aws

Implementation Guide

Media Services Application Mapper on AWS

Media Services Application Mapper on AWS: Implementation Guide

Table of Contents

Browser application to visualize relationships between your media services

Publication date: *December 2018 (*[*last update*](#)*: October 2023)*

Many Amazon Web Services (AWS) customers use media services to prepare, process, and deliver broadcast and over-the-top (OTT) streaming video from the AWS Cloud. While these services allow you to innovate, test, and deploy video services, it is difficult to quickly understand the relationships among several media services resources that are configured to operate together. Especially when multiple resources are reporting alerts or alarms through [Amazon CloudWatch](#), it can be a challenge to identify the source of the problem.

Media Services Application Mapper (MSAM) on AWS helps you visualize media service relationships and the real-time status of linear video services. This solution features a browser application that visualizes the relationships between your inventory and allows you to display the logical connections between media services. It also provides the status of related alerts and alarms that visualize error conditions in complex workflows to help you quickly navigate to the root cause for problematic workflows. The application features a tile view and a diagram view.

This solution monitors the following AWS Elemental products:

- [AWS Elemental Link](#)
- [AWS Elemental Live](#)
- [AWS Elemental MediaStore](#)
- [AWS Elemental MediaConnect](#)
- [AWS Elemental MediaLive](#)
- [AWS Elemental MediaPackage](#)
- [AWS Elemental MediaTailor](#)

This solution also monitors the following supporting services that are often used with media services: [Amazon CloudFront](#), [Amazon Simple Storage Service](#) (Amazon S3), and [Amazon Elastic Compute Cloud](#) (Amazon EC2) and on-premises hardware configured with the [AWS Systems Manager](#) agent.

This implementation guide describes architectural considerations and configuration steps for deploying the Media Services Application Mapper on AWS in the AWS Cloud. It includes links to an

[AWS CloudFormation](#) template that launches, configures, and runs 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 with media workflows and architecting in the AWS Cloud.

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

If you want to . . .	Read . . .
Know the cost for running this solution The estimated cost for running this solution in the US East (N. Virginia) Region is USD $ 32.00 per month.	[Cost](#)
Understand the security considerations for this solution	[Security](#)
Know which AWS Regions are supported for this solution	[Supported AWS Regions](#)
View or download the AWS CloudFormation template included in this solution to automatically deploy the infrastructure resources (the "stack") for this solution	[AWS CloudFormation templates](#)

Features and benefits

The Media Services Application Mapper (MSAM) on AWS solution provides the following features:

Channel tiles

Aggregate resources into a single channel tile that represents a single streaming video channel. Each tile displays the aggregated media service configuration information for all resources included in the tile.

Workflow diagrams

Visualize the relationships between your media services and the status of your resources and pipelines as workflow diagrams. You can also customize the solution to include custom node types or on-premise equipment.

Customizable

You can customize or extend this solution for your specific needs. For example, add custom node types with the browser application, cache them into a database through a cloud-side task, and extend connection discovery and visualization overlay functionality to on-premise equipment.

Integration with AWS 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 AWS Service Catalog AppRegistry and AWS Systems Manager Application Manager. With this integration, you can centrally manage the solution's resources.

Use cases

Media Processing and quality control

As content moves into and through the media supply chain, the produced content must go through a number of quality and accuracy control reviews. Media Services Application Mapper on AWS automatically provision resources to visualize workflow steps, their relationships, and their real-time status. This solution helps you identify the root cause of a problem when multiple services are used, and maintain a high level of quality each step of the way.

Concepts and definitions

Tile

A logical group of cloud resources that represents a single streaming video channel or other related collection of resources.

Workflow diagram

A graphical representation of cloud resources and the connections between them assembled by the user.

Microservices

A set of AWS Lambda functions that provide the business logic and data access layer for all operations in the solution.

For a general reference of AWS terms, see the [AWS glossary](#) in the AWS General Reference.

Architecture overview

This section provides a reference implementation architecture diagram for the components deployed with this solution.

Architecture diagram

Deploying this solution with the default parameters builds the following configuration of resources in the AWS Cloud.

Media Services Application Mapper architecture on AWS

The AWS CloudFormation template deploys the following serverless infrastructure and web content for Media Services Application Mapper (MSAM) on AWS:

1. An Amazon CloudFront distribution to serve the Media Services Application Mapper on AWS web application.

2. An Amazon Simple Storage Service (Amazon S3) bucket as an origin to serve the solution's web application.

3. Amazon API Gateway for inventory and monitoring queries to the solution's backend.

4. AWS Lambda functions to handle client REST API requests from API Gateway, periodically scan for media services resources, and receive notifications of changes in a resource's status. Two Lambda functions are custom resources for the browser application stack and one for the DynamoDB tables stack that are used during installation, update, and deletion.

5. Amazon CloudWatch scheduled rules or event patterns to invoke the primary Lambda functions.

6. Amazon DynamoDB tables to store the solution's backend and front-end configuration and state.

7. The AWS Cloud services and devices that this solution scans. The solution uses the AWS Elemental MediaLive API to discover AWS Elemental Link devices and the AWS Systems Manager API to discover Elemental Live encoders.

