
Multi-Region Asynchronous Object Replication Solution

Implementation Guide



Multi-Region Asynchronous Object Replication Solution: Implementation Guide

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Home	1
Overview	2
Cost	2
Architecture	2
Components	5
Regional Bucket Creation	5
Multipart Upload Model	5
Metrics and Alarms	6
Base Regional Resources	6
Regional Bucket Metrics	6
Regional Whitelist	7
Considerations	8
Performance Testing	8
Critical Workloads	8
Replication Order	8
Amazon S3 Bucket Naming	8
Replication to Publicly Accessible Buckets	9
Cross-Account Replication	9
Regional Deployments	9
Template	10
Deployment	11
What We'll Cover	11
Step 1. Launch the Stack	11
Step 2. Create Base Regional Resources in Other Regions	12
Step 3. Replicate a Large Object	13
Security	14
IAM Roles	14
Encryption	14
Bucket Deletion	14
Resources	15
Appendix A: Cost Breakdown	16
Appendix B: AWS CloudFormation Template Resources	17
Appendix C: Uninstall the Solution	20
Appendix D: Operational Metrics	21
Source Code	22
Revisions	23
.....	23

Deploy a solution that helps minimize latency when accessing objects in different geographic regions, meet compliance requirements, and increase operational efficiency in the AWS Cloud

February 2020

This implementation guide discusses architectural considerations and configuration steps for deploying the Multi-Region Asynchronous Object Replication Solution in the Amazon Web Services (AWS) Cloud. It includes links to an [AWS CloudFormation](#) template that launches and configures the AWS services required to deploy this solution using AWS best practices for security and availability.

The guide is intended for IT infrastructure architects, administrators, and DevOps professionals who have practical experience architecting in the AWS Cloud.

Overview

[Amazon Simple Storage Service](#) (Amazon S3) offers cross-region replication, a bucket-level feature that enables automatic, asynchronous copying of objects from a bucket in one AWS Region to a bucket in a different region. This feature can help you minimize latency when accessing objects in different geographic regions, meet compliance requirements, and increase operational efficiency. However, it can be a complex and time-consuming process to manually configure the buckets to replicate objects across multiple regions.

To help make it easier to set up a multi-region replication configuration, AWS offers the Multi-Region Asynchronous Object Replication Solution (MARS). This solution automatically provisions and configures all the AWS services necessary to set up a multi-region replication configuration with Amazon S3 buckets. For example, you can create objects in Oregon, rename them in Singapore, and delete them in Dublin, and the changes are replicated to all other regions.

Important

MARS leverages Amazon S3 event notifications, which typically delivers events in seconds. However, event delivery can take a minute or longer, and events can be lost. For customers with strict replication requirements, we do not recommend using this solution. For more information, see [Critical Workloads](#) (p. 8).

This solution is designed for workloads that can tolerate lost events and variations in replication speed. For example, you can use this solution to provide local copies of data for multi-regional applications or distributed teams, or to create a global view of applications that run on active/active clusters across multiple regions.

You can also extend this solution's functionality by configuring [Amazon Kinesis Data Firehose](#) to load streaming data into this solution's regional buckets. The streaming data would then be replicated across all regional buckets, creating a consistent dataset in each region for local analytics.

If you must maintain objects in multiple regions to comply with business or regulatory requirements, you can use this solution as a reference to build an architecture that can help automate that process.

Cost

You are responsible for the cost of the AWS services used while running this reference deployment. As of the date of publication, the cost for running this solution with default settings in the US East (N. Virginia) Region is approximately **\$16.37 per month**. This estimate assumes your multi-region configuration will consist of three regions with 10 users each uploading 200 objects per day with an average size of 1 MB. Prices are subject to change. For full details, see the pricing webpage for each AWS service you will be using in this solution. For a detailed cost breakdown, see [Appendix A](#) (p. 16).

Architecture Overview

Deploying this solution builds the following environment in the AWS Cloud.

