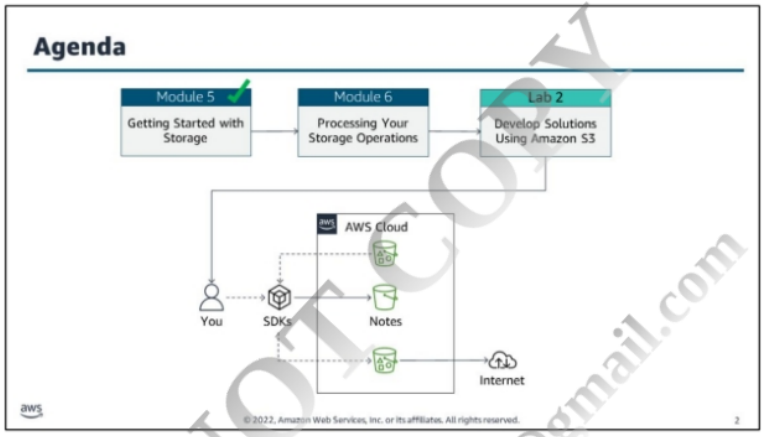
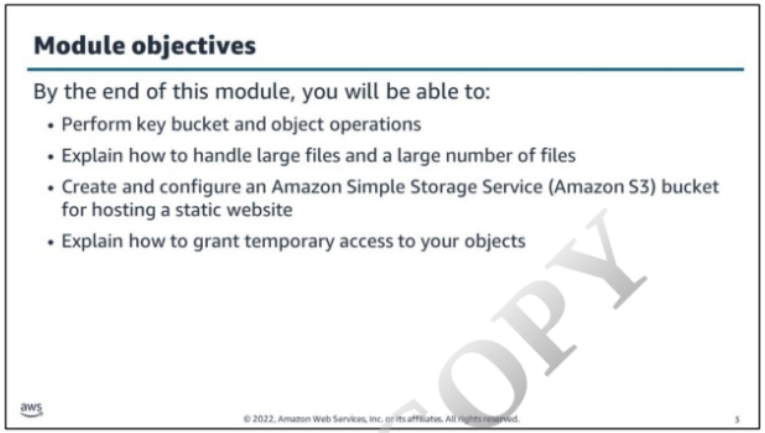
Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification

Module 6: Processing Your Storage Operations





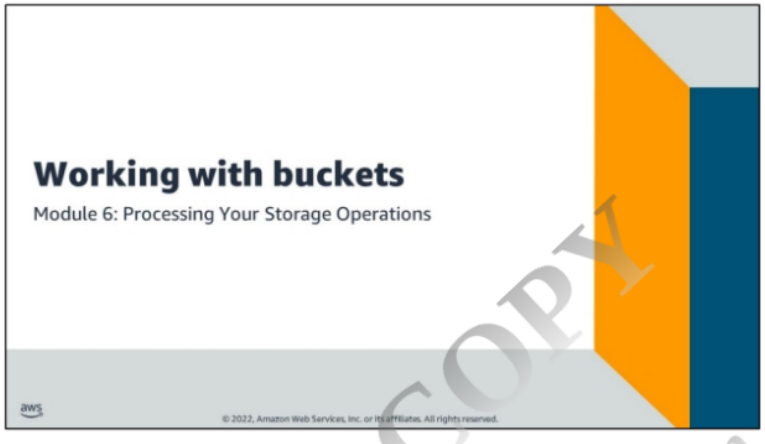
In the previous module, you learned about configuring the SDK and creating the service client for accessing Amazon Simple Storage Service (Amazon S3). This module focuses on working with the data. You will learn how to create a bucket programmatically and perform CRUD (create, read, update, and delete) operations on Amazon S3 objects. You will look at batch operations to manage the S3 objects at scale.

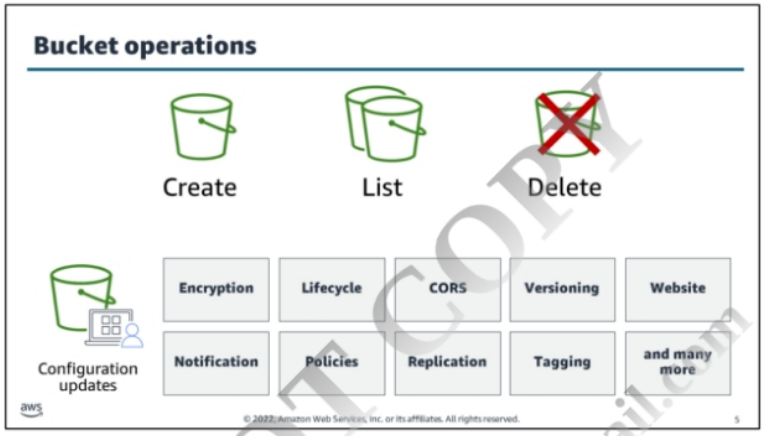
When working with S3 objects, you will be working with others in your organization. In the last module, you learned how you can grant access to buckets and objects through bucket policies and access control lists (ACLs). In this module, you will learn how to provide temporary access to your objects with presigned URLs. You will learn how to do the following:

- Configure a bucket for hosting a website
- Set up cross-origin resource sharing (CORS) to allow selective cross-origin access to your Amazon S3 resources

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification Module 6: Processing Your Storage Operations





After the AWS SDK or the AWS Command Line Interface (AWS CLI) is configured, you are ready to work with Amazon S3 through your application. With the right permissions, you can create a bucket, list buckets, delete a bucket, and configure a bucket.

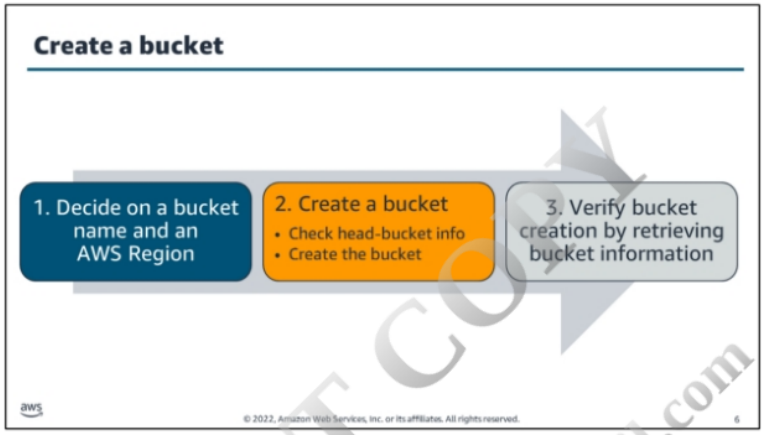
Configuring Amazon S3 resources

When configuring a bucket, you can set the following types of configurations:

- **Permissions** – Manage access permissions and object ownership, from ACLs, to Bucket policies, to Amazon S3 access points.
  Properties – Specify the way buckets function and manage objects.
  You can also set properties, such as enabling versioning for the objects of the bucket, setting event notifications, logs, website hosting, and more.
- **Management** – Manage objects.
  Manage the data by creating rules for replication of objects to automatically and asynchronously copying objects across Amazon S3 buckets. Alternatively, create lifecycle rules to define actions that Amazon S3 applies to a group of objects. Examples include transitioning objects to another storage class, archiving them, or deleting them after a specified period of time.

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

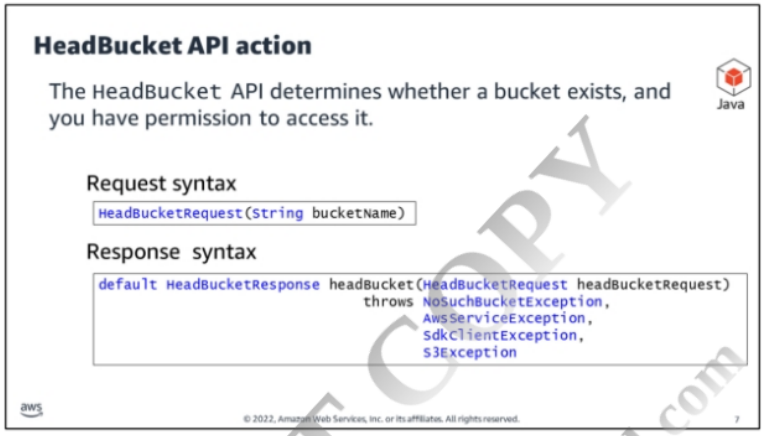
AWS Training and Certification Module 6: Processing Your Storage Operations



You will need a bucket to store and work with objects. After you create a bucket, you cannot change its name or AWS Region. The AWS account that creates the bucket owns it. An AWS account can create up to 100 buckets by default, but can request a service limit increase to a maximum of 1,000 buckets.