8. CloudWatch events to invoke the regional Lambda functions when there are operational changes in the sources of inventory or changes in state to subscribed CloudWatch alarms.

9. Lambda functions to monitor source services for CloudWatch events in additional AWS Regions.

> ⓘ **Note**
>
> Requests to and responses from the APIs are logged in Amazon CloudWatch. Lambda functions invoked by scheduled events or notification events also send log data to Amazon CloudWatch.

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 we architected this solution using the principles and best practices of the operational excellence pillar.

The Media Services Application Mapper on AWS solution pushes metrics to Amazon CloudWatch at various stages to provide observability into the infrastructure; Lambda functions, AWS S3 buckets, and the rest of the solution components.

Security

This section describes how we architected this solution using the principles and best practices of the security pillar.

All browser actions are performed through an authenticated and SSL encrypted REST API hosted in the cloud. The API can be used by other tools to perform activities such as preloading tile definitions or adding custom content to the cache.

To support multiple users in Media Services Application Mapper on AWS, you can create multiple API keys, each with different names that are provided to separate groups. To share the browser application with a colleague, provide the browser application URL, core endpoint URL, and an API key.

Reliability

This section describes how we architected this solution using the principles and best practices of the reliability pillar.

Media Services Application Mapper on AWS uses AWS serverless services wherever possible (for example: Lambda, API Gateway, S3 , and DynamoDB) to ensure high availability and quick recovery from service failure.

Performance efficiency

This section describes how we architected this solution using the principles and best practices of the performance efficiency pillar.

Media Services Application Mapper on AWS, as mentioned earlier, uses serverless architecture throughout the solution. It can be launched in any region that supports the AWS services used in the solution such as: AWS Lambda, Amazon API Gateway, AWS S3, and the AWS Elemental services that it supports.

This 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 we architected this solution using the principles and best practices of the cost optimization pillar.

Media Services Application Mapper on AWS uses serverless architecture; therefore, customers only get charged for what they use. The total cost of the solution depends on the size of the resource inventory and the monitoring activity.

Sustainability

This section describes how we architected this solution using the principles and best practices of the sustainability pillar.

Media Services Application Mapper on AWS utilizes managed and serverless services to minimize the environmental impact of the backend services. The serverless design using Lambda, API Gateway, S3, and DynamoDB is aimed at reducing carbon footprint compared to the footprint of continually operating on-premise servers.
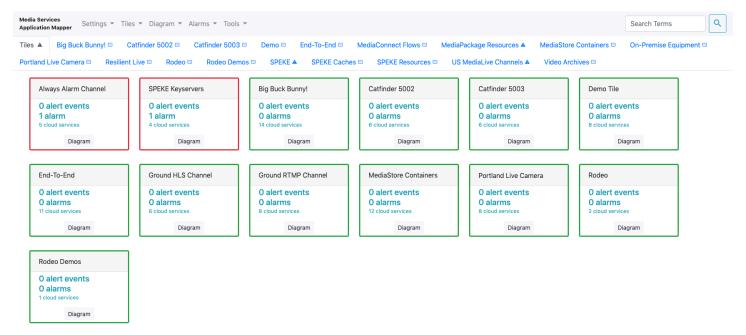
Architecture details

Browser application

The browser application visualizes the relationships between your inventory. Configure your environment with the tile view and diagram view according to your needs.

Tiles

Tiles aggregate several cloud resources into a single visual item that represents a single streaming video channel or other related collection of resources. Multiple square tiles are displayed in a grid and can be filtered by state. You can create single tiles through the browser application or multiple tiles in bulk with the solution's RESTful API.