Multi-Region Asynchronous Object Replication Solution Implementation Guide

Architecture

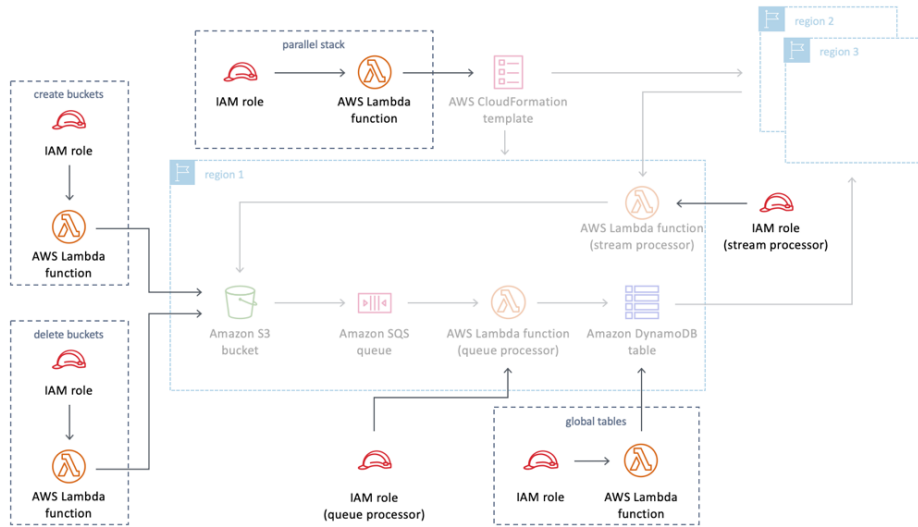


Figure 1: Multi-Region Asynchronous Object Replication Solution resources

The AWS CloudFormation template deploys the [AWS Lambda](#) functions and [AWS Identity and Access Management](#) (IAM) roles necessary to set up a multi-region asynchronous object replication environment in your account. The template creates four Lambda functions: one function that enables you to create regional Amazon Simple Storage Service (Amazon S3) buckets, one function that enables you to delete regional buckets, one function that configures the regional [Amazon DynamoDB](#) tables that track replication changes, and one function that deploys AWS CloudFormation templates that deploy the solution's *base regional resources*.

Base regional resources consist of an Amazon S3 bucket, an [Amazon Simple Queue Service](#) (Amazon SQS) queue, the *queue processor* and *stream processor* Lambda functions, and a DynamoDB table.

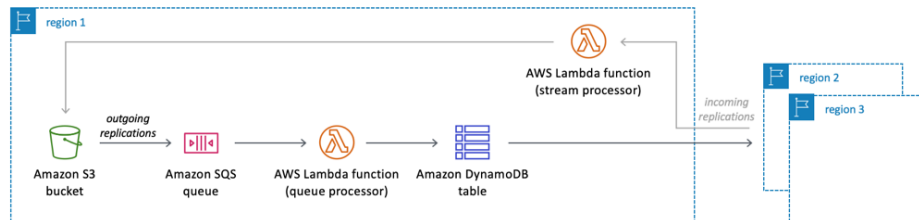


Figure 2: Base regional resources

When you upload objects to a regional bucket, a message is sent through the Amazon SQS queue to the *queue processor*. The *queue processor* adds the message to a DynamoDB global table. The global table replicates the message to all other regions in your configuration which triggers the *stream processor* functions in those regions. Each *stream processor* copies the object from the source region to its regional bucket.

