Cross-Region Replication in Amazon S3 with KMS Encryption

Introduction to S3 Cross-Region Replication

What is Amazon S3?

Amazon Simple Storage Service (S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. It is used to store and retrieve any amount of data at any time.

What is Cross-Region Replication (CRR)?

Cross-Region Replication is an S3 feature that automatically replicates objects from one S3 bucket (source) to another bucket (destination) in a different AWS region.

- Replication is **asynchronous**
- Used to reduce **latency**, increase **availability**, and support **disaster recovery**
- Requires **versioning enabled** on both source and destination buckets

Page 2: Use Cases of S3 Cross-Region Replication

1. Disaster Recovery

In case of regional service disruption, CRR ensures that the data is available from another AWS region.

2. Compliance Requirements

Certain regulations require data copies to be stored in geographically separate locations.

3. Latency Reduction

Replicating data to regions closer to users improves access performance.

4. Data Sovereignty

CRR can help store a copy of the data in a country that mandates data localization.

5. Backup Strategy

Secondary copy of your data acts as a version-controlled backup.

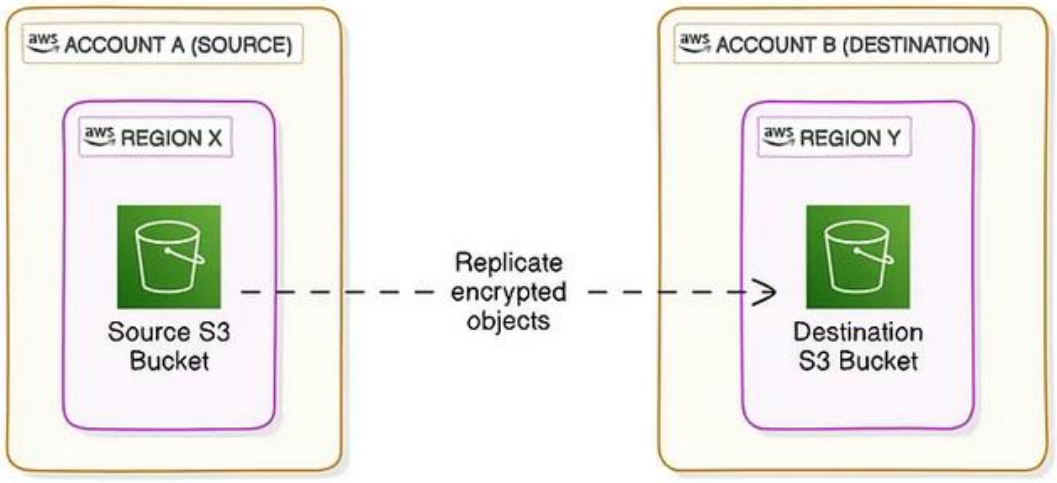
6. Application Data Sync

Applications deployed in multiple AWS Regions can access the same data quickly if replicated.

Prerequisites & Architecture

Prerequisites

Requirement	Description
AWS Account	Active AWS account with IAM access
S3 Buckets	One source and one destination bucket
Versioning	Must be enabled on both buckets
Permissions	IAM roles or bucket policies for replication



Cross Region, Cross Account S3 Replication

Cross-Account, Cross-Region S3 Replication Architecture with KMS Encryption

This architectural setup demonstrates a highly secure and reliable method for replicating data between two Amazon S3 buckets that are located in **different AWS accounts** and **different regions**, while using **AWS Key Management Service (KMS)** for encryption. It is designed to meet both **compliance** and **business continuity** requirements by ensuring that data is not only **replicated across geographies** but also **secured at every stage** of the process.

Source and Destination AWS Accounts

The architecture is based on **two separate AWS accounts**:

- **Account A (Source)** – This account contains the **source S3 bucket**, which is located in **Region X**.
- **Account B (Destination)** – This account houses the **destination S3 bucket**, placed in **Region Y**.

By leveraging **cross-account permissions**, this configuration ensures that data can be securely transferred from one account to another while also achieving **cross-region replication** for enhanced **data durability**, **availability**, and **disaster recovery readiness**.

Source Bucket Encryption

All objects that are uploaded to the **source S3 bucket** are automatically encrypted using a **KMS key that belongs to Account A** (referred to as the **KMS Key for Source**). This ensures that data is **secure at rest** in the source location.

To enable replication, a **dedicated IAM Role** (Replication Role) in Account A is granted **decrypt permissions** using this KMS key. This is crucial because the objects must be decrypted and re-encrypted before being sent to the destination bucket in another account and region.

IAM Role for Replication

A specialized **IAM Role** in the **Source Account (Account A)** is created specifically to handle the replication process. This role has a **trust relationship** with the **S3 service**, enabling Amazon S3 to assume the role during the replication process.

The role has the following permissions:

- **Read access** to the source S3 bucket.
- **Decrypt permissions** using the **Source KMS key**.
- **Write access** to the destination S3 bucket in **Account B**.
- **Encrypt permissions** using the **Destination KMS key** (belonging to Account B).

Additionally, the **destination account (Account B)** must grant the necessary permissions to this role to allow **cross-account replication**, including access to the KMS key used for encrypting the replicated data.

Destination Bucket Encryption

Once the data reaches the **destination S3 bucket**, it is **re-encrypted** using a **KMS key specific to Account B** (the **KMS Key for Destination**). This ensures that data remains encrypted at rest, following the security policies and compliance requirements of the destination environment.

This re-encryption step is crucial because:

- It ensures **ownership and control** of the replicated data by the destination account.
- It maintains **encryption consistency** according to the best practices for cross-region and cross-account replication.

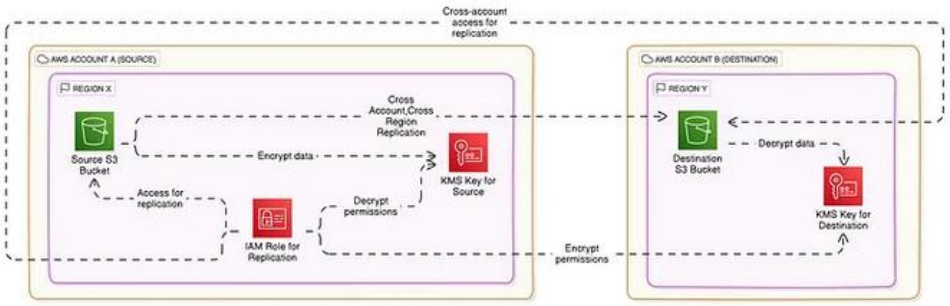
End-to-End Secure Replication

The overall process provides **end-to-end security** by ensuring that:

- Data is **encrypted at rest** in both the source and destination locations using **account-specific KMS keys**.
- The **IAM Role** involved in replication has **minimum necessary privileges**, adhering to the **principle of least privilege**.
- Replication happens **automatically and securely** behind the scenes once it is configured, requiring no manual intervention.

This type of setup is ideal for organizations that need to:

- Replicate sensitive or critical data across accounts for **segregation of duties**.
- Maintain **geo-redundancy**.
- Ensure **strong encryption** and **auditable control** using AWS KMS.

