



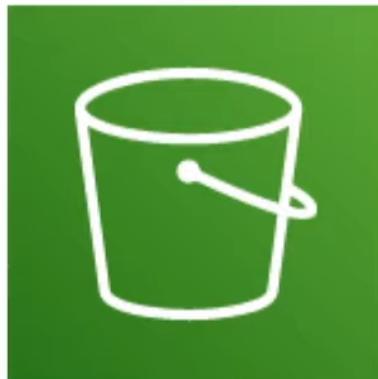
AWS Developer Associate

Simple Storage Service (S3)



Introduction to S3

Simple Storage Service (S3)



Object-based storage service.

Serverless storage in the cloud.

Don't worry about filesystems or disk space.



Introduction to S3

What is Object Storage (Object-based Storage)?

data storage architecture that manages data as objects, **as opposed** to other storage architectures:

- **file systems** which manages data as a files and file hierarchy, and
- **block storage** which manages data as blocks within sectors and tracks.

S3 provides you with **unlimited storage**. You don't need to think about the underlying infrastructure

The S3 Console provides an interface for you to upload and access your data



S3 Object

Objects contain your data. They are like files.

Object may consist of:

- **Key** this is the name of the object
- **Value** the data itself made up of a sequence of bytes
- **Version ID** when versioning enabled, the version of object
- **Metadata** additional information attached to the object

You can store data from **0 Bytes** to **5 Terabytes** in size



S3 Bucket

Buckets hold objects. Buckets can also have folders which in turn hold objects

S3 is a universal namespace so bucket names must be unique (think like having a domain name)



AWS Developer Associate

Simple Storage Service (S3)



Storage Classes



S3 - Storage Classes

Trade **Retrieval Time, Accessibility and Durability** for **Cheaper Storage**

11 9's (Eleven Nines) = 99.9999999999%

9 9's (Nine Nines) = 99.99999999%

Cheaper it gets

Standard (default) Fast! 99.99% Availability, **11 9's** Durability. Replicated across at least three AZs

Intelligent Tiering* Uses ML to analyze your object usage and determine the appropriate storage class. Data is moved to the most cost-effective access tier, without any performance impact or added overhead.

Standard Infrequently Accessed (IA) Still Fast! Cheaper if you access files less than once a month. Additional retrieval fee is applied. **50% less** than Standard (reduced availability)

One Zone IA Still Fast! Objects only exist in one AZ. Availability (is 99.5%). but cheaper than Standard IA by 20% less (Reduce durability) Data could get destroyed. A retrieval fee is applied.



Glacier

For long-term cold storage. Retrieval of data can take minutes to hours but the off is very cheap storage

Glacier Deep Archive The lowest cost storage class. Data retrieval time is 12 hours.





AWS Developer Associate

Simple Storage Service (S3)



Storage Class Comparision



S3 – Storage Classes Comparison

	Standard	Intelligent Tiering	Standard IA	One-Zone IA	Glacier	Glacier Deep Archive
Durability	11 9's	11 9's	11 9's	11 9's	11 9's	11 9's
Availability	99.99%	99.9%	99.9%	99.5%	N/A	N/A
Availability SLA	99.99%	99%	99%	99%	N/A	N/A
AZs	>3	>3	>3	1	>3	>3
Min Capacity charge per object	N/A	N/A	128kb	128kb	40kb	40kb
Min storage duration charge	N/A	30 days	30 days	30 days	90 days	180 days
Retrieval fee	N/A	N/A	Per GB	Per GB	Per GB	Per GB
First byte latency	ms	ms	ms	ms	mins to hours	hours

S3 Guarantees

Platform is built for 99.99% availability
Amazon guarantee 99.9% availability
Amazon guarantees 11' 9s of durability



AWS Developer Associate

Simple Storage Service (S3)



S3 Security



S3 – Security

All new buckets are **PRIVATE** when created by default

Logging per request can be turned on a bucket

Log files are generated and saved in a different bucket.
(even a bucket in a different AWS account if desired)

Access control is configured using **Bucket Policies** and
Access Control Lists (ACL)

Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through *new* access control lists (ACLS)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through *any* access control lists (ACLS)
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through *new* public bucket policies
S3 will block new bucket policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through *any* public bucket policies
S3 will ignore public and cross-account access for buckets with policies that grant public access to buckets and objects.

