

# TASK – 6 Create S3 bucket and apply cross account replication

## Step 1 — In Account A (Source): Create Source Bucket

1. Log in to AWS Account A.
2. Go to S3 console.
3. Click Create bucket.
4. Enter a unique bucket name
5. Choose a Region.
6. Scroll down → Enable Versioning.
7. Leave other settings as default.
8. Click Create bucket.

The screenshot shows the 'Create bucket' page in the AWS S3 console. Under 'General configuration', the 'AWS Region' is set to 'US East (N. Virginia) us-east-1'. The 'Bucket type' dropdown has 'General purpose' selected. The 'Bucket name' field contains 'source-bucket-kalpesh'. Below the name, a note states: 'Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn More](#)'.

**Object Ownership** (Info)  
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

**ACLs disabled (recommended)**  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only [Bucket Policies](#).

**ACLs enabled**  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

The screenshot shows the 'Create bucket' page in the AWS S3 console. Under 'Bucket Versioning', the 'Bucket Versioning' dropdown has 'Enable' selected. A note explains: 'Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)'.

**Tags - optional (0)**  
You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

[Add new tag](#)

You can add up to 50 tags.

**Default encryption** (Info)  
Server-side encryption is automatically applied to new objects stored in this bucket.

**Encryption type** (Info)  
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

Server-side encryption with Amazon S3 managed keys (SSE-S3)  
 Server-side encryption with AWS Key Management Service keys (SSE-KMS)  
 Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

**Bucket Key**

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

## Step 2 — In Account B (Destination): Create Destination Bucket

1. Log in to AWS Account B.
2. Go to S3 console.
3. Click Create bucket.
4. Enter a unique bucket name.
5. Choose a Region.
6. Scroll down → Enable Versioning.
7. Click Create bucket.

The screenshot shows the 'Create bucket' wizard in the AWS S3 console. The 'General configuration' section is visible, featuring fields for 'Bucket name' (set to 'destination-bucket-kalpesh') and 'AWS Region' (set to 'Asia Pacific (Mumbai) ap-south-1'). Under 'Bucket type', the 'General purpose' option is selected, with a note explaining it's recommended for most use cases. The 'Directory' option is also available. Other sections like 'Object Ownership' and 'Bucket Versioning' are partially visible at the bottom.

The screenshot shows the 'Create bucket' wizard in the AWS S3 console, focusing on 'Bucket Versioning'. The 'Enable' option is selected. Other sections like 'Tags - optional (0)', 'Default encryption', and 'Encryption type' are visible at the bottom.

## Step 3 — In Account A: Create Replication Role

1. In Account A, go to IAM console.
2. Click Roles → Create role.
3. Select AWS Service → choose S3.
4. Click Next.
5. Attach permissions → choose AmazonS3FullAccess.
6. Click Next.
7. Click Create role.
8. Open the role and copy the Role ARN (in Account B).

The screenshot shows the 'Select trusted entity' step of the IAM role creation wizard. It has three tabs: 'Select trusted entity' (selected), 'Add permissions' (disabled), and 'Name, review, and create' (disabled). Under 'Trusted entity type', the 'AWS service' option is selected, showing a sub-description: 'Allow AWS services like EC2, Lambda, or others to perform actions in this account.' Other options include 'AWS account', 'Web Identity', 'SAML 2.0 federation', and 'Custom trust policy'. Below this is a 'Use case' section with a dropdown menu set to 'S3'. Under 'Service or use case', 'S3' is selected with the sub-description: 'Allows S3 to call AWS services on your behalf.' Other options are 'S3 Batch Operations' with its sub-description: 'Allows S3 Batch Operations to call AWS services on your behalf.' At the bottom, there are links for CloudShell, Feedback, and cookie preferences.

The screenshot shows the 'Add permissions' step of the IAM role creation wizard. It has three tabs: 'Select trusted entity' (disabled), 'Add permissions' (selected), and 'Name, review, and create' (disabled). Under 'Permissions policies', the 'AmazonS3FullAccess' policy is selected and highlighted in blue. Other policies listed include 'AmazonDMSRedshiftS3Role', 'AmazonS3ObjectLambdaExecutionRole', 'AmazonS3OutpostsFullAccess', 'AmazonS3OutpostsReadOnlyAccess', 'AmazonS3ReadOnlyAccess', 'AmazonS3TablesFullAccess', 'AmazonS3TablesLakeFormationService...', 'AmazonS3TablesReadOnlyAccess', and 'AWSBackupServiceRolePolicyForS3Bac...'. The search bar at the top is set to 's3'. At the bottom, there are links for CloudShell, Feedback, and cookie preferences.