To create a bucket, follow these steps:

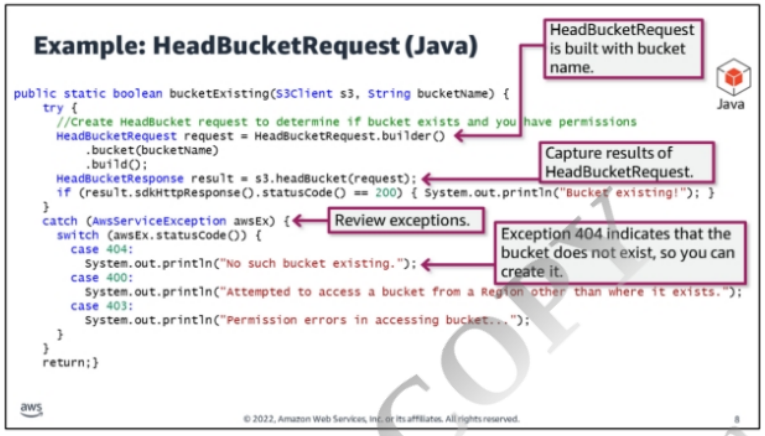
1. Decide on a bucket name and AWS Region for the bucket.
2. To create the bucket on the AWS Cloud, you must do the following:
   a) Create a client, specifying the AWS Region where the bucket will exist.
   b) Determine whether the bucket exists. You can do this by checking the head-bucket information. If the bucket exists and you have permissions to access it, the HEAD request returns the status code `200 OK`. If the bucket does not exist or you do not have permissions to access it, the HEAD request returns a status code `404 Not Found` or `403 Forbidden`.
   c) If you determine that the bucket does not exist, send a `CreateBucket` request. The bucket is created in the Region specified by the client.
3. Verify the bucket creation by retrieving bucket information and checking the bucket location.



The HeadBucket operation is an HTTP HEAD request to the bucket. It returns the headers that an HTTP GET operation would return.

A HeadBucket **request** is useful in determining whether a bucket exists and if you have permission to access it. If the bucket exists and you have permission to access it, the action returns a code 200 OK. If the bucket does not exist or you do not have permission to access it, the HEAD request returns a generic 404 Not Found or 403 Forbidden code. A message body is not included, so you cannot determine the exception beyond these error codes.

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Here is a Java code example in which a HeadBucket request is made on the specified bucket name. The results of the API call are captured and interpreted.

Example: HeadBucketRequest (Python)

Python

```python
def verifyBucketName(s3Client, bucket):
    try:
        ## check if a bucket already exists in AWS
        s3Client.head_bucket(Bucket=bucket)
        # If the previous command is successful, the bucket is already in your account.
        raise SystemExit('This bucket has already been created')
    except botocore.exceptions.ClientError as e:
        error_code = int(e.response['Error']['Code'])
        if error_code == 404:
            ## If you receive a 404 error code, a bucket with that name
            ##  does not exist anywhere in AWS.
            print('Existing Bucket Not Found, please proceed')
        if error_code == 403:
            ## If you receive a 403 error code, a bucket with that name exists
            ##  in another AWS account.
            raise SystemExit('This bucket has already owned by another AWS Account')
```

HeadBucketRequest is made to determine whether the bucket exists.

Here is a Python code example in which a `HeadBucket` request is made on the specified bucket name. The results of the API call are captured and interpreted.

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Example: HeadBucketRequest (.NET)

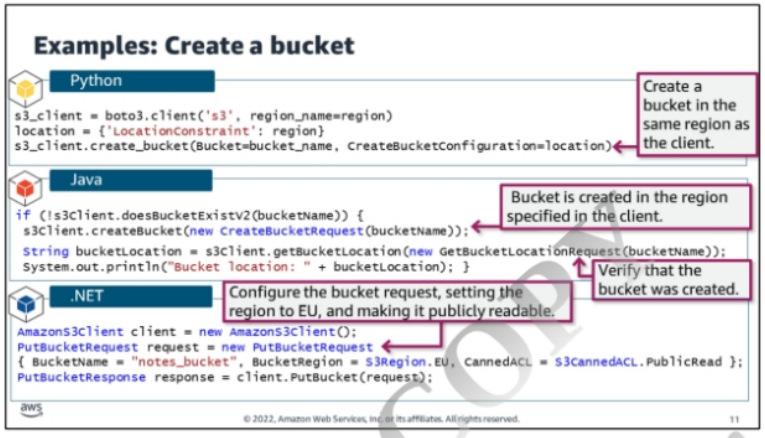
.NET

```dotnet
async Task VerifyBucketName(IAmazonS3 s3Client, string bucketName)
{
    bool exists = false;
    // Check if a bucket already exists in AWS
    exists = await AmazonS3Util.DoesS3BucketExistV2Async(s3Client, bucketName);
    if (exists)
    {
        // DoesS3BucketExistV2Async returns true if the bucket exists, but that does not
        // necessarily mean it belongs to your account as the method catches AccessDenied
        // and other exceptions.
        Console.WriteLine("This bucket already exists in your, or someone else's, account."
);
        Environment.Exit(0);
    }
    else
    {   Console.WriteLine("The bucket does not exist."); }
```

HeadBucketRequest with bucket name.

Here is a .NET code example in which a HeadBucket request is made on the specified bucket name. The results of the API call are captured and interpreted.

DO NOT COPY
sameersheik68@gmail.com

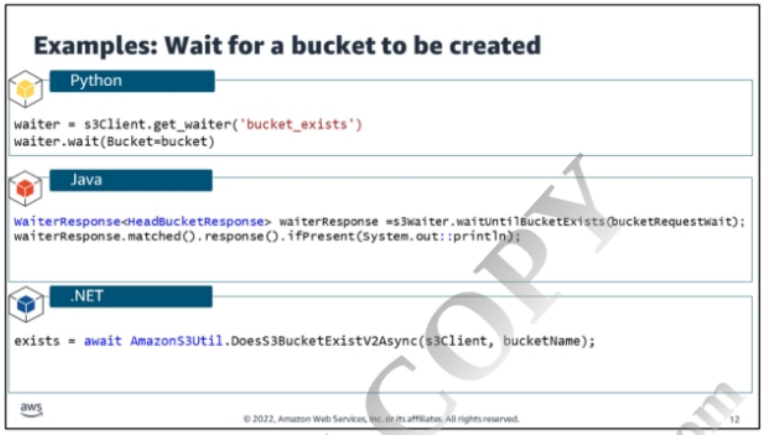


Some SDKs have methods that use the `HeadBucket` operation, but they provide limited feedback. For example, if you are interested in knowing only whether a bucket exists and not whether you own the bucket, you can use `doesBucketExistV2` in Java.

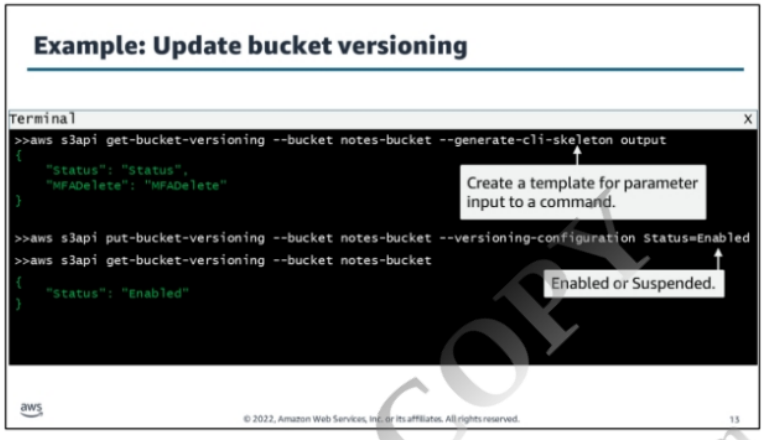
In the .NET example, some configurations are set programmatically when you construct the request.

