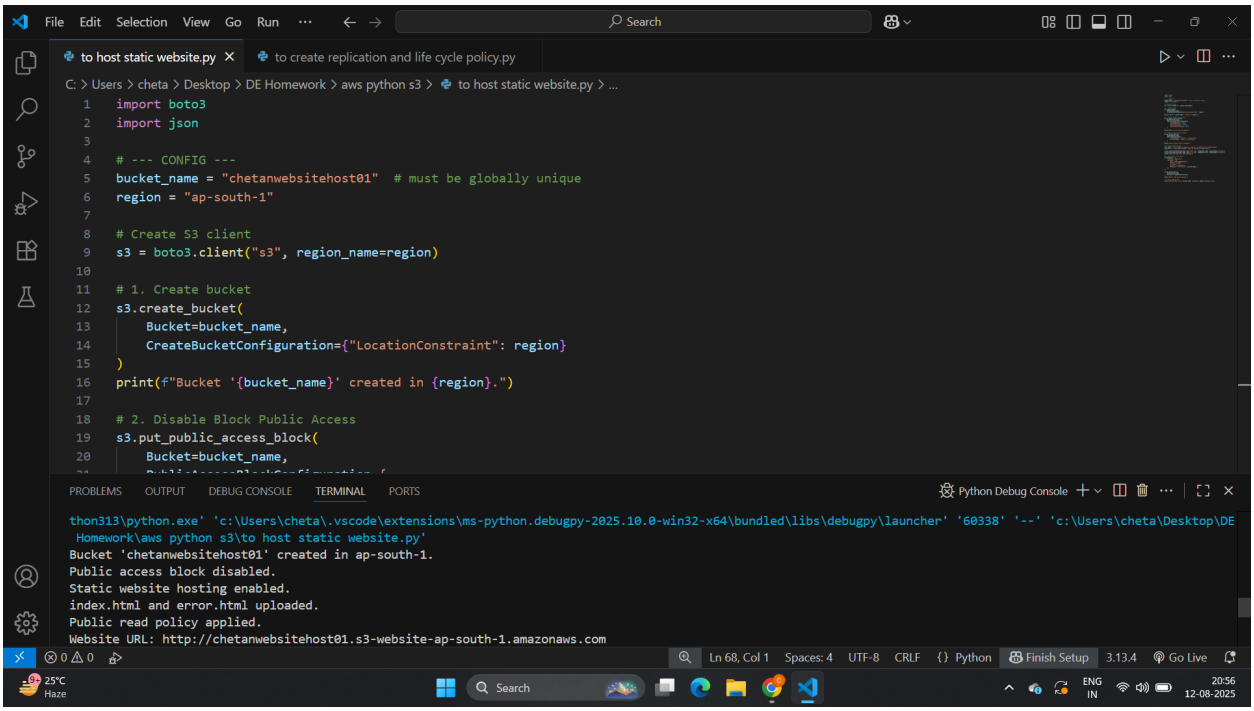
Python (boto3):