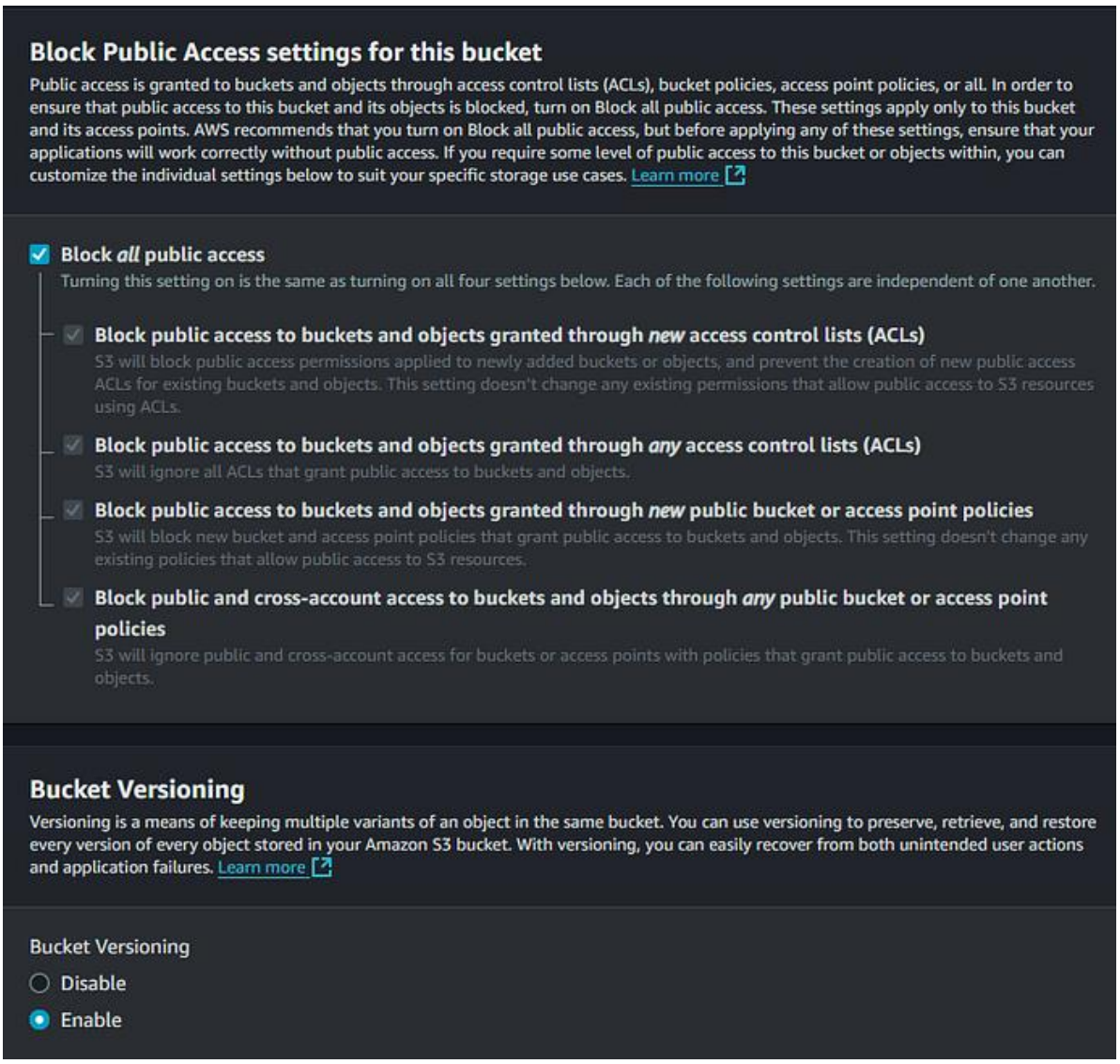


Overview

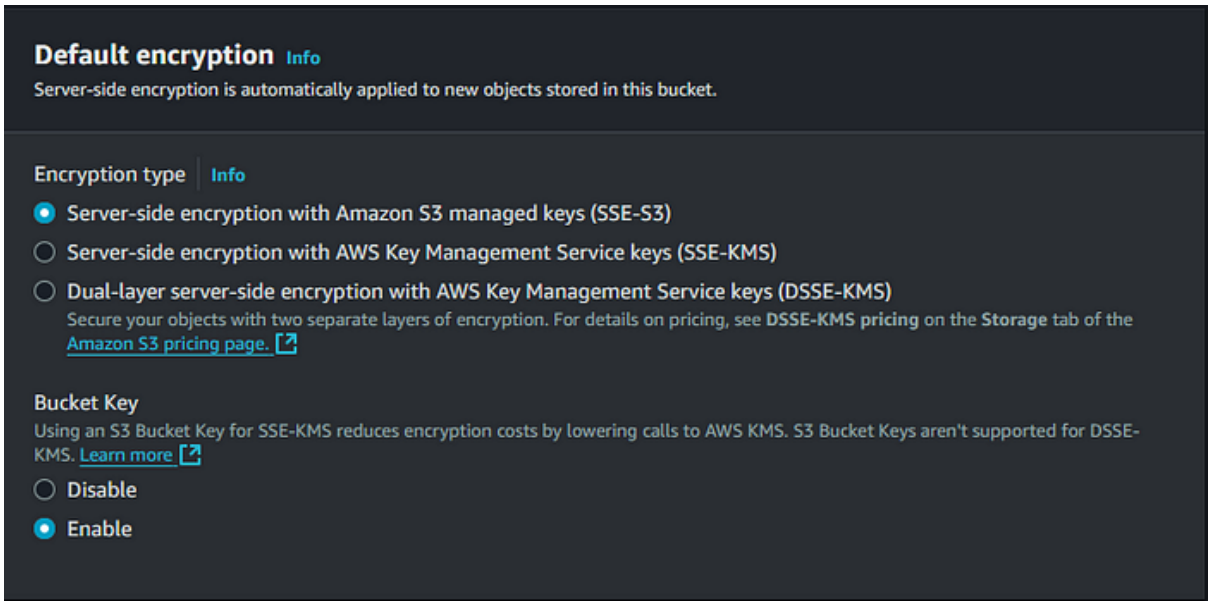
1.1 Create the Source Bucket in Account A

Follow the steps below to create the source bucket in AWS Account A:

1. Log in to the AWS Management Console using the credentials for Account A.
2. In the AWS Console's search bar at the top, type "S3" and select Amazon S3 from the results to navigate to the S3 dashboard.
3. Click the "Create bucket" button to begin creating a new S3 bucket.
4. On the Create bucket page, perform the following actions:
    - Bucket Name: Enter a globally unique name for the source bucket. Example: my-source-bucket-account-a.
    - Region: Choose the AWS Region where you want this source bucket to reside.
      For instance, you might choose Asia Pacific (Mumbai) ap-south-1 if your primary operations are in India.
5. Under the section titled "Bucket settings for Block Public Access", leave all four settings enabled (checked). This ensures that your bucket remains private and secure, blocking all public access unless explicitly allowed later.
6. Scroll down to the section labeled "Bucket Versioning":
    - Click on "Enable" to turn on versioning.
    - Why? Versioning is a mandatory requirement for enabling replication in Amazon S3. It allows the bucket to keep multiple versions of objects, which is crucial for tracking changes and replicating data reliably.
7. You can leave other settings (like tags, default encryption, advanced options) unchanged for now unless you have specific requirements (e.g., applying encryption settings at this stage).
8. Finally, scroll to the bottom and click the "Create bucket" button to provision your source bucket.