*For Python, to create a bucket to us-east-1, do not include the location constraint property because it will cause an error. Not including the location constraint will create the bucket in us-east-1.

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



You could continuously loop through with the `headbucket` command until a bucket is created. Alternatively, you can use the waiters, which wait until the bucket exits.

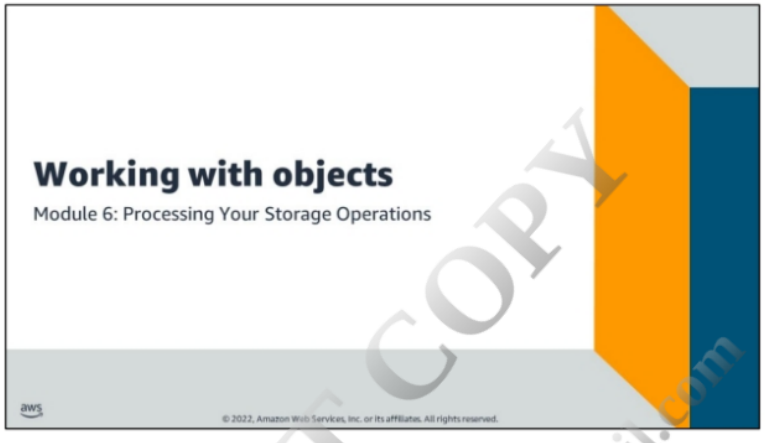


The --generate-cli-skeleton parameter causes a command not to run, but instead to generate and display a parameter template with all of the parameters that the command supports.

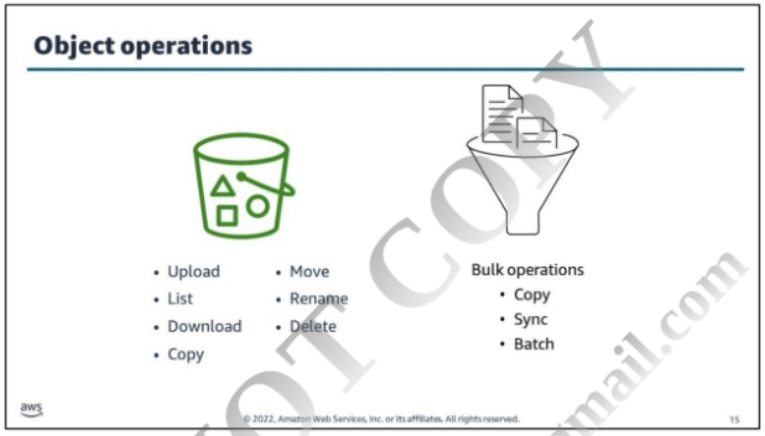
You can change a bucket's configuration even after it is created.
In the example, the bucket's versioning is enabled. You can also prevent accidental or intentional deletion of objects in an S3 bucket by making the bucket MFA-protected. This provides an extra layer of protection by requiring a multi-factor authentication (MFA) token from an authorized user.

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification Module 6: Processing Your Storage Operations



DO NOT COPY
sameersheik68@gmail.com



You can perform the following operations on objects: Upload, List, Download, Copy, Move, Rename, Delete.

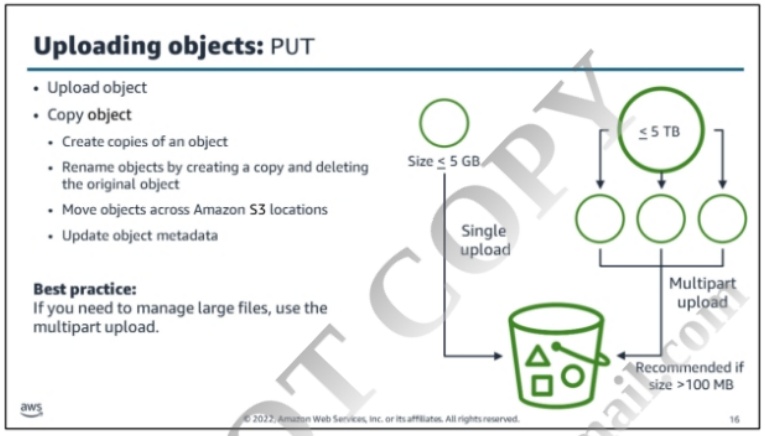
Bulk operations include: Copy, Sync and Batch.

Object operations can be done one at a time or in batches.

Be aware of bucket properties such as encryption, versioning, and lifecycle. They may require additional header or subresource information. For example, by default, the GET action returns the current version of an object. To return a different version, use the versioned subresource.

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification
Module 6: Processing Your Storage Operations



Amazon S3 delivers strong read-after-write consistency. After a successful write of a new object or an overwrite of an existing object, any subsequent read request immediately receives the latest version of the object. Amazon S3 also provides strong consistency for list operations. Therefore, after a write, you can immediately perform a listing of the objects in a bucket with any changes reflected.

You can upload or copy objects of up to 5 GB in a single PUT operation. For larger objects up to 5 TB, you must use the multipart upload API.

Multipart uploads
It is a best practice to use `aws s3` commands (such as `aws s3 cp`) for multipart uploads and downloads. They automatically perform multipart uploading and downloading based on the file size.

You can also use `aws s3api` commands to create a multipart upload. Using the `aws s3api` commands for multipart upload gives you more flexibility because you can upload a single object

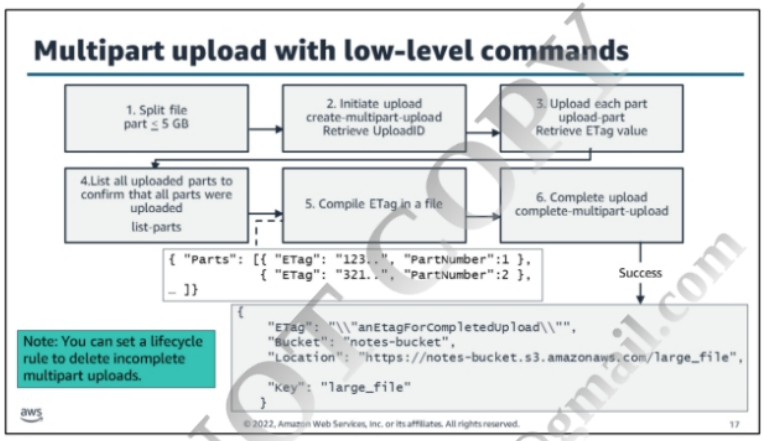
as a set of parts. For example, you might be uploading from multiple servers to one or more buckets. If one of the parts fails to upload, you can retransmit that particular part without retransmitting the remaining parts. Or you might need to stop or resume a large multipart upload manually. When you stop an upload, Amazon S3 deletes all the parts that were already uploaded and frees up storage.

Stopping uploads

Amazon S3 retains all the parts on the server until you complete or stop the upload. To avoid unnecessary storage costs related to incomplete uploads, complete or stop an upload. We recommend that you enable the `AbortIncompleteMultipartUpload` lifecycle rule on your Amazon S3 buckets. This rule directs Amazon S3 to stop multipart uploads that do not complete within a specified number of days after being initiated. When the set time limit is exceeded, Amazon S3 stops the upload and then deletes the incomplete upload data.

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification Module 6: Processing Your Storage Operations



Consider using multipart upload for objects larger than 100 MB. With multipart uploads, you can perform the following tasks:

- Upload parts in parallel to improve throughput
- Recover quickly from network issues
- Pause and resume object uploads
- Begin an upload before you know the final size of an object

Commands for multipart uploads

There are special commands for multipart uploads that rely on `ETag` and `UploadID` values. You send a special command to initialize the multipart upload. After all the parts of your object are uploaded to the server, you must send a complete multipart upload request that indicates that multipart upload has been completed. Amazon S3 then assembles these parts and creates the complete object.