Example channel tiles

Each tile displays the aggregated media service configuration information for all resources included in that tile. You can also select the tile to view the tile's resources on an existing diagram, or choose to create one on-demand. When you choose the diagram button on the tile, it searches for diagrams containing all the tile's resources, and allows you to select a matching diagram or generate a new diagram with the tile's resources.
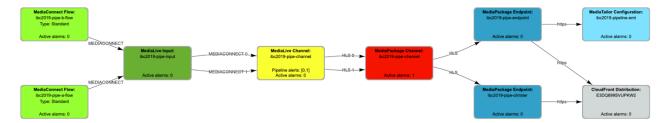
The following image shows the aggregated JSON configuration of all the cloud resources assigned to the tile. This output is shown in the **Selected Item** compartment when the tile is selected with the mouse.

```
Channel: PDX Live on Broadway
-[
  -{
      "Id": "E1AOHZYWIAXZBK",
      "ARN": "arn:aws:cloudfront::        :distribution/E1AOHZYWIAXZBK",
      "Status": "Deployed",
      "LastModifiedTime": "2018-08-31 17:13:32.356000+00:00",
      "DomainName": "dcw3cwpfmf2pb.cloudfront.net",
      "Aliases": +{1 item},
      "Origins": +{2 items},
      "DefaultCacheBehavior": +{12 items},
      "CacheBehaviors": +{1 item},
      "CustomErrorResponses": +{1 item},
      "Comment": "Services Live Stream. NAB2018 MediaPackage.",
      "PriceClass": "PriceClass_All",
      "Enabled": true,
      "ViewerCertificate": +{3 items},
      "Restrictions": +{1 item},
      "WebACLId": "",
      "HttpVersion": "HTTP2",
      "IsIPV6Enabled": true
  },
  -{
      "Arn": "arn:aws:medialive:us-west-2:        :channel:66440",
      "Destinations": +[1 item],
      "EgressEndpoints": +[2 items],
      "Id": "66440",
      "InputAttachments": +[1 item],
      "InputSpecification": +{3 items},
      "Name": "CES Live Camera Demo - End to End Live Page",
      "PipelinesRunningCount": 2,
      "RoleArn": "arn:aws:iam::        :role/AllowMediaLiveAccessRole",
      "State": "RUNNING"
  },
  -{
      "Arn": "arn:aws:medialive:us-west-2:        :input:9172618",
      "AttachedChannels": +[1 item],
      "Destinations": +[2 items],
      "Id": "9172618",
      "Name": "CES Camera End to End Input",
      "SecurityGroups": +[1 item],
      "Sources": [],
```

Tile elements in JSON configuration file

Diagrams

The application visualizes your resources and connections as workflow diagrams that are assembled by the user. Diagrams can be organized in any way that makes sense for your environment and required level of detail. The browser application includes features to navigate, find, and build diagrams of related cloud resources. It visualizes cloud resources as nodes and the logical connections between resources as directed edges.

Example workflow diagram

The application includes two default automatic layout configurations: vertical (top to bottom) and horizontal (left to right). You can also select and move individual items or groups of items to new
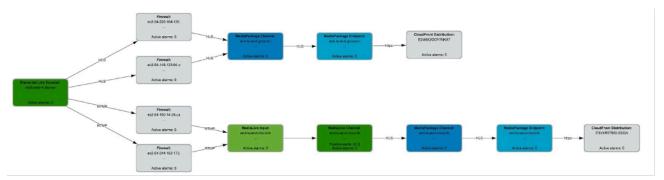
locations to create your own custom layout. When you move items, the layout is saved in the cloud so all users will be able to view the same diagrams at the same endpoint.

Furthermore, the solution's diagram tool features the following:

- Edges (connections) on diagrams generally represent some type of data flow in the direction of the arrowhead.
- Resources with redundant data paths are drawn with multiple, labeled connections.
- Different types of service resources are visualized with unique color and textual indicators.
- Graphical tools available through the menu to quickly build diagrams of related resources.
- The solution provides configuration data for each resource on the diagram in the **Selected Item** compartment at bottom of the page.

The solution is designed to be extended with new node types, connection discovery, visualization overlays, and tools. You can add custom nodes within the browser application directly, or cached into a database through a cloud-side task.

The diagram shows on-premises equipment: AWS Elemental Live encoder and several firewalls with connectivity to cloud resources.

Workflow diagram with on-premise equipment incorporated

The following image shows the JSON configuration of a diagram node when it is selected in the diagram. This is the same configuration you would receive from the response to a List or Describe API call directed to the service.

```
 Status     Search       Selected Item     Monitor

MediaLive Channel: portland-timbers-channel    AWS Console   AWS CloudWatch
Updated: Wed Sep 26 2018 09:50:58 GMT-0700 (Pacific Daylight Time)

-{
    "Arn": "arn:aws:medialive:us-west-2:          :channel:7853890",
    "Destinations": +[1 item],
    "EgressEndpoints": +[2 items],
    "Id": "7853890",
    "InputAttachments": +[1 item],
    "InputSpecification": +{3 items},
    "Name": "portland-timbers-channel",
    "PipelinesRunningCount": 2,
    "RoleArn": "arn:aws:iam::          :role/MediaLiveAccessRole",
    "State": "RUNNING"
}
```

Diagram elements in JSON configuration file

Solution microservices

The Media Services Application Mapper on AWS microservices are a set of AWS Lambda functions
that provide the business logic and data access layer for all solution operations. The browser
application and other custom applications can interact with solution data securely through the
RESTful APIs to perform activities such as creating diagrams and tiles, associating inventory items
with alarms, and receiving alert notifications when a monitored item changes state.

Each Lambda function assumes an AWS Identity and Access Management (IAM) role with
least-privilege access to perform its designated functions. The following sections outline each
microservice.

API Handler microservice

The `APIHandler` Lambda function handles all browser requests and interacts with the solution's
Amazon DynamoDB tables in the browser application. The `APIHandler` Lambda function also
handles all administrative services including layout management and general settings.

Inventory discovery microservice

An Amazon CloudWatch event invokes the `UpdateNodes`, `UpdateSSMNodes`, and
`UpdateFromTags` Lambda functions every five minutes. These functions scan for your media
services and related services resources, and store information about the resources in the Content
DynamoDB table.

