Implementation Guide

Media Exchange on AWS

Media Exchange on AWS: Implementation Guide

Table of Contents

Solution overview

Publication date: June 2021 (last update: October 2023)

This implementation guide provides an overview of the Media Exchange on AWS solution, its reference architecture and components, considerations for planning the deployment, configuration steps for deploying the Media Exchange on AWS solution to the Amazon Web Services (AWS) Cloud.

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 AWS Snow 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.

This guide is intended for solution architects, business decision makers, DevOps engineers, data scientists, and cloud professionals who want to implement Media Exchange on AWS in their environment.

Use this navigation table to quickly find answers to these questions:

If you want to . . .	Read . . .
Know the cost for running this solution. The cost for using this solution with the default settings in the US East (N. Virginia) is approximately $84.10 for a 10 TB transfer.	Cost
Understand the security considerations for this solution.	Security
Know how to plan for quotas for this solution.	Quotas
Know which AWS Regions are supported for this solution. This solution uses the Service Catalog which isn’t available in all regions	Supported AWS Regions
View or download the AWS CloudForm ation template included in this solution to automatically deploy the infrastructure resources (the “stack”) for this solution.	AWS CloudFormation template

Features and benefits

The Media Exchange on AWS solution provides the following features:

Secure asset transfer

Publishers and subscribers do not share credentials. Assets are encrypted at rest and in transit.

Reduced data transfer costs

There are no per GB data egress and transfer costs within the same AWS Region and no servers to manage or licensing fees.

Mitigate quality control issues

You can get up to 100GB/s in transfer speeds in the same Region because assets never leave the S3 data plane. In addition, S3 conducts asset integrity checks before, during, and after transfer.

Asset tracking and delivery receipts

All operations on assets are tracked via Amazon S3 server access logs. Publishers and subscribers receive automatic notifications.

Integration with Service Catalog AppRegistry and AWS Systems Manager Application Manager

This solution includes a Service Catalog AppRegistry resource to register the solution's CloudFormation template and its underlying resources as an application in both Service Catalog AppRegistry and AWS Systems Manager Application Manager. With this integration, you can centrally manage the solution's resources.

Use cases

Media content exchange

Media and Entertainment companies need a fast, secure, and cost-effective way to transfer files internally and to partners that also use AWS services. Rather than using costly and labor-intensive file transfer protocols, they can use Media Exchange on AWS which is a cloud-based media file sharing solution. The solution virtually eliminate the fees associated with removing data from storage, mitigate the associated labor resources, and reduce the transfer time for multiterabyte files from weeks to hours.

Concepts and definitions

This section describes key concepts and defines terminology specific to this solution:

Publisher

The AWS account being used to transmit assets to other AWS Accounts

Subscriber

An AWS account that receives assets

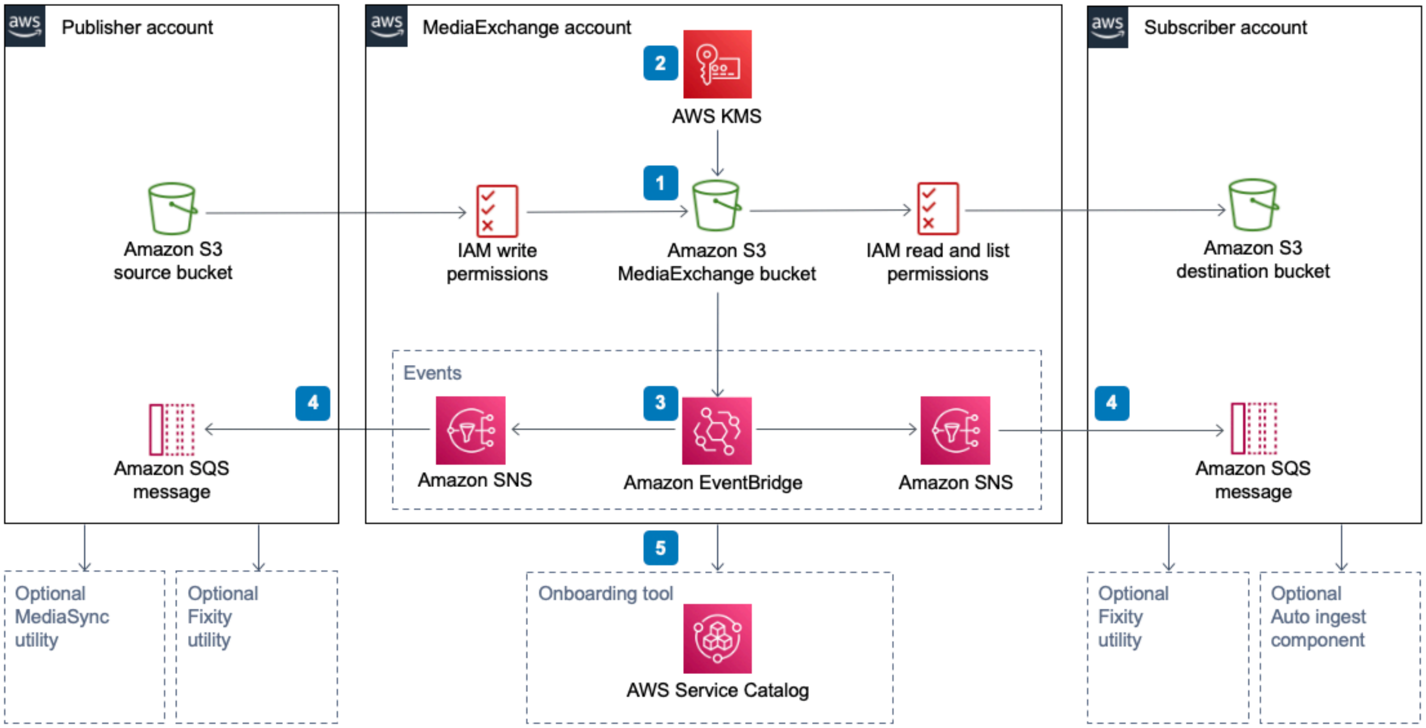
MediaExchange account

The central account that acts as a middleman between publishers and subscribers

Checksum

Used to verify assets have been delivered with 100% accuracy

Architecture overview



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. An Amazon EventBridge bus to receive notifications from Amazon S3.
4. Amazon SNS topics and Amazon Simple Queue Service (Amazon SQS) messages for subscriber and publisher notifications.
5. A set of products in Service Catalog to onboard publishers and subscribers, and to set up new transfer agreements.