Multi-Region Asynchronous Object Replication Solution Implementation Guide Architecture

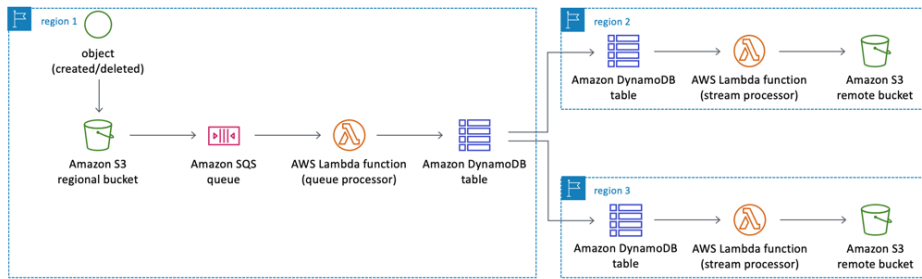


Figure 3: Outgoing replication process

Small objects are copied immediately. Large objects (16 MB or more) are divided into smaller parts by the stream processor and replicated using a parallel, multipart upload pattern. For more information, see [Multipart Upload Model \(p. 5\)](#).

Solution Components

Regional Bucket Creation

After you deploy the solution, you can create your Amazon Simple Storage Service (Amazon S3) regional buckets with the `MARS-CreateMARSBucket` AWS Lambda function. This function launches AWS CloudFormation templates to build the base regional resources and bucket-specific resources for your regional bucket.

The `MARS-CreateMARSBucket` payload has two properties:

- `BaseName`: The base name for the regional bucket.
- `Regions`: A comma-separated list of AWS Region codes to deploy the solution to.

The following example payload deploys buckets, bucket-specific resources, and the base regional resources to three regions: US West (Oregon), EU (Ireland), and Asia Pacific (Singapore).

```
{
  "BaseName": "yourbucketbasename",
  "Regions": "us-west-2,eu-west-1,ap-southeast-1"
}
```

The buckets are named using the `mars-<BaseName>-<Region>` pattern. In the previous example, the three buckets will be named:

- `mars-yourbucketbasename-us-west-1`
- `mars-yourbucketbasename-eu-west-1`
- `mars-yourbucketbasename-ap-southeast-1`

When you execute the `MARS-CreateMARSBucket` function, the function checks to see if the base regional resources are already deployed in the region. For regions that do not have base regional resources, the solution creates one AWS CloudFormation template (`<bucketname>-BaseRegionalResources-<timestamp>`) that uses a `ParallelStack` custom resource to deploy the resources to those regions simultaneously. This template deploys a single AWS CloudFormation template in each region defined in the `Regions` property.

After the base resources are deployed in each applicable region, another AWS CloudFormation template (`MARS-<BaseName>`) deploys the bucket resources. Continuing from the previous example, the template name would be `MARS-yourbucketbasename`. This template deploys templates (`MARS-<BaseName>-RegionalResources`) in each applicable region that launch the bucket resources.

Multipart Upload Model

The Multi-Region Asynchronous Object Replication Solution uses a parallel, multipart upload pattern for large objects (16 MB or more). When a large object is uploaded to Amazon Simple Storage Service (Amazon S3), the regional `stream processor` AWS Lambda function divides the object into smaller

parts. The smaller parts are sent to an Amazon Simple Queue Service (Amazon SQS) queue where the queue processor Lambda function downloads the parts in parallel. The queue processor updates the DynamoDB table which triggers the stream processor to copy the object to the regional bucket.