Cancel Save

Access Control Lists

Everyone

Access to the objects

- List objects
- Write objects

Access to this bucket's ACL

- Read bucket permissions
- Write bucket permissions

Cancel Save

Legacy feature (but not deprecated) of controlling access to buckets and objects.

Simple way of granting access

Bucket Policies

Use a policy to define complex rule access.

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "PublicReadGetObject",  
6       "Effect": "Allow",  
7       "Principal": "*",  
8       "Action": "s3:GetObject",  
9       "Resource": "arn:aws:s3:::www.exampro.co/*"  
10    }  
11  ]  
12 }
```



AWS Developer Associate

Simple Storage Service (S3)



S3 Encryption



S3 – Encryption

Encryption In Transit

Traffic between your local host and S3 is achieved via **SSL/TLS**

Server Side Encryption (SSE) - Encryption At Rest

Amazon help you encrypt the object data

S3 Managed Keys - (Amazon manages all the keys)

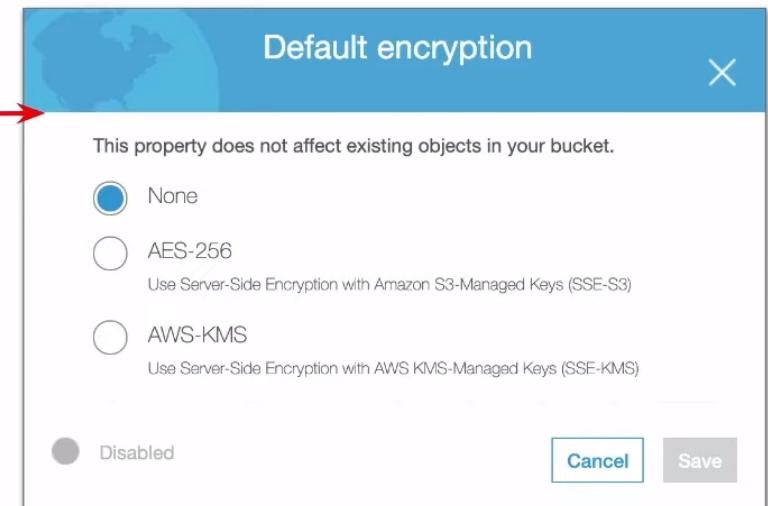
SSE-AES S3 handles the key, uses AES-256 algorithm

SSE-KMS Envelope encryption, AWS KMS and you manage the keys

SSE-C Customer provided key (you manage the keys)

Client-Side Encryption

You encrypt your own files before uploading them to S3





AWS Developer Associate

Simple Storage Service (S3)



Data Consistency



S3 – Data Consistency

New Objects (PUTS)

Read After Write Consistency

When you upload a new S3 object you are able **read immediately** after writing.

Overwrite (PUTS) or Delete Objects (DELETES)

Eventual Consistency

When you overwrite or delete an object it takes time for S3 to replicate versions to AZs.

If you were to read immediately, S3 may return you an old copy. You need to generally wait a few seconds before reading.



AWS Developer Associate

Simple Storage Service (S3)

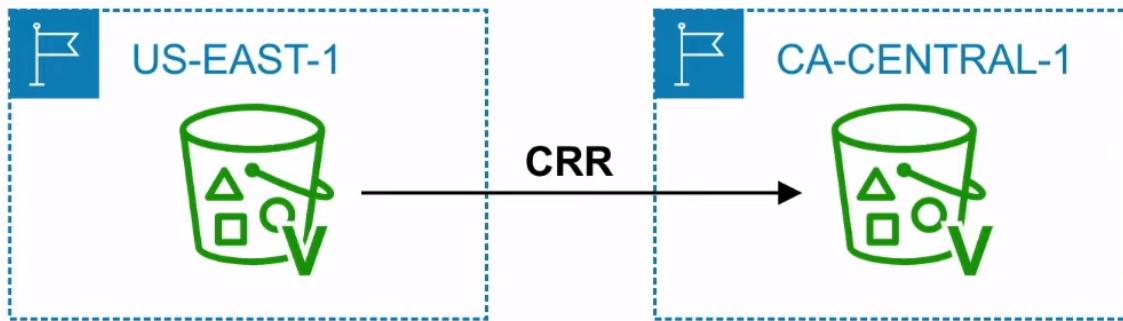


Cross-Region Replication



S3 – Cross Region Replication (CRR)

When enabled, any object that is uploaded will be **automatically replicated** to another region(s)
Provides higher durability and potential disaster recovery for objects



You must have **versioning** turned on both the **source** and **destination** buckets.
You can have CRR replicate to another AWS account