AWS Well-Architected design considerations

This solution was designed with best practices from the AWS Well-Architected Framework which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how the design principles and best practices of the Well-Architected Framework were applied when building this solution.

Operational excellence

This section describes how the principles and best practices of the operational excellence pillar were applied when designing this solution.

The Media Exchange on AWS solution tracks all assets via AWS CloudTrail and S3 logs to provide observability into the infrastructure; Amazon S3 buckets and the rest of the solution components.

Security

This section describes how the principles and best practices of the security pillar were applied when designing this solution.

The assets are encrypted by default at rest and in transit, and Media Exchange on AWS uses AWS KMS to store a customer managed key (CMK) that has been established with specific account level permissions. The MediaExchange S3 bucket is configured with a lifecycle policy to delete the shared files after a configurable number of days.

Additionally, Media Exchange on AWS uses AWS Identity and Access Management (IAM) roles to allow customers to assign granular access policies and permissions to services and users.

Reliability

This section describes how the principles and best practices of the reliability pillar were applied when designing this solution.

Media Exchange on AWS uses AWS serverless services wherever possible (for example, Amazon S3, Amazon EventBridge) to ensure high availability and quick recovery from service failure.

Media Exchange on AWS uses constant verification checks throughout the process to ensure that data has been transferred correctly, including use of the fixity tool.

Performance efficiency

This section describes how the principles and best practices of the performance efficiency pillar were applied when designing this solution.

Media Exchange on AWS, as mentioned earlier, uses serverless architecture throughout the solution.

Media Exchange on AWS can be launched in any region that supports the AWS services used in the solution such as: Amazon S3, Amazon EventBridge, AWS KMS, Amazon SQS and Amazon SNS. It supports up to 100 GB/s in transfer speeds in the same region to provide users with the best experience.

The solution is automatically tested and reviewed by solutions architects and subject matter experts for areas to experiment and improve.

Cost optimization

This section describes how the principles and best practices of the cost optimization pillar were applied when designing this solution.

Media Exchange on AWS uses a serverless architecture, therefore customers only get charged for what they use. Additionally, there are no per GB data egress and transfer costs within the same AWS Region.

Customers can measure the efficiency of the workloads, and the costs associated with delivery, by using Application Manager.

Sustainability

This section describes how the principles and best practices of the sustainability pillar were applied when designing this solution.

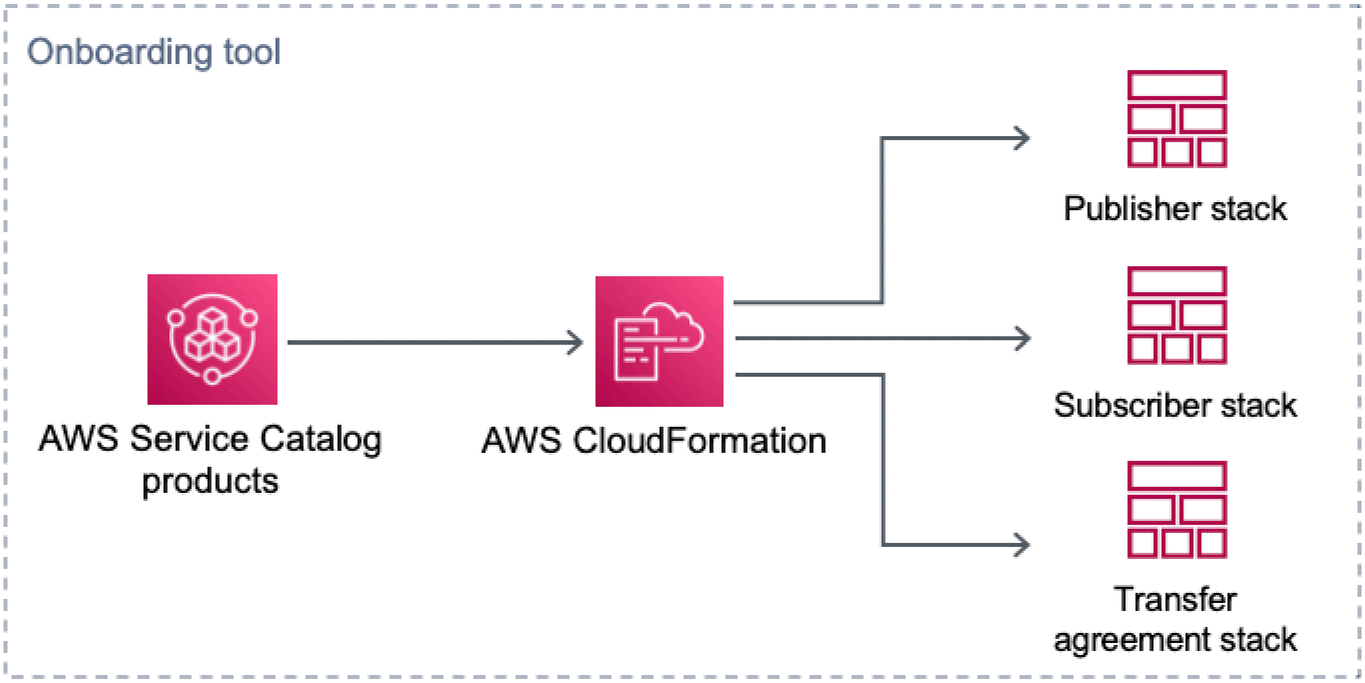
Media Exchange on AWS uses managed and serverless services, to minimize the environmental impact of the backend services. The serverless design of Media Exchange on AWS is aimed at reducing carbon footprint compared to the footprint of continually operating on-premises servers.

