Boto3 S3 Task Documentation

Purpose

This document details the implementation of a Python script developed for FrogTech to demonstrate proficiency with the AWS Boto3 SDK. The script performs S3 operations in the us-east-1 region, meeting specific requirements: listing buckets, uploading/downloading a dummy file, verifying object existence with a waiter, listing objects with a paginator, and using a modular design. Additionally, it includes functionality to inspect file contents before cleanup. This document serves as a personal reference to revisit the task, understand the code, and refresh knowledge on Boto3 and S3 operations.

Script Overview

The script (s3_operations.py) uses Boto3 to interact with AWS S3, performing the following tasks:

1. **List Buckets**: Retrieves all S3 buckets in us-east-1 using a resource call.
2. **Create and Upload Dummy File**: Creates a local text file with a timestamp, uploads it to a selected bucket, and stores it under a test/ prefix.
3. **Download File**: Downloads the uploaded file to verify the operation.
4. **Inspect File Contents**: Prints the contents of the local dummy file and downloaded file.
5. **Waiter**: Uses the ObjectExists waiter to confirm the uploaded object's availability (10-second delay, 10 attempts).
6. **Paginator**: Lists all objects in the bucket using the ListObjectsV2 paginator.
7. **Pause and Cleanup**: Pauses for manual file inspection and cleans up local files.
8. **Modular Design (Bonus)**: Organizes operations into separate functions with a main() function orchestrating execution.

Key Components

Imports and Initialization

- **Libraries**:
  - boto3: AWS SDK for Python to interact with S3.
  - botocore.exceptions.ClientError: Handles AWS API errors (e.g., permission issues).
  - uuid: Generates unique file names to avoid conflicts.
  - os: Manages local file operations (creation, deletion).
  - datetime: Adds timestamps to dummy file contents.
- **S3 Resource and Client**:
  - s3_resource = boto3.resource('s3', region_name='us-east-1'): High-level interface for bucket and object operations.

- s3_client = boto3.client('s3', region_name='us-east-1'): Low-level interface for waiters and paginators.

Functions

1. **list_buckets()**:
   - Uses s3_resource.buckets.all() to list all S3 buckets.
   - Returns a list of bucket names or None if no buckets exist or an error occurs.
   - Handles ClientError for permission or connectivity issues.
2. **create_dummy_file(file_path)**:
   - Creates a local text file with a timestamp (e.g., Dummy file created at 2025-04-30T12:34:56.789123).
   - Returns True on success, False on I/O errors.
3. **read_local_file(file_path)**:
   - Reads and prints the contents of a local file (e.g., dummy file or downloaded file).
   - Returns the content or None on I/O errors.
4. **upload_file(bucket_name, file_path, object_key)**:
   - Uploads a file to an S3 bucket using Bucket.upload_file.
   - Stores the file under test/<filename> in the bucket.
   - Returns True on success, False on ClientError.
5. **download_file(bucket_name, object_key, download_path)**:
   - Downloads an S3 object to a local path using Bucket.download_file.
   - Returns True on success, False on ClientError.
6. **wait_for_object(bucket_name, object_key)**:
   - Uses the object_exists waiter to confirm the object's availability.
   - Configured with a 10-second delay and 10 attempts.
   - Returns True if the object exists, False on WaiterError.
7. **list_objects_paginator(bucket_name)**:
   - Uses the ListObjectsV2 paginator to list all objects in the bucket.
   - Handles pagination automatically, printing each object's key and size.
   - Handles ClientError for permission or bucket issues.
8. **main()**:

- Orchestrates the workflow: lists buckets, selects a bucket, creates/uploads/downloads a file, inspects contents, verifies the object, lists objects, pauses, and cleans up.
- Uses try-finally to ensure local file cleanup.