Leave the **Default encryption** setting as it is for now, as we'll return to set the encryption key once the KMS key is created in the next step.



1.2. Create Destination Bucket in Account B

To store replicated data securely in another AWS account and region, perform the following steps:

1. Log in to **Account B** using your AWS credentials.
2. Navigate to the **S3 Console** and click **Create bucket**.
3. Enter the **bucket name** (e.g., my-destination-bucket-account-b). Ensure the bucket name is globally unique.
4. Choose a different region from the source bucket (for cross-region replication). For example, if the source is in **Asia Pacific (Mumbai)**, you can choose **US East (N. Virginia)** for the destination.
5. Under **Bucket Versioning**, **enable versioning**, as it is mandatory for replication.
6. Leave the **Default encryption** setting as is for now. Encryption will be managed via KMS configuration later.
7. Optionally configure tags and permissions as per your security policy.
8. Click **Create Bucket**.

At this point, you have successfully created both source and destination buckets in different AWS accounts and regions.

Step 2: Create Customer Managed Keys (CMKs) for Encryption

To support server-side encryption using AWS KMS, especially in a **cross-account replication** scenario, you need to create **CMKs (Customer Managed Keys)** in both accounts.

2.1. Create a KMS Key in Source Region (Account A)

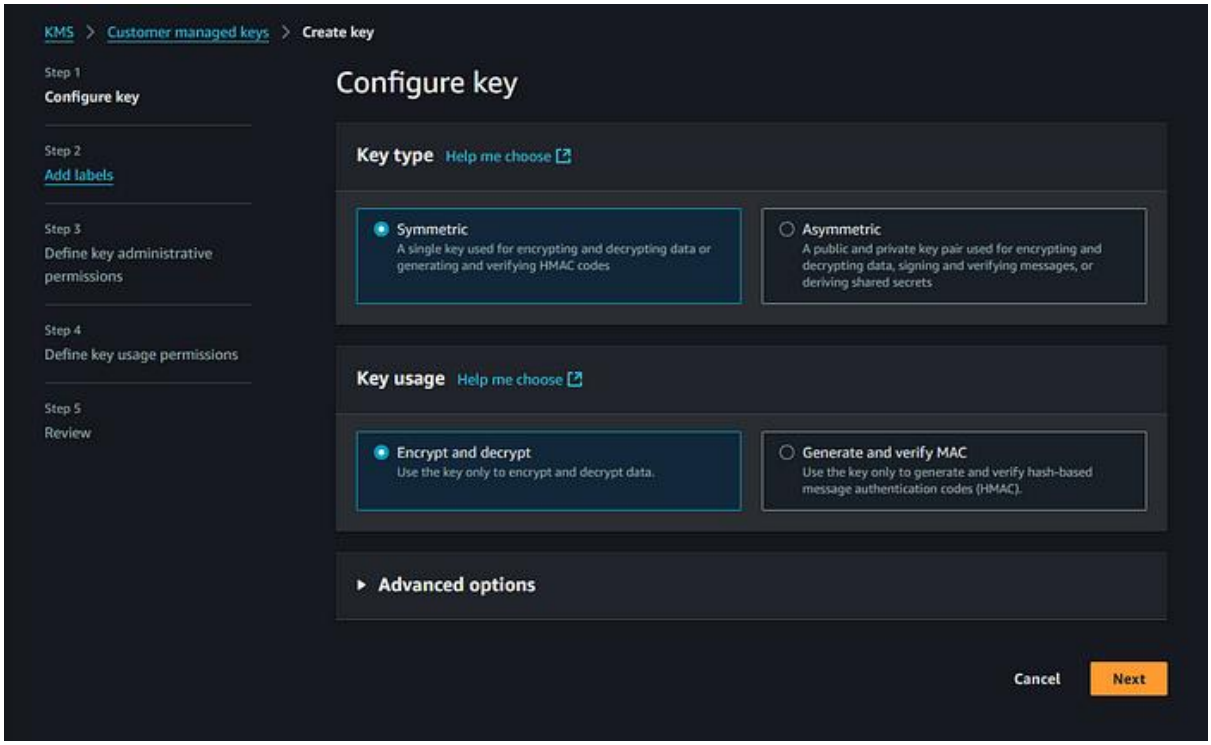
Follow these steps to create a CMK in **Account A**, where your source bucket resides:

1. Navigate to **AWS Key Management Service (KMS)** in the **same region** as your source bucket (e.g., ap-south-1 for Mumbai).
2. Click on **Create key**.
3. **Key Type:**
    - Select **Symmetric** key (default).
    - Reason: S3 uses symmetric encryption with KMS; asymmetric keys are not supported for this purpose.
4. **Key Usage:** Choose **Encrypt and decrypt**.

5. **Key Alias:**
   Provide a friendly name such as s3-crr-key-account-a.
6. **Tags (Optional):**
   Add tags such as Environment: Production or Department: IT.
7. **Key Administrative Permissions:**
   Choose the IAM users or roles in Account A that should have administrative access to the key (e.g., your root user or a dedicated S3 replication IAM role).
8. **Define Key Usage Permissions:**
   - Add IAM users/roles that will use this key to encrypt/decrypt S3 objects.
   - Ensure the IAM role used for replication has **kms:Encrypt, kms:Decrypt, kms:GenerateDataKey, kms:ReEncrypt*** permissions.
9. **Review & Create:**
   Confirm all settings and click **Finish** to create the key.

Would you like me to continue with:

- **2.2. Creating a KMS Key in Destination Region (Account B)**
- **3. IAM Role Setup and Bucket Policy**
- **4. Configuring Replication Rules**