Architecture details

This section describes the components and AWS services that make up this solution and the architecture details on how these components work together.

Onboarding tool

When you deploy the Media Exchange on AWS solution, it adds deployable products on Service Catalog. 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.

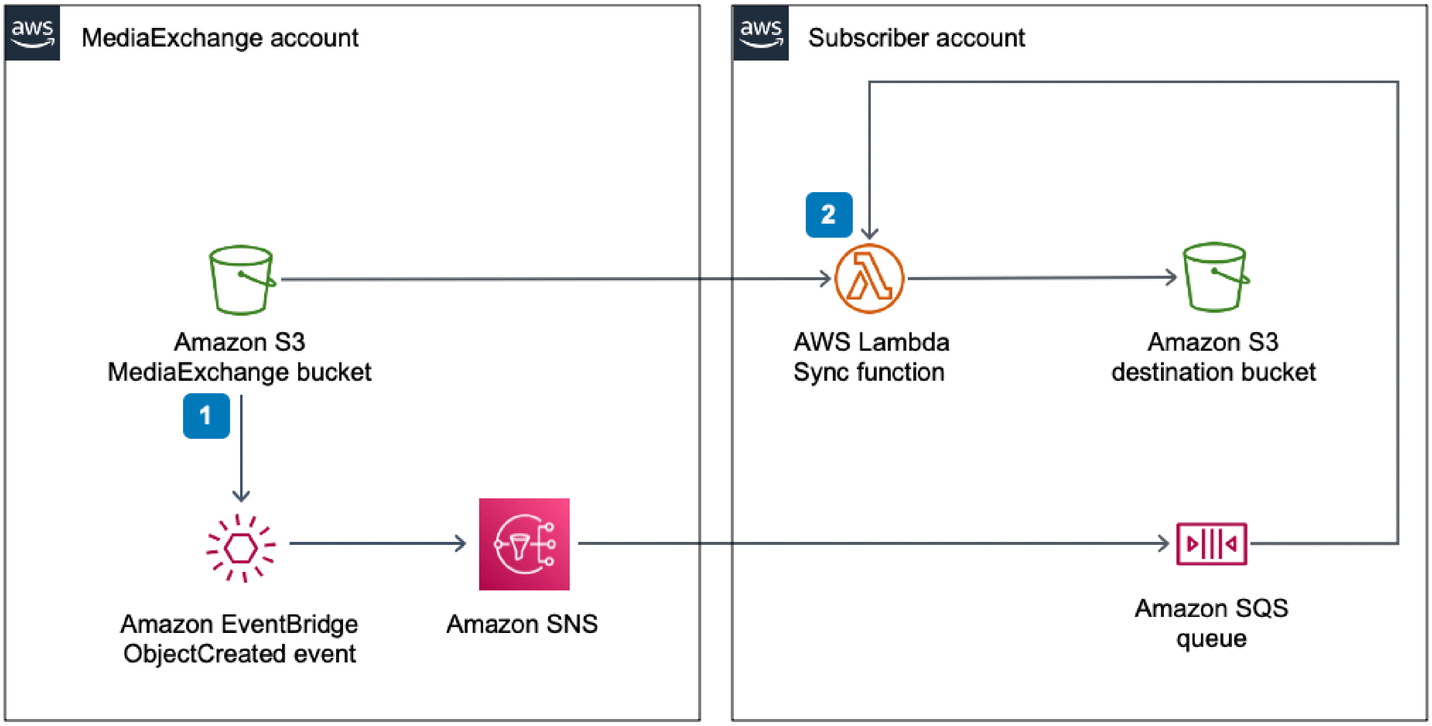


Onboarding tool architecture

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.



Auto ingest component workflow

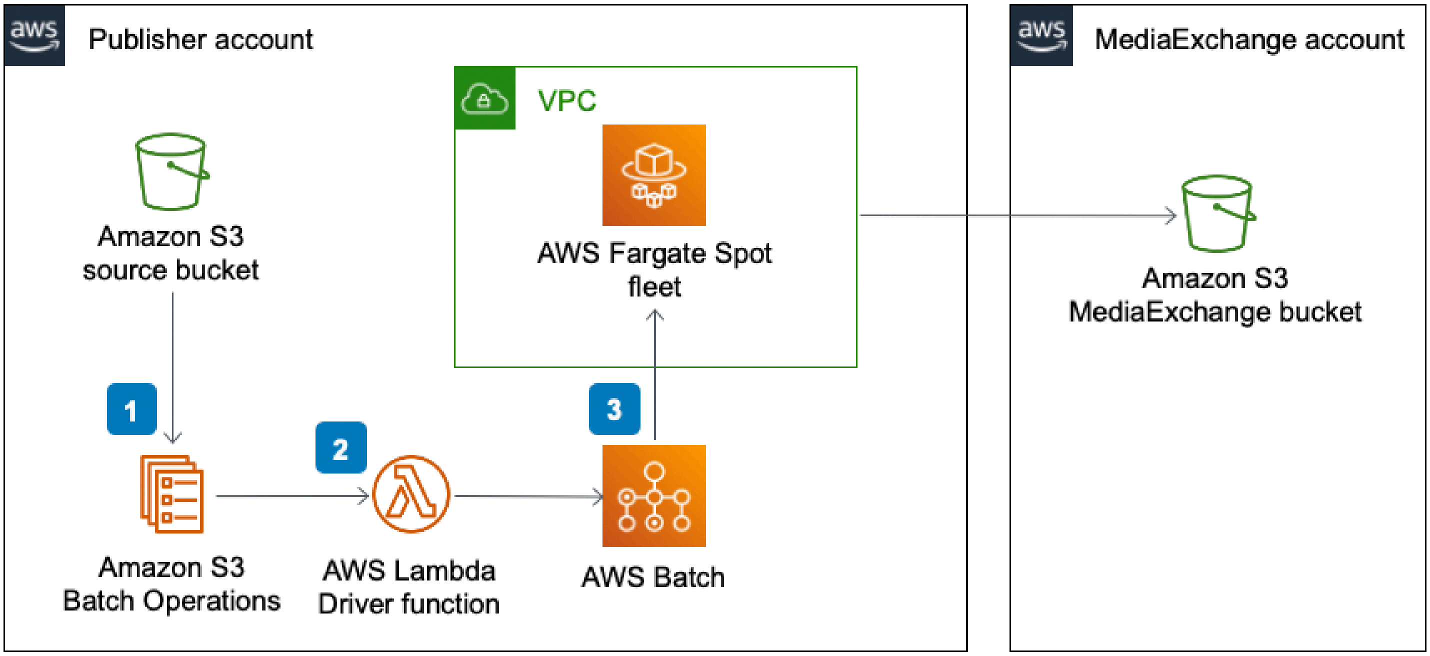
1. When assets are added to the `MediaExchange` S3 bucket, `ObjectCreated` Amazon EventBridge events are sent through Amazon SNS to an Amazon Simple Queue Service (Amazon SQS) queue.
2. The `Sync` Lambda function copies the object from the `MediaExchange` S3 bucket to the destination S3 bucket.

> **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 Lambda function options in the *AWS Lambda Developer Guide*.

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.

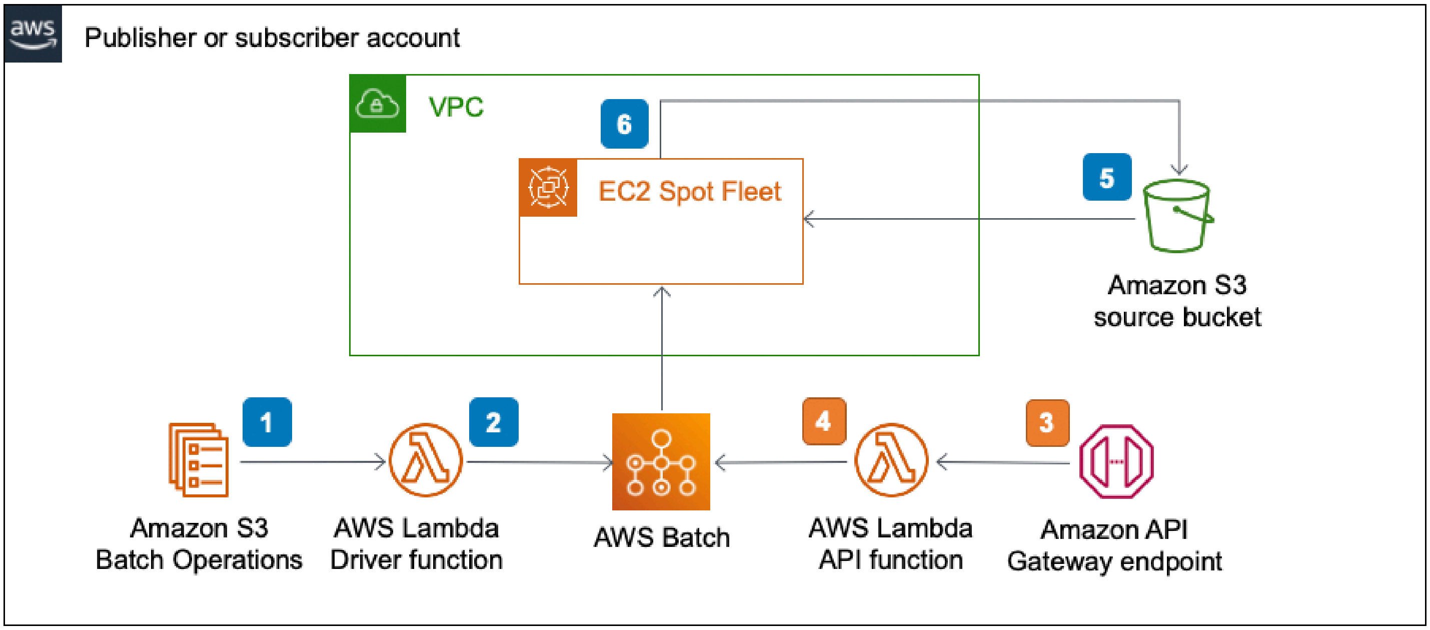


MediaSync utility workflow

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.



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.

AWS services in this solution

AWS service	Description
Amazon S3	**Core.** S3 is where data is stored in Publisher, Subscriber, and MediaExchange accounts
AWS IAM	**Core.** IAM is used to give appropriate cross-account permissions to facilitate data transfer.
AWS KMS	**Core.** Allows encryption configuration of data.
AWS CloudFormation	**Core.** Used to deploy this solution.
AWS Service Catalog	**Core.** Allows onboarding of publishers / subscribers and the setup of new transfer agreements.
Amazon EventBridge	**Supporting.** Receives notifications from AWS S3 .
Amazon SNS	**Supporting.** Used to send notifications to publishers and subscribers.
Amazon SQS	**Supporting.** Used to send notifications to publishers and subscribers.

AWS service	Description
AWS Systems Manager	**Supporting.** Provides application-level resource monitoring and visualization of resource operations and cost data.
AWS Fargate	**Optional.** The compute environment for MediaSync utility.
AWS Lambda	**Optional.** Used to copy data between buckets and run validation checks.
AWS EC2	**Optional.** Used by Fixity utility to compute checksums.
AWS Batch	**Optional.** Used to invoke Lambda validations and trigger Fargate instances.
AWS API Gateway	**Optional.** Used to invoke Lambda validations.

Plan your deployment

This section describes the Cost, Security, Regions, and other considerations prior to deploying the solution.

Cost

The owner of the MediaExchange account is responsible for the cost of the AWS services used while running this solution. As of this revision, the cost for using this solution with the default settings in the US East (N. Virginia) is approximately **$84.10 for a 10 TB transfer**.

AWS service	Cost for 10 TB transfer [USD]
Amazon S3	$79.55
AWS KMS	$4.05
Amazon SNS	$0.50
Total for 10 TB transfer:	**$84.10**

⚠ **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.

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 KMS to store a customer managed 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 `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

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).

Security groups

The security groups created in this solution are designed to control and isolate network traffic between the Lambda functions, CSR instances, and remote VPN endpoints. We recommend that you review the security groups and further restrict access as needed once the deployment is up and running.

Supported AWS Regions

This solution uses the Service Catalog service, which is not currently available in all AWS Regions. You must launch this solution in an AWS Region where Service Catalog is available. For the most current availability of AWS services by Region, see the AWS Regional Services List.

Media Exchange on AWS is supported in the following AWS Regions:

Region name	
US East (Ohio)	Asia Pacific (Tokyo)
US East (N. Virginia)	Canada (Central)
US West (Northern California)	Europe (Frankfurt)
US West (Oregon)	Europe (Ireland)
Africa (Cape Town)	Europe (London)
Asia Pacific (Hong Kong)	Europe (Milan)
Asia Pacific (Jakarta)	Europe (Paris)
Asia Pacific (Mumbai)	Europe (Stockholm)
Asia Pacific (Osaka)	Middle East (Bahrain)
Asia Pacific (Seoul)	South America (São Paulo)
Asia Pacific (Singapore)	AWS GovCloud (US-East)
Asia Pacific (Sydney)	AWS GovCloud (US-West)

Quotas

Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account.

Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the services implemented in this solution. For more information, see AWS service quotas.

Use the following links to go to the page for that service. To view the service quotas for all AWS services in the documentation without switching pages, view the information in the Service endpoints and quotas page in the PDF instead.

AWS CloudFormation quotas

Your AWS account has AWS CloudFormation quotas that you should be aware of when launching the stack in this solution. By understanding these quotas, you can avoid limitation errors that would prevent you from deploying this solution successfully. For more information, see AWS CloudFormation quotas in the in the *AWS CloudFormation User's Guide*.

Deploy the solution

This solution uses AWS CloudFormation templates and stacks to automate its deployment. The CloudFormation template to describes the AWS resources included in this solution and their properties. The CloudFormation stack provisions the resources that are described in the template.

Prerequisites

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

> ⚠ **Important**
>
> This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered though this survey. Data collection is subject to the Privacy Notice.
> To opt out of this feature, download the template, modify the AWS CloudFormation mapping section, and then use the AWS CloudFormation console to upload your updated template and deploy the solution. For more information, see the Anonymized data collection section of this guide.

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:

View template

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 on AWS.

> **Note**
>
> If you have previously deployed this solution, see Update the solution for update instructions.

Before you launch the solution, review the cost, architecture, network security, and other considerations discussed earlier in this guide.

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 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.

   Launch solution

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 AWS STS quotas, name requirements, and character 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 deploymen t between development, staging, and production.
Owner	`mediaops`	Name of the group responsib le 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.

Post-deployment tasks

After the stack has successfully deployed, complete these post-deployment tasks.

Post-deployment overview

Use the following steps to deploy to add publishers and subscribers. For detailed instructions, follow the links for each step.

Step 1. Sign in to Media Exchange on AWS

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

Step 2. Add publishers and subscribers

- Provision **Publisher**, **Subscriber**, and **Transfer** agreement product.
- Review the template parameters, and adjust if necessary.

Step 1. 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 2. 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 Sign in to Media Exchange on AWS.
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 Your AWS account ID and its alias 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 Service Catalog page. For details, refer to Sign in to Media Exchange on AWS.
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 deploymen t environment. This lets you isolate your deploymen t 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

Parameter	Default	Description
		to Your AWS account ID and its alias 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 Service Catalog page. For details, refer to Sign in to Media Exchange on AWS.
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

Parameter	Default	Description
		your 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.
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.
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.

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.

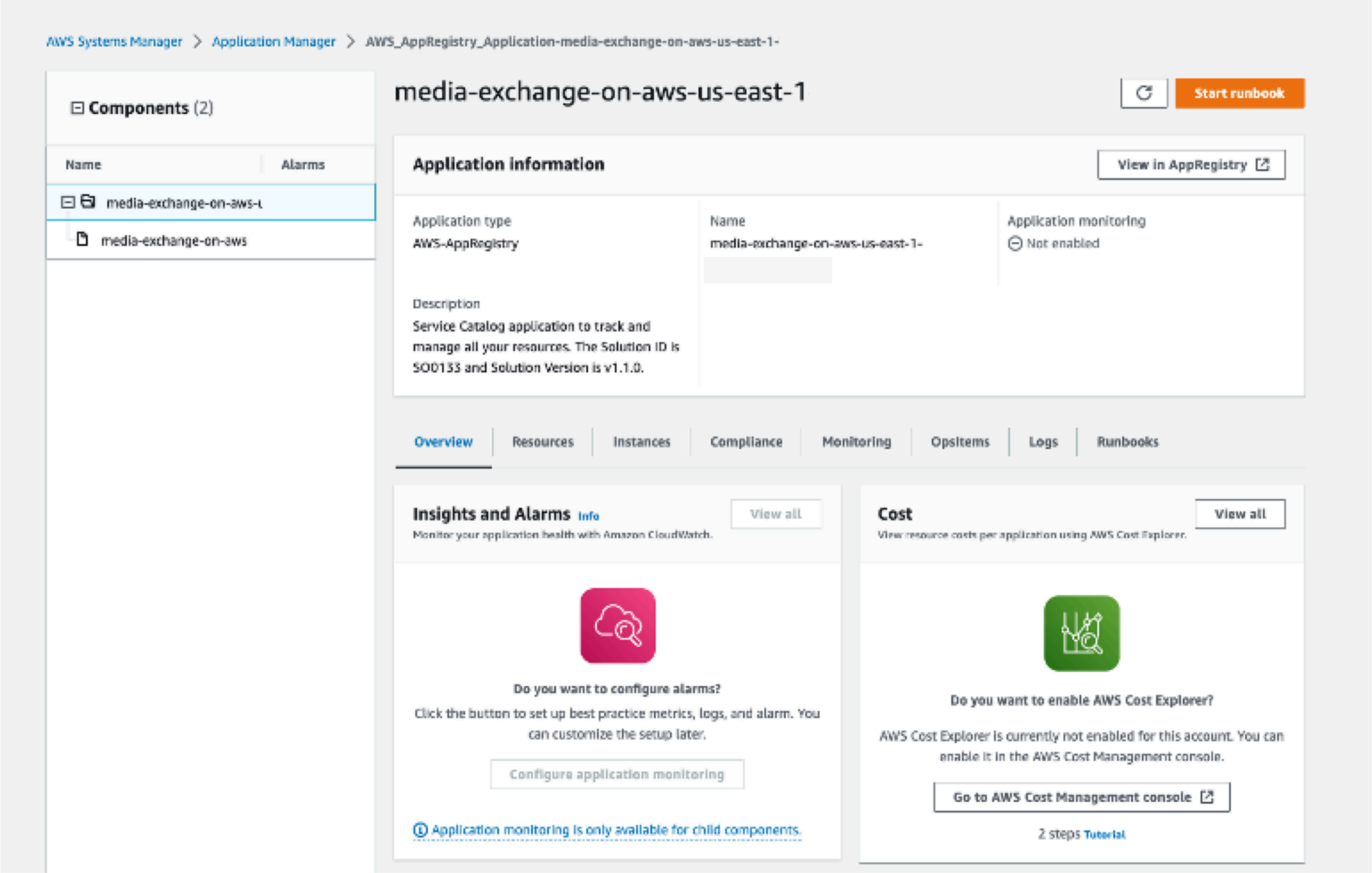
Monitoring the solution with Service Catalog AppRegistry

The Media Exchange on AWS solution includes a Service Catalog AppRegistry resource to register the CloudFormation template and underlying resources as an application in both Service Catalog AppRegistry and AWS Systems Manager Application Manager.

AWS Systems Manager Application Manager gives you an application-level view into this solution and its resources so that you can:

- Monitor its resources, costs for the deployed resources across stacks and AWS accounts, and logs associated with this solution from a central location.
- View operations data for the resources of this solution in the context of an application. For example, deployment status, CloudWatch alarms, resource configurations, and operational issues.

The following figure depicts an example of the application view for the Media Exchange on AWS stack in Application Manager.



media-exchange-on-aws-us-east-1

Application information

Application type
AWS-AppRegistry

Name
media-exchange-on-aws-us-east-1-

Application monitoring
⊖ Not enabled

Description
Service Catalog application to track and manage all your resources. The Solution ID is SO0133 and Solution Version is v1.1.0.

Insights and Alarms Info
Monitor your application health with Amazon CloudWatch.

Do you want to configure alarms?
Click the button to set up best practice metrics, logs, and alarm. You can customize the setup later.

ⓘ Application monitoring is only available for child components.

Cost
View resource costs per application using AWS Cost Explorer.

Do you want to enable AWS Cost Explorer?
AWS Cost Explorer is currently not enabled for this account. You can enable it in the AWS Cost Management console.

Media Exchange on AWS stack in Application Manager

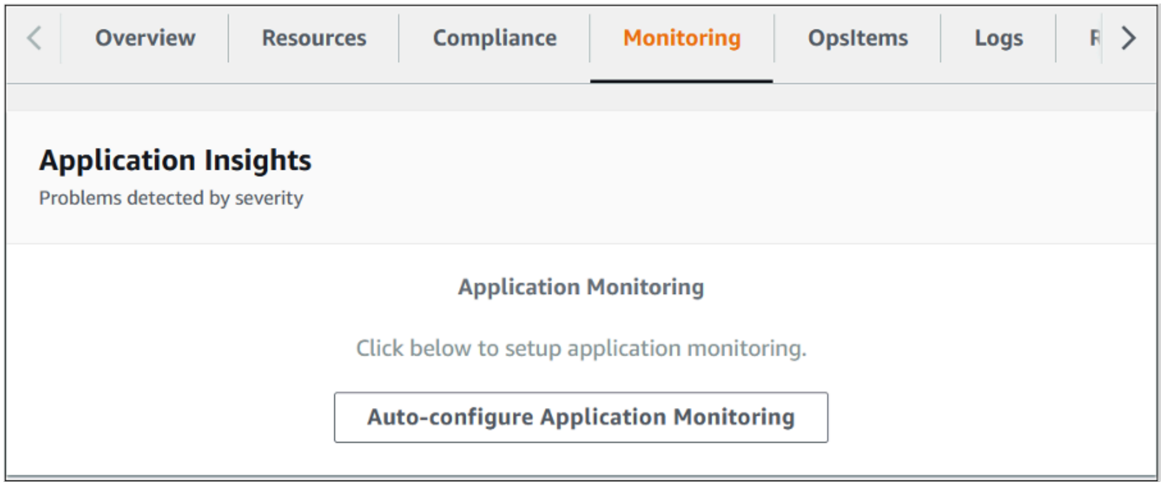
> **Note**
>
> You must activate CloudWatch Application Insights, AWS Cost Explorer, and cost allocation tags associated with this solution. They are not activated by default.

Activate CloudWatch Application Insights

1. Sign in to the Systems Manager console.
2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose **AppRegistry applications**.
4. In **AppRegistry applications**, search for the application name for this solution and select it.

The next time you open Application Manager, you can find the new application for your solution in the **AppRegistry application** category.

5. In the **Components** tree, choose the application stack you want to activate.
6. In the **Monitoring** tab, in **Application Insights**, select **Auto-configure Application Monitoring**.



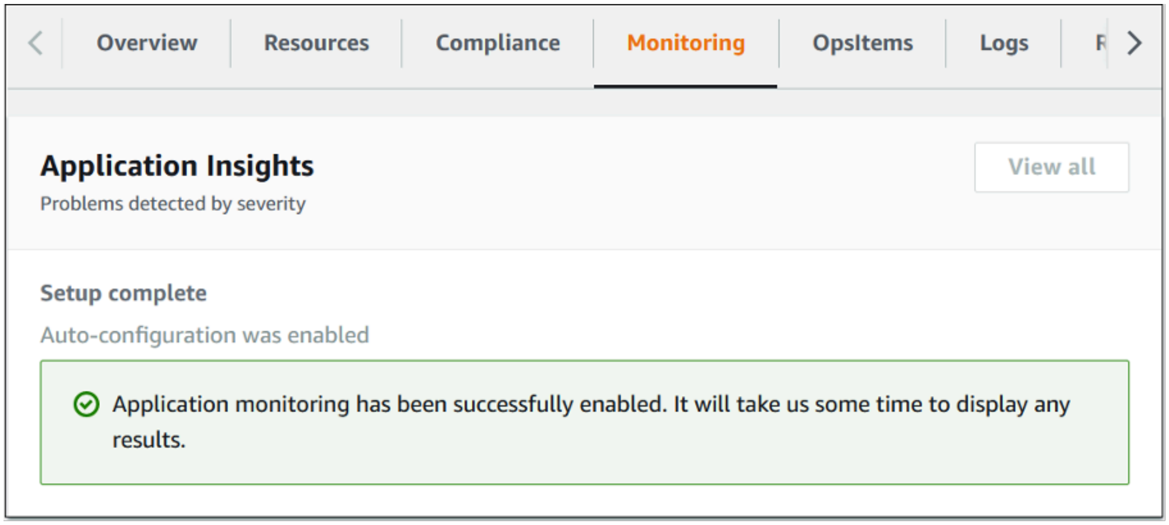
Application Insights

Problems detected by severity

Application Monitoring

Click below to setup application monitoring.

Monitoring for your applications is now activated and the following status box appears:



Application Insights

Problems detected by severity

Setup complete

Auto-configuration was enabled

Application monitoring has been successfully enabled. It will take us some time to display any results.

Activate AWS Cost Explorer

You can see the overview of the costs associated with the application and application components within the Application Manager console through integration with AWS Cost Explorer which must be first activated. Cost Explorer helps you manage costs by providing a view of your AWS resource costs and usage over time. To activate Cost Explorer for the solution:

1. Sign in to the AWS Cost Management console.
2. In the navigation pane, select **Cost Explorer**.
3. On the **Welcome to Cost Explorer** page, choose **Launch Cost Explorer**.

The activation process can take up to 24 hours to complete. Once activated, you can open the Cost Explorer user interface to further analyze cost data for the solution.

Activate cost allocation tags associated with the solution

After you activate Cost Explorer, you must activate the cost allocation tags associated with this solution to see the costs for this solution. The cost allocation tags can only be activated from the management account for the organization. To activate cost allocation tags:

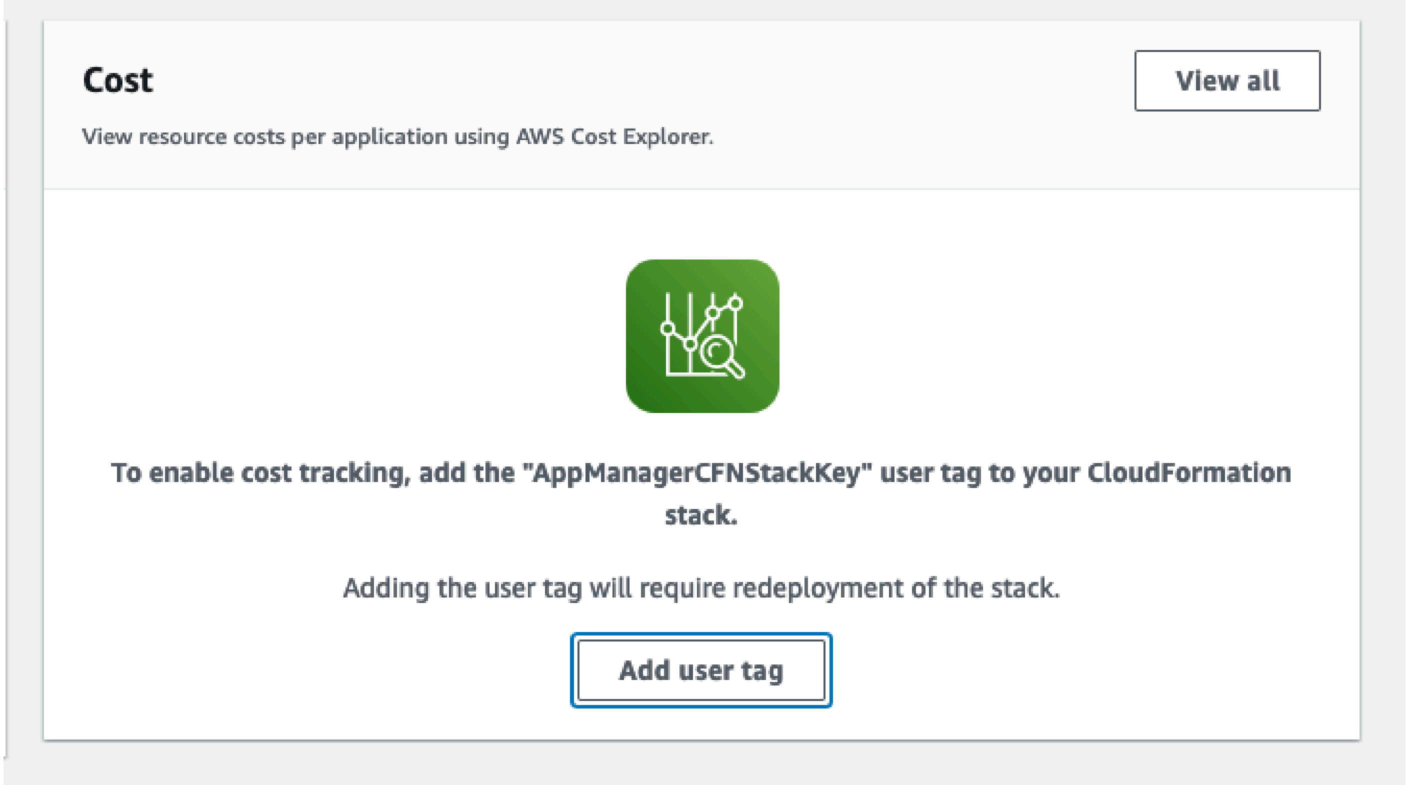
1. Sign in to the AWS Billing and Cost Managementconsole.
2. In the navigation pane, select **Cost Allocation Tags**.
3. On the **Cost allocation tags** page, filter for the `AppManagerCFNStackKey` tag, then select the tag from the results shown.
4. Choose **Activate**.

The activation process can take up to 24 hours to complete and the tag data to appear.

Confirm cost tags associated with the solution

After you activate cost allocation tags associated with the solution, you must confirm the cost allocation tags to see the costs for this solution. To confirm cost allocation tags:

1. Sign in to the Systems Manager console.
2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose the application name for this solution and select it.
4. In the **Overview** tab, in **Cost**, select **Add user tag**.



Cost

View resource costs per application using AWS Cost Explorer.

To enable cost tracking, add the "AppManagerCFNStackKey" user tag to your CloudFormation stack.

Adding the user tag will require redeployment of the stack.

5. On the **Add user tag** page, enter `confirm`, then select **Add user tag**.

The activation process can take up to 24 hours to complete and the tag data to appear.

Update 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 Launch the stack 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 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. 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 Service Catalog Console using the **ConsoleURL** link from the updated CloudFormation stack. For details, refer to Sign in to Media Exchange on AWS.
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 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 Sign in to Media Exchange on AWS.
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.

Developer guide

This section provides the source code for the solution.

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.md file for additional information.

Reference

This section includes information about an optional feature for collecting unique metrics for this solution, pointers to related resources, and a list of builders who contributed to this solution.

Anonymized data collection

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

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

AWS owns the data gathered though this survey. Data collection is subject to the Privacy Notice. To opt out of this feature, complete the following steps before launching the AWS CloudFormation template.

1. Download the AWS CloudFormation template to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

   ```
   AnonymizedData:
   SendAnonymizedData:
   Data: Yes
   ```

   to:

   ```
   AnonymizedData:
   SendAnonymizedData:
   Data: No
   ```

4. Sign in to the AWS CloudFormation console.
5. Select Create stack.

6. On the Create stack page, Specify template section, select Upload a template file.
7. Under **Upload a template file**, choose **Choose file** and select the edited template from your local drive.
8. Choose **Next** and follow the steps in Launch the stack in the Deploy the solution section of this guide.
9. To opt out of operational metrics for the optional Lambda deployments, follow the same process for the corresponding template file (Fixity, Mediasync, Autoingest).

Related resources

- The Media2Cloud Solution is a similar to Media Exchange on AWS but its default implementation is designed for enterprise customers to move assets to the cloud.

Contributors

- Provanshu Dey
- Leah Siddall
- Brandon Dold
- Eric Thoman
- San Dim Ciin

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.
July 2022	Release v1.1.0: Remove Auto ACL utility. For more information, refer to the CHANGELOG.md file in the GitHub repository.
June 2023	Release v1.2.0: Fixed dependency security vulnerability and added integration with Service Catalog AppRegistry and AWS Systems Manager Application Manager. For more information, refer to the CHANGELOG.md file in the GitHub repository.
July 2023	Release v1.2.1: Added operational metrics reporting for main stack deployment, including the ability for customers to opt out of anonymized data collection. Also added the ability for customers to opt out of anonymous data collection of the three optional Lambda deployments. For more information, refer to the CHANGELOG.md file in the GitHub repository.
September 2023	Release v1.2.2: Updated the library and improved unit tests. For more information, refer to the CHANGELOG.md file in the GitHub repository.

Date	Change
October 2023	Release v1.2.3: Updated package versions to resolve security vulnerabilities. For more information, refer to the CHANGELOG.md file in the GitHub repository.

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.