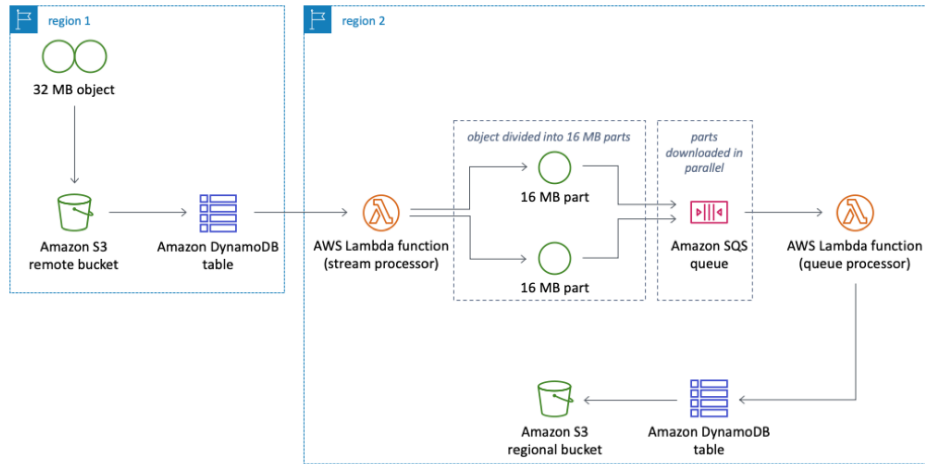


Figure 4: Large object replication process

Metrics and Alarms

This solution leverages [Amazon CloudWatch](#) to collect and track metrics, to generate alarms, and to display those metrics and alarms in a single view. Enabling CloudWatch will incur additional charges.

Base Regional Resources

If enabled, CloudWatch collects metrics for the queue processor and stream processor Lambda functions, the Amazon SQS queue, and the multipart upload DynamoDB table.

CloudWatch also generates alarms that watch the queue processor and stream processor Lambda function metrics in each region and sends an Amazon Simple Notification Service (Amazon SNS) notification to subscribed email addresses when a threshold is breached. You can use the Amazon SNS console to subscribe to the solution's SNS topic.

CloudWatch also creates a dashboard (`MARS-Base-Regional-Resources-<region>`) in each region that displays these metrics and alarms for the region.

Regional Bucket Metrics

If enabled, CloudWatch also collects the following metrics for your regional Amazon S3 buckets.

Parameter	Description	
Requests	The number of requests for the bucket	
Bytes	The size, in bytes, of uploads and downloads	
Latency	The per-request time, in milliseconds, to complete a request	
Errors	The number of bucket errors	

Parameter	Description	
Storage	The total size of all objects in the bucket. There are updated once per day and may not be in sync from region to region. For the most accurate bucket storage metrics, use the console to list all objects. For customers with more than 1 million objects, use the AWS CLI.	

Regional Whitelist

With this solution, you can create regional buckets in nine specific regions. These regions are listed in a whitelist that prevents the solution from being deployed in regions where Amazon DynamoDB global tables are not available. For more information, see [Regional Deployments \(p. 9\)](#). You can remove regions from the whitelist, but you cannot add additional regions that were not specified in the original list. To remove regions from this whitelist, modify the **SupportedRegions** environment variable in the `MARS-CreateMARSBucket` AWS Lambda function.

Design Considerations

Performance Testing

This solution has been tested for objects up to 25 GB, replication of 1 million objects in a bucket, and a configuration including nine regions. While we expect the solution to work for file sizes larger than 25 GB and more than 1 million objects, we recommend you use a rigorous performance testing process to test the solution at the scale for your use case.

You may experience variations in performance with this solution.

Critical Workloads

This solution leverages [Amazon S3 event notifications](#), which typically delivers events in seconds but can sometimes take a minute or longer. We do not recommend using this solution for critical workloads that require 100% guaranteed replication or replication within a specific timeframe. We recommend using this solution for workloads that require high levels of availability and partition tolerance. For example, use this solution to provide local copies of data for either multi-regional applications or distributed teams. You can also use this solution as a reference for building your own application to maintain objects in multiple regions in order to comply with business or regulatory requirements.

For customers with strict replication requirements, consider leveraging [LIST Objects API](#) or [Amazon S3 Inventory](#) reports for your application. The API and inventory are subject to [eventual consistency](#) and may not reflect recently added or deleted objects.

Replication Order

This solution replicates objects in the order the replications are received in each region. But, replications may be received in different orders in different regions, if an object is modified two different ways in two different regions at the exact same time. For example, if an object is deleted in US East (N. Virginia) and, at the same time, renamed in US West (Oregon), the object may be deleted in EU (Ireland) and renamed in EU (Frankfurt) because the replications may be received in different orders in those regions. To mitigate the risk of this occurring, you should limit modifications to one region at a time.