Bucket Selection Logic

- **Method**: Selects the first bucket from the list returned by list_buckets() (e.g., bucket_list[0]).
- **Reason**: Chosen for simplicity to meet the "random bucket" requirement without adding complexity. The first bucket is a pseudo-random choice since the AWS API determines the order.
- **Example**: In the provided output, backend-bucket-terraform was selected as the first bucket.
- **Alternatives**:
    - **User Input**: Prompt the user to choose a bucket.
    - **Random Selection**: Use random.choice() for a truly random bucket.
    - **Filtered Selection**: Choose based on name patterns (e.g., theme-park*) or permissions.
    - **Hardcoded**: Specify a bucket like backend-bucket-terraform.

File Handling

- **Dummy File**: Named dummy_<uuid>.txt (e.g., dummy_087b46cbab434332a1c05795d439fc2d.txt) to ensure uniqueness.
- **S3 Key**: Stored as test/<filename> in the bucket.
- **Downloaded File**: Named downloaded_<filename>.txt.
- **Inspection**: Contents are printed after creation and download, with a pause (input()) for manual inspection.
- **Cleanup**: Local files are deleted in the finally block to avoid clutter.

Requirements Fulfillment

1. **List us-east-1 Buckets (Resource Call)**:
    - Implemented in list_buckets() using s3_resource.buckets.all().
    - Output example: Lists buckets like backend-bucket-terraform, codepipeline-us-east-1-580313038498, etc.
2. **Upload/Download Dummy File (Resource Call)**:
    - create_dummy_file() creates a file with a timestamp.
    - upload_file() uploads it to test/<filename> in the selected bucket.

   - download_file() downloads it to downloaded_<filename>.txt.
   - Output example: Uploaded dummy_087b46cbab434332a1c05795d439fc2d.txt to s3://backend-bucket-terraform/test/....

3. **Waiter (ObjectExists)**:
   - wait_for_object() uses the object_exists waiter with a 10-second delay and 10 attempts.
   - Output example: Object s3://backend-bucket-terraform/test/... exists..

4. **Paginator (ListObjectsV2)**:
   - list_objects_paginator() uses the ListObjectsV2 paginator to list bucket objects.
   - Output example: Lists objects like terraform.tfstate, test/dummy_087b46cbab434332a1c05795d439fc2d.txt.

5. **Bonus (Modular Design)**:
   - Functions are separated for each operation, with main() handling dependencies.
   - Error handling is centralized using try-except and try-finally.

6. **File Inspection**:
   - read_local_file() prints the dummy and downloaded file contents.
   - A pause (input()) allows manual inspection before cleanup.

Challenges and Solutions

1. **Challenge**: Choosing a bucket without specific criteria.
   - **Solution**: Selected the first bucket for simplicity. Added validation to ensure the bucket list isn't empty.