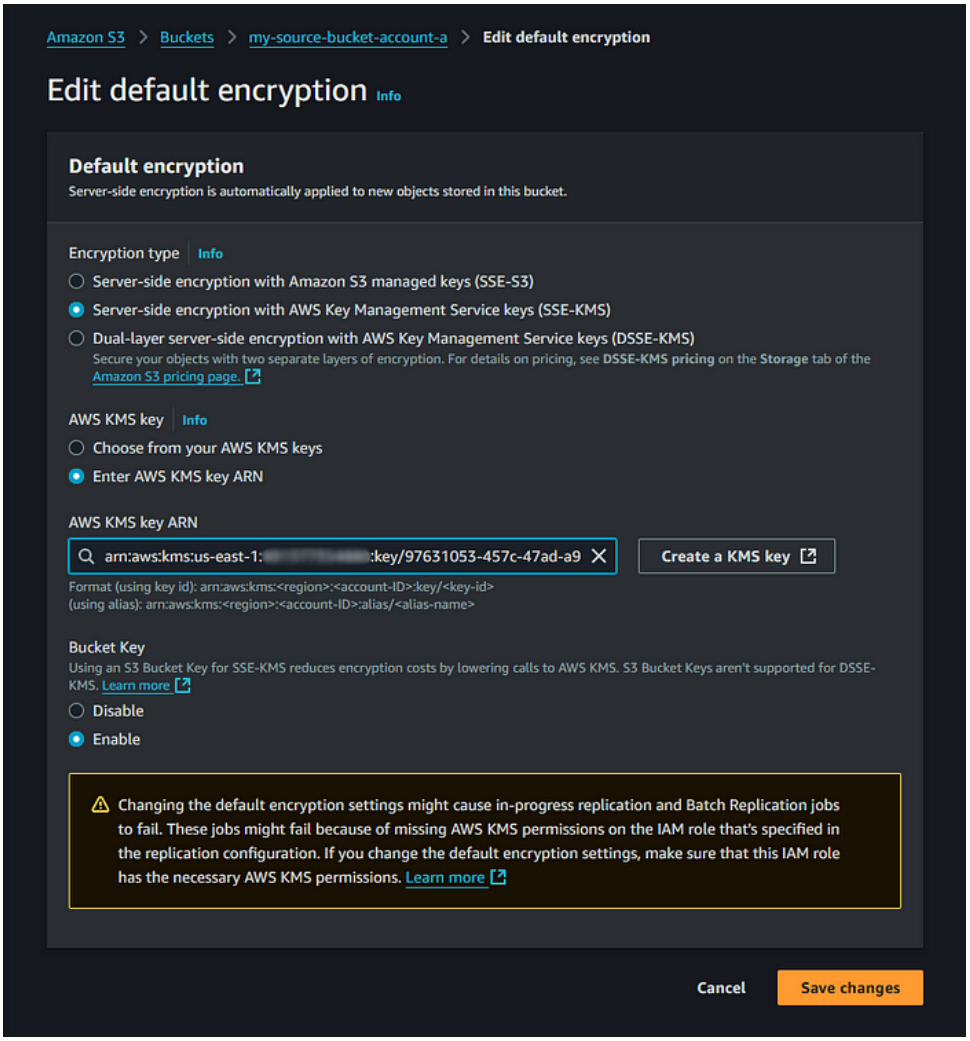


- Enter a name and description (e.g., SourceBucketKeyAccountA).

- Set **Key Administrators**: Assign IAM users or roles that need permissions to administer this key.
- **Add Key Usage Permissions**: Specify the IAM roles or users who will use this key for encryption/decryption (for cross-account replication, you'll adjust this in a later step).
- Complete the key setup and click **Create key**.

2.2. Set Default Encryption on Source Bucket:

- Navigate to the **Amazon S3 Console** and open your **Source Bucket**.
- Go to the **Properties** tab, and scroll down to **Default encryption**.
- Select **AWS Key Management Service key (SSE-KMS)**, then enter or select the **KMS key ARN** created in the previous step.
- Save the changes to set KMS as the default encryption method for the bucket.



2.3. Create a KMS Key in the Destination Region (Account B)

- Follow the same procedure as outlined earlier to create a new AWS Key Management Service (KMS) key in Account B. You may name it something like DestinationBucketKeyAccountB for reference.

2.4. Configure Default Encryption on the Destination Bucket

- Again, repeat the steps used previously to enable default encryption for the destination bucket. Select AWS Key Management Service key (SSE-KMS) as the encryption method and provide the ARN of the KMS key you just created in the destination account.

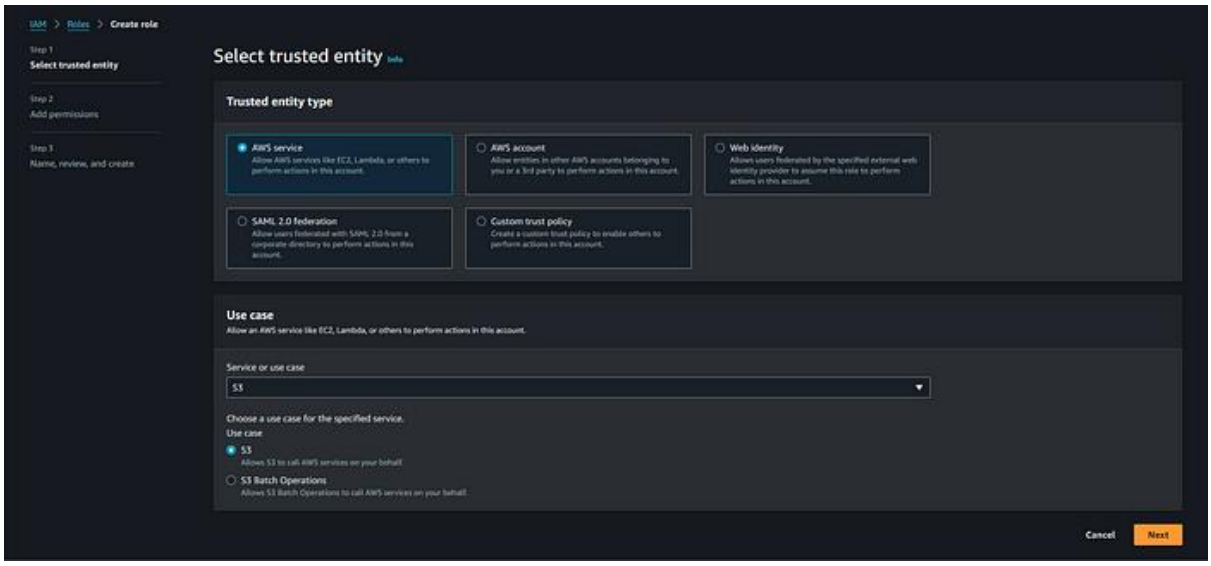
Step 3: Create an IAM Role with a Trust Policy in Source Account (Account A)

3.1. Open IAM Console in Account A

- Navigate to the IAM console, go to the Roles section, and click Create role to begin the setup.

