

---

# AWS Data Exchange Publisher Coordinator Implementation Guide



## **AWS Data Exchange Publisher Coordinator: 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 overview .....	2
Components .....	4
Design considerations .....	5
Amazon S3 buckets .....	5
New Amazon S3 bucket .....	5
Existing Amazon S3 buckets .....	5
AWS Data Exchange data product provider .....	5
Security .....	6
Lambda access to Amazon S3 .....	6
AWS CloudFormation template .....	7
Automated deployment .....	8
Prerequisites .....	8
AWS Data Exchange product .....	8
IAM Permissions .....	8
Logging and asset Amazon S3 buckets .....	8
Create a logging Amazon S3 bucket .....	8
Deployment .....	9
Step 1. Launch the stack .....	9
Step 2. Create and upload a manifest file .....	10
Dataset revision manifest file .....	11
Additional resources .....	12
Troubleshooting .....	13
Uninstall the solution .....	14
Using the AWS Management Console .....	14
Using AWS Command Line Interface .....	14
Deleting the Amazon S3 buckets .....	14
Using the AWS Management Console .....	14
Using AWS Command Line Interface .....	14
Operational metrics .....	16
Source code .....	17
Contributors .....	18
Revisions .....	19
Notices .....	20

# AWS Data Exchange Publisher Coordinator

## **AWS Implementation Guide**

Publication date: *December 2020*

This implementation guide describes architectural considerations and provides configuration steps for deploying AWS Data Exchange Publisher Coordinator 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

The AWS Data Exchange Publisher Coordinator solution automates the steps to create a new revision for an existing dataset in the AWS Marketplace used by [AWS Data Exchange](#). Each revision for a dataset can consist of one or more data files. Currently, to manage AWS Data Exchange dataset revisions for file-based data products, you would manually upload and download your dataset revisions. Alternatively, you could develop and maintain your own solutions for automation. With this solution, you can reduce these manual processes and publish file objects in one step.

Your subscribers can use the [AWS Data Exchange Subscriber Coordinator](#) solution to retrieve the new data to their [Amazon Simple Storage Service](#) (Amazon S3) bucket location as soon as the data publication completes.

## Cost

You are responsible for the cost of the AWS services used while running this solution. As of the date of publication, the estimated cost for running this solution containing four AWS Lambda functions and one AWS Step Functions with default settings in the US East (N. Virginia) Region is **\$0.01 per month** for 10 publications per day for 30 days. This calculation assumes that there is no free-tier consumption available in the account.

AWS service	Quantity	Cost
AWS Lambda	1,500 runs in a month at 3000 ms	\$0.01
AWS Step Functions	300 runs in a month	\$0.00002
Total cost		<b>\$0.01</b>

This cost estimate does not account for Amazon S3 storage or AWS Data Exchange usage. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service you will be using in this solution.

## Architecture overview

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

# AWS Data Exchange Publisher Coordinator Implementation Guide Architecture overview

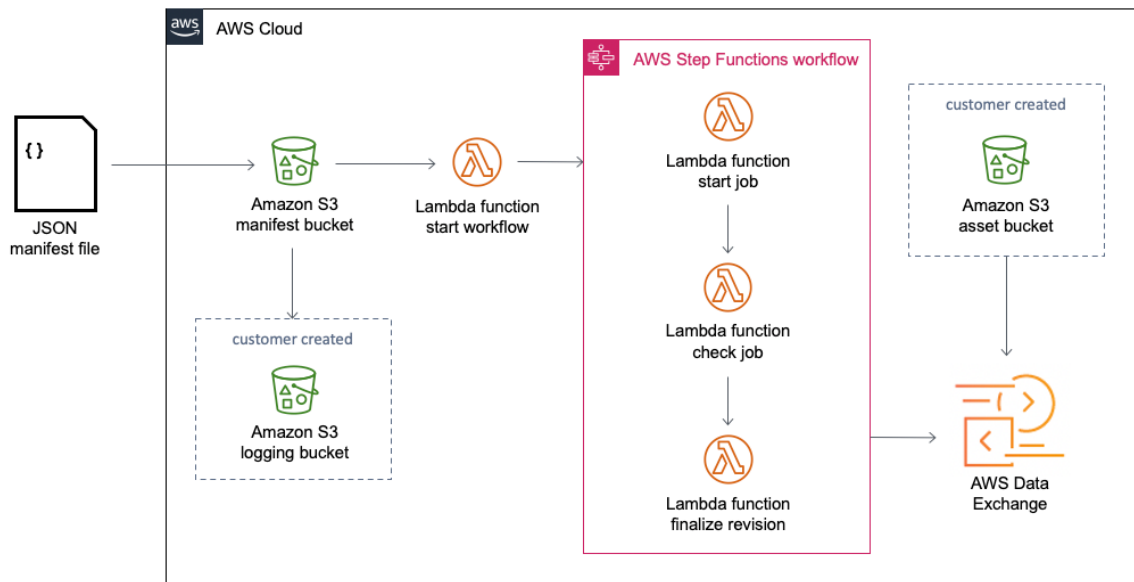


Figure 1: AWS Data Exchange Publisher Coordinator architecture on AWS

The AWS CloudFormation template creates an Amazon S3 bucket for submitting manifests when a dataset is ready for publication. It deploys the serverless infrastructure necessary to coordinate the publication steps required for AWS Data Exchange, which includes the following:

- An Amazon S3 bucket, where manifest files are uploaded.
- Four [AWS Lambda](#) functions: one Lambda function initiates the [AWS Step Function](#); and three Lambda functions that create a revision of an AWS Data Exchange product, monitor job runs, and finalize and publish the revision.
- An AWS Step Functions to orchestrate the management of the publishing process.
- [AWS Identity and Access Management](#) (IAM) roles that set up permissions for AWS Step Functions and the AWS Lambda functions to run their API calls.

This solution does not deploy either AWS Data Exchange resources or the Amazon S3 asset bucket and logging bucket. However, AWS Data Exchange retrieves the objects from the S3 asset bucket, which is referenced in the manifest file.

# Components

After you have stored the data in the Amazon S3 asset bucket (not created by refer AWS CloudFormation template), create a JSON manifest file referencing the location of each of the files that you plan to publish.

When you upload the manifest file to the Amazon S3 manifest bucket (created by this template), it initiates the process to publish a new revision. It invokes the creation of a new job in AWS Data Exchange, adds each of the files referenced in the manifest file to a new revision, then finalizes the revision once all of the assets have been added. Upon job completion, your customers can access the new dataset revision.

# Design considerations

## Amazon S3 buckets

This solution uses three Amazon Simple Storage Service (Amazon S3) buckets. The first is the manifest bucket, where you upload manifest files to. The second is the logging bucket where manifest logging is sent to. The third is the asset bucket where you store your data files which the solution transfers to AWS Data Exchange. This solution creates the manifest bucket; however, before launching the AWS CloudFormation template, identify the names of the asset and manifest logging buckets.

### New Amazon S3 bucket

This solution creates the Amazon S3 `ManifestBucket-<solution-ID>` as a destination to upload manifest files to. This Amazon S3 bucket invokes an AWS Lambda function when manifest files are uploaded. For auditing purposes, the `ManifestBucket` has Amazon S3 server access logging enabled. The access logs are sent to the `<manifest-bucket-logging-bucket-name>`.

### Existing Amazon S3 buckets

The `<manifest-bucket-logging-bucket-name>` Amazon S3 bucket must be created before launching the CloudFormation template. Enter the name of this S3 bucket in the `ManifestBucketLoggingBucket` parameter value when you [launch the stack \(p. 9\)](#). You can use an existing Amazon S3 bucket in your account where logging activity is already being written to (Amazon S3 log delivery group write access) or you can create a new Amazon S3 bucket. Refer to [Create a logging Amazon S3 bucket \(p. 8\)](#) for details.

The Amazon S3 `<asset-bucket-name>` asset bucket serves as the source bucket for the data files and must also be created before launching the CloudFormation template. Enter the name of this S3 bucket in the `AssetBucket` parameter value when you [launch the stack \(p. 9\)](#). Since the Lambda function that creates the dataset revision in AWS Data Exchange must have access to this Amazon S3 bucket, you must specify the `<asset-bucket-name>` name so an Identity and Access Management (IAM) role can be created for the Lambda function. You can use an existing Amazon S3 bucket.

## AWS Data Exchange data product provider

To publish file-based data products through the AWS Marketplace using AWS Data Exchange, you must register as a provider and publish a product. For detailed instructions on becoming a provider and publishing a product, refer to [Providing Data Products on AWS Data Exchange](#) in the *AWS Data Exchange User Guide*.



# Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model reduces 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 [AWS Cloud Security](#).

## Lambda access to Amazon S3

The `CreateRevisionFunction-<solution-ID>` Lambda function uses the `CreateRevisionFunctionRole` AWS IAM role to access the `<asset-bucket-name>` Amazon S3 bucket. You can extend this solution by allowing the Lambda function access to additional asset S3 buckets by listing new buckets in the IAM role policy.

# AWS CloudFormation template

To automate deployment, this solution uses AWS CloudFormation. It includes the following CloudFormation template, which you can download before deployment:

[View  
Template](#)

**aws-data-exchange-publisher-coordinator.template:** Use this template to launch AWS Data Exchange Publisher Coordinator and all associated components. The default configuration deploys AWS Lambda functions, AWS Step Functions, an Amazon Simple Storage Service (Amazon S3) bucket, and associated AWS Identity and Access Management (IAM) roles. You can customize this template to meet your specific needs.

# Automated deployment

Before you launch the solution, review the architecture, configuration, network security, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the AWS Data Exchange Publisher Coordinator solution into your account.

**Time to deploy:** Approximately five minutes.

## Prerequisites

### AWS Data Exchange product

The AWS Data Exchange product and dataset must be created prior to uploading manifest files to the `ManifestBucket` Amazon S3 bucket because they are both referenced in the manifest file itself. If a manifest file is uploaded to the `ManifestBucket` S3 bucket without a valid product ID or dataset ID, the AWS Step Functions run will end in an error condition.

### IAM Permissions

To deploy this solution, you must have an IAM user role with permissions to launch AWS CloudFormation templates that create Step Functions, AWS Lambda functions, and Amazon S3 buckets; and set the associated permissions.

### Logging and asset Amazon S3 buckets

If you have two Amazon S3 buckets that meet the criteria, you can use them and go straight to [Deployment \(p. 9\)](#). Refer to [Existing Amazon S3 buckets \(p. 5\)](#) for details about the two S3 buckets.

### Create a logging Amazon S3 bucket

Follow these instructions to create the `<manifest-bucket-logging-bucket-name>` S3 bucket.

1. Sign in to the [Amazon S3 console](#).
2. Choose **Create bucket**.
3. Enter a unique name for the manifest access logging bucket.

The bucket name must:

- Be unique across all of Amazon S3
- Be between 3 and 63 characters long
- Not contain uppercase characters
- Start with a lowercase letter or number

4. In **Region**, choose the AWS Region you will be using to deploy this solution.
5. Choose **Create bucket**.
6. Select the newly created bucket by typing its name in the **Find bucket by name** search field. Choose the bucket then choose **Permissions**. Scroll down to **Access control list (ACL)** and choose **Edit**.

7. Under the **S3 log delivery group**, select **List** and **Write** in the **Objects** column and select **Read** in the **Bucket ACL** column.
8. Choose **Save changes**.

**Note**

To create an `<asset-bucket-name>` S3 bucket, repeat steps 1-5.

## Deployment

### Step 1. Launch the stack

1. Sign in to the AWS Management Console and select the button below to launch the in this guide to launch the aws-data-exchange-publisher-coordinator AWS CloudFormation template.



Alternatively, you can [download the template](#) as a starting point for your own implementation.

2. The template launches in the US East (N. Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to [IAM and STS quotas](#) in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
<b>AssetBucket</b>	<code>&lt;Requires input&gt;</code>	An existing Amazon S3 bucket used to store data assets prior to being uploaded to AWS Data Exchange. The <code>aws-data-exchange-publisher-coordinator</code> AWS CloudFormation template uses this information to grant read access to the AWS Lambda function that creates the revision.
<b>LoggingLevel</b>	INFO	Specifies the level of information detail to log in CloudWatch Logs when running. Other options include: <code>DEBUG</code> , <code>WARNING</code> , <code>ERROR</code> , and <code>CRITICAL</code> .

Parameter	Default	Description
<b>ManifestBucket</b>	<Requires input>	A new globally unique Amazon S3 bucket name created by this solution. <i>.manifest</i> files are uploaded to this bucket after the assets are staged in the <code>AssetBucket</code> S3 bucket.
<b>ManifestBucketLoggingBucket</b>	<Requires input>	An existing Amazon S3 bucket that will be used to store activity logs associated with the use of the <code>ManifestBucket</code> .
<b>ManifestBucketLoggingPrefix</b>	<Optional input>	An optional prefix to add to the server access log objects.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the boxes 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 receive a `CREATE_COMPLETE` status in approximately five minutes.

**Note**

This solution deploys an AWS Lambda function, `solution-helper`, which runs only during initial configuration or when resources are updated or deleted. You can view the `solution-helper` function in the AWS Lambda console. Do not delete this function, as it is needed to manage associated resources.

## Step 2. Create and upload a manifest file

The CloudFormation stack runs the publishing automation steps when a manifest file is uploaded to the `ManifestBucket` Amazon S3 bucket specified in the parameters. Refer to Dataset revision manifest file for the details of the required file structure.

# Dataset revision manifest file

The automation for the publication of the AWS Data Exchange product is controlled by the dataset revision manifest file. When the assets are in their source location, you can upload the manifest file to the `ManifestBucket` Amazon S3 bucket to start the automated process of publishing a new revision. The name of the object key must end in `.manifest` and can be preceded by other information to help maintain order.

The contents of the [JSON](#) manifest file list the assets for a new dataset revision. For example:

```
{ "Manifest":  
  { "Version": "1.0",  
    "ProductId": "<product id>",  
    "DatasetId": "<dataset id>",  
    "Assets": [  
      { "Bucket": "<bucket>", "Key": "<file1>" },  
      { "Bucket": "<bucket>", "Key": "<file2>" },  
      { "Bucket": "<bucket>", "Key": "<file3>" }  
    ]  
  }  
}
```

**Note**

You can locate the product ID and the dataset ID in the details pages of the AWS Data Exchange console for your AWS Data Exchange product.

The Amazon S3 bucket listed in the manifest file is the asset bucket specified as a parameter when deploying this solution. To use additional Amazon S3 buckets, refer to [Lambda access to Amazon S3 \(p. 6\)](#).

# Additional resources

## AWS services

- |  |  |
|--|--|
| <ul style="list-style-type: none"><li>• <a href="#">Amazon Simple Storage Service</a></li><li>• <a href="#">AWS CloudFormation</a></li><li>• <a href="#">AWS Data Exchange</a></li></ul> | <ul style="list-style-type: none"><li>• <a href="#">AWS Identity and Access Management</a></li><li>• <a href="#">AWS Lambda</a></li><li>• <a href="#">AWS Step Functions</a></li></ul> |
|--|--|

## Related AWS Solutions Implementation

Your subscribers can use the [AWS Data Exchange Subscriber Coordinator](#) solution to automate the subscription of new revisions and have them delivered to their Amazon S3 bucket when you publish your revision using this solution.

# AWS Data Exchange Publisher Coordinator troubleshooting

This solution requires that you give Amazon Simple Storage Service (Amazon S3) access to send logging information to the manifest logging S3 bucket (`ManifestBucketLoggingBucket`). If this was not configured before launching the solution, you will receive an error similar to the following:

```
You must give the log-delivery group WRITE and READ_ACP permissions to the target bucket  
(Service: Amazon S3; Status Code: 400; Error Code: InvalidTargetBucketForLogging;)
```

To grant Amazon S3 log delivery group write access to this bucket, refer to step 6 under [Create a logging Amazon S3 bucket \(p. 8\)](#).



# Uninstall the solution

To uninstall the AWS Data Exchange Publisher Coordinator solution, you must delete the CloudFormation stack which will delete all the resources created by the template. The `ManifestBucket` S3 bucket will only be deleted by CloudFormation if it is empty, otherwise it will be retained when the solution stack is deleted. If you have uploaded manifest files to the `ManifestBucket` S3 bucket, you can use either the AWS Management Console or the AWS Command Line Interface (AWS CLI) to empty, then delete the S3 bucket.

## Using the AWS Management Console

1. Sign in to the [AWS CloudFormation console](#).
2. On the **Stacks** page, select the solution stack.
3. Choose **Delete**.

## Using AWS Command Line Interface

Determine whether the AWS CLI is available in your environment. For installation instructions, refer to [What Is the AWS Command Line Interface](#) in the *AWS CLI User Guide*. After confirming that the AWS CLI is available, run the following commands.

```
$ aws cloudformation delete-stack --stack-name <your-stack-name>
```

Replace `<your-stack-name>` with the name of your CloudFormation stack.

## Deleting the Amazon S3 buckets

### Using the AWS Management Console

1. Sign in to the [Amazon S3 console](#).
2. Select the bucket name you provided as a parameter for `ManifestBucket`.
3. Choose **Empty** and follow the instructions to permanently delete all the objects in the bucket.
4. Navigate to the AWS CloudFormation console.
5. Select this solution's installation stack by the name used for the stack when it was deployed.
6. Choose **Delete**.

### Using AWS Command Line Interface

To empty the `ManifestBucket` Amazon S3 bucket, run the following command replacing `<manifest-bucket-name>` with the name of the manifest bucket used as a parameter when the stack was launched. This step will permanently delete all the objects in the S3 bucket that you specify.

```
$ aws s3 rm <manifest-bucket-name> --recursive
```

To delete the stack, run the following command replacing *<installation-stack-name>* with the name of the stack.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

# 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 AWS Data Exchange Publisher Coordinator deployment
- **Timestamp:** Data-collection timestamp
- **Version:** The version of the solution
- **Metrics:** The metrics include lifecycle metrics for the solution (create, run, update, delete) and the number of assets published

AWS owns the data gathered through this survey. Data collection is subject to the [AWS Privacy Policy](#). To opt out of this feature, modify the AWS CloudFormation template mapping section from:

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

to:

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

# Source code

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

# Contributors

The following individuals contributed to this document:

- Chris Marshall
- Colin Marden

# Revisions

Date	Change
December 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.

AWS Data Exchange Publisher Coordinator is licensed under the terms of the MIT No Attribution at [The Software Package Data Exchange](#).