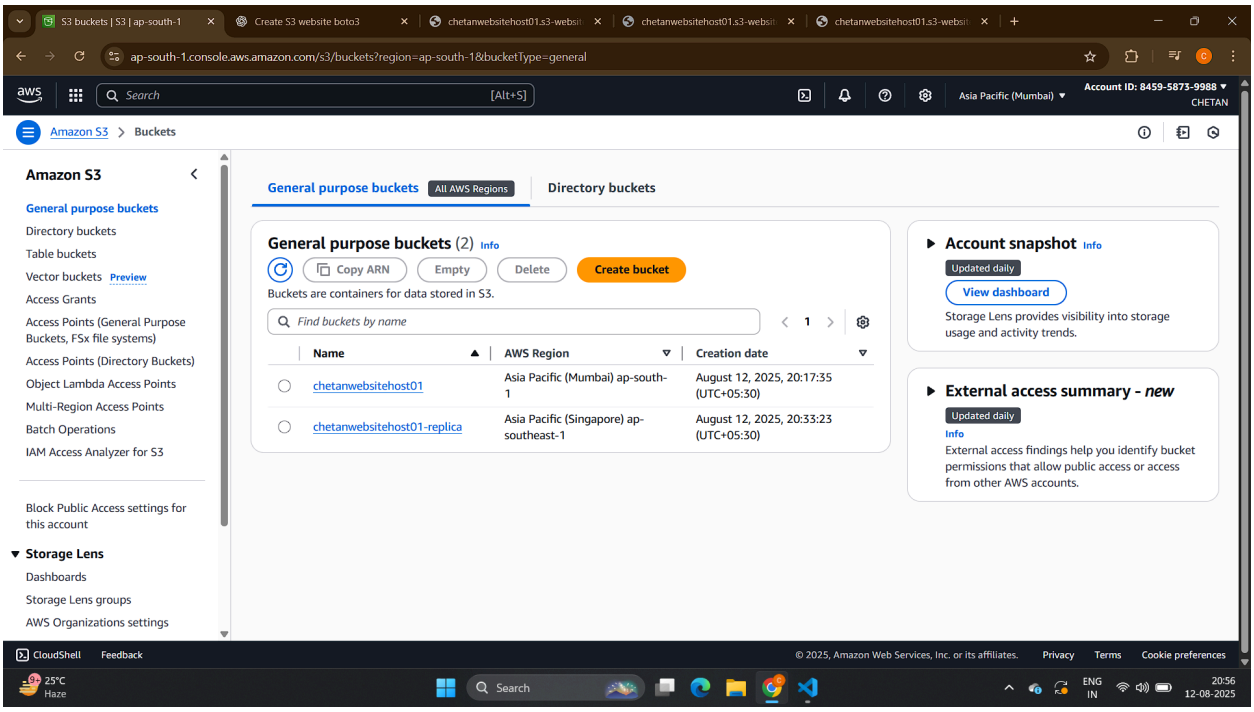
```
python
CopyEdit
import boto3

s3 = boto3.client('s3')

bucket_name = "chetanwebsitehost01"

website_config = {
    'IndexDocument': {'Suffix': 'index.html'},
    'ErrorDocument': {'Key': 'error.html'}
}

s3.put_bucket_website(Bucket=bucket_name, WebsiteConfiguration=website_
config)
print("Static website hosting enabled.")
```

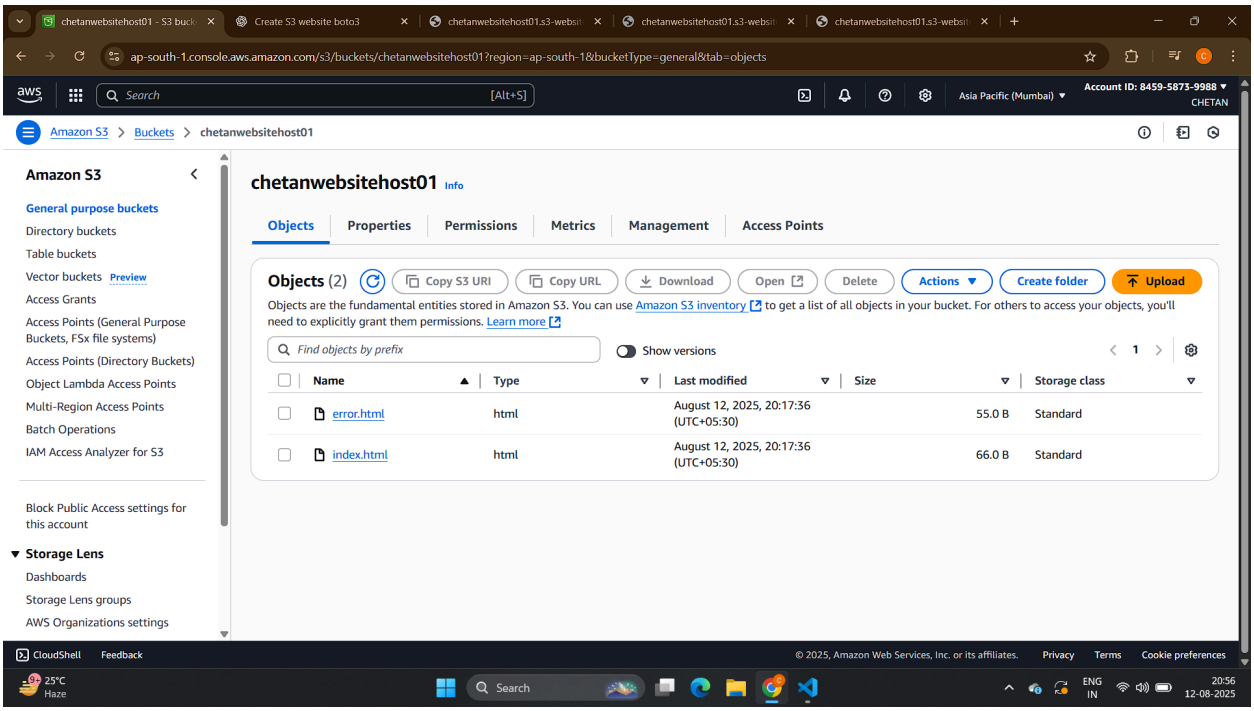
AWS CLI:

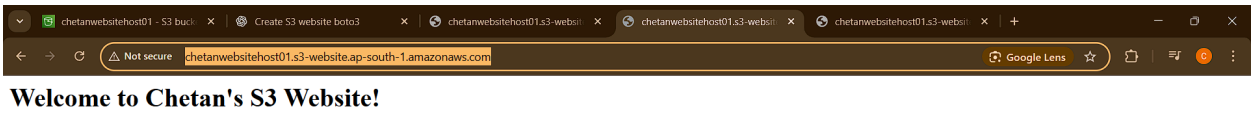
```
bash
CopyEdit
aws s3 website s3://chetanwebsitehost01/ --index-document index.html --err
or-document error.html
```

Explanation: This enables the S3 bucket to serve files as a website. You must upload `index.html` and optionally `error.html`.









2. Configure Public Access

Purpose: Allow public reads for website hosting.

Python:

```
python
CopyEdit
# Allow public read
s3.put_bucket_acl(Bucket=bucket_name, ACL='public-read')
```

AWS CLI:

```
bash
CopyEdit
aws s3api put-bucket-acl --bucket chetanwebsitehost01 --acl public-read
```

Explanation: Without public access, browsers can't view your static site files.

---

3. Enable Versioning

Purpose: Keep multiple versions of objects for backup/restore.

Python:

```
python
CopyEdit
s3.put_bucket_versioning(
    Bucket=bucket_name,
    VersioningConfiguration={'Status': 'Enabled'}
)
```

AWS CLI:

```
bash
CopyEdit
aws s3api put-bucket-versioning --bucket chetanwebsitehost01 --versioning-
```

```
configuration Status=Enabled
```

---

4. Lifecycle Policies

Purpose: Automatically transition/delete old objects.

Python:

```
python
CopyEdit
lifecycle_config = {
    'Rules': [
        {
            'ID': 'MoveOldFilesToIA',
            'Filter': {'Prefix': ''},
            'Status': 'Enabled',
            'Transitions': [{'Days': 30, 'StorageClass': 'STANDARD_IA'}],
            'Expiration': {'Days': 365}
        }
    ]
}

s3.put_bucket_lifecycle_configuration(
    Bucket=bucket_name,
    LifecycleConfiguration=lifecycle_config
)
```

AWS CLI:

```
bash
CopyEdit
aws s3api put-bucket-lifecycle-configuration --bucket chetanwebsitehost01 -
```

```
-lifecycle-configuration file://lifecycle.json
```