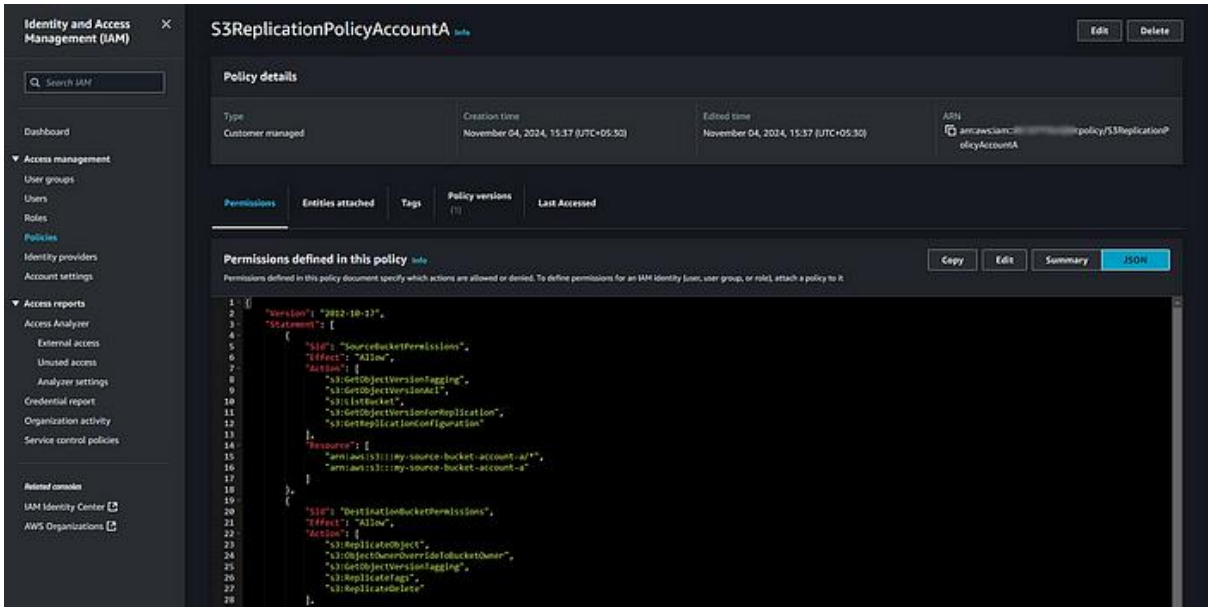
3.2. Define the Trusted Entity

- Choose AWS service as the trusted entity type and select S3 as the service. This allows Amazon S3 to assume the role for performing replication tasks.
- Under Use case, select S3 (bucket replication) and proceed by clicking Next.



3.3. Add Permissions to the Role:

- Attach a custom policy with the following permissions to allow access to both the source and destination buckets:
  - **Source bucket permissions** for replication configuration access.
  - **Destination bucket permissions** to replicate and store objects.
  - KMS **Encrypt/Decrypt permissions** for both the source and destination bucket KMS keys.



Define a policy similar to this (replace placeholders with actual bucket and key ARNs):

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "SourceBucketPermissions",
   "Effect": "Allow",
   "Action": [
    "s3:GetObjectVersionTagging",
    "s3:GetObjectVersionAcl",
    "s3:ListBucket",
    "s3:GetObjectVersionForReplication",
    "s3:GetReplicationConfiguration"
   ],
   "Resource": [
    "arn:aws:s3:::<SourceBucketName>/*",
    "arn:aws:s3:::<SourceBucketName>"
   ]
  },
  {
```

```
    "Sid": "DestinationBucketPermissions",
    "Effect": "Allow",
    "Action": [
     "s3:ReplicateObject",
     "s3:ObjectOwnerOverrideToBucketOwner",
     "s3:GetObjectVersionTagging",
     "s3:ReplicateTags",
     "s3:ReplicateDelete"
    ],
    "Resource": [
     "arn:aws:s3:::<DestinationBucketName>/*"
    ]
   },
   {
    "Sid": "SourceBucketKMSKey",
    "Action": [
     "kms:Decrypt",
     "kms:GenerateDataKey"
    ],
    "Effect": "Allow",
    "Resource": "<SourceBucketKMSKeyARN>"
   },
   {
    "Sid": "DestinationBucketKMSKey",
    "Action": [
     "kms:Encrypt",
     "kms:GenerateDataKey"
    ],
    "Effect": "Allow",
    "Resource": "<DestinationBucketKMSKeyARN>"
   }
  ]
}
```

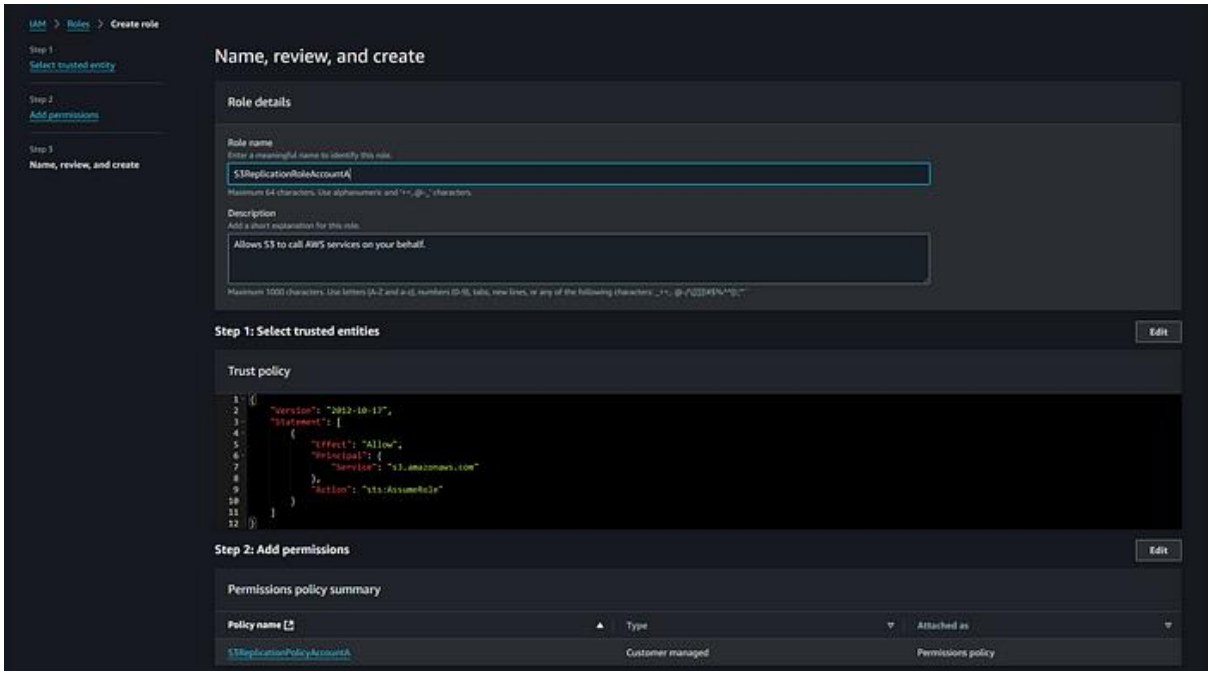
3.4. Set the Trust Policy:

- Click **Trust relationships** and **Edit trust relationship**.
- Ensure the trust relationship allows S3 to assume this role:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Principal": {
```

```
      "Service": "s3.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }
 ]
}
```

Zoom image will be displayed



3.5. Name and Create the Role:

- Name the role (e.g., S3ReplicationRoleAccountA), and click **Create role** to complete the setup.