An Amazon CloudWatch event invokes the `UpdateConnections` Lambda function every five minutes. The function identifies connections between your media services, and stores any connections it could determine in the `Content` DynamoDB table.

The `UpdateFromTags` Lambda function is also invoked every five minutes and updates diagrams and tiles based on tags present in the discovered inventory.

The `SSMRunCommand` and `ProcessSSMRunCommand` Lambda functions are invoked every five minutes and used to periodically gather state from AWS Systems Manager managed instances.

When a user accesses the browser application, the `APIHandler` Lambda function retrieves the resource and state information from the DynamoDB tables and displays the resources, their connections and state in the browser through diagrams and tiles.

CloudWatch Events microservices
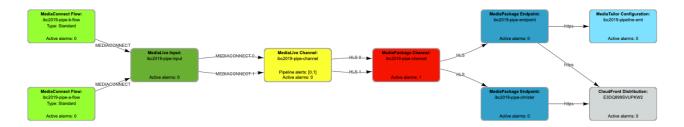
CloudWatch Events rules receive an event when there is a change in your media services or their status, and then route the event to the `Collector` or `AlarmUpdater` Lambda functions. These functions track alert or alarm state for inventory nodes in the DynamoDB tables. When a user accesses the browser application, the `APIHandler` Lambda function retrieves the alert or alarm information from the `CloudWatchEvents`, `Events`, or `Alarms` table and displays it in the browser.

Resource monitoring

The solution can be configured to automatically display MediaLive Channel, Multiplex, and MediaConnect alerts. It integrates with CloudWatch alarms to indicate operational problems from a top-down view. Any CloudWatch alarm can be associated with any node on the diagram. CloudWatch alarm indicators are visualized as color and text on the node. CloudWatch high-resolution alarms can be used for frequent ten-second notification intervals.

The following image shows MediaConnect flows sending to a MediaLive input, a MediaLive channel with an alert status on one of two pipelines shown in yellow, and a MediaPackage channel with a custom alarm condition.

Diagram with alerts for monitoring

The tabs shown in the lower compartment are based on the resource type selected in the diagram. MSAM provides three tabs in the lower compartment for **Pipeline Alerts**, **Subscribed Alarms**, and **Recent CloudWatch Events** history. Select the **Recent CloudWatch Events** tab to view event details for the currently selected diagram node.
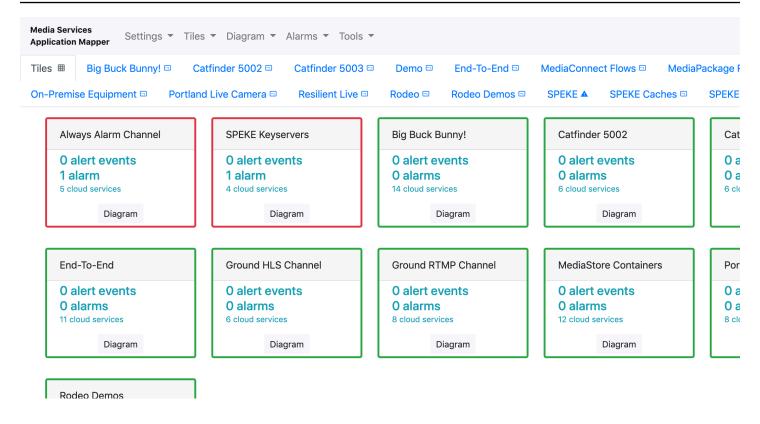
> ⓘ **Note**
>
> The **Pipeline Alerts** tab only shows when you select MediaLive resources.

A tile aggregates all structure, configuration, and status from the underlying assigned resources. If any resource's CloudWatch alarm assigned to a tile goes into alarm status, the tiles associated with that resource will also show the same status. The same is true for any MediaLive resource contained by the tile that sets an alert state.

Tiles with alarms

Custom resources

The `DefaultSettingsResource`, `WebContentResource` and `WebInvalidationResource` Lambda functions are custom resources used by CloudFormation during installation of the solution.

Amazon DynamoDB

The Media Services Application Mapper on AWS provisions several DynamoDB tables during initial deployment: `Alarms`, `Channels`, `CloudWatchEvents`, `Content`, `Events`, `Layout`, `ResourceNotes`, and `Settings`.

DynamoDB tables and their purposes:

- `Alarms` - stores alarms states

- `Channels` - stores tile definitions

- `CloudWatchEvents` - stores recent events for monitored inventory

- `Content` - stores cached inventory collected from the various AWS services

- `Events` - stores media services alerts and their status
- `Layout` - stores diagram layout information for the browser application
- `ResourceNotes` - stores notes for tiles, monitored inventory, and its connections
- `Settings` - stores general settings information for the cloud processes and browser application

Read and write capacity

By default, this solution sets the Amazon DynamoDB read/write capacity mode to on-demand for all tables created for the application. This setting allows the table read and write capacity to scale for variable sizes of inventory while remaining billed by total reads and writes. For more information, refer to Read/write capacity mode in the *Amazon DynamoDB Developer Guide*.