Amazon S3 Bucket Naming

When a regional bucket is created, a bucket name that is unique for each region is generated with following the pattern: `mars-<BaseName>-<Region>`. `BaseName` is the base name you specified for the bucket and `Region` is the AWS Region code where the bucket is created. The `MARS-CreateMARSBucket` AWS Lambda function generates a dynamic AWS CloudFormation template that attempts to deploy the bucket in each region you specified.

If a specific bucket name is not available in a region because the name is already in use, the regional AWS CloudFormation template deployment fails and the bucket template (the parallel stack deployment) is notified of the failure and rolls back all the other regional deployments. This is to make sure the solution cannot create a regional bucket if the generated bucket name is not available in each region to prevent the solution from being partially deployed.

Replication to Publicly Accessible Buckets

If you delete one of this solution's regional buckets, and another user creates a bucket with the same name as the deleted bucket with public read and write access, the solution may replicate your objects in the publicly accessible bucket.

If this happens, unauthorized users can read object key names, but they cannot download objects or view object metadata because they do not have access to the [AWS Key Management Service \(AWS KMS\)](#) key used to encrypt the objects at rest.

To delete one of this solution's regional buckets, delete the bucket's contents. Then, use the AWS CloudFormation template (`MARS-DeleteMARSBucket`) to delete the bucket. This prevents the solution from replicating objects in publicly accessible buckets with the same names as this solution's regional buckets.

Cross-Account Replication

Currently, this solution is designed to work in a single account. The solution does not replicate objects in Amazon S3 buckets that are not in the account in which the solution is deployed.

Regional Deployments

This solution uses Amazon DynamoDB global tables which are available in specific AWS Regions only. Therefore, you must deploy this solution in a region where global tables are available.

AWS CloudFormation Template

This solution uses AWS CloudFormation to automate the deployment of the Multi-Region Asynchronous Object Replication Solution in the AWS Cloud. It includes the following AWS CloudFormation template, which you can download before deployment.

[View
Template](#)

multi-region-asynchronous-object-replication-solution.template: Use this template to launch the Multi-Region Asynchronous Object Replication Solution and all associated components. The default configuration deploys AWS Lambda functions and AWS Identity and Access Management (IAM) roles, but you can also customize the template based on your specific needs.

Automated Deployment

Before you launch the automated deployment, please review the architecture, configuration, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the Multi-Region Asynchronous Object Replication Solution into your account.

Time to deploy: Approximately five minutes

What We'll Cover

The procedure for deploying this architecture on AWS consists of the following steps. For detailed instructions, follow the links for each step.

[Step 1. Launch the Stack \(p. 11\)](#)

- Launch the AWS CloudFormation template into your AWS account.
- Review the template parameters, and adjust if necessary.

[Step 2. Create Base Regional Resources in Other Regions \(p. 12\)](#)

- Launch the AWS Lambda function to create a regional bucket and base regional resources.

[Step 3. Replicate a Large Object \(p. 13\)](#)

- Replicate a large object to test the functionality.

Step 1. Launch the Stack

This automated AWS CloudFormation template deploys the Multi-Region Asynchronous Object Replication Solution in the AWS Cloud.

Note

You are responsible for the cost of the AWS services used while running this solution. See the [Cost \(p. 2\)](#) section for more details. For full details, see the pricing webpage for each AWS service you will be using in this solution.

1. Sign in to the AWS Management Console and click the button below to launch the `multi-region-asynchronous-object-replication-solution` AWS CloudFormation template.



You can also [download the template](#) as a starting point for your own implementation.

2. The template is launched in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the region selector in the console navigation bar.

Note

This solution uses Amazon DynamoDB global tables, which are currently available in specific AWS Regions only. Therefore, you must launch this solution in an AWS Region where global tables are available.