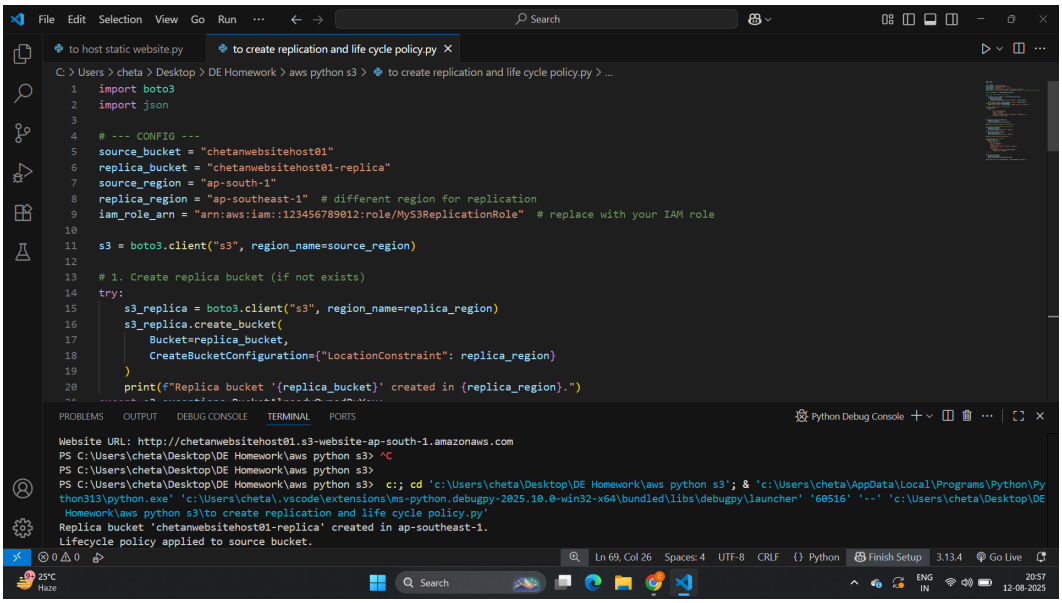
---

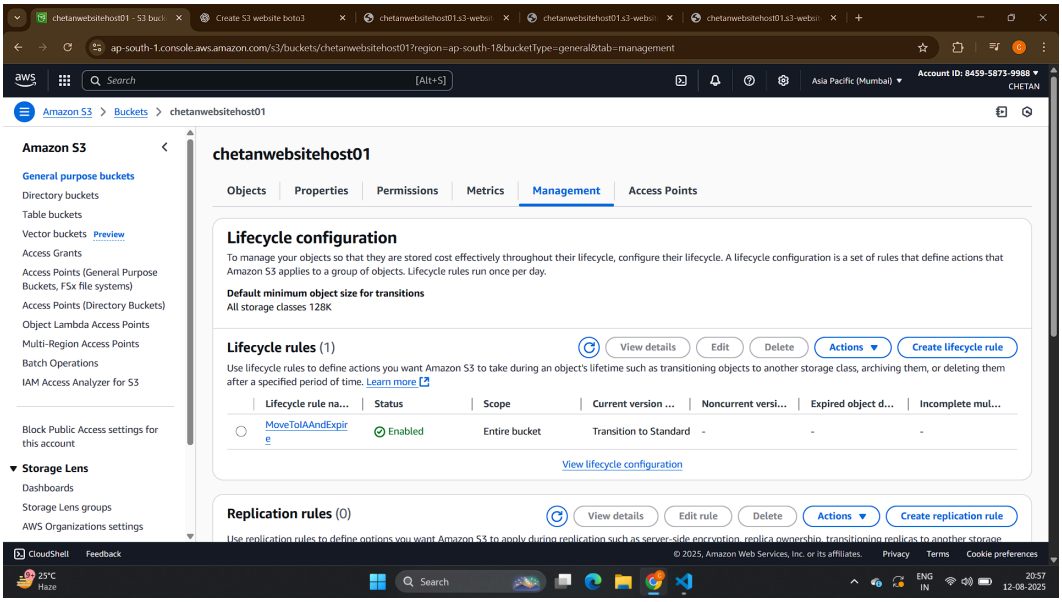
5. Cross-Region Replication

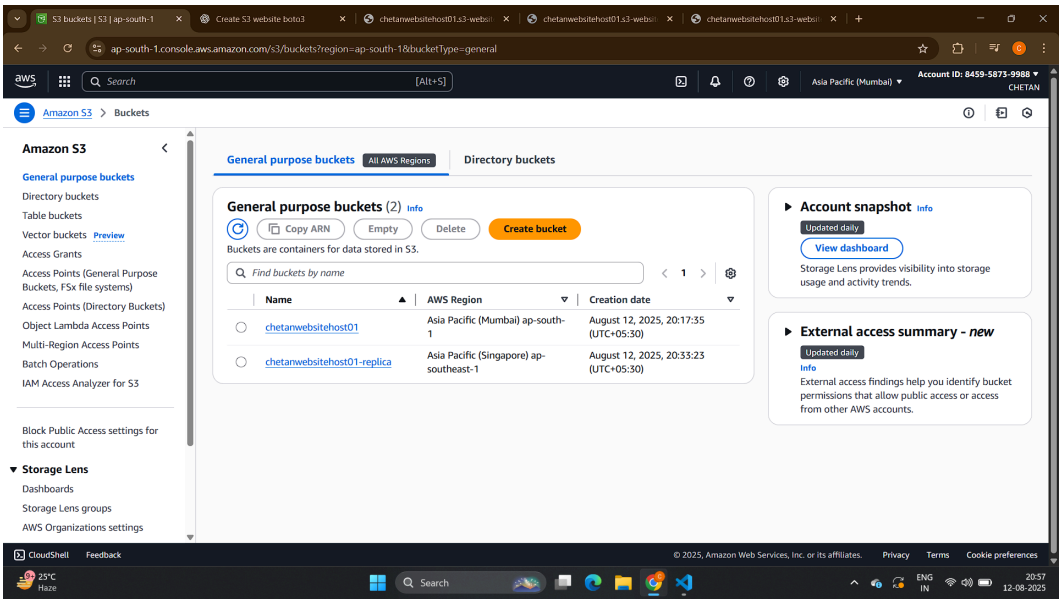
Purpose: Automatically copy objects to another bucket in another region.

Python:

```
python
CopyEdit
replication_config = {
    'Role': 'arn:aws:iam::<account-id>:role/MyS3ReplicationRole',
    'Rules': [{
        'ID': 'CRR',
        'Status': 'Enabled',
        'Priority': 1,
        'Filter': {'Prefix': ''},
        'Destination': {
            'Bucket': 'arn:aws:s3:::chetanwebsitehost01-replica',
            'StorageClass': 'STANDARD'
        }
    }]
}

s3.put_bucket_replication(
    Bucket=bucket_name,
    ReplicationConfiguration=replication_config
)
```


to host static website.py
to create replication and life cycle policy.py
C: > Users > cheta > Desktop > DE Homework > aws python s3 > to create replication and life cycle policy.py > ...
import boto3
import json
# --- CONFIG ---
source_bucket = "chetanwebsitehost01"
replica_bucket = "chetanwebsitehost01-replica"
source_region = "ap-south-1"
replica_region = "ap-southeast-1" # different region for replication
iam_role_arn = "arn:aws:iam::123456789012:role/MyS3ReplicationRole" # replace with your IAM role
s3 = boto3.client("s3", region_name=source_region)
# 1. Create replica bucket (if not exists)
try:
s3_replica = boto3.client("s3", region_name=replica_region)
s3_replica.create_bucket(
Bucket=replica_bucket,
CreateBucketConfiguration={"LocationConstraint": replica_region}
)
print(f"Replica bucket '{replica_bucket}' created in {replica_region}.")
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python Debug Console
Website URL: http://chetanwebsitehost01.s3-website-ap-south-1.amazonaws.com
PS C:\Users\cheta\Desktop\DE Homework\aws python s3> ^C
PS C:\Users\cheta\Desktop\DE Homework\aws python s3>
PS C:\Users\cheta\Desktop\DE Homework\aws python s3> c:; cd 'c:\Users\cheta\Desktop\DE Homework\aws python s3'; & 'c:\Users\cheta\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\cheta\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '60516' '--' 'c:\Users\cheta\Desktop\DE Homework\aws python s3\to create replication and life cycle policy.py'
Replica bucket 'chetanwebsitehost01-replica' created in ap-southeast-1.
Lifecycle policy applied to source bucket.