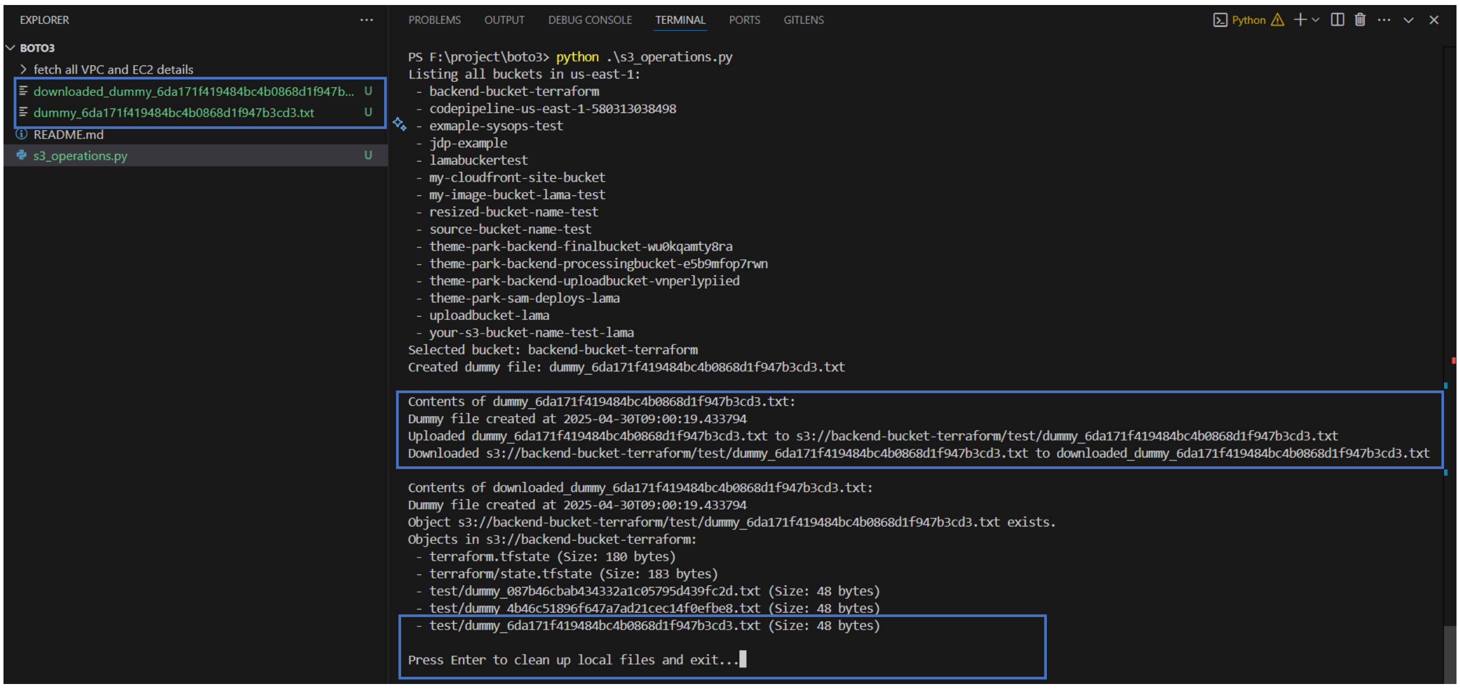
2. **Challenge**: Ensuring file content visibility before deletion.
   - **Solution**: Added read_local_file() and a pause (input()) for inspection.

3. **Challenge**: Handling S3 eventual consistency.
   - **Solution**: Used the ObjectExists waiter to confirm object availability.

4. **Challenge**: Managing large bucket listings.
   - **Solution**: The paginator automatically handles pagination.

5. **Challenge**: Permissions errors.
   - **Solution**: Assumed valid credentials; added ClientError handling for robustness.