AWS Developer Associate

Simple Storage Service (S3)

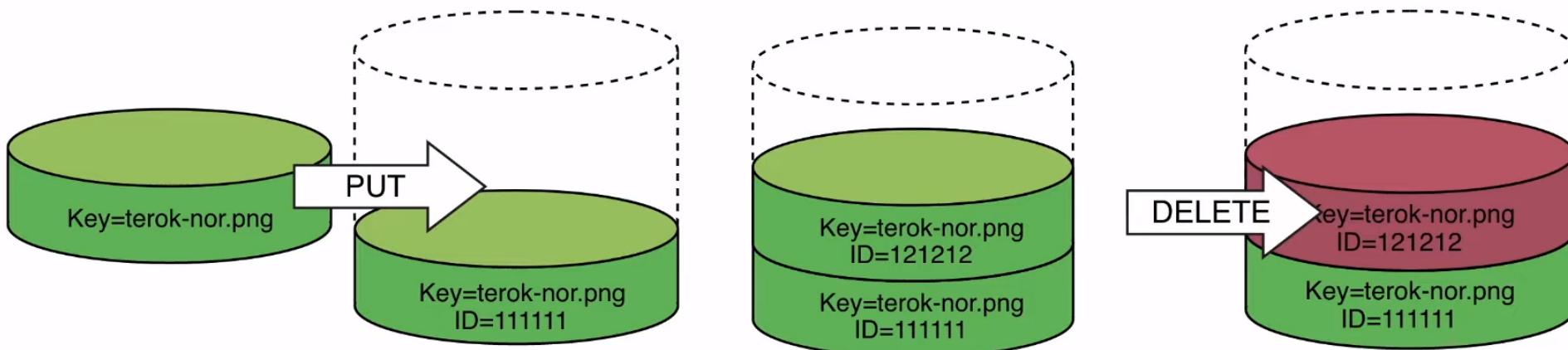
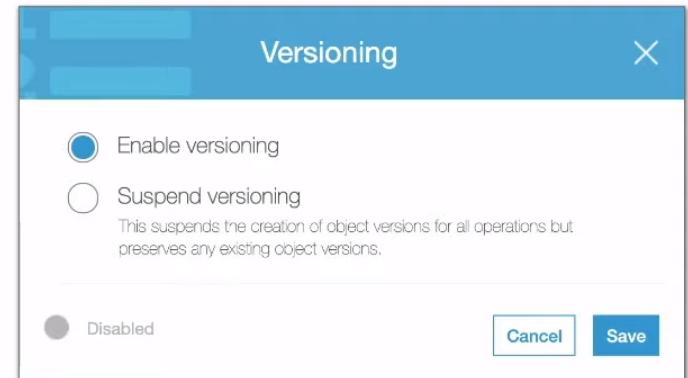


Versioning



S3 Versioning

- Store all versions of an object in S3
- Once enabled it cannot be disabled, only suspended on the bucket
- Fully integrates with S3 Lifecycle rules
- MFA Delete feature provides extra protection against deletion of your data





AWS Developer Associate

Simple Storage Service (S3)



Lifecycle Management



S3 Lifecycle Management

Automate the process of moving objects to different Storage classes or deleting objects all together.

Can be used together with **versioning**

Can be applied to both **current** and **previous** versions



Lifecycle rule

(1) Name and scope (2) Transitions (3) Expiration (4) Review

Storage class transition

There are [per-request fees](#) when using lifecycle to transition data to any S3 or S3 Glacier storage class. [Learn more](#) or see [Amazon S3 pricing](#)

Current version Previous versions

For current versions of objects [+ Add transition](#)

Object creation Days after creation

Transition to Glacier after 7 X

⚠️ Transitioning small objects to Glacier or Glacier Deep Archive will increase costs.

Before creating a lifecycle rule that transitions small objects to Glacier or Glacier Deep Archive, consider how many objects will be transitioned and how



AWS Developer Associate

Simple Storage Service (S3)



Transfer Acceleration



S3 - Transfer Acceleration

Fast and secure transfer of files **over long distances** between your end users and an S3 bucket.



Utilizes

CloudFront's distributed



Edge Locations.

Instead of uploading to your bucket, users use a **distinct URL** for an Edge Location

As data arrives at the Edge Location it is automatically routed to S3 over a specially optimized network path. (Amazon's backbone network)





AWS Developer Associate

Simple Storage Service (S3)



Presigned URLs



S3 – PresignedUrls

Generate a url which provides you temporary access to an object to either upload or download object data. Presigned Urls are commonly used **to provide access to private objects**. You can use AWS CLI or AWS SDK to generate Presigned Urls.