> ⓘ **Note**
>
> We recommend setting alarms and notifications for throttling errors, if utilization exceeds 75%, or if any table exceeds read/write capacity units.

Encryption

Each DynamoDB table and all indexes are configured for encryption at rest using AWS Key Management Service (AWS KMS) and the KMS key for DynamoDB in your account.

Point-in-time recovery

Each DynamoDB table is configured for point-in-time recovery. The recovery window, set by AWS, is up to five calendar weeks. This window size is set by AWS. You can use this type of recovery to restore one or more copies of a table to a known place in time, and then use the DynamoDB table migration tool to copy data back into the live tables after reviewing the restored results.

For more information, refer to Point-in-Time Recovery for DynamoDB in the *Amazon DynamoDB Developer Guide*.

REST API

All browser actions are performed through an authenticated and SSL encrypted REST API hosted in the cloud. The API can be used by other tools to perform activities such as preloading tile definitions or adding custom content to the cache.

For details, refer to the REST_API.md file in the GitHub Repository.

AWS services in this solution

The following AWS services are included in this solution:

AWS service	Description
Amazon API Gateway	**Core**. Supports the inventory and monitoring queries to the solution's backend.
Amazon CloudFront	**Core**. Serves the Media Services Application Mapper on AWS web application.
Amazon CloudWatch	**Core**. Monitors the scheduled rules or event patterns to invoke the primary Lambda functions.
Amazon DynamoDB	**Core**. Stores the solution's backend and frontend configuration and state.
AWS Lambda	**Core**. Handles client REST API requests from API Gateway, periodically scans for media services resources, and receives notification of changes in resource statuses.
Amazon S3	**Core** . Provides the origin bucket to serve the solution's web application.
AWS Elemental Link	**Optional**. One of the AWS Elemental services supported in the solution.
AWS Elemental Live	**Optional**. One of the AWS Elemental services supported in the solution.
AWS Elemental MediaStore	**Optional**. One of the AWS Elemental services supported in the solution.

AWS service	Description
AWS Elemental MediaConnect	**Optional**. One of the AWS Elemental services supported in the solution.
AWS Elemental MediaLive	**Optional**. One of the AWS Elemental services supported in the solution.
AWS Elemental MediaPackage	**Optional**. One of the AWS Elemental services supported in the solution.
AWS Elemental MediaTailor	**Optional**. One of the AWS Elemental services supported in the solution.
AWS Systems Manager	**Supporting**. Provides application-level resource monitoring and visualization of resource operations and cost data.

Plan your deployment

This section describes the cost, architecture, network security, and other considerations prior to deploying the solution.

Cost

You are responsible for the cost of the AWS services used while running this solution. The total cost of this solution depends on the size of your resource inventory and monitoring activity. As of this revision, the cost for running this solution with default settings for about five channels in the US East (N. Virginia) Region is approximately **$32 per month.** This amount can vary due to data storage charges for larger resource inventories, monitoring multiple AWS Regions, or monitoring at higher-than-default intervals. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution.

AWS service	Quantity	Cost (USD per month)
Amazon API Gateway	~ 5,000,000 requests	$15.00
Amazon DynamoDB	~ 5,500,000 reads/writes	$6.00
AWS Lambda	~ 750,000 seconds	$9.00
Amazon CloudFront		No cost (free tier)
Amazon S3		No cost (free tier)
Amazon CloudWatch	~ 20-30 GB logs, 14 metrics	$2.00
Total per month:		**$32.00**

We recommend creating a [budget](#) through [AWS Cost Explorer](#) to help manage costs. Prices are subject to change. For full details, see the pricing webpage for each AWS service used in this solution.

Sample cost table

The following table provides a sample cost breakdown for deploying this solution with the default parameters in the US East (N. Virginia) Region for one month.

AWS service	Dimensions	Cost (USD)
Amazon API Gateway	1,000,000 REST API calls per month	$3.50
Amazon Cognito	1,000 active users per month without the advanced security feature	$0.00
Amazon CloudFront	1,000,000 requests — 1 request per download with the static bucket size of 100KB and approximately 20 GB data transfer	$8.50
Amazon S3	100 GB data transfer + 1,000,000 requests — 100 records x 100 KB from Kinesis	$3.27
AWS Lambda	2,000,000 requests with 200 ms duration	$1.23
Systems Manager Parameter Store	2,000,000 requests with 10 standard parameters	$0.00
Amazon Lex	100,000 text requests per month	$75.00

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This [shared model](#) can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the

physical security of the facilities in which the services operate. For more information about security on AWS, visit AWS Cloud Security.

Multiple users and access control

You can create multiple API keys, each with different names that are provided to separate groups. To share the browser application with a colleague, provide the browser application URL, core endpoint URL, and an API key.