Sample Output

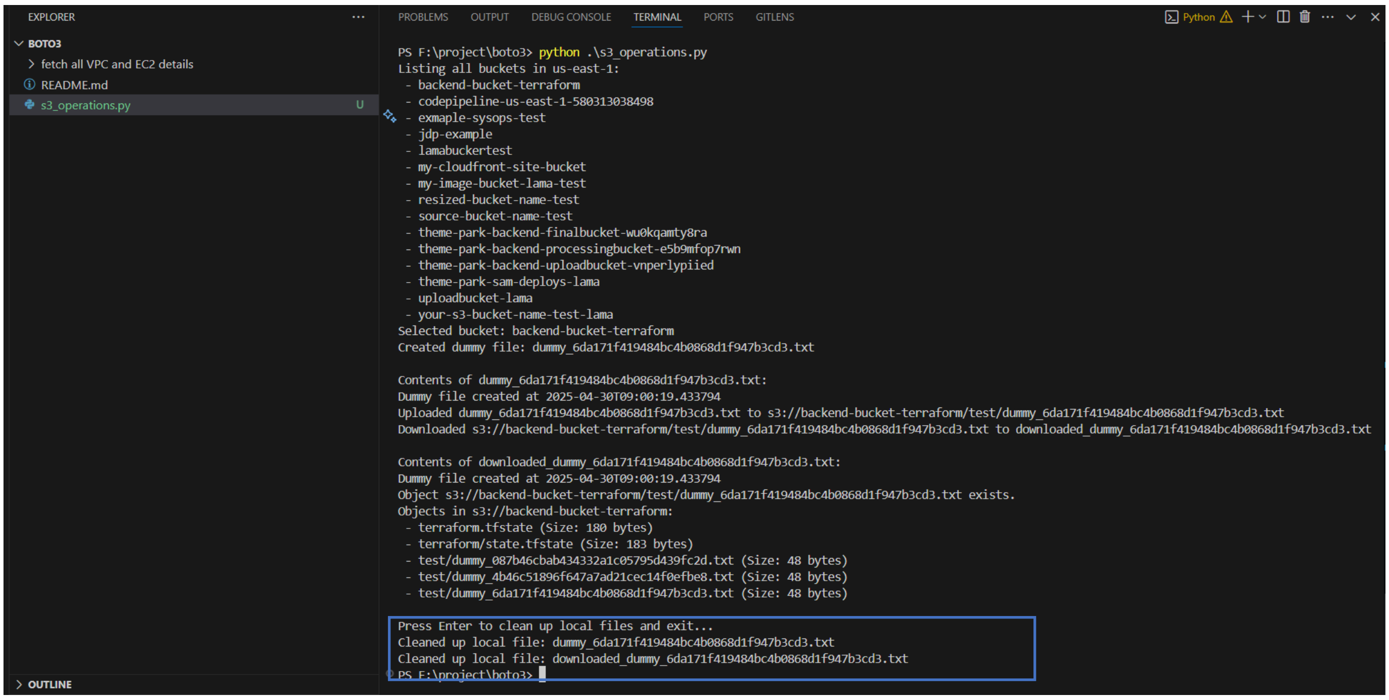

EXPLORER
BOTO3
fetch all VPC and EC2 details
downloaded_dummy_6da171f419484bc4b0868d1f947b...
dummy_6da171f419484bc4b0868d1f947b3cd3.txt
README.md
s3_operations.py
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
Python

```
PS F:\project\boto3> python .\s3_operations.py
Listing all buckets in us-east-1:
 - backend-bucket-terraform
 - codepipeline-us-east-1-580313038498
 - exmaple-sysops-test
 - jdp-example
 - lamabuckertest
 - my-cloudfront-site-bucket
 - my-image-bucket-lama-test
 - resized-bucket-name-test
 - source-bucket-name-test
 - theme-park-backend-finalbucket-wu0kqamty8ra
 - theme-park-backend-processingbucket-e5b9mfop7rwn
 - theme-park-backend-uploadbucket-vnperlypiied
 - theme-park-sam-deploys-lama
 - uploadbucket-lama
 - your-s3-bucket-name-test-lama
Selected bucket: backend-bucket-terraform
Created dummy file: dummy_6da171f419484bc4b0868d1f947b3cd3.txt

Contents of dummy_6da171f419484bc4b0868d1f947b3cd3.txt:
Dummy file created at 2025-04-30T09:00:19.433794
Uploaded dummy_6da171f419484bc4b0868d1f947b3cd3.txt to s3://backend-bucket-terraform/test/dummy_6da171f419484bc4b0868d1f947b3cd3.txt
Downloaded s3://backend-bucket-terraform/test/dummy_6da171f419484bc4b0868d1f947b3cd3.txt to downloaded_dummy_6da171f419484bc4b0868d1f947b3cd3.txt

Contents of downloaded_dummy_6da171f419484bc4b0868d1f947b3cd3.txt:
Dummy file created at 2025-04-30T09:00:19.433794
Object s3://backend-bucket-terraform/test/dummy_6da171f419484bc4b0868d1f947b3cd3.txt exists.
Objects in s3://backend-bucket-terraform:
 - terraform.tfstate (Size: 180 bytes)
 - terraform/state.tfstate (Size: 183 bytes)
 - test/dummy_087b46cbab434332a1c05795d439fc2d.txt (Size: 48 bytes)
 - test/dummy_4b46c51896f647a7ad21cec14f0efbe8.txt (Size: 48 bytes)
 - test/dummy_6da171f419484bc4b0868d1f947b3cd3.txt (Size: 48 bytes)

Press Enter to clean up local files and exit...
```