If the upload is successful, you get a success message similar to this:

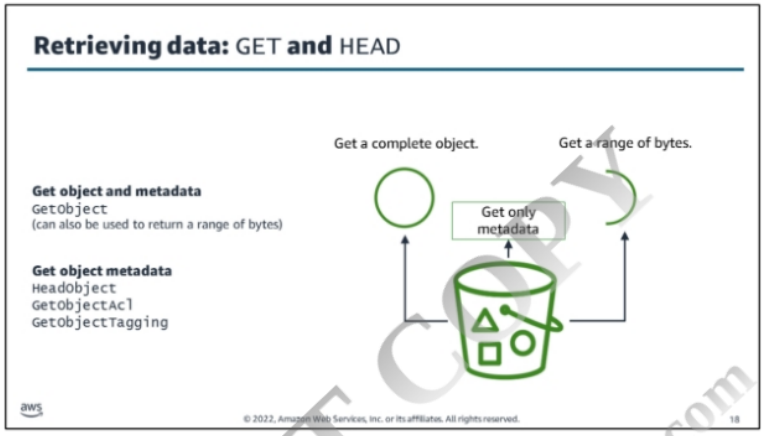
```
{
    "ETag": "\\"anEtagForCompletedUpload\\"",
    "Bucket": "notes-bucket",
    "Location": "https://notes-bucket.s3.amazonaws.com/large_file",

    "Key": "large_file"
  }
```

If the upload was not successful, some cleanup is required to avoid being charged. You can set a lifecycle rule to delete incomplete multipart uploads.

For more information, see "How do I use the AWS CLI to perform a multipart upload of a file to Amazon S3?" in the Amazon S3 Knowledge Center (https://aws.amazon.com/premiumsupport/knowledge-center/s3-multipart-upload-cli/).

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification Module 6: Processing Your Storage Operations

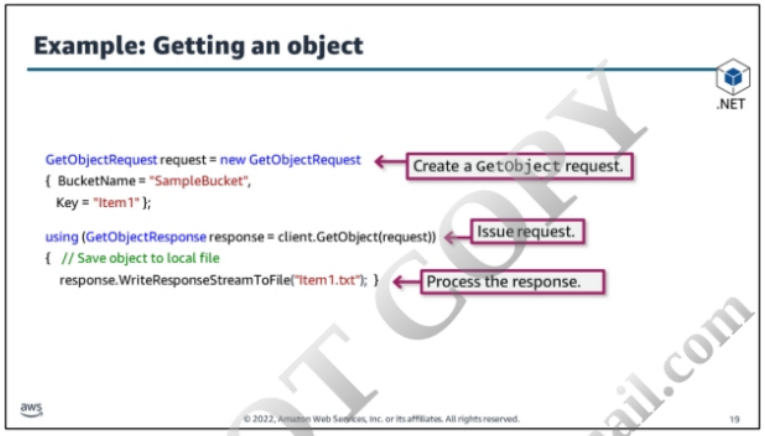


You can retrieve a complete object in a single GET request. You can also retrieve an object in parts by specifying the range of bytes needed. This is useful in scenarios in which network connectivity is poor or your application can or must process only subsets of object data.

By default, the GET action returns the current version of an object. To return a different version, use the `versionId` subresource.

Sometimes you might be interested in only the metadata of an object and not the content of the object. A HEAD request returns the headers that contain the metadata information of an object in the same way as the GET request. However, the HEAD does not return the body of the object.

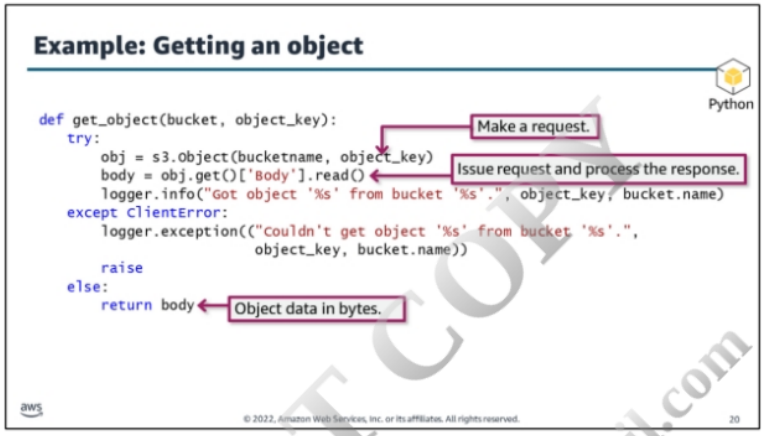
DO NOT COPY
sameersheik68@gmail.com



When downloading an object from an S3 bucket, use the `GetObject` method to get an object from a bucket. Then use of the `GetObjectResponse` methods to process the data stream. The object is streamed directly from Amazon S3. This means that your network connection will remain open until you read all the data or close the input stream.

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

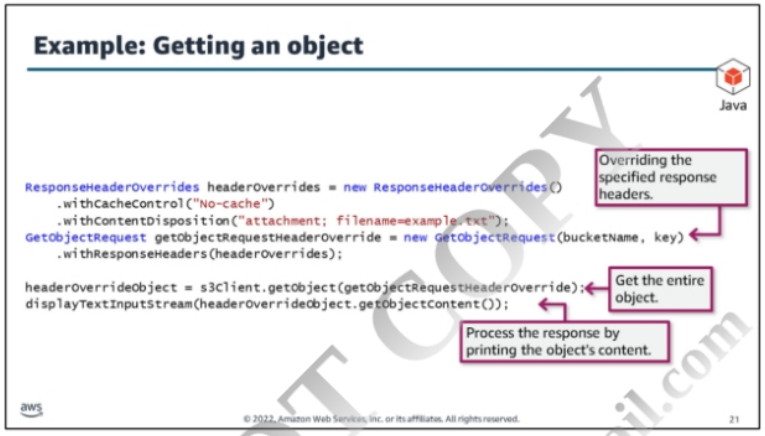
AWS Training and Certification Module 6: Processing Your Storage Operations



Create an object resource with the command:
`object = s3.Object('bucket_name','key')`
Where the parameters are the Object's bucket_name identifier, and the Object's key identifier.

The command returns an object resource.

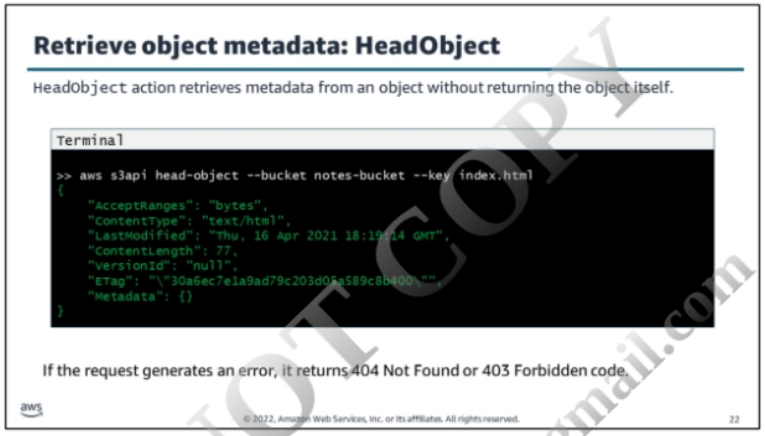
The get() command retrieves the object from Amazon S3. The return type is a dictionary which contains information such as the object data, the delete marker and more. The object is streamed directly from Amazon S3. This means that your network connection will remain open until you read all the data or close the input stream.

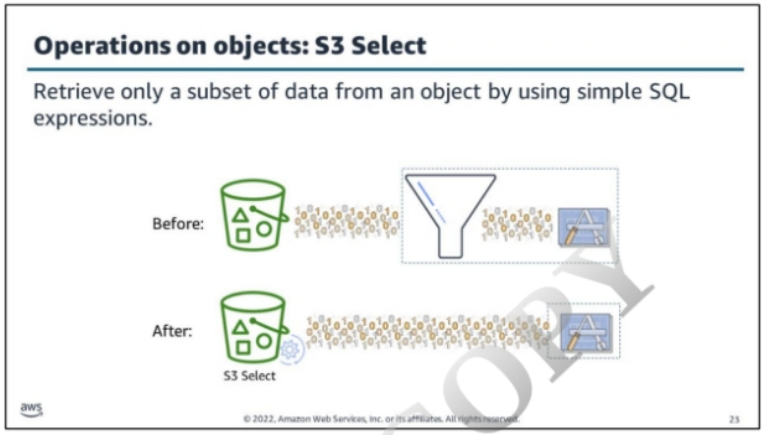


When downloading an object from an S3 bucket, use the `GetObject` method to get an object from a bucket. Then use of the `GetObjectResponse` methods to process the data stream. The object is streamed directly from Amazon S3. This means that your network connection will remain open until you read all the data or close the input stream.

It is possible, when getting an object, to override aspects of a response header. For example, a client could dynamically change the Content-Disposition header of a single object, so that it appears to have a different file name for different callers. One client could be configured return the object with
Content-Disposition: attachment; filename=FileName1.exe while another could return that same object with headers Content-Disposition: attachment; filename=FileName2.pdf

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



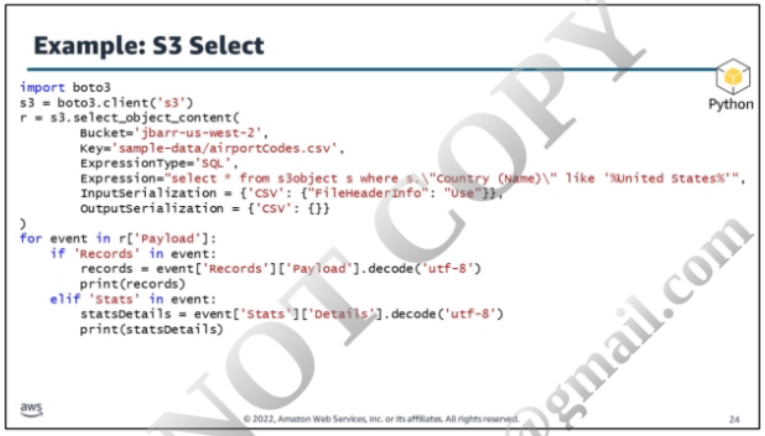


Amazon S3 Select analyzes and processes data in an object in Amazon S3 buckets faster and cheaper than the GET method. With GET, you retrieve the whole object in an Amazon S3 bucket. With SELECT, you can retrieve a subset of data from an object by using simple SQL expressions. Your applications do not have to use compute resources to scan and filter the data from an object. Not using compute resources to scan or filter potentially increases query performance and reduces cost.

To use Amazon S3 Select, change your application code to use SELECT instead of GET.

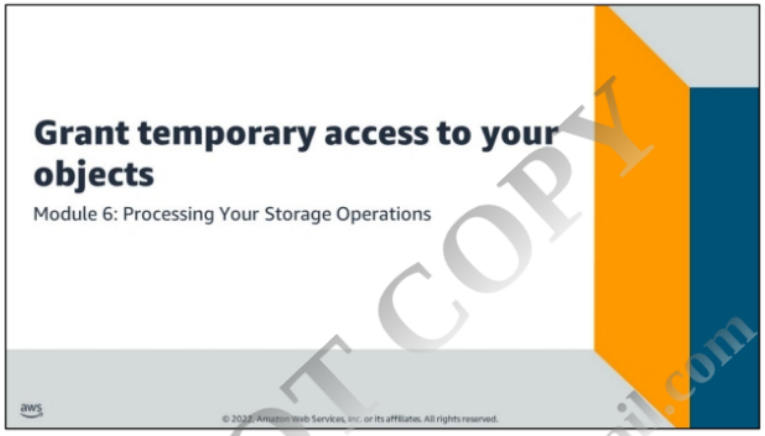
Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification Module 6: Processing Your Storage Operations



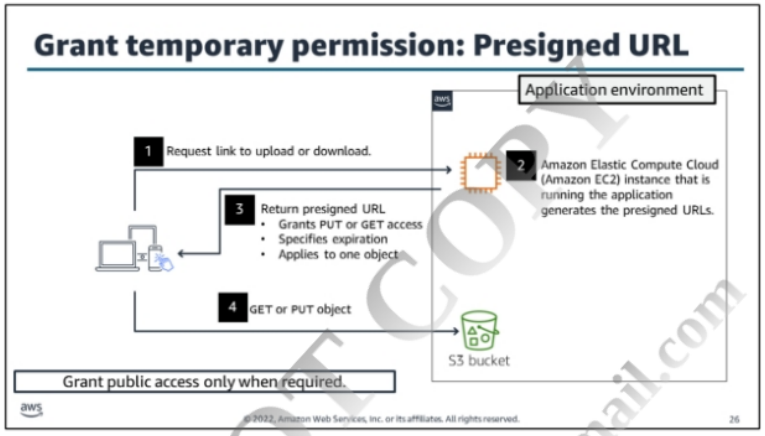
The Python example shows how to retrieve data from an object in the first column in CSV format.

Without S3 Select, you must download, decompress, and process the entire CSV file so that you can get the data you need. With S3 Select, you can use a SQL expression to return only the data that you're interested in. This means that you're dealing with an order of magnitude less data, which improves the performance of your underlying applications.



Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification
Module 6: Processing Your Storage Operations



All objects and buckets are private by default. Presigned URLs are useful if you want your user to retrieve (GET) or upload (PUT) a specific object to your bucket without requiring AWS security credentials or permissions. When you create a presigned URL, you must provide the following information:

- Your security credentials, including:
  - AWS Identity and Access Management (IAM) user
  - IAM instance profile
  - AWS Security Token Service (AWS STS)
- Bucket name
- Object key
- HTTP method (PUT for uploading objects, GET for retrieving objects)
- Expiration date and time

The presigned URLs are valid only for the specified duration. The time-limited URL is valid to anyone who has the URL.

Note: Do not share your AWS security credentials (secret access key ID) with others.

Use case

Suppose that in the application that you are developing, you used presigned URLs so that users could share their favorite list of MP3 files with each other. This feature is not one that you are building in the course, but it is a relevant use case.

Example

The following is an AWS CLI command that generates a presigned URL for the `readme.txt` object in the S3 bucket named `notes-bucket`. The presigned URL expires in one hour.

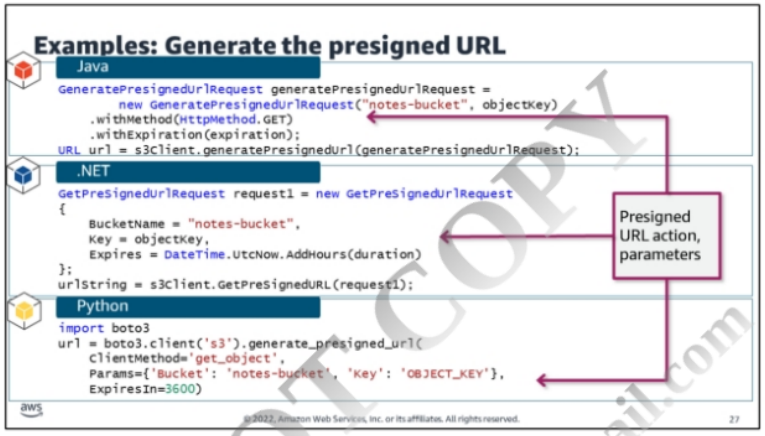
```
>> aws s3 presign s3://notes-bucket/readme.txt --expires-in 3600
```

The returned URL looks like this:

```
https://notes-
bucket.s3.amazonaws.com/test2.txt?AWSAccessKeyId=AKIAEXAMPLEACCESSK
EY&
Signature=EXHCcBe%EXAMPLEKnz3r8O0AgEXAMPLE&Expires=1556132848
```


Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification Module 6: Processing Your Storage Operations



DO NOT COPY
sameersheik68@gmail.com

Demo

Module 6: Processing Your Storage Operations

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

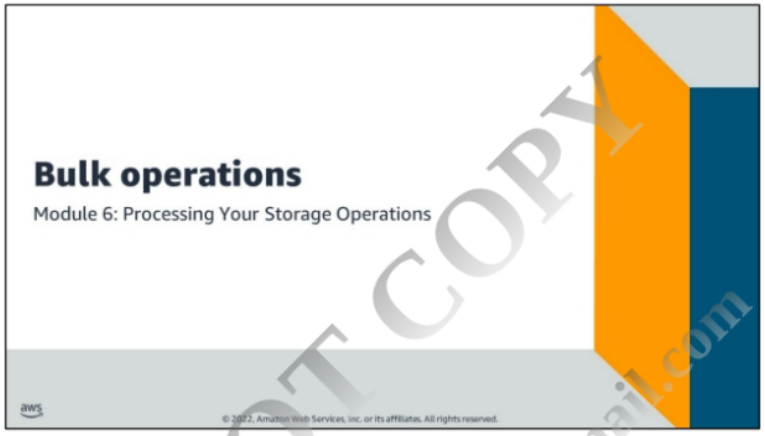
AWS Training and Certification
Module 6: Processing Your Storage Operations

Product demonstration

- Using SDK for CRUD operations
- Using the AWS CLI for presigned URL and continuation token

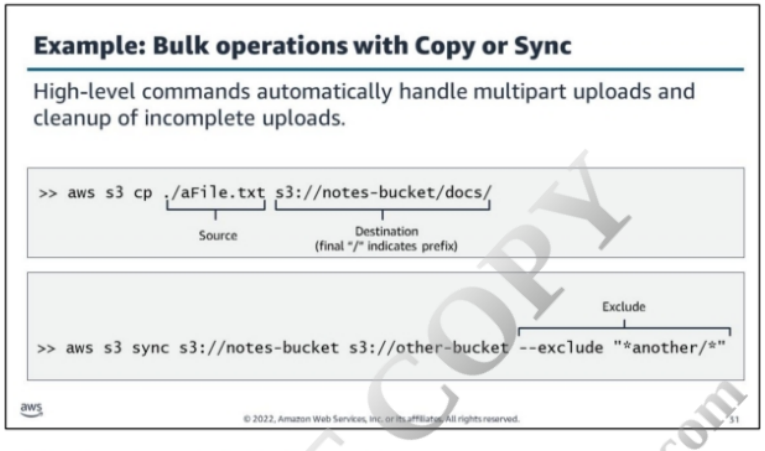
29

DO NOT COPY
sameersheik68@gmail.com



Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification Module 6: Processing Your Storage Operations



Use the `copy` operation for the following actions:

- Copy a local object or Amazon S3 object to another location
- Create copies of an object
- Rename an object
- Move an object to a different Amazon S3 location
- Update an object's metadata

Copy example

```
>> aws s3 cp ./aFile.txt s3://notes-bucket/docs/
```

The first example copies a file (aFile.txt) from the local directory to the S3 bucket (notes-bucket) and into the prefix `docs`. The last backslash (/) is required. If it is missing, the file will be renamed to `docs`. If you need to copy multiple files, the `aws s3 cp` command requires the `--recursive` parameter.

```
>> aws s3 cp ./ s3://notes-bucket/ --recursive --exclude "*" --include "*.jpg" --include "*.txt"
```

This example copies all files ending in .jpg or .txt recursively, from the current local directory to the S3 bucket named notes-bucket. Because the command includes all files, if you want only a subset, exclude all files and then use the `include` option.

Sync example

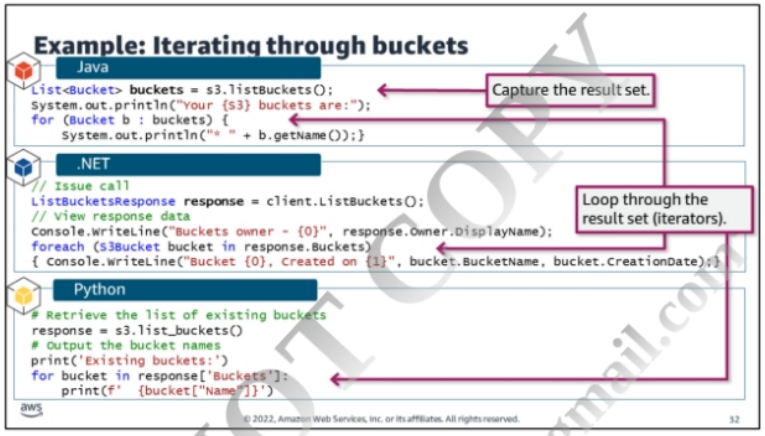
```
>> aws s3 sync s3://notes-bucket s3://other-bucket --exclude
"*another/*"
```

The second example illustrates the `sync` command. When using the `aws s3 sync` command, it copies an entire directory by default. Subsequent use of the `sync` command copies new or modified files.

You can also use the `copy` operation to change the storage class of an object from standard to reduced redundancy or vice versa. For information about storage classes, see "Using Amazon S3 storage classes" in the *Amazon Simple Storage Service (S3) User Guide* (https://docs.aws.amazon.com/AmazonS3/latest/dev/ChgStoClsOfObj.html).

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

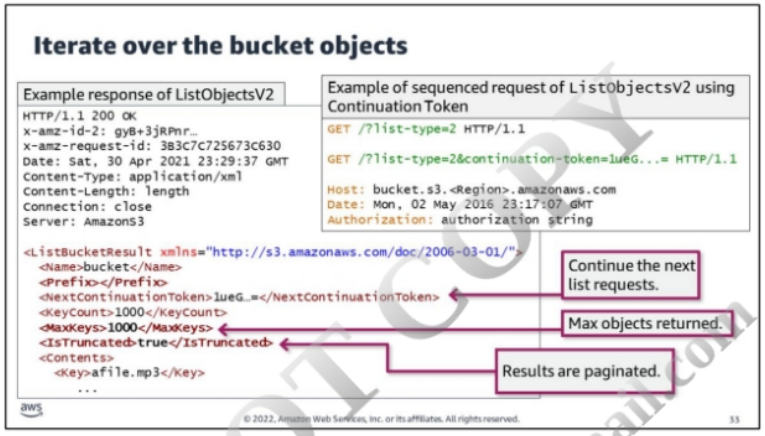
AWS Training and Certification Module 6: Processing Your Storage Operations



In these examples, `ListBucket` returns a list of all buckets owned by the authenticated sender of the request.
Notice the use of iterators to loop through the result set.

DO NOT COPY sameersheik68@gmail.com

© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved.



Some AWS operations return results that are paginated, meaning that the results are listed one page at a time. Using a continuation token, you can process the next set of paginated results. For example, listing the contents of your Amazon S3 bucket returns paginated results, up to 1000 objects at a time. This means that if you have 1001 objects, you get a page of results listing the first 1000. Therefore, to retrieve the next *page* of results, you must send subsequent requests with a continuation token. You then repeat this process until you've read the complete dataset.

The example code illustrates the use of the `ListObjectV2` command, which lists objects contained in a bucket. With `ListObjectsV2`, you can list up to 1000 objects in a bucket. Notice that the value of the `MaxKeys` parameter is set to 1000, but you can use this parameter to further adjust the number of objects returned.

In the response, pay attention to this value:

- **`IsTruncated`** – A value of true indicates that the response is incomplete, and it was truncated to the number of objects specified by `MaxKeys`.
- **`ContinuationToken`** – If sent with the request, it is included in the response.
- **`NextContinuationToken`** – Sent when `IsTruncated` is true, which means that the bucket contains more keys that can be listed. In the request, use the value of the **`NextContinuationToken`** as the value for new continuation-token, to continue the next list requests to Amazon S3.

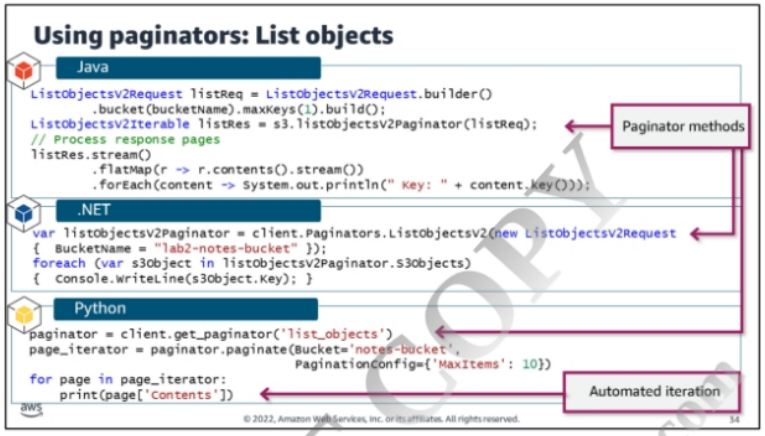
Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification Module 6: Processing Your Storage Operations

When making the request, you can use the continuation token, `prefix`, `Delimiter`, and `start-after` token parameters to control which objects to return.

- **`start-after`** – Indicates lexicographically starting after a string.

© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved.

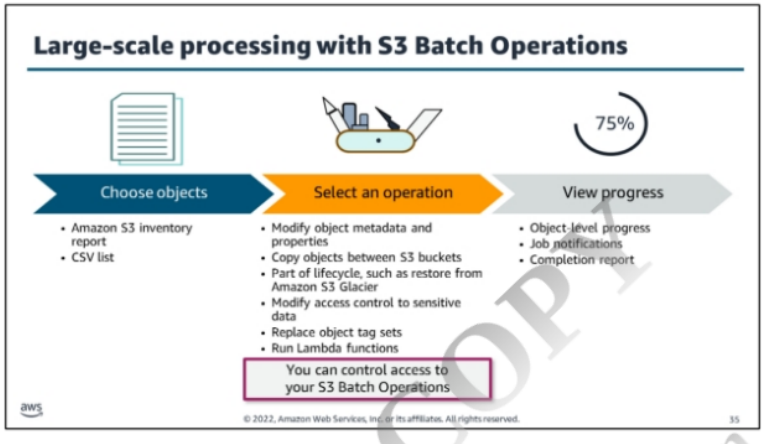


To retrieve all the data of a paginated response, you can create a loop that makes multiple requests. The operation replaces the next token in each iteration. However, when using paginators, this extra loop isn't necessary. AWS SDKs include the paginator feature to abstract the process of iterating over a complete result set of a truncated API operation.

Paginators can access the **full responses** or only the **key results** for the AWS resource.

- **Responses** is a property on the paginator that you can use to iterate through full responses from AWS for the operation. You can access any properties contained in the response.
- **Key results** are unique to each operation and represent the properties in the response whose data is most likely to be truncated because of length. For the `ListObjectsV2` paginator, the key results are `S3Objects` and `CommonPrefixes`.

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

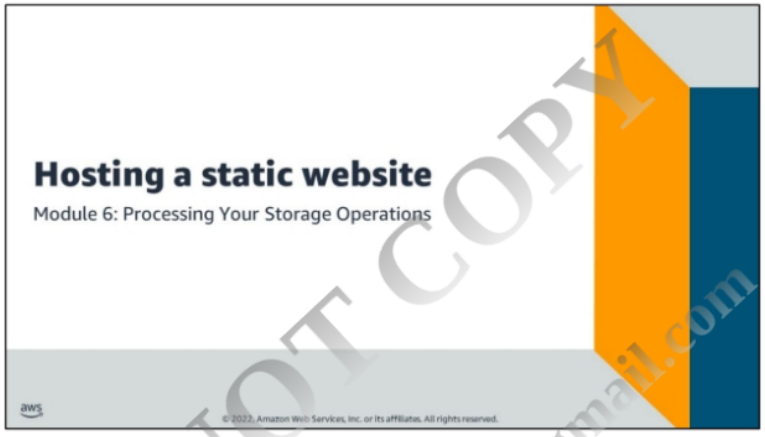


S3 Batch Operations performs large-scale operations on Amazon S3 objects. Batch operations are performed from the AWS Management Console or though the API. You can label and control access to your S3 Batch Operations jobs.

The objects for the batch operations, are specified through an Amazon S3 inventory report or a custom CSV file. These files, known as manifest files, contain a list of object keys that you want Amazon S3 to act on. Each row in the file includes the bucket name, object key, and (optionally) the object version. Version IDs must be included for all objects or omitted for all objects. Object keys must be URL-encoded.

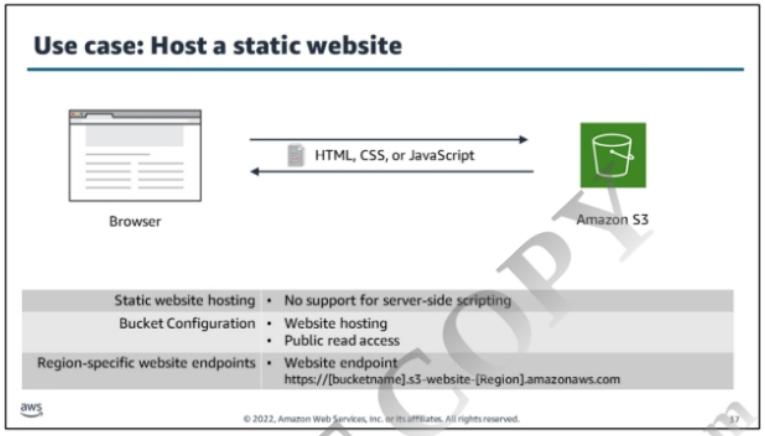
S3 Batch Operations supports the following operations:

- Put object copy
- Initiate restore object
- Put object ACL
- Put object tagging
- Manage Object Lock retention dates
- Manage Object Lock legal hold
- Run a custom Lambda operation



Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

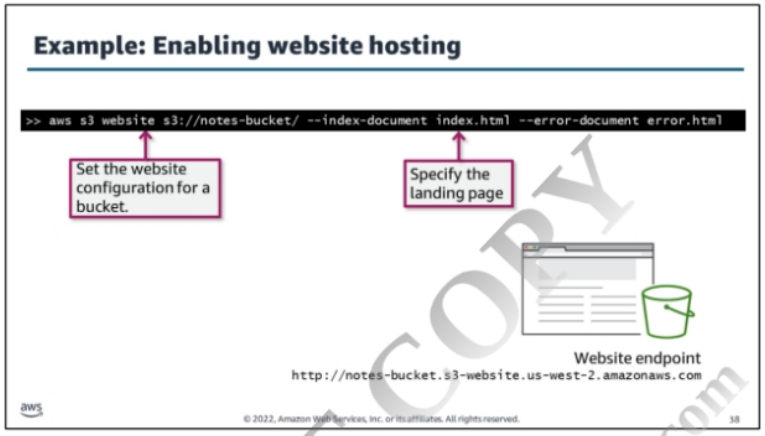
AWS Training and Certification
Module 6: Processing Your Storage Operations



You can configure an S3 bucket as a static website. A static website contains static resources such as HTML or images, but it does not contain server-side processing or scripting.

The website is available at the AWS Region-specific website endpoint of the bucket.

For the website to be publicly accessible, the bucket must have public read access.



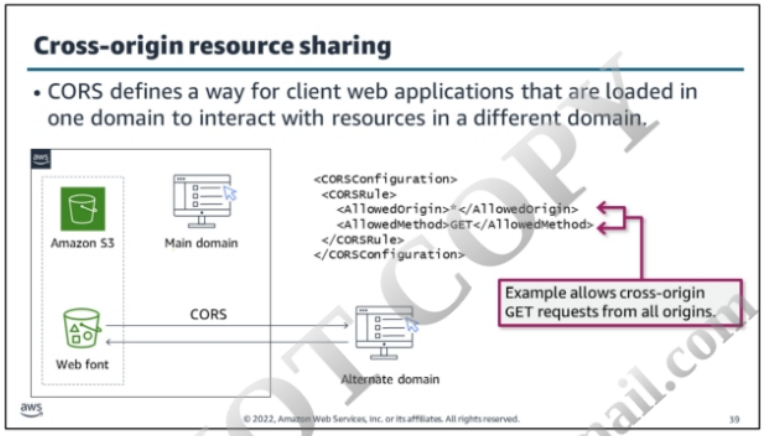
The API-level `s3api` command `website` is used to configure the S3 bucket as a static website.

Use `website` to set the website configuration for a bucket. The website is available at the AWS Region-specific website endpoint of the bucket. An Amazon S3 website endpoint is optimized for access from a web browser. The index-document options indicate the landing page where visitors will be directed.

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification

Module 6: Processing Your Storage Operations



With CORS support in Amazon S3, you can build web applications with Amazon S3 and selectively allow cross-origin access to your Amazon S3 resources.

To enable CORS, create a CORS configuration XML file. This file contains rules that identify the origins that can access your bucket, the operations (HTTP methods) that you support for each origin, and other operation-specific information. You can add up to 100 rules to the configuration. You can apply the CORS configuration to the S3 bucket by using the AWS SDK.

Examples

Consider the following examples:

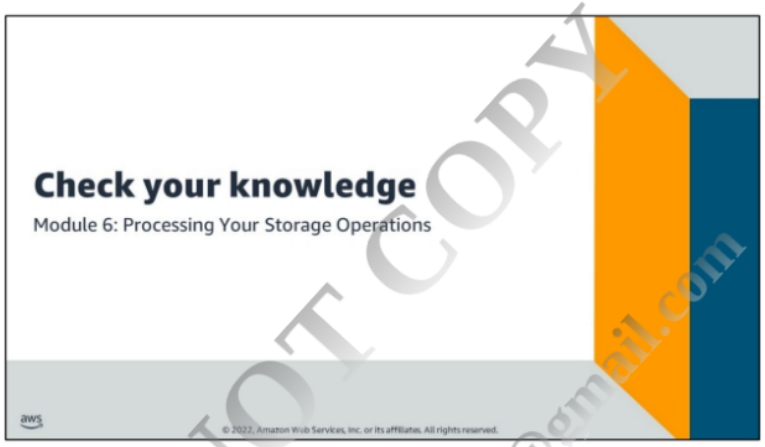
- You want to host a web font in your S3 bucket. A webpage in an alternate domain might try to use this web font. Before the browser loads this webpage, it performs a CORS check to make sure that the domain from which the page is being loaded is allowed to access S3 bucket resources.
- JavaScript in one domain's webpages (http://www.example.com) wants to use resources from your S3 bucket by using the endpoint website.s3.amazonaws.com. The browser allows such cross-domain access only if CORS is enabled on your bucket.

For more information, see the following in the *Amazon Simple Storage Service (S3) User Guide*:

- "Using cross-origin resource sharing (CORS)" (https://docs.aws.amazon.com/AmazonS3/latest/dev/cors.html)
- "CORS configuration" (https://docs.aws.amazon.com/AmazonS3/latest/dev/ManageCorsUsing.html)

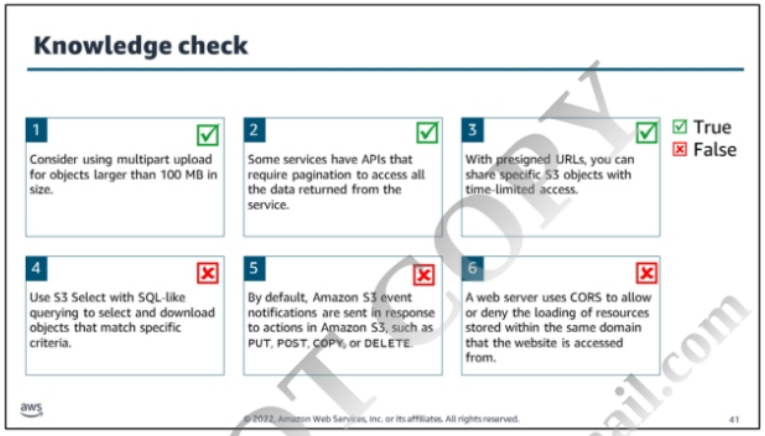
Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification Module 6: Processing Your Storage Operations



DO NOT COPY sameersheik68@gmail.com

© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved.

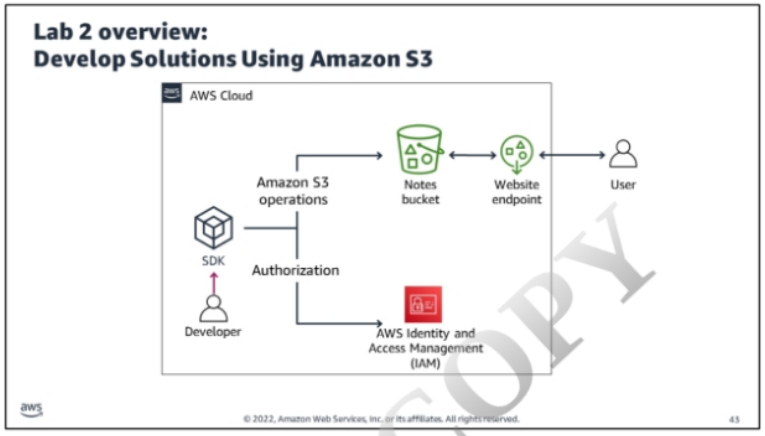


1. (True) It is a best practice to consider using multipart upload for objects larger than 100 MB. This is true even though the largest object that can be uploaded in a single PUT method is 5 GB.
2. (True) You reviewed an example of paginated results when listing the content of a bucket.
3. (True) Using presigned URL, you can grant users temporary access to specific S3 objects.
4. (False) With S3 Select, you can use SQL-like syntax to retrieve only a subset of data from an object. To get a subset of objects that match criteria, such as file types or prefixes, use the Amazon S3 GET command.
5. (False) You can configure a bucket to send Amazon S3 event notifications in response to actions in Amazon S3, such as PUT, POST, COPY, or DELETE.
6. (False) Cross-Origin Resource Sharing (CORS) protects your data from being accessed from other domains. In CORS, the origin is a web server, and a resource is an asset at a different domain. With CORS a web server indicates the origins (domain, scheme, or port) other than its own from which a browser should permit loading of resources.

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification Module 6: Processing Your Storage Operations

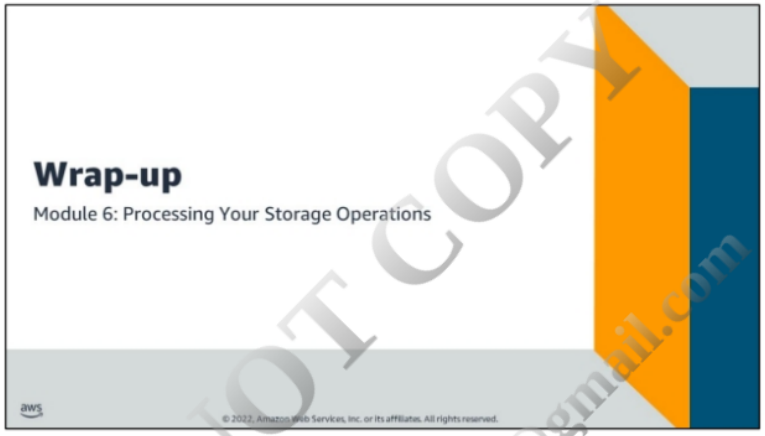
© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved.



Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification

Module 6: Processing Your Storage Operations



Module summary

You are now able to:

- Perform key bucket and object operations
- Explain how to handle large files and a large number of files
- Create and configure an S3 bucket for hosting a static website
- Explain how to grant temporary access to your objects

Printed by: sameersheik68@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

AWS Training and Certification
Module 6: Processing Your Storage Operations

Thank you

Corrections, feedback, or other questions?
Contact us at https://support.aws.amazon.com/#/contacts/aws-training.
All trademarks are the property of their owners.

DO NOT COPY
sameersheik68@gmail.com

© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved.