Media Exchange on AWS

Implementation Guide

Media Exchange on AWS: Implementation Guide

Table of Contents

Welcome 1
Cost 2
Architecture overview 3
Solution components 4
    Onboarding tool 4
    Auto ingest 4
    Auto ACL utility 5
    MediaSync utility 6
    Fixity utility 7
Security 8
    Amazon S3 buckets 8
    IAM roles 8
Design considerations 9
AWS CloudFormation template 10
Automated deployment 11
    Prerequisites 11
    Deployment overview 11
    Step 1. Launch the stack 11
    Step 2. Sign in to Media Exchange on AWS 12
    Step 3. Add publishers and subscribers 12
        Provision publisher product 13
        Provision subscriber product 13
        Provision transfer agreement 14
Additional resources 16
Update the stack 17
Update products 18
Uninstall the solution 19
    Using the AWS Management Console 19
    Deleting the Amazon S3 buckets 19
    Using AWS Command Line Interface 20
Source code 21
Revisions 22
Contributors 23
Notices 24

Create a mechanism for transferring media assets to your customers and partners

Publication date: *June 2021 (last update (p. 22): July 2021)*

Traditional file transfer services for media supply chain are expensive, can add unnecessary hours to a workflow, and are not optimized for in-the-cloud media ecosystems. The Media Exchange on AWS solution provides Amazon Web Services (AWS) customers and Independent Software Vendor (ISV) partners a common method for direct media asset transfer within the ecosystem.

This solution creates a shared object storage area using an Amazon Simple Storage Service (Amazon S3) bucket between publishers and subscribers in a separate, secured AWS account. Publishers and subscribers do not share credentials. Publishers copy the assets into this shared S3 bucket and create permissions to allow subscribers to pull the content from the shared resource. As a result, assets do not leave the S3 data plane. In addition to being secure (assets are encrypted at rest and in transit), this process provides the following benefits:

- No per GB data egress and transfer costs within the same AWS Region.
- No egress, schedule, or delivery time dependencies.
- No servers to manage.
- Asset tracking and receipt confirmation. All operations on the assets are tracked via AWS CloudTrail and Amazon S3 server access logs.
- Asset quality and fidelity.

Each publisher-subscriber transfer relationship gets its own S3 bucket to share assets. Publishers have *write* permissions to this bucket, a subscriber can only view assets under their assigned bucket and import assets that are shared with them. The assets shared in the account are lifecycle deleted after a publisher-defined period.

This solution can work seamlessly with non-S3 storage systems as a source and destination. It does not require the assets to be coming from or going to Amazon S3.

This solution can integrate with native AWS file transfer offerings, such as AWS DataSync and AWSSnow Family devices to help transfer assets between physical locations for customers who do not currently integrate Amazon S3 into the media supply chain. You can also use it to move content across large geographical distances by leveraging AWS global infrastructure.

This implementation guide describes architectural considerations and configuration steps for deploying Media Exchange on AWS in the 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 architects, developers, DevOps, MediaOps, and other technology professionals who have practical experience architecting in the AWS Cloud.

Cost

The owner of the MediaExchange account is responsible for the cost of the AWS services used while running this solution. As of June 2021, the cost for using this solution with the default settings in the US East (N. Virginia) is approximately **$84.68 for a 10 TB transfer**.

AWS service	Cost for 10 TB transfer
Amazon Simple Storage Service (Amazon S3)	$79.55
AWS Key Management Service (AWS KMS)	$4.05
AWS Lambda	$0.58
Amazon Simple Notification Service (SNS)	$0.50
Total for 10 TB transfer:	**$84.68**

Important
This cost estimate accounts for an average number of Amazon S3 PUT and GET requests (163,840 PUT and 163,840 GET), which can vary per scenario because of the variance in file sizes.

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

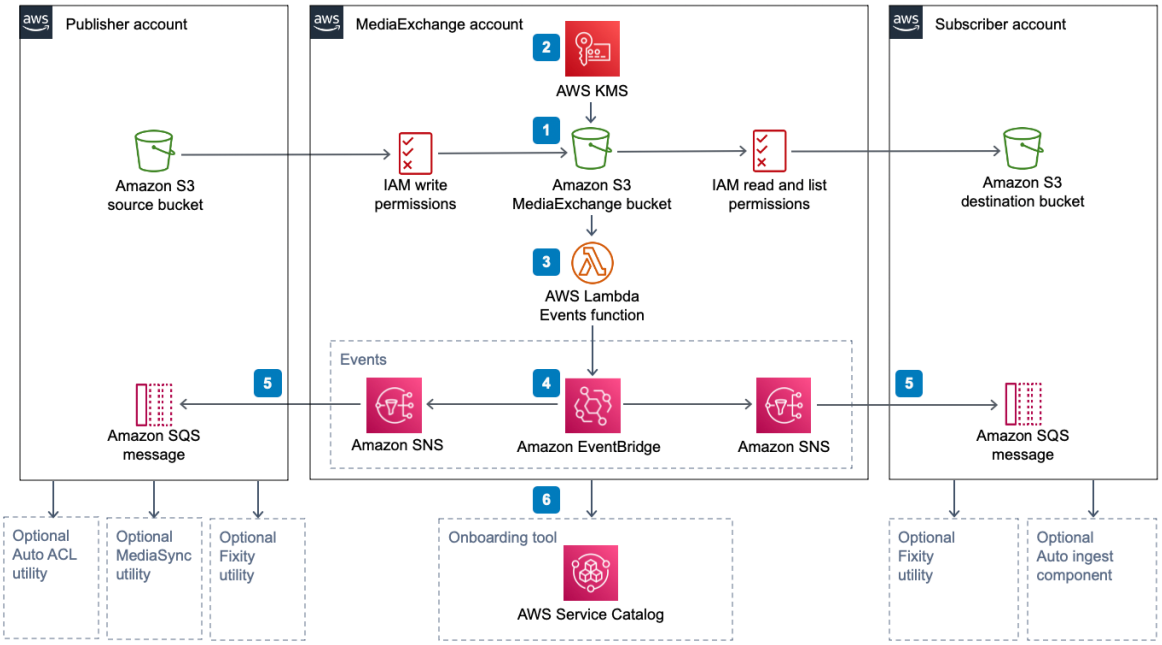

Figure 1: Media Exchange on AWS architecture

The AWS CloudFormation template deploys the following infrastructure on AWS:

1. A `MediaExchange` Amazon S3 bucket. The publisher selects assets from the user-created Source S3 bucket in their account and copies them to the `MediaExchange` S3 bucket. The subscriber copies assets from the shared `MediaExchange` S3 bucket to the user-created Destination S3 bucket in their account.
2. An Amazon KMS key to configure default encryption for the `MediaExchange` S3 bucket.
3. A Lambda function to forward notifications when objects are created in the `MediaExchange` S3 bucket.
4. An Amazon EventBridge bus to receive notifications from the `Events` Lambda function.
5. Amazon SNS topics and Amazon Simple Queue Service (Amazon SQS) messages for subscriber and publisher notifications.
6. A set of products in AWS Service Catalog to onboard publishers and subscribers, and to set up new transfer agreements.

Solution components

Onboarding tool

When you deploy the Media Exchange on AWS solution, it adds deployable products on AWS Service Catalog. AWS Service Catalog deploys infrastructure for a number of publisher and subscriber transfers by deploying a unique, isolated set of resources for each of the transfer relationships.

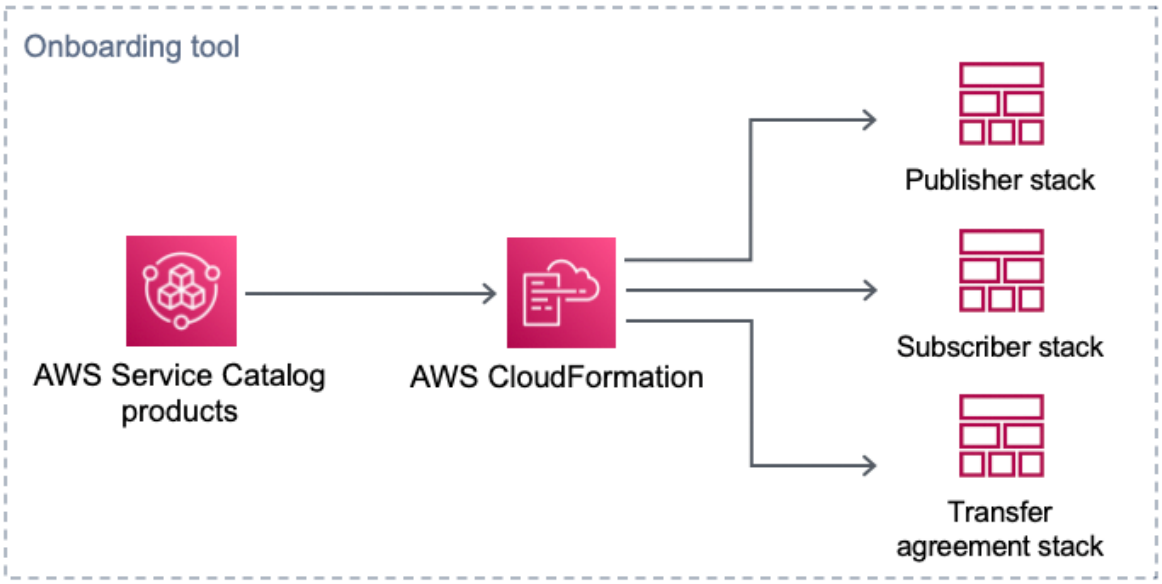

Figure 2: Onboarding tool architecture

To onboard new publishers and subscribers or to set up a new transfer agreement, an account administrator logs in to the solution using the Administrator IAM role for the solution. They then provision the corresponding product (Publisher, Subscriber, or Transfer agreement) that uses AWS CloudFormation to deploy the necessary infrastructure components for that product.

Auto ingest

Subscribers to a `MediaExchange` Amazon S3 bucket have the option to automatically ingest using this component. It automatically moves assets shared through Media Exchange into a subscriber-owned S3 bucket. This optional component is deployed in the subscriber's account.

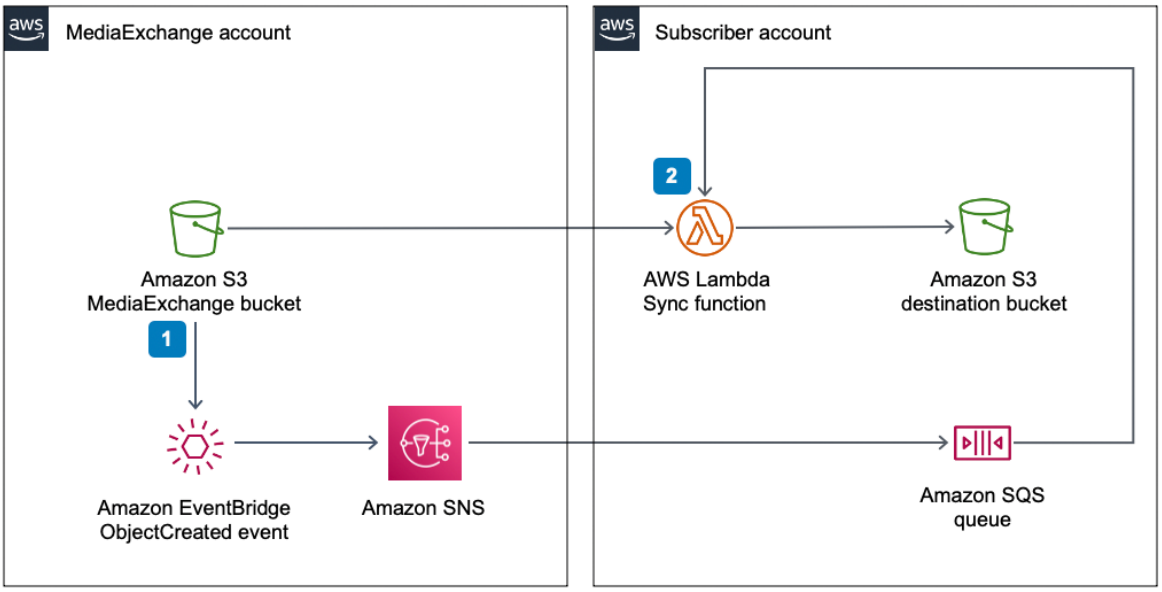

Figure 3: Auto Ingest component workflow

1. When assets are added to the `MediaExchange` S3 bucket, `ObjectCreated` Amazon EventBridge events are sent through Amazon SNS to an Amazon Simple Queue Service (Amazon SQS) queue.
2. The `Sync` Lambda function copies the object from the `MediaExchange` S3 bucket to the destination S3 bucket.

> **Note**
> The `Sync` Lambda function is set up to use S3 server-side copy and multipart uploads. In most cases, the default 15-minute timeout is sufficient to copy files as large as 5 TB in the same AWS Region. However, if you are using this component to ingest assets in a different Region, increase the timeout accordingly. For details, refer to Configuring functions in the console in the *AWS Lambda Developer Guide*.

Auto ACL utility

This optional utility is deployed in the publisher's account so that existing Amazon S3 workflows can adopt to MediaExchange-based transfer workflows without making code changes. The existing workflows treat the `MediaExchange` S3 bucket like any other S3 bucket with write permissions. The Auto ACL utility automates permissions and asset sharing from the `MediaExchange` S3 bucket so that objects copied into the `MediaExchange` S3 bucket have their permissions set for the target subscriber.

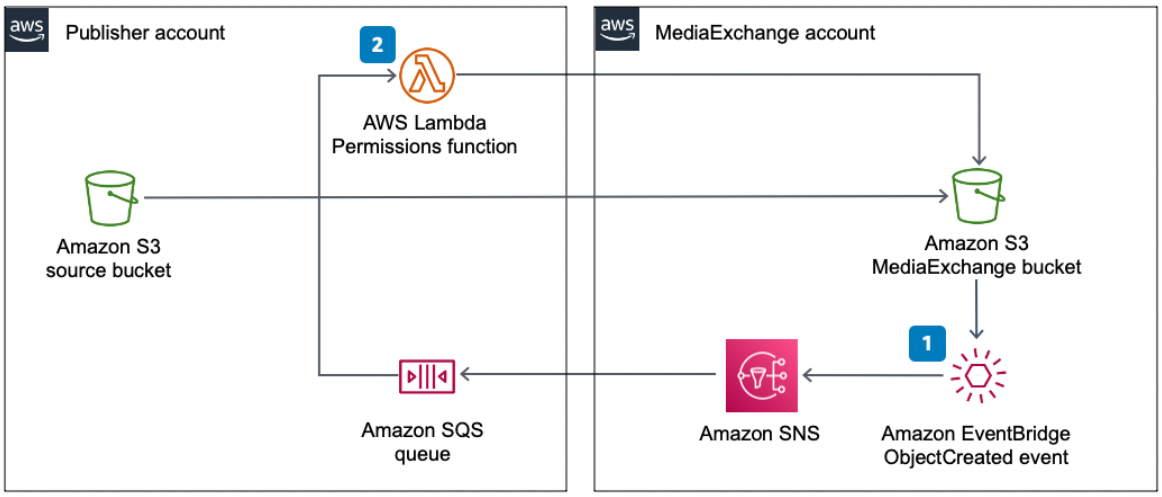

Figure 4: Auto ACL utility workflow

1. Every time an object is copied into the `MediaExchange` S3 bucket, EventBridge sends an `ObjectCreated` event through Amazon SNS to an Amazon SQS queue.
2. The `Permissions` Lambda function is invoked for each of the SNS messages. The `Permissions` Lambda function then makes an API call to set the read ACL on the object from the event. This function is configured to set the permissions for a specific publisher-subscriber relationship at the deployment time.

MediaSync utility

This optional utility moves assets between Amazon S3 buckets. When you deploy the solution, it enables a new toolset in the AWS Management Console that helps move large (100s of GBs) files or hundreds of thousands of small files. The MediaSync utility scales up by running the copy operation in parallel to thousands of concurrent processes. It can handle file sizes up to 5 TB, is resilient, and cost effective. The utility uses S3 server-side copy to move assets between buckets and AWS Fargate Spot for its compute environment.

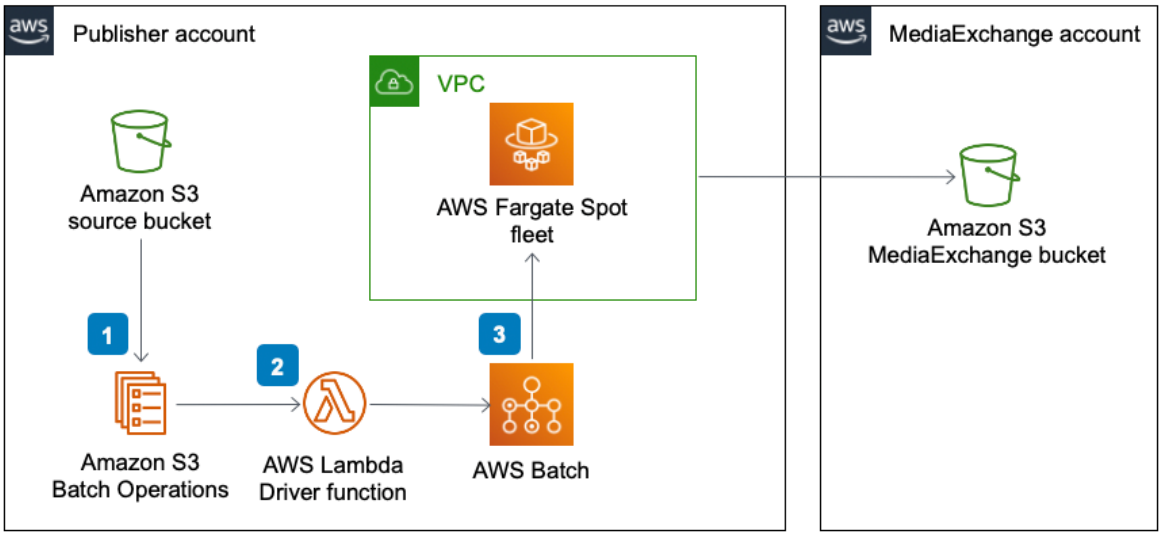

Figure 5: MediaSync utility workflow

1. The publisher selects a list of objects from the source S3 bucket. Then they use Amazon S3 Batch Operations to invoke a `Driver` Lambda function for each one of the objects.
2. The `Driver` Lambda function validates storage class and permissions on the source object. Upon successful validation, it hands off the actual copy operation to a job in AWS Batch.
3. The AWS Batch compute environment is configured to run jobs in Docker containers on AWS Fargate Spot. The default configuration scales to the limit of the maximum number of GET/PUT operations in an S3 partition.

Fixity utility

This optional standalone utility computes checksums at scale by publishers (at source) or by subscribers (at destination) to ensure file integrity. Often checksum computation is required as part of contractual agreements. The Fixity utility uses AWS Batch and Amazon Elastic Compute Cloud (Amazon EC2) Spot Instances to orchestrate the infrastructure. There are no servers to manage. Moreover, the utility calculates checksums by streaming the objects directly from Amazon S3, so that there is no dependency on local storage. In the case of larger files, it can achieve 85% of the theoretical maximum speed - roughly 550 MB/s for md5 on an Intel Skylake/Cascade Lake CPU.

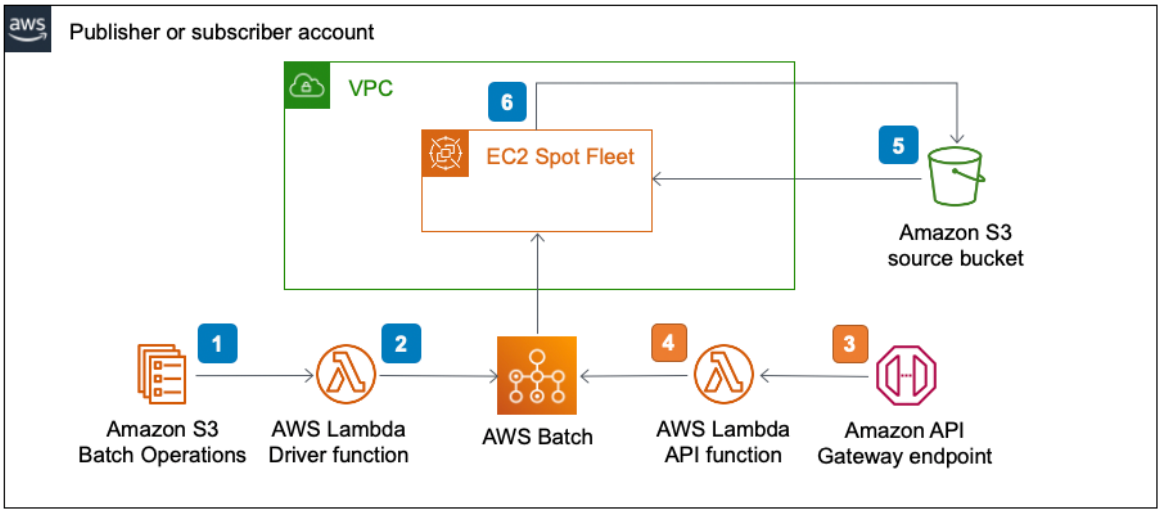

Figure 6: Fixity utility workflow

There are two ways to initiate a checksum. If you have a list of objects, you can use the Amazon S3 Batch Operations interface to initiate the process. Otherwise, use the standalone API.

1. The S3 Batch Operations invokes the `Driver` Lambda function that performs certain validations on source for storage class and/or permissions.
2. After the validations are successful, the `Driver` Lambda function hands off the actual checksum operation to a job in AWS batch. A smaller object gets fewer CPUs and a larger object gets more CPUs.
3. The API invokes the `API` Lambda function that performs certain validations on source for storage class and/or permissions.
4. After the validations are successful, the `API` Lambda function hands off the actual checksum operation to a job in AWS batch.
5. The AWS Batch compute environment is configured to run containers on Amazon EC2 Spot Fleet, which streams the object from the Amazon S3 bucket (source bucket for publishers, destination bucket for subscribers).
6. EC2 Spot Fleet computes checksums. The default configuration calculates md5, sha1, and xxhsum. The results are then saved as custom tags against the object in the source S3 bucket.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate. For more information about AWS security, visit AWS Cloud Security.

Amazon S3 buckets

This solution uses AWS best practices for securing the assets shared through the object storage area. Assets are encrypted by default at rest and in transit. This solution uses AWS Key Management Service (AWS KMS) to store a customer master key (CMK) that has been established for specific account level permissions. The publisher account can use the CMK to encrypt and the subscriber account can use the CMK to decrypt the Amazon S3 bucket level data keys that are used to secure each of the assets in the shared `MediaExchange` Amazon S3 bucket. The S3 bucket is configured with specific permissions so that the publisher account can write to it and the subscriber account can read from it. The objects in the `MediaExchange` S3 bucket are owned by the publisher account and the MediaExchange account does not have any (read or write) permissions to the assets passing through it. The publisher, when sharing through the `MediaExchange` S3 bucket, sets read ACLs for subscriber account by specifying the canonical user id of the subscriber. The `MediaExchange` S3 bucket is configured with a lifecycle policy to delete the shared assets after a configurable number of days. In addition, the actions on assets in the `MediaExchange` S3 bucket are tracked by access logs that are delivered to the `Logs` S3 bucket, which is made available to the publisher account.

IAM roles

AWS Identity and Access Management (IAM) roles allow customers to assign granular access policies and permissions to services and users in the AWS Cloud. This solution creates a role attached to the Amazon EC2 instance with least privilege, allowing access to specific S3 buckets (when applicable).

Design considerations

We recommended that you deploy this solution in a dedicated AWS account. This recommendation is based on security best practices for transferring assets between businesses. However, for evaluation purposes, it is acceptable to deploy the solution in a shared account.

AWS CloudFormation template

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

media-exchange-on-aws: Use this template to launch the solution and all associated components. The default configuration deploys a Service Catalog portfolio that lets you onboard publishers and subscribers in Media Exchange.

Automated deployment

Before you launch the solution, review the architecture, security, and design considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy: Approximately eight minutes

Prerequisites

Create a new AWS account for production deployment. Don't use a shared account for production work.

Deployment overview

Use the following steps to deploy this solution on AWS. 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. Sign in to Media Exchange on AWS (p. 12)

- Sign in to access the AWS Service Catalog products for this solution.

Step 3. Add publishers and subscribers (p. 12)

- Provision Publisher, Subscriber, and Transfer Agreement Product.
- Review the template parameters, and adjust if necessary.

Step 1. Launch the stack

This automated AWS CloudFormation template deploys Media Exchange on AWS in the AWS Cloud.

> **Note**
> You are responsible for the cost of the AWS services used while running this solution. For more details, visit the Cost (p. 2) section in this guide, and refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console and select the button to launch the `media-exchange-on-aws.yaml` 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 Limits 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
Environment	`dev`	Name of the deployment environment. This lets you isolate your deployment between development, staging, and production.
Owner	`mediaops`	Name of the group responsible for administration and maintenance of Media Exchange on AWS.
Owner Email	`mediaops@mycompany.com`	An email address for the maintainer group.

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

Step 2. Sign in to Media Exchange on AWS

After launching the solution, obtain the console URL to sign in.

1. Sign in to the AWS CloudFormation console.
2. Select this solution's installation stack.
3. Choose the **Outputs** tab and record the value for **ConsoleURL**.
4. Select the link and choose **Switch Role**.

The AWS Service Catalog page lists three products for this solution: Transfer agreement, Publisher, and Subscriber.

Step 3. Add publishers and subscribers

After launching the stack and signing in, you are ready to add new publishers and subscribers to the solution.

Provision publisher product

Follow this procedure to onboard a publisher into Media Exchange on AWS.

1. Go to the solution's AWS Service Catalog page. For details, refer to Step 2. Sign in to Media Exchange on AWS (p. 12).
2. Select **Publisher** and then choose **Launch product** to onboard an account that can share assets through the MediaExchange S3 bucket.
3. Enter a product name or select the **Generate name** box.
4. Under **Product versions**, select **latest**.
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
Environment	`dev`	Specify a name for the deployment environment. This lets you isolate your deployment between development, staging, and production.
Publisher Name	*<Requires input>*	Enter the name of the content publisher. Must be alphanumeric, between 3 and 10 characters.
Publisher Account ID	*<Requires input>*	Enter the AWS account ID of the publisher account. Refer to Finding your AWS account ID for details.
Publisher Role		(Optional) Enter the subscriber's IAM role for automated testing.

6. (Optional) Add tags to resources by specifying them as key-value pairs.
7. Choose **Launch product**.

You should receive a **Succeeded** status in approximately five minutes.

Provision subscriber product

Follow this procedure to on-board a subscriber into Media Exchange on AWS.

1. Go to the solution's AWS Service Catalog page. For details, refer to Step 2. Sign in to Media Exchange on AWS (p. 12).
2. Select **Subscriber** and then choose **Launch product** to onboard an account that can receive assets.
3. Select **Subscriber** and then choose **Launch product**.
4. Enter a product name or select the **Generate name** box.
5. Under **Product versions**, select **latest**.
6. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Environment	`dev`	A name of the deployment environment. This lets you isolate your deployment between development, staging and production.
Subscriber Name	`<Requires input>`	Enter the name of the content subscriber. Must be alphanumeric, between 3 and 10 characters.
Subscriber Account ID	`<Requires input>`	Enter the AWS account ID of the subscriber account. Refer to Finding your AWS account ID for details.
Canonical ID	`<Requires input>`	Enter the Canonical user ID of the subscriber account. Refer to Finding your canonical user ID for details.
Email	`<Requires input>`	Enter the subscriber email address that should receive transfer events.
Subscriber Role		(Optional) Enter the subscriber's IAM role for automated testing.

7. (Optional) add tags to resources by specifying them as key-value pairs.
8. Choose **Launch product**.

   You should receive a **Succeeded** status in approximately five minutes.

Provision transfer agreement

Follow this procedure to set up the transfer agreement between a publisher and subscriber.

1. Go to the solution's AWS Service Catalog page. For details, refer to Step 2. Sign in to Media Exchange on AWS (p. 12).
2. Select **Transfer agreement** and then choose **Launch product** to onboard an account that can receive assets through MediaExchange.
3. Enter a product name or select the **Generate name** box.
4. Under **Product versions**, select **latest**.
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
Environment	`dev`	Specify a name for the deployment Environment. This lets you isolate your

Parameter	Default	Description
		deployment between development, staging and production.
Publisher Name	*<Requires input>*	Enter a name for the content publisher specified during provisioning. Must be alphanumeric, between 3 and 10 characters. For details, refer to provision publisher product (p. 13).
Subscriber Name	*<Requires input>*	Enter a name for the content subscriber specified during provisioning. Must be alphanumeric, between 3 and 10 characters. For details, refer to provision subscriber product (p. 13).
Expiration in Days	`5`	Specifies the timeframe (between 1 and 30 days) when assets are automatically removed from the `MediaExchange` Amazon S3 bucket after expiration.
Email Notifications	`no`	Select `yes` to forward all notifications to subscriber's email. If `no` (default), the notifications are available via Amazon EventBridge and Amazon SNS.
Deactivate S3 Events	`no`	Select `yes` to turn off S3 notifications. If you select `yes`, only the notifications originating from publisher are processed. If the email notifications are turned off, they are only available via Amazon EventBridge and Amazon SNS.

6. (Optional) add tags to resources by specifying them as key-value pairs.
7. Choose **Launch product**.

   You should receive a **Succeeded** status in approximately four minutes.

Additional resources

AWS services	
• Amazon Simple Storage Service	• AWS Key Management Service
• Amazon EventBridge	• AWS Service Catalog
• Amazon Simple Notification Service	• AWS CloudFormation
• Amazon Simple Queue Service	• AWS Lambda
• AWS CloudTrail	• AWS Identity and Access Management
• Amazon Elastic Compute Cloud	• AWS Batch

Update the stack

If you previosly deployed the solution, complete the following steps to update your AWS CloudFormation stack to the current version.

1. Sign in to the AWS Cloudformation console, select your existing MediaExchangeOnAWS CloudFormation stack, and choose **Update**.
2. Select **Replace current template**.
3. Under **Specify template**:
   a. Select **Amazon S3 URL**.
   b. Copy the link of the latest template.
   c. Paste the link in the **Amazon S3 URL** box.
   d. Verify that the correct template URL shows in the **Amazon S3 URL** text box, and choose **Next**. Choose **Next** again.
4. Under **Parameters**, review the parameters for the template and modify them as necessary. Refer to Step 1. Launch the Stack (p. 11) for details about the parameters.
5. Choose **Next**.
6. On the **Configure stack options** page, choose **Next**.
7. On the **Review** page, review and confirm the settings. Be sure to check the box acknowledging that the template might create AWS Identity and Access Management (IAM) resources.
8. Choose **View change set** and verify the changes.
9. Choose **Update 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 status of **UPDATE_COMPLETE** in approximately 15 minutes.

The updated CloudFormation stack deploys new versions of the provisioned products in Service Catalog, but does not automatically update the previously launched products. For instructions, refer to Update the provisioned products (p. 18). All new product launches after an update to the product's template get the newer version of the product by default.

Update the provisioned products

Follow these instructions after updating the stack to update the previously deployed products.

1. Sign in to the AWS Service Catalog Console using the **ConsoleURL** link from the updated CloudFormation stack. For details, refer to Step 2. Sign in to Media Exchange on AWS (p. 12).
2. Navigate to **Provisioned Products**.
3. Select one of the provisioned MediaExchangeOnAWS products. Go to **Actions** and then choose **Update**.
4. Under **Product Versions**, select **latest**.
5. Under **Parameters**, review the parameters for the template and modify them as necessary.
6. Choose **Update**.

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

Uninstall the solution

You can uninstall this solution from the AWS Management Console, or by using the AWS Command Line Interface (AWS CLI). You must manually delete the Amazon Simple Storage Service (Amazon S3) buckets created by this solution.

Using the AWS Management Console

1. Sign in to the solution's Service Catalog products. For details, refer to Step 2. Sign in to Media Exchange on AWS (p. 12).
2. From the left navigation pane select **Provisioned products**.
3. Select one of the provisioned Transfer agreement products, select **Actions**, and then choose **Terminate** to de-provision the product.
4. In the confirmation text box enter `terminate` and then choose **Terminate provisioned product**.
5. Repeat steps 2 through 4 to terminate all the Transfer agreement products.
6. From the left navigation pane select **Provisioned products**.
7. Select one of the provisioned Subscriber products, select **Actions**, and then choose **Terminate** to de-provision the product.
8. In the confirmation text box enter `terminate` and then choose **Terminate provisioned product**.
9. Repeat steps 6 through 8 to terminate all the **Subscriber** products.
10. From the left navigation pane select **Provisioned products**.
11. Select one of the provisioned Publisher products, select **Actions**, and then choose **Terminate** to de-provision the product.
12. In the confirmation text box enter `terminate` and then choose **Terminate provisioned product**.
13. Repeat steps 10 through 12 to terminate all the Publisher products.
14. Navigate to the AWS CloudFormation console.
15. Select the `mediaexchange-servicecatalog` stack.
16. Choose **Delete**.

Deleting the Amazon S3 buckets

After uninstalling the solution, manually delete the Amazon S3 buckets.

1. Sign in to the Amazon S3 console.
2. From the left navigation pane choose **Buckets**.
3. Locate all the S3 buckets tagged with the `createdby` key containing the `media-exchange-on-aws/1.0.0` value.
4. Select the S3 buckets and choose **Delete**.

To delete the S3 buckets using AWS CLI, run the following command:

```
$ aws s3 rb s3://<bucket-name> --force
```

Using AWS Command Line Interface

Visit our GitHub repository for instructions on how to uninstall and clean up the installation using the AWS CLI.

Source code

Visit our GitHub repository to download the source files for this solution and to share your customizations with others. Refer to the README file for additional information.

Revisions

Date	Change
June 2021	Initial release
July 2021	Release v1.0.1: bug fixes. For more information, refer to the CHANGELOG.md file in the GitHub repository.

Contributors

- Provanshu Dey
- Leah Siddall

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 AWS current 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. AWS responsibilities and liabilities 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.

Media Exchange on AWS is licensed under the terms of the of the Apache License Version 2.0 available at The Apache Software Foundation.