Step 4: Edit Key Policy of Both Keys

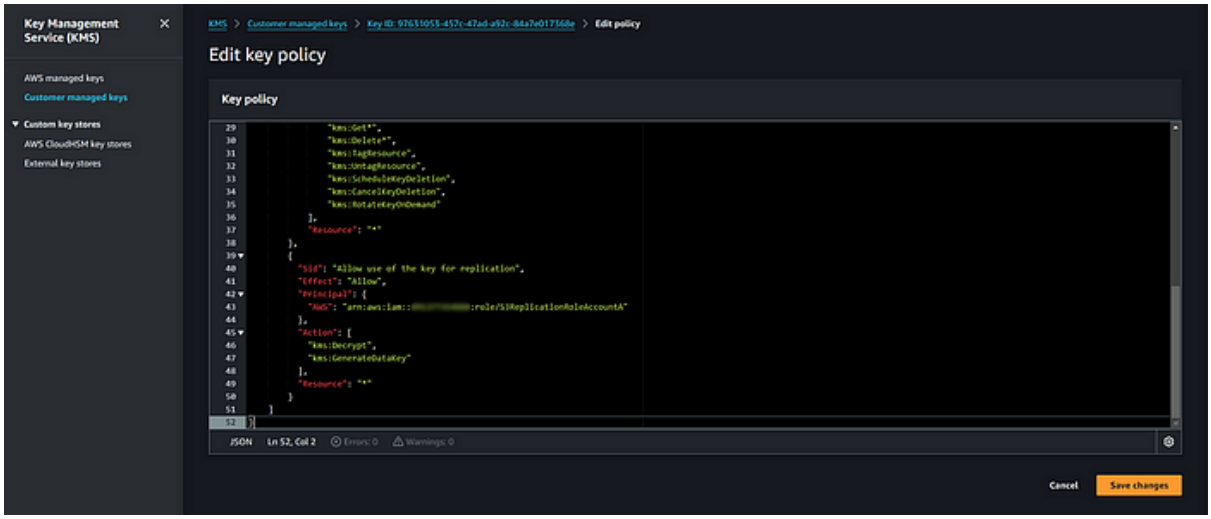
To allow cross-account replication, the KMS keys in both the source and destination accounts need key policies that grant the necessary permissions to the IAM role from the source account (Account A). These permissions enable the replication process to encrypt and decrypt objects using the KMS keys associated with each bucket.

4.1. Edit the Key Policy in the Source Account (Account A):

- Navigate to **AWS Key Management Service (KMS)** in Account A and select the KMS key you created earlier (e.g., SourceBucketKeyAccountA).
- In the key's **Key policy** section, choose **Edit**.
- Add a new policy statement that allows the IAM role created in Step 3 to perform kms:Decrypt and kms:GenerateDataKey actions on this key. Replace placeholders with actual role and key ARNs.

```
{
 "Version": "2012-10-17",
 "Id": "key-consolepolicy-3",
 "Statement": [
  {
   "Sid": "Allow use of the key for replication",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::AccountA:role/S3ReplicationRoleAccountA"
   },
   "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
   ],
   "Resource": "*"
  }
 ]
}
```

Zoom image will be displayed



4.2. Edit the Key Policy in the Destination Account (Account B):

- Log in to Account B, navigate to **AWS KMS**, and select the KMS key for the destination bucket (e.g., DestinationBucketKeyAccountB).
- Edit the key policy by adding a new statement similar to the one above, but for kms:Encrypt and kms:GenerateDataKey actions. Replace placeholders with actual ARNs for Account A's IAM role.

```
{
 "Version": "2012-10-17",
 "Id": "key-consolepolicy-4",
```

```
 "Statement": [
  {
   "Sid": "Allow use of the key for replication from Account A",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::AccountA:role/S3ReplicationRoleAccountA"
   },
   "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey"
   ],
   "Resource": "*"
  }
 ]
}
```

Zoom image will be displayed

4.3. Save the Policy:

- After editing the policy in each account, click **Save changes**.

Step 5: Edit the Bucket Policy of the Destination Bucket

5.1. Navigate to the S3 Console in Account B:

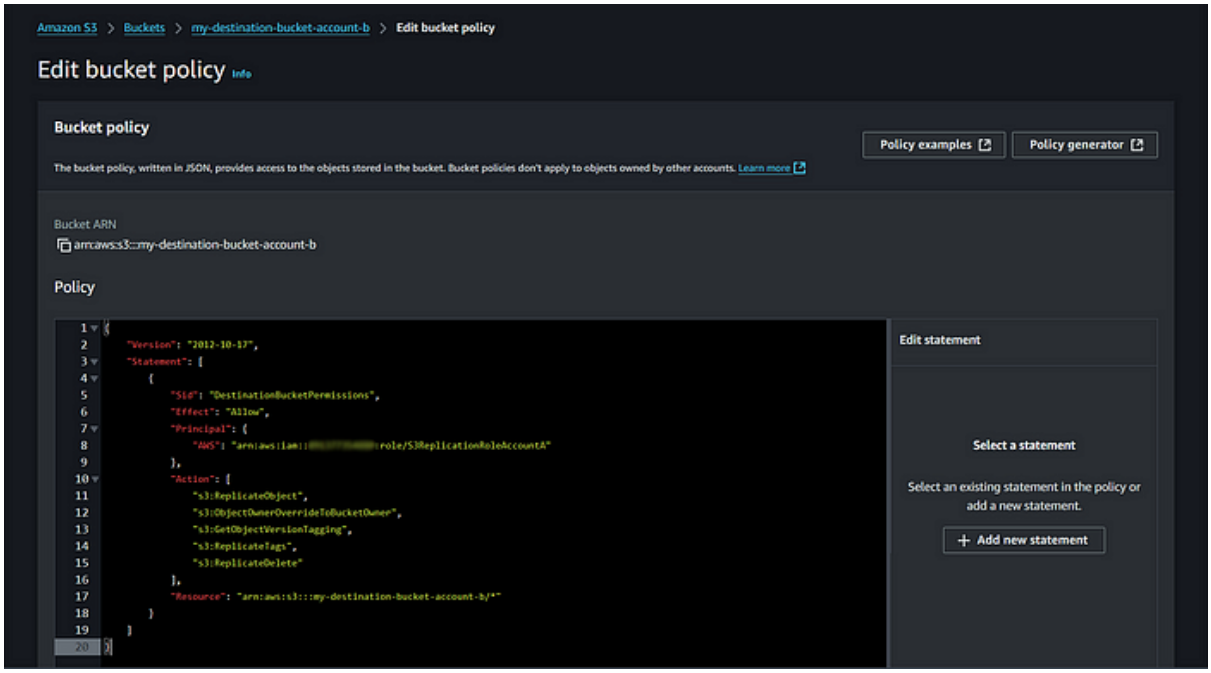
- In Account B, go to the **S3 console** and select the destination bucket.