```
aws s3 presign s3://mybucket/myobject --expires-in 300
```



```
https://mybucket.s3.amazonaws.com/myobject?AWSAccessKeyId=AKIAJXXXXXXXXXXXXXX&Expires=1503602631&Signature=ib0GfAovnhIF13DALdAgsdtg2s%3D
```

You have a web-application which needs to allow users to download files from a password protected part of your web-app. Your web-app generates presigned url which expires after 5 seconds. The user downloads the file.



AWS Developer Associate

Simple Storage Service (S3)



MFA Delete



S3 - MFA Delete

MFA Delete ensures users cannot delete objects from a bucket unless they provide their MFA code.

MFA Delete can only be enabled under these conditions

1. The **AWS CLI** must be used to turn on MFA
2. The bucket must have **versioning turned on**



```
aws s3api put-bucket-versioning \
--bucket bucketname \
--versioning-configuration Status=Enabled,MFADelete=Enabled \
--mfa "your-mfa-serial-number mfa-code" \
```

Only the bucket owner logged in as Root User can DELETE objects from bucket



AWS Developer Associate

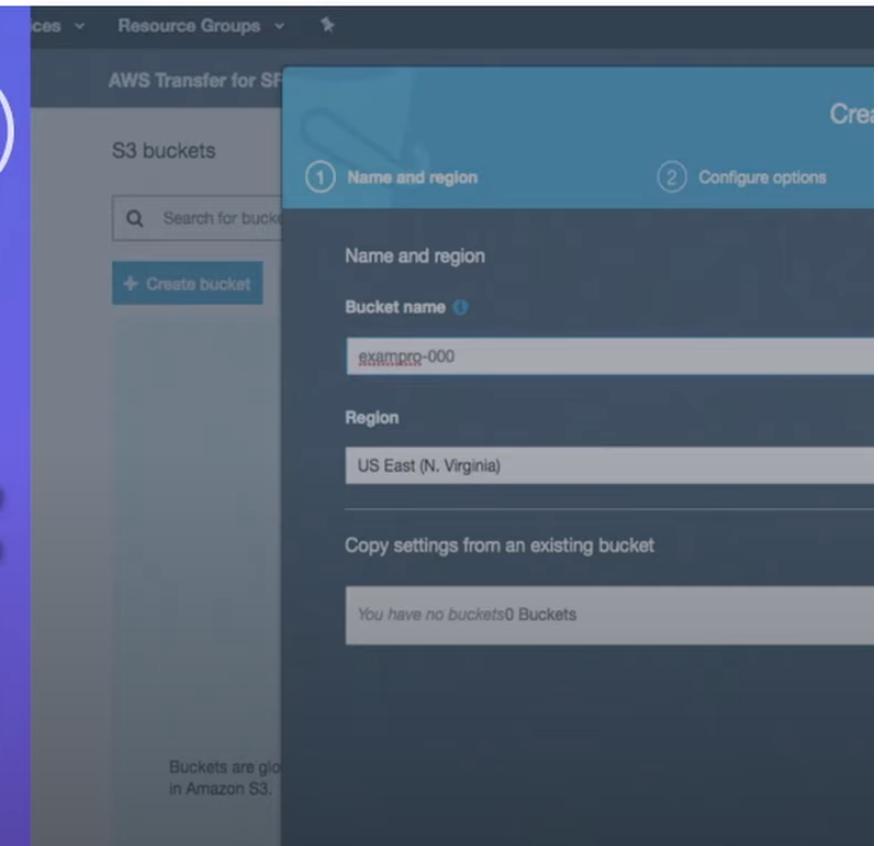
Simple Storage Service (S3)



Create And Delete Bucket



Follow Along





AWS Developer Associate

Simple Storage Service (S3)



Upload Files And Make Public



Follow Along

The screenshot shows the AWS S3 Bucket Properties page for a bucket named 'xampro-000'. The 'Block public access' tab is selected. A success message at the top states 'Public access settings updated successfully'. Below it, under 'Block all public access', the setting is set to 'Off'. There are four sub-options: 'Block public access to buckets and objects granted through new access control lists (ACLs)' (Off), 'Block public access to buckets and objects granted through any access control lists (ACLs)' (Off), 'Block public access to buckets and objects granted through new public bucket policies' (Off), and 'Block public and cross-account access to buckets and objects through any public bucket policies' (Off).



AWS Developer Associate

Simple Storage Service (S3)



Versioning



Follow Along

The screenshot shows the AWS S3 console with the bucket 'exampro-000' selected. The 'Versioning' tab is open, displaying three options: 'Enable versioning', 'Suspend versioning', and 'Disabled'. The 'Suspend versioning' option is selected, with a tooltip explaining that it suspends object version creation while preserving existing versions. Below the versioning section is a 'Default encryption' section, which is currently disabled. At the bottom right of the dialog are 'Cancel' and 'Save' buttons.



AWS Developer Associate

Simple Storage Service (S3)



Encryption



Follow Along