3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template, and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Enable Bucket Metrics	true	Choose whether to enable Amazon S3 bucket metrics. Note Enabling this feature will incur additional charges. For more information, see Monitoring Metrics with CloudWatch .
Create Regional Dashboards	true	Choose whether to create Amazon CloudWatch dashboards in each region. Note Enabling this feature will incur additional charges. For more information, see the CloudWatch pricing page .
Code Bucket Base Name	solutions	The name of the bucket that contains the solution code. We do not recommend changing this parameter.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Be sure to check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
9. Choose **Create stack** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of **CREATE_COMPLETE** in approximately five minutes.

Step 2. Create Base Regional Resources in Other Regions

Use this procedure to create the solution's base regional resources in other regions.

1. Navigate to the [AWS CloudFormation console](#).
2. Select the solution stack.
3. Choose the **Resources** tab.
4. Under **Physical ID**, select **MARS-CreateMARSBucket**.
5. On the **MARS-CreateMARSBucket** Lambda function page, select **Test** (in the upper right corner).
6. On the **Configure Test Events** page, choose **Create new test event**.
7. For **Event name**, enter `yourbucketbasename`.
8. Replace the sample event code with the following code.

```
{
  "BaseName": "yourbucketbasename",
  "Regions": "us-west-2, eu-west-1, ap-southeast-1"
}
```

9. Choose **Create**.
10. Select **Test**.

The solution checks each region for base regional resources. For regions that do not have base regional resources, the solution deploys those resources. For more information, see [Regional Bucket Creation](#) (p. 5).

Step 3. Replicate a Large Object

Use this procedure to replicate a large object to test the functionality of the solution.

1. Upload a large file (100 MB or more) to the bucket in the US West (Oregon) Region.
2. Use the region selector in the console navigation bar to switch to the EU (Ireland) Region.
3. Navigate to the [Amazon DynamoDB console](#).
4. In the navigation pane, select **Tables**.
5. Select the **MultiPartUpload** table.
6. Choose the **Items** tab.
7. Select the refresh button periodically until the **Status** changes to **Complete**.
8. Verify that the large object was successfully replicated in the bucket in the EU (Ireland) Region.
9. Verify that the large object was successfully replicated in the bucket in the Asia Pacific (Singapore) Region.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. For more information about security on AWS, visit the [AWS Security Center](#).

IAM Roles

AWS Identity and Access Management (IAM) roles enable customers to assign granular access policies and permissions to services and users on the AWS Cloud. This solution creates IAM roles that grants the solution's AWS Lambda functions access to create regional resources. This role is necessary to allow the solution to manage the resources.

Encryption

AWS highly recommends that customers encrypt sensitive data in transit and at rest. The Amazon Simple Storage Service (Amazon S3) buckets this solution creates are encrypted with [Amazon S3 Server-Side Encryption \(SSE\)](#) using AWS Key Management Service (AWS KMS). The solution's Amazon Simple Queue Service (Amazon SQS) queue and Amazon DynamoDB tables are also encrypted at rest using SSE. To encrypt data in transit, this solution uses signed requests and the Transport Layer Security (TLS) standards used by the AWS Node.js SDK.

Bucket Deletion

This solution includes an AWS CloudFormation template (`MARS-DeleteMARSBucket`) you can use to delete the regional Amazon S3 buckets. If you do not use the template to delete regional buckets, and a user creates a bucket with the same name as the deleted bucket with public read and write access, the solution may replicate your objects in the publicly accessible objects.

If this happens, unauthorized users can read object key names, but they cannot download objects or view object metadata because they do not have access to the AWS KMS key used to encrypt the objects at rest.