5.2. Edit Bucket Policy:

- Go to the **Permissions** tab, then **Bucket policy**.
- Add the following policy to allow the IAM role from Account A to replicate objects into this bucket. Update the Principal section with the ARN of the IAM role created in Account A and replace the bucket name.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "DestinationBucketPermissions",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::AccountA:role/S3ReplicationRoleAccountA"
   },
   "Action": [
    "s3:ReplicateObject",
    "s3:ObjectOwnerOverrideToBucketOwner",
    "s3:GetObjectVersionTagging",
    "s3:ReplicateTags",
    "s3:ReplicateDelete"
   ],
   "Resource": "arn:aws:s3:::<DestinationBucketName>/*"
  }
 ]
}
```

Zoom image will be displayed



5.3. Save the Policy:

- After entering the policy, click **Save changes**.

Step 6: Set Up the S3 Replication Rule

6.1. Navigate to the Source Bucket:

- In Account A, go to **Amazon S3** and select the source bucket.

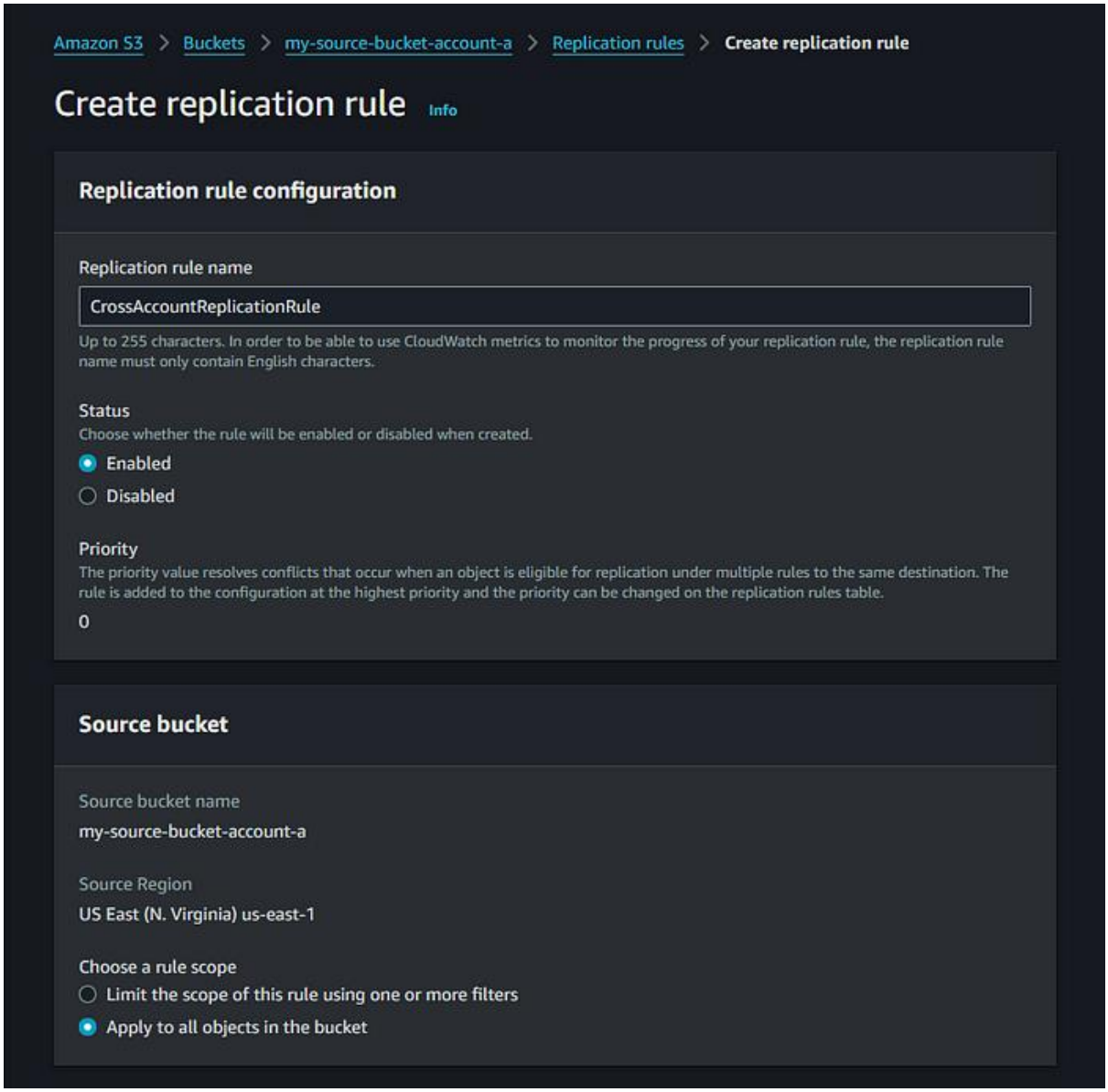
6.2. Configure Replication Rule:

- Go to the **Management** tab, scroll down to **Replication rules**, and select **Create replication rule**.
- **Rule name**: Enter a descriptive name for the rule (e.g., CrossAccountReplicationRule).

6.3. Source Configuration:

- Under **Source bucket**, specify whether to apply replication to the entire bucket or only to objects with a specific prefix.

Zoom image will be displayed



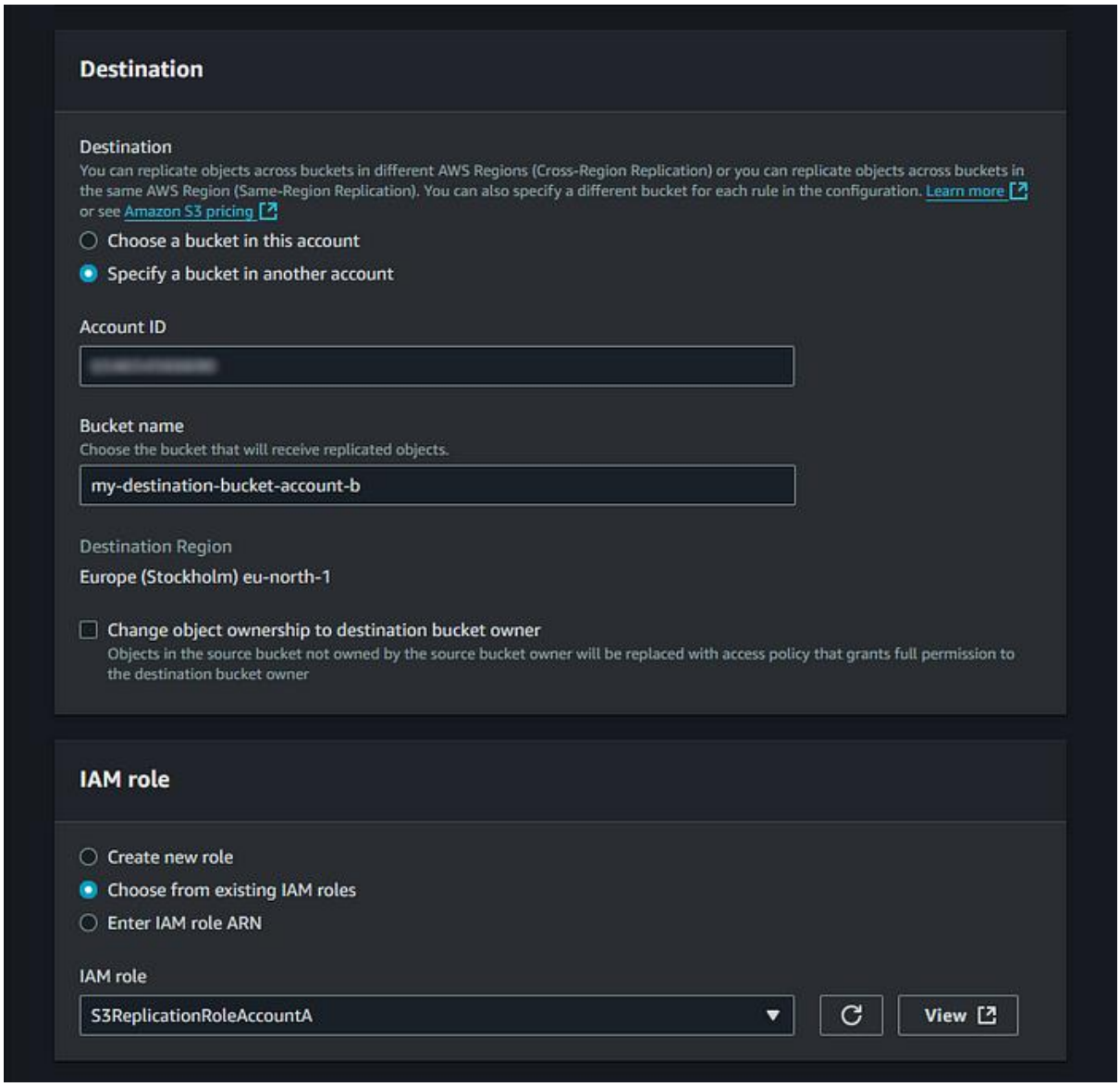
6.4. Destination Configuration:

- In the **Destination** section, select **Choose a bucket in another account**, and specify the ARN of the destination bucket in Account B.

6.5. Choose or Create an IAM Role:

- Under **IAM role**, select **Choose from existing IAM roles** and select the role you created earlier in Account A for replication (e.g., S3ReplicationRoleAccountA).

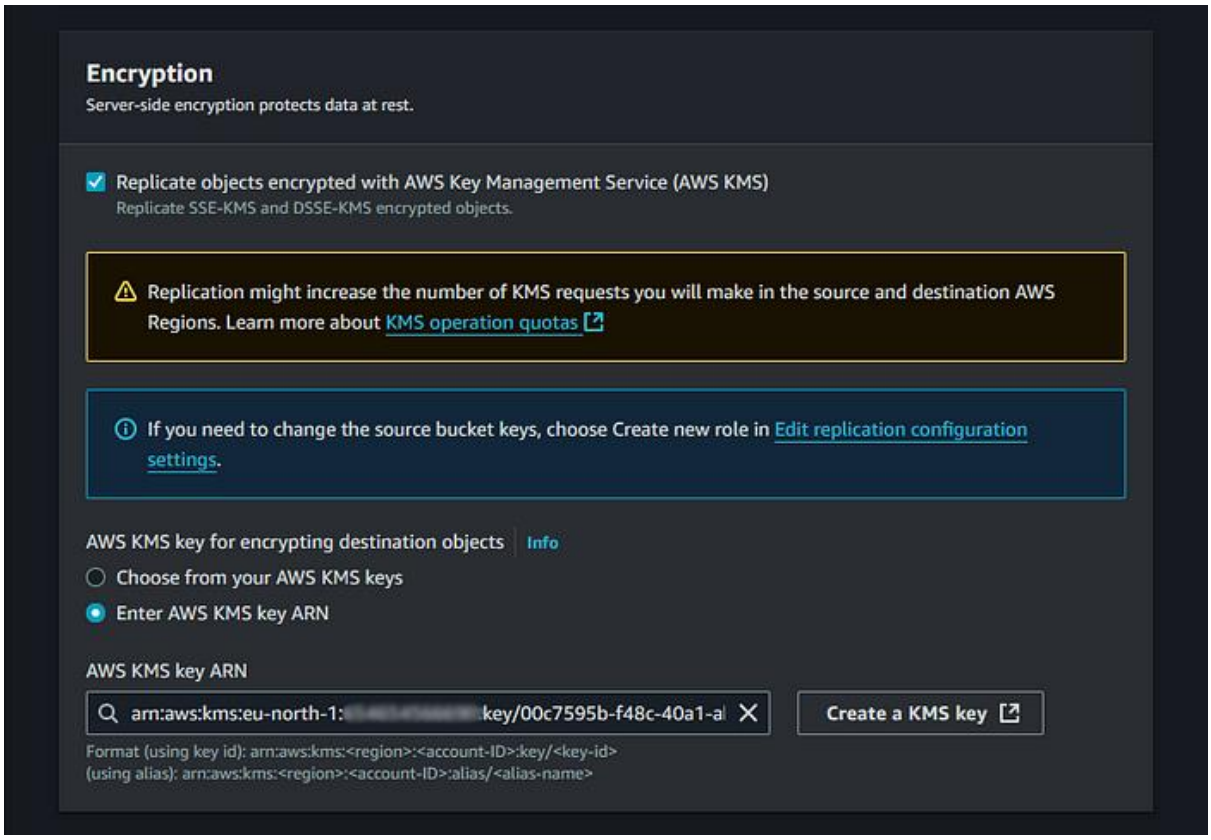
Zoom image will be displayed



6.6. Encryption Settings:

- Under **Encryption**, choose **Replicate objects encrypted with AWS KMS** and enter the destination bucket KMS key ARN of Account B.