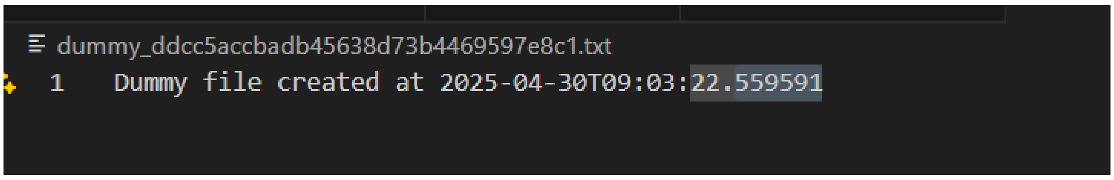


```
PS F:\project\boto3> python .\s3_operations.py
Listing all buckets in us-east-1:
 - backend-bucket-terraform
 - codepipeline-us-east-1-580313038498
 - exmaple-sysops-test
 - jdp-example
 - lamabuckertest
 - my-cloudfront-site-bucket
 - my-image-bucket-lama-test
 - resized-bucket-name-test
 - source-bucket-name-test
 - theme-park-backend-finalbucket-wu0kqamty8ra
 - theme-park-backend-processingbucket-e5b9mfop7rwn
 - theme-park-backend-uploadbucket-vnperlypiied
 - theme-park-sam-deploys-lama
 - uploadbucket-lama
 - your-s3-bucket-name-test-lama
Selected bucket: backend-bucket-terraform
Created dummy file: dummy_6da171f419484bc4b0868d1f947b3cd3.txt

Contents of dummy_6da171f419484bc4b0868d1f947b3cd3.txt:
Dummy file created at 2025-04-30T09:00:19.433794
Uploaded dummy_6da171f419484bc4b0868d1f947b3cd3.txt to s3://backend-bucket-terraform/test/dummy_6da171f419484bc4b0868d1f947b3cd3.txt
Downloaded s3://backend-bucket-terraform/test/dummy_6da171f419484bc4b0868d1f947b3cd3.txt to downloaded_dummy_6da171f419484bc4b0868d1f947b3cd3.txt

Contents of downloaded_dummy_6da171f419484bc4b0868d1f947b3cd3.txt:
Dummy file created at 2025-04-30T09:00:19.433794
Object s3://backend-bucket-terraform/test/dummy_6da171f419484bc4b0868d1f947b3cd3.txt exists.
Objects in s3://backend-bucket-terraform:
 - terraform.tfstate (Size: 180 bytes)
 - terraform/state.tfstate (Size: 183 bytes)
 - test/dummy_087b46cbab434332a1c05795d439fc2d.txt (Size: 48 bytes)
 - test/dummy_4b46c51896f647a7ad21cec14f0efbe8.txt (Size: 48 bytes)
 - test/dummy_6da171f419484bc4b0868d1f947b3cd3.txt (Size: 48 bytes)

Press Enter to clean up local files and exit...
Cleaned up local file: dummy_6da171f419484bc4b0868d1f947b3cd3.txt
Cleaned up local file: downloaded_dummy_6da171f419484bc4b0868d1f947b3cd3.txt
PS F:\project\boto3> 
```



```
dummy_ddcc5accbadb45638d73b4469597e8c1.txt
Dummy file created at 2025-04-30T09:03:22.559591
```