> ⚠️ **Important**
>
> If an API key is stolen or lost, create a new API key and delete the previous one. Send the updated API key by method of your preference to all users that require access. All users will have to update in the browser application. This solution's browser application prompts the user to update the endpoint and key if the previously used settings are unable to access the back-end.

Installation permissions

The CloudFormation templates provided for solution installation require permissions to create and configure several different types of cloud resources. The user launching the CloudFormation templates must have permissions to create the resources defined in the templates. CloudFormation will assume that user's permissions temporarily during installation to complete the steps.

There are several options for the user installing the templates to have the correct permissions:

1. A user with `AdministratorAccess` role attached.

2. A user that belongs to the `installationGroup` created by the IAM template.

3. Using the root user if no other options are available.

> ⚠️ **Important**
>
> Using the root user is generally discouraged. For details, refer to AWS account root user in the *IAM User Guide*.

IAM resources

The solution's IAM template installs a group with an inline policy with permissions sufficient to install all the templates of the solution. The IAM group is named `<stackname>`-`installationGroup-<ID>`. Add the users to this group who will be installing the solution if your organization has strict requirements about the `AdministratorAccess` managed policy.

Supported AWS Regions

This solution uses the MediaLive and MediaPackage support service, which is not currently available in all AWS Regions. You must launch this solution in an AWS Region where these services are available.For the most current availability of AWS services by Region, see the [AWS Regional Services List](#).

Media Services Application Mapper is supported in the following AWS Regions:

Region name	
US East (Ohio)	Asia Pacific (Tokyo)
US East (N. Virginia)	Europe (Frankfurt)
US West (Northern California)*	Europe (Ireland)
US West (Oregon)	Europe (London)
Asia Pacific (Mumbai)	Europe (Paris)
Asia Pacific (Seoul)	Europe (Stockholm)
Asia Pacific (Singapore)	South America (Sao Paulo)
Asia Pacific (Sydney)	

MediaPackage only*

Deploy the solution

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

Customization

The Media Services Application Mapper on AWS is designed to be customized or extended. For example, you can add custom node types with the browser application, using REST API, or directly into the database through a cloud-side task. You can also extend connection discovery and visualization overlay functionality to different types of cloud services. Refer to the [EXTENDING_MSAM.md](#) file in the GitHub repository for guidance about customizing the application.

Raw web content

The Media Services Application Mapper on AWS browser application is available in zipped form. This file can be extracted into a web server or another type of hosting environment. Take this approach if you decide not to use the CloudFormation template and prefer not to host the application in an Amazon S3 bucket.

URL for zipped file:

```
https://solutions-us-west-2.s3.us-west-2.amazonaws.com/media-services-
application-mapper/vX.X.X/msam-web.zip
```

> ⓘ **Note**
>
> Update *vX.X.X* with the current version number of the solution; for example, v1.11.0.

> ⓘ **Note**
>
> If you have previously deployed this solution, see [Update the solution](#) for update instructions.

AWS CloudFormation templates

You can download the CloudFormation template for this solution before deploying it.

[View template] **aws-media-services-application-mapper-release.template:** This is the solution's root template that installs all of the nested templates (listed below) with a single AWS CloudFormation launch. This template's default configuration deploys API Gateway, CloudFront, CloudWatch, DynamoDB, Lambda, and Amazon S3. We recommend this template for most installations. You can also customize the template based on your specific needs.

[View template] **aws-msam-iam-roles-release.template:** This nested template configures the solution's Amazon IAM policies and roles. You can also customize the template based on your specific needs.

[View template] **aws-msam-dynamodb-release.template:** This nested template configures the solution's DynamoDB tables that store all solution-related data. You can also customize the template based on your specific needs.

[View template] **aws-msam-core-release.template:** This nested template configures all solution microservices. The default configuration deploys Lambda functions that provide the business logic and data access layer for all solution operations. You can also customize the template based on your specific needs.

[View template] **aws-msam-events-release.template:** This nested template configures the solution's event handler microservices. You can also customize the template based on your specific needs.

View template

aws-msam-browser-app-release.json: This nested template configures the solution's browser application for visualizing your media service resources and connections. You can also customize the template based on your specific needs.

> ⓘ **Note**
>
> For information about the various templates and how to customize them, refer to [Individual templates](#).

Deployment process overview

Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Time to deploy:

Approximately 10 minutes, depending on the AWS Region. Some Regions can take approximately 40 minutes.

[Step 1. Launch the root template](#)

- Launch the AWS CloudFormation template into your AWS account.
- Enter value for required parameter: **Stack Name**.

[Step 2. (Optional) Install Event Handler stack in other Regions](#)

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

> ⚠ **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 [AWS Privacy Policy](#).

> 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.

Step 1. Launch the root template

This automated AWS CloudFormation template deploys the Media Services Application Mapper on AWS solution in the AWS Cloud. The root template nests and deploys the existing five templates automatically.

> ⓘ **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 `aws-media-services-application-mapper-release.template` AWS CloudFormation template.