Zoom image will be displayed



6.7. Change destination storage class:

- Under **Destination storage class**, select the desired storage class for replicated objects (e.g., **Standard** or **Intelligent-Tiering**).

6.8. Additional Replication Options (Optional):

- Configure additional options, such as **Replicate delete markers** if you want deletion replication across accounts.

6.9. Review and Create Rule:

- Review the replication rule settings, then click **Save**.

Step 7: Verify Replication

1. **Upload a test file** into **Bucket A**.
2. After a few minutes, check **Bucket B** for the replicated object.
3. Verify that the object is encrypted using the KMS key from **Account B**.
4. Check CloudWatch logs or enable S3 replication metrics to monitor replication health.

roubleshooting Tips for Common Errors

- **Permission Issues**: Misconfigured IAM roles, trust policies, or bucket policies can cause replication failures. Ensure that roles from both accounts have the necessary permissions.
- **KMS Key Policies**: Make sure that the KMS keys' policies allow cross-account use for both encryption and decryption.
- **S3 Replication Rule Configuration**: Ensure the replication rule specifies the correct destination bucket, IAM role, and replication scope. If the replication status shows "Failed", revisit each configuration to ensure that permissions, roles, and policies align with the setup requirements.

Conclusion

Setting up Cross-Region, Cross-Account Replication with KMS encryption is a powerful approach for organizations needing robust data security and disaster recovery solutions. By following the steps above, you enable secure, compliant, and resilient data replication across accounts and Regions in AWS. The replication setup ensures data redundancy, optimized latency, and compliance support, with KMS encryption adding an essential layer of security. This approach demonstrates AWS's capacity to handle complex, secure data transfers across its global infrastructure.