The screenshot shows the AWS S3 console with the 'exampro-000' bucket selected. The 'Properties' tab is active. In the 'Default encryption' section, the 'None' option is selected, indicated by a blue radio button. Other options include 'AES-256' and 'AWS-KMS'. A modal window titled 'Default encryption' is open, providing details about each option and showing the 'None' option as selected. Buttons for 'Cancel' and 'Save' are at the bottom right of the modal.

Amazon S3 > exampro-000

Overview Properties Permissions

Versioning

Keep multiple versions of an object in the same bucket.

Learn more

Enabled

Server access logging

Set up access log records that provide details about access requests.

Learn more

Disabled

Default encryption

This property does not affect existing objects in your bucket.

None

AES-256
Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)

AWS-KMS
Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)

Disabled

Cancel Save



AWS Developer Associate

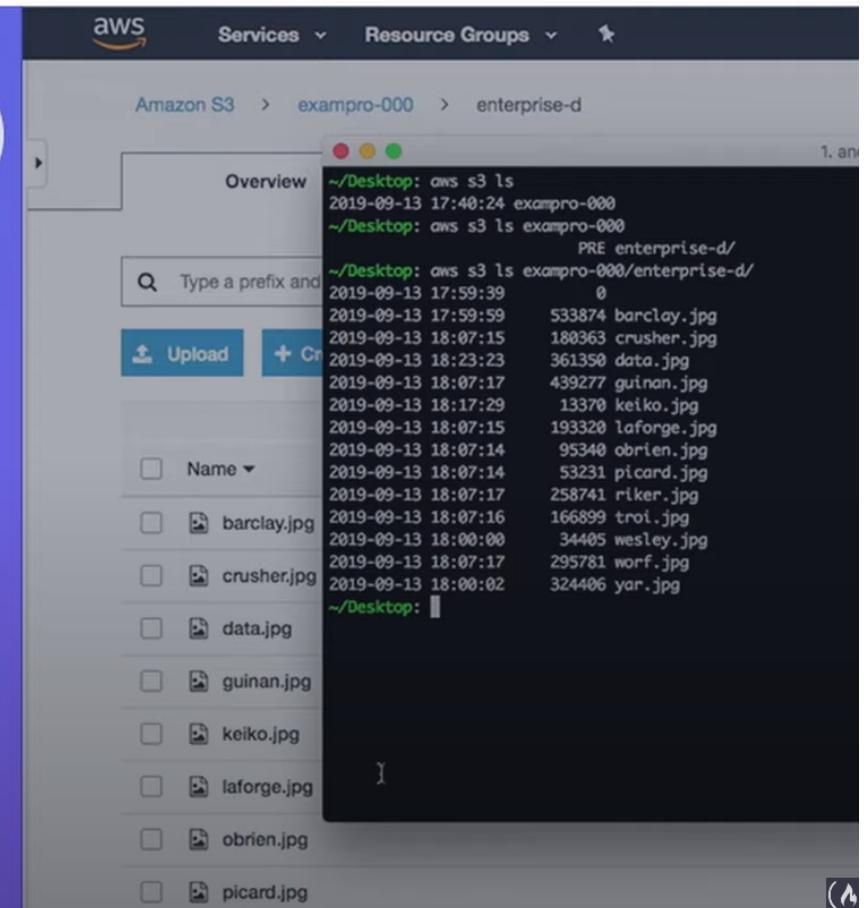
Simple Storage Service (S3)



S3 CLI



Follow Along



The screenshot shows the AWS S3 console interface. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and a star icon. The main path 'Amazon S3 > exampro-000 > enterprise-d' is displayed. On the left, there's an 'Overview' section with a search bar labeled 'Type a prefix and...' and two buttons: 'Upload' and '+ Create'. Below this is a table with columns for 'Name' and 'Last modified'. The table lists several files:

Name	Last modified
barclay.jpg	2019-09-13 18:07:15
crusher.jpg	2019-09-13 18:07:14
data.jpg	2019-09-13 18:07:17
guinan.jpg	2019-09-13 18:07:29
keiko.jpg	2019-09-13 18:07:15
laforge.jpg	2019-09-13 18:07:14
obrien.jpg	2019-09-13 18:07:14
picard.jpg	2019-09-13 18:07:17
troi.jpg	2019-09-13 18:07:16
wesley.jpg	2019-09-13 18:00:00
worf.jpg	2019-09-13 18:07:17
yar.jpg	2019-09-13 18:00:02

At the bottom right of the screenshot, there is a small '(A)' icon.



AWS Developer Associate

Simple Storage Service (S3)



Lifecycle Policies



Follow Along

Lifecycle rule

(1) Name and scope (2) Transitions (3) Expiration

Storage class transition

There are per-request fees when using lifecycle to transition data to any S3 storage class. [Learn more](#) or see [Amazon S3 pricing](#)

Current version Previous versions

For current versions of objects [+ Add transition](#)

! A minimum of 30 days is required before transitioning to Standard-IA storage class
Enter an integer value greater than or equal to 30.

Object creation Days after creation

Transition to Standard-IA after 7 X



AWS Developer Associate

Simple Storage Service (S3)



Cross-Region Replication



Follow Along