<div align="right">

Launch solution

</div>

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 and leave the **CacheItemTTL** value as the default. For information about naming character limitations, refer to IAM and STS quotas in the *AWS Identity and Access Management User Guide*.

5. Choose **Next.**

6. On the **Configure stack options** page, choose **Next**.

7. 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.

8. Choose **Create stack** to deploy the root stack, which also deploys the five nested stacks.

   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 10 minutes.

Step 2. (Optional) Install Event Handler stack in other Regions

> ⚠️ **Important**
>
> Install the CloudWatch event handler stack in each Region Media Services are configured. Otherwise, you won't be able to monitor these resources.

For detailed instructions on how to deploy this template into additional Regions, refer to Template 4: CloudWatch event handler.

Post-deployment tasks

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

Obtain the browser and endpoint URLs

After the stack deploys, record the browser and endpoint URLs.

1. Sign in to the [AWS CloudFormation console](#).

2. On the **Stacks** page, select the solution's root stack.

3. Choose the **Outputs** tab and record the values for **MSAMBrowserURL** and **EndpointURL**.

Obtain the API Gateway API key

Media Services Application Mapper on AWS requires an API key to authorize access.

1. Open a new browser tab using the link shown for **APIKeyID**. This will open the API Gateway console to the page for this stacks API key.

2. In the main panel, in the **API key row**, choose **Show**.

   The API key is hidden by default and only appears when you choose **Show**.

3. Record the API key for making API calls. Save it with the browser and endpoint URL values.

Configure the browser application

When you launch this solution's browser application it prompts you for an endpoint URL and API key. The tool confirms that the connection information is valid before continuing. You can save several endpoints in browser storage for reuse later. If you launch the browser application and it finds a previous connection, it will automatically use that connection on the next launch.

1. Open a new browser tab and use the URL value from the **MSAMBrowserURL** output obtained earlier.

2. In the API Endpoint Connection box, enter the **EndpointURL** value and **API key**, also obtained earlier.

> ⓘ **Note**
>
> Select the **Remember** check box to store the endpoint and API key for recall after the browser is closed. Uncheck **Remember** to use this endpoint and API key only during the current browser session.

3. Choose **Connect**.

The dialog box closes and the **Status** compartment at the bottom of the page shows progress loading the inventory collected so far.

Individual templates

Depending on your needs, the solution allows you to install the CloudFormation templates separately. This might be the preferred option for you if you are performing a customized installation that requires special handling between stacks, or if you have distributed responsibilities within your organization for managing your AWS accounts.

The order in which to create the stacks are as follows:

1. Create the IAM resources stack.

2. Create the DynamoDB tables stack.

3. Create the Core API and periodic tasks stack.

4. Create the CloudWatch event handler stack.

5. Create the Browser application stack.

Install the IAM, DynamoDB, Core, and Web stacks in your main, or most accessed AWS Region only. The IAM resources installed by CloudFormation are global, although the stack you use to install them should be run in the same Region as the other stacks.

> ⚠️ **Important**
>
> Install the CloudWatch event handler stack in each Region Media Services are configured. Otherwise, you won't be able to monitor these resources.

Template 1: IAM resources

This template creates a stack for the IAM roles needed by the solution and a group needed by operators for installing the stacks.

1. Sign in to the AWS Management Console and select the button to launch the `msam-iam-roles-release.template` AWS CloudFormation template.

Launch solution

You can also [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 the stack.

5. Choose **Next.**

6. On the **Configure stack options** page, choose **Next**.

7. 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.

8. 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 two minutes.

Outputs

After the stack is deployed, select the **Outputs** tab and copy the four role ARNs to a notepad. You need these as input parameters for the next templates. Also, the **InstallationGroupLink** link takes you to the IAM console to review the group created for granting installation permissions to users.

Outputs (5)

Key	Value	Description
CoreRoleARN	arn:aws:iam::▬▬▬▬▬▬▬▬:role/newthings-CoreRole-7AZQKRYVVQZ7	IAM Role ARN for the Core module Lambda functions
DynamoDBRoleARN	arn:aws:iam::▬▬▬▬▬▬▬:role/newthings-DynamoDBRole-R6NTZ4JUM9B6	IAM Role ARN for the DynamoDB module Lambda functions
EventsRoleARN	arn:aws:iam::▬▬▬▬▬▬▬:role/newthings-EventsRole-N3DZLYQMN19W	IAM Role ARN for the Events module Lambda functions
InstallationGroupLink	https://console.aws.amazon.com/iam/home?#/groups/newthings-InstallationGroup-1JAO3AOVMW82F	Link to IAM to view and assign users to the installation group
WebRoleARN	arn:aws:iam::▬▬▬▬▬▬▬:role/newthings-WebRole-1EUV4U21F56XU	IAM Role ARN for the Web app module Lambda functions

IAM stack Outputs tab

Template 2: DynamoDB tables

This template creates a stack for the tables, indices, and on-demand capacity settings. The first time a stack is created from this template, defaults are added to scan and display AWS Cloud resources in the current Region only. Update these settings in the tool to expand the inventory coverage to other Regions.