Additional Resources

AWS services

- [Amazon Simple Storage Service](#)
- [AWS CloudFormation](#)
- [AWS Lambda](#)
- [Amazon DynamoDB](#)
- [Amazon Simple Queue Service](#)
- [Amazon CloudWatch](#)
- [Amazon Simple Notification Service](#)

Appendix A: Cost Breakdown

You are responsible for the cost of the AWS services used while running this solution. The total cost to run this solution depends on the number of buckets and regions in your configuration, and the amount of data you replicate. For a customer with a configuration that includes three regions with 10 users uploading 200 objects per day with an average size of 1 MB, the total cost is **\$16.37 per month** as of the date of publication. This estimate is based on the following assumptions:

- **Amazon Simple Storage Service (Amazon S3) storage:** 10 users each uploading 200 MB of data per day in three regions: 6 GB per month
- **Amazon S3 data transfer:** 6 GB per month replicated to two regions: 12 GB data transfer per month
- **Amazon S3 requests:** 10 users each making 200 replications (200 GET, 200 PUT) per day to 2 replicated regions: 120,000 GET requests and 120,000 PUT requests per month
- **AWS Key Management Service (AWS KMS):** three customer master keys (one per region)
- **Amazon DynamoDB:** 3 regional writes and two reads * 6,000 S3 Object Events
- **AWS Lambda requests:** 6,000 S3 Object Events per region * 3 regions: 18,000 AWS Lambda requests + 3 regions * Lambda SQS poller (1 per second) = 2.6 million requests
- **Lambda duration:** 256 MB of memory allotted to the function, 150,000 executions per month for one second each
- **Amazon CloudWatch logs:** 1 KB log statement * 2.6 million log statements per month: 2.6 GB per month
- **Amazon CloudWatch dashboard:** 3 dashboards (1 per region)

Service	Total cost per month
Amazon S3 storage	\$0.14
Amazon S3 data transfer	\$0.06
Amazon S3 GET requests	\$0.05
Amazon S3 PUT requests	\$0.60
AWS KMS customer master keys	\$3.00
Amazon DynamoDB	\$0.03
AWS Lambda requests	\$1.56
AWS Lambda duration	\$0.63
Amazon CloudWatch logs	\$1.30
Amazon CloudWatch dashboard	\$9.00
Total	\$16.37

For a customer with a configuration that includes nine regions with 100,000 users uploading 10 MB of objects per day with an average size of 1 MB, the total cost is **\$14,362 per month**.

You can optimize costs by changing the Amazon S3 storage type to a lower cost tier and using fewer regions in your configuration.

Appendix B: AWS CloudFormation Template Resources

The following resources are created as part of the Multi-Region Asynchronous Object Replication Solution stack.

Resource Name	Type	Description
MARS-CreateMARSBucket	<code>AWS::Lambda::Function</code>	The Lambda function that creates regional buckets.
MARS-DeleteMARSBucket	<code>AWS::Lambda::Function</code>	The Lambda function that deletes regional buckets.
MARS-CreateMARSBucket-Role	<code>AWS::IAM::Role</code>	The IAM role that allows the <code>MARS-CreateMARSBucket</code> Lambda function to create regional buckets.
MARS-CustomResource-GlobalTable	<code>AWS::Lambda::Function</code>	The custom resource in the solution-generated AWS CloudFormation templates that configures the DynamoDB journal tables.
MARS-DeleteMARSBucket-Role	<code>AWS::IAM::Role</code>	The IAM role that allows the <code>MARS-DeleteMARSBucket</code> Lambda function to delete regional buckets.
MARS-GlobalTable-Role	<code>AWS::IAM::Role</code>	The IAM role that allows the <code>MARS-CustomResource-GlobalTable</code> Lambda function to configure the DynamoDB tables used to track replication events.
MARS-ParallelStackSet-Role	<code>AWS::IAM::Role</code>	The IAM role that the <code>MARS-CustomResource-ParallelStackSet</code> Lambda function uses to create base regional resources in each region.
MARS-QueueProcessor-Role	<code>AWS::IAM::Role</code>	The IAM role that allows the <code>MARS-QueueProcessor</code> Lambda function to copy multipart upload tables from buckets, write the state to DynamoDB, and pull messages from the Amazon SQS queue.
MARS-StreamProcessor-Role	<code>AWS::IAM::Role</code>	The IAM role that allows the <code>MARS-StreamProcessor</code>

Resource Name	Type	Description
		Lambda function to read messages from DynamoDB streams, manage the multipart upload state in DynamoDB, and write messages to the Amazon SQS queue for multipart upload processing.
MARS-CustomResource-ParallelStackSet	<code>AWS::Lambda::Function</code>	The Lambda function that deploys solution-generated AWS CloudFormation templates to multiple regions simultaneously.