The screenshot shows the 'Replication rule' configuration page. At the top, there are four steps: 1 Set source, 2 Set destination, 3 Configure rule options, and 4 Review. Step 1 is active, showing 'Set source' with the option 'Entire bucket exapro-000' selected. Step 2 shows 'Set destination' with a dropdown menu open. Step 3 shows 'Configure rule options' with a checkbox for 'Replicate objects encrypted with AWS KMS'. A note at the bottom of step 1 explains the new replication schema: 'Your replication rule will be created using the new schema. Replication now has a new schema that supports replication based on prefixes, one or more object tags or a combination of the two. As part of the new schema, you can set overlapping rules with priorities. The new schema does not support delete marker replication, which would prevent any delete actions from replicating.' Buttons for 'Cancel', 'Next', and 'Get started' are at the bottom right.



AWS Developer Associate

Simple Storage Service (S3)



Bucket Policies



Follow Along

The screenshot shows the AWS S3 Bucket Policy editor interface. At the top, there's a navigation bar with 'Amazon S3' and 'exampro-000'. Below it is a tab bar with 'Overview', 'Properties', 'Permissions' (which is selected), and 'Manage'. Under the 'Permissions' tab, there are four buttons: 'Block public access', 'Access Control List', 'Bucket Policy' (which is highlighted in blue), and 'CORS configuration'. A sub-header reads 'Bucket policy editor ARN: arn:aws:s3:::exampro-000' with a note 'Type to add a new policy or edit an existing policy in the text area below.' Below this is a code editor containing the following JSON policy:

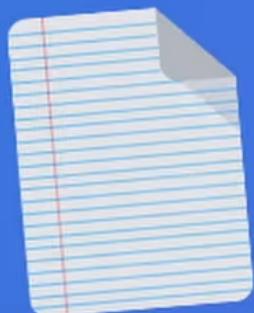
```
1 {  
2     "Id": "Policy1568422943891",  
3     "Version": "2012-10-17",  
4     "Statement": [  
5         {  
6             "Sid": "Stmt1568422942086",  
7             "Action": [  
8                 "s3:PutObject",  
9                 "s3:PutObjectAcl"  
10            ],  
11            "Effect": "Deny",  
12            "Resource": "arn:aws:s3:::exampro-000/*",  
13            "Principal": "*"  
14        }  
15    ]  
16}
```

At the bottom of the editor, there are links for 'Documentation' and 'Policy generator'.



AWS Developer Associate

Simple Storage Service (S3)



S3 Cheat Sheet



S3 CheatSheet

- **Simple Storage Service (S3)** Object-based storage. Store **unlimited** amount of data without worry of underlying storage infrastructure
 - S3 replicates data across at least 3 AZs to ensure 99.99% Availability and 11' 9s of durability
 - Objects contain your data (they're like files)
 - Objects can be size anywhere from **0 Bytes** up to 5 Terabytes
 - Buckets contain objects. Buckets can also contain folders which can in turn contain objects.
 - Bucket names are unique across all AWS accounts. Like a domain name.
 - When you upload a file to S3 successfully you'll receive a HTTP 200 code
- Lifecycle Management** Objects can be moved between storage classes or objects can be deleted automatically based on a schedule