1. Sign in to the AWS Management Console and select the button to launch the `msam-dynamodb-release.template` AWS CloudFormation template.
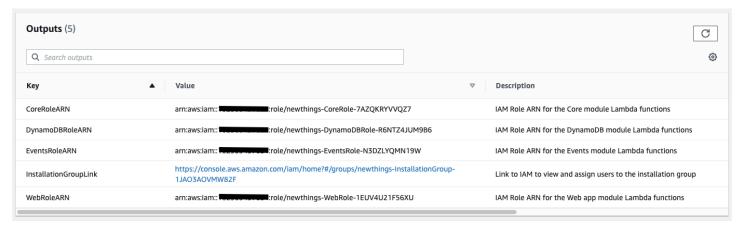
   Launch solution

   You can also [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 the stack.

5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.
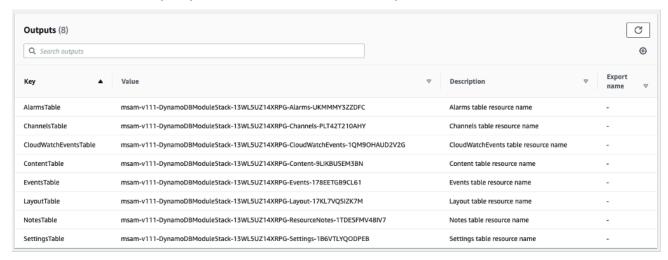
Parameter	Default	Description
DynamoDBIAMRoleARN	*\<Requires input\>*	Provide the ARN for the DynamoDB IAM role created by the IAM resources template.

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 two minutes.

Outputs

After the stack is deployed, select the **Outputs** tab and copy the seven table names to a notepad. You need these as input parameters for the next templates.

Key	Value	Description	Export name
AlarmsTable	msam-v111-DynamoDBModuleStack-13WL5UZ14XRPG-Alarms-UKMMMY3ZZDFC	Alarms table resource name	-
ChannelsTable	msam-v111-DynamoDBModuleStack-13WL5UZ14XRPG-Channels-PLT42T210AHY	Channels table resource name	-
CloudWatchEventsTable	msam-v111-DynamoDBModuleStack-13WL5UZ14XRPG-CloudWatchEvents-1QM9OHAUD2V2G	CloudWatchEvents table resource name	-
ContentTable	msam-v111-DynamoDBModuleStack-13WL5UZ14XRPG-Content-9LIKBUSEM3BN	Content table resource name	-
EventsTable	msam-v111-DynamoDBModuleStack-13WL5UZ14XRPG-Events-178EETGB9CL61	Events table resource name	-
LayoutTable	msam-v111-DynamoDBModuleStack-13WL5UZ14XRPG-Layout-17KL7VQSIZK7M	Layout table resource name	-
NotesTable	msam-v111-DynamoDBModuleStack-13WL5UZ14XRPG-ResourceNotes-1TDESFMV48IV7	Notes table resource name	-
SettingsTable	msam-v111-DynamoDBModuleStack-13WL5UZ14XRPG-Settings-1B6VTLYQODPEB	Settings table resource name	-

DynamoDB tables stack Outputs tab

Template 3: Core API and periodic tasks

This template creates a stack for the core REST API and periodic tasks used to refresh the content cache and discover logical resource connections.

> ℹ️ **Note**
>
> After installing the DynamoDB stack (Template 2), you can install the remaining stacks (Templates 3, 4, and 5) concurrently. There is no need to wait for each to finish before starting the next.

1. Sign in to the AWS Management Console and select the button to launch the `msam-core-release.template` AWS CloudFormation template.

Launch
solution

You can also 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 the stack.

5. Choose **Next.**

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
CoreIAMRoleARN	*<Requires input>*	Provide the ARN for the Core IAM role created by the IAM resources template.
CacheItemTTL	*<Requires input>*	Specify the time-to-live for the content, event, and alarm tables; content not refreshed before this time will be removed from the cache.
RootStackName	*<Requires input>*	The name of the root stack when used in a root/nested template configuration. When deploying templates separately, use this stack's name.

| Parameter	Default	Description
AlarmsTableName	*<Requires input>*	The name of the DynamoDB `Alarms` table from the DynamoDB stack.
ChannelsTableName	*<Requires input>*	The name of the DynamoDB `Channels` table from the DynamoDB stack.
CloudWatchEventsTableName	*<Requires input>*	The name of the DynamoDB `CloudWatch Events` table from the DynamoDB stack.
ContentTableName	*<Requires input>*	The name of the DynamoDB `Content` table from the DynamoDB stack.
EventsTableName	*<Requires input>*	The name of the DynamoDB `Events` table from the DynamoDB stack.
LayoutTableName	*<Requires input>*	The name of the DynamoDB `Layout` table from the DynamoDB stack.
NotesTableName	*<Requires input>*	The name of the DynamoDB `ResourceNotes` table