The following resources are created as part of the base regional resources stack.

Resource Name	Type	Description
DLQueue	<code>AWS::SQS::Queue</code>	The dead-letter queue.
Dashboard	<code>AWS::CloudWatch::Dashboard</code>	The CloudWatch dashboard.
MultiPartUploadTable	<code>AWS::DynamoDB::Table</code>	The DynamoDB tables that tracks multipart upload progress.
Queue	<code>AWS::SQS::Queue</code>	The Amazon SQS queue that captures change events and stores multipart upload messages. After 10 failed retries, messages are moved to the dead-letter queue.
QueuePolicy	<code>AWS::SQS::QueuePolicy</code>	The policy that gives the Lambda functions and Amazon S3 buckets access to the Amazon SQS queue.
QueueProcessor	<code>AWS::Lambda::Function</code>	The Lambda function that processes messages from the Amazon SQS queue.
QueueSubscription	<code>AWS::Lambda::EventSourceMapping</code>	The IAM role that the <code>MARS-CustomResource-ParallelStackSet</code> Lambda function uses to create base regional resources in each region.
StreamProcessor	<code>AWS::Lambda::Function</code>	The Lambda function that processes messages from the journal DynamoDB stream.
Key	<code>AWS::KMS::Key</code>	The customer master key (CMK) used to encrypt Amazon SQS messages at rest.

Resource Name	Type	Description
KeyAlias	AWS::KMS::Alias	The human-readable name for the CMK for this solution.
Bucket	AWS::S3::Bucket	The Amazon S3 bucket that stores objects.
DynamoDBTable	AWS::DynamoDB::Table	The journal table that replicates events across regions.
StreamSubscription	AWS::Lambda::EventSourceMapping	A subscription that connects the regional journal table to the stream processor Lambda function.

Appendix C: Uninstall the Solution

To uninstall the Multi-Region Asynchronous Object Replication Solution, you must delete the templates in the reverse order you installed them. Use this procedure to uninstall the solution.

1. Delete all Amazon Simple Storage Service (Amazon S3) bucket contents from all regional buckets.
2. Delete the `MARS-<BaseName>` AWS CloudFormation stack in each region.
3. Delete the base regional resources.
4. Delete the stack you created using the `<bucketname>-BaseRegionalResources-<timestamp>` template.
5. Delete the stack you created using the `multi-region-asynchronous-object-replication-solution` template.

Appendix D: Collection of Operational Metrics

This solution includes an option to send anonymous operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When enabled, the following information is collected and sent to AWS:

- **Solution ID:** The AWS solution identifier
- **Unique ID (UUID):** Randomly generated, unique identifier for each solution deployment
- **Timestamp:** Data-collection timestamp

Note that AWS will own the data gathered via this survey. Data collection will be subject to the [AWS Privacy Policy](#). To opt out of this feature, modify the AWS CloudFormation template mapping section as follows:

```
"Send" : {  
  "AnonymousUsage" : { "Data" : "Yes" }  
},
```

to

```
"Send" : {  
  "AnonymousUsage" : { "Data" : "No" }  
},
```


Source Code

You can visit our [GitHub repository](#) to download the templates and scripts for this solution, and to share your customizations with others.

Document Revisions

Date	Change	
February 2020	Initial release	

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

The Multi-Region Asynchronous Object Replication Solution is licensed under the terms of the Apache License Version 2.0 available at <https://www.apache.org/licenses/LICENSE-2.0>