## Step 4 — In Account B: Allow Account A to Write

1. In Account B, go to S3 → Destination bucket → Permissions.

2. Scroll to Bucket policy → Edit.

3. Use Policy generator wizard

Effect → Allow

Actions → select:

ReplicateObject

ReplicateDelete

ReplicateTags

ARN → destination bucket

4. Save policy.

The screenshot shows the AWS S3 Bucket Policy editor. On the left, there's a sidebar with navigation links like 'Amazon S3', 'General purpose buckets', 'Storage Lens', and 'CloudShell'. The main area displays a JSON policy document. The policy allows replication from a specific IAM role to the destination bucket. The right side of the screen shows sections for 'Edit statement', 'Add actions', and 'Included' services like S3, AI Operations, AMP, and API Gateway.

```
1 Version: "2012-10-17",
2 Statement: [
3     {
4         Sid: "AllowReplicationFromSource",
5         Effect: "Allow",
6         Principal: [
7             "ARN: arn:aws:iam::722313191117:role/Replication-policy"
8         ],
9         Action: [
10            "s3:ReplicateObject",
11            "s3:ReplicateDelete",
12            "s3:ReplicateTags"
13        ],
14        Resource: "arn:aws:s3:::destination-bucket-kalpesh/*"
15    }
16 ]
17 }
18 }
```

## Step 5 – In Account A: Create Replication Rule

1. In Account A, go to S3 console → open Source bucket.
2. Go to Management tab → Replication rules → Create replication rule.
3. Enter rule name.
4. Choose Entire bucket.
5. Destination → choose Bucket in another account.
6. Enter:
  - Account B's Account ID.
  - Destination bucket name.
7. IAM Role → select the role that created.
8. Click Save.

The screenshot shows the AWS IAM Replication-policy page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' and 'Access management' sections. The main area displays a 'Replication-policy' card with a summary of its creation date (September 03, 2025) and maximum session duration (1 hour). It also shows an ARN copied message and an 'Edit' button. Below this is a 'Permissions' tab showing one managed policy named 'AmazonS3FullAccess'. The bottom section is titled 'Permissions boundary (not set)'.

The screenshot shows the AWS S3 source-bucket-kalpesh Replication rules page. It includes a 'Replication configuration settings' section for the source bucket and an 'IAM role' section for the replication rule. The main area shows a table of replication rules. One rule is listed: 'ReplicationRule1' with status 'Enabled', destination bucket 's3://destination-bucket-kalpesh', destination region 'Asia Pacific (Mumbai) ap-south-1', priority 0, scope 'Entire bucket', storage class 'Same as source', replica owner 'Same as source', replication time control 'Disabled', KMS-encrypted objects 'Disabled', and replica modification sync 'Disabled'.

## Step 6 — Test Replication

1. In Account A, upload a file into source bucket.
2. Wait a short time (replication is async).
3. In Account B, open destination bucket → check file is appeared or not. Done! Cross-account replication works.

The screenshot shows the AWS S3 console with the URL [us-east-1.console.aws.amazon.com/s3/](https://us-east-1.console.aws.amazon.com/s3/). The navigation bar includes tabs for IAM, EC2, and S3. The main content area is titled "Upload" and shows a table of files being uploaded. One file, "Kalpesh\_AWS.pdf", is listed with a size of 76.7 KB and type application/pdf. Below the table, there's a section for "Destination" settings, which points to the destination bucket "s3://source-bucket-kalpesh". The "Permissions" section indicates public access is granted. At the bottom, there are links for CloudShell, Feedback, and cookie preferences.

The screenshot shows the AWS S3 console with the URL [us-east-1.console.aws.amazon.com/s3/](https://us-east-1.console.aws.amazon.com/s3/). The navigation bar includes tabs for IAM, EC2, and S3. The main content area is titled "Upload: status" and displays a green success message: "Upload succeeded. For more information, see the Files and folders table." Below this, there's a summary table with two rows: "Succeeded" (1 file, 76.7 KB) and "Failed" (0 files, 0 B). The "Files and folders" tab is selected, showing a table of uploaded files. The file "Kalpesh\_AWS.pdf" is listed with a status of "Succeeded". The bottom of the screen includes links for CloudShell, Feedback, and cookie preferences.

(9) ZERO to HERO in Excel | destination-bucket-kalpesh - S3 | AWS Policy Generator

ap-south-1.console.aws.amazon.com/s3/buckets/destination-bucket-kalpesh?region=ap-south-1&bucketType=general&tab=objects

AWS Account ID: 7556-8259-4374 | All Bookmarks | Asia Pacific (Mumbai) | Amar D

IAM S3

Amazon S3 > Buckets > destination-bucket-kalpesh

## Amazon S3

### General purpose buckets

- Directory buckets
- Table buckets
- Vector buckets
- Access Grants
- Access Points (General Purpose Buckets, FSx file systems)
- Access Points (Directory Buckets)
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- IAM Access Analyzer for S3

Block Public Access settings for this account

▼ Storage Lens

- Dashboards

CloudShell Feedback

## destination-bucket-kalpesh Info

Objects Properties Permissions Metrics Management Access Points

### Objects (1)

Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix Show versions

Name	Type	Last modified	Size	Storage class
Kalpesh_AWS.pdf	pdf	September 3, 2025, 17:08:55 (UTC+05:30)	76.7 KB	Standard

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences





