ap-south-1.console.aws.amazon.com/s3/buckets/chetanwebsitehost01?region=ap-south-1&bucketType=general&tab=management
Amazon S3 > Buckets > chetanwebsitehost01
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Vector buckets Preview
Access Grants
Access Points (General Purpose Buckets, FSx file systems)
Access Points (Directory Buckets)
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
chetanwebsitehost01
Objects
Properties
Permissions
Metrics
Management
Access Points
Lifecycle configuration
To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their lifecycle. A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. Lifecycle rules run once per day.
Default minimum object size for transitions
All storage classes 128K
Lifecycle rules (1)
View details
Edit
Delete
Actions
Create lifecycle rule
Use lifecycle rules to define actions you want Amazon S3 to take during an object's lifetime such as transitioning objects to another storage class, archiving them, or deleting them after a specified period of time. Learn more
Lifecycle rule na...
Status
Scope
Current version ...
Noncurrent versi...
Expired object d...
Incomplete mul...
MoveToIAAndExpire
Enabled
Entire bucket
Transition to Standard
View lifecycle configuration
Replication rules (0)
View details
Edit rule
Delete
Actions
Create replication rule
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



AWS CLI:

```
bash
CopyEdit
aws s3api put-bucket-replication --bucket chetanwebsitehost01 --replication-configuration file://replication.json
```

Note: Requires IAM role with S3 replication permissions.

---

6. Encryption

Purpose: Secure data at rest.

Python:

```
python
CopyEdit
s3.put_bucket_encryption(
    Bucket=bucket_name,
    ServerSideEncryptionConfiguration={
        'Rules': [{
            'ApplyServerSideEncryptionByDefault': {'SSEAlgorithm': 'AES256'}
```

```
        }]
    }
)
```

AWS CLI:

```
bash
CopyEdit
aws s3api put-bucket-encryption --bucket chetanwebsitehost01 --server-side
-encryption-configuration file://encryption.json
```

---

7. Logging

Purpose: Track access requests.

Python:

```
python
CopyEdit
s3.put_bucket_logging(
    Bucket=bucket_name,
    BucketLoggingStatus={
        'LoggingEnabled': {
            'TargetBucket': 'my-log-bucket',
            'TargetPrefix': 'logs/'
        }
    }
)
```

---

8. CORS Rules

Purpose: Allow cross-origin requests for web apps.

Python:

```
python
CopyEdit
cors_config = {
    'CORSRules': [{
        'AllowedHeaders': ['*'],
        'AllowedMethods': ['GET', 'POST'],
        'AllowedOrigins': ['*'],
        'ExposeHeaders': ['ETag'],
        'MaxAgeSeconds': 3000
    }]
}

s3.put_bucket_cors(Bucket=bucket_name, CORSConfiguration=cors_config)
```

---

9. Event Notifications

Purpose: Trigger Lambda when object is uploaded.

Python:

```
python
CopyEdit
notification_config = {
    'LambdaFunctionConfigurations': [{
        'LambdaFunctionArn': 'arn:aws:lambda:ap-south-1:account-id:function:m
yLambda',
        'Events': ['s3:ObjectCreated:*']
    }]
}

s3.put_bucket_notification_configuration(
    Bucket=bucket_name,
    NotificationConfiguration=notification_config
```

```
)
```

---

10. Object Tagging

Purpose: Label objects for management.

Python:

```
python
CopyEdit
s3.put_object_tagging(
    Bucket=bucket_name,
    Key='test.txt',
    Tagging={'TagSet': [{'Key': 'Project', 'Value': 'Cycle08'}]}
)
```

Testing

- After enabling website hosting & public access, visit:

```
arduino
CopyEdit
http://chetanwebsitehost01.s3-website.ap-south-1.amazonaws.com
```

- Use `aws s3 ls` to confirm objects exist.
- Verify replication by checking the replica bucket in another region.