- **Versioning** Objects are given a Version ID. When new objects are uploaded the old objects are kept. You can access any object version. When you delete an object the previous object is restored. Once Versioning is turned on it cannot be turned off, only suspended.
 - **MFA Delete** enforces DELETE operations to require MFA token in order to delete an object. Must have versioning turned on to use. Can only turn on MFA Delete from the AWS CLI. Root Account is only allowed to delete objects
 - All new buckets are **private by default**
 - Logging can be turned on for a bucket to log operations performed on objects
 - **Access control** is configured using **Bucket Policies** and **Access Control Lists (ACL)**
 - **Bucket Policies** are JSON documents which let you write complex control access
 - **ACLs** are the legacy method (not deprecated) where you grant access to objects and buckets with simple actions



S3 CheatSheet

- **Security in Transit** Uploading files is done over SSL
- **SSE** stands for Server Side Encryption. S3 has **3 options** for SSE.
- **SSE-AES** S3 handles the key, uses AES-256 algorithm
- **SSE-KMS** Envelope encryption via AWS KMS and you manage the keys
- **SSE-C** Customer provided key (you manage the keys)
- **Client-Side Encryption** You must encrypt your own files before uploading them to S3
- **Cross Region Replication (CRR)** allows you to replicate files across regions for greater durability. You must have versioning turned on in the source and destination bucket. You can have CRR replicate to bucket in another AWS Account
- **Transfer Acceleration** provide faster and secure uploads from anywhere in the world. data is uploaded via distinct url to an Edge Location. Data is then transported to your S3 bucket via AWS backbone network.
- **PresignedUrls** is a url generated via the AWS CLI and SDK. It provides temporary access to write or download object data. Presigned Urls are commonly used to access private objects.



S3 CheatSheet

- S3 has **6 different** Storage Classes:
 - **Standard** Fast! 99.99% Availability, 11 9's Durability. Replicated across at least three AZs
 - **Intelligent Tiering** Uses ML to analyze your object usage and determine the appropriate storage class. Data is moved to the most cost-effective access tier, without any performance impact or added overhead.
 - **Standard Infrequently Accessed (IA)** Still Fast! Cheaper if you access files less than once a month. Additional retrieval fee is applied. 50% less than Standard (reduced availability)
 - **One Zone IA** Still Fast! Objects only exist in one AZ. Availability (is 99.5%). but cheaper than Standard IA by 20% less (Reduce durability) Data could get destroyed. A retrieval fee is applied.
 - **Glacier** For long-term cold storage. Retrieval of data can take minutes to hours but the off is very cheap storage
 - **Glacier Deep Archive** The lowest cost storage class. Data retrieval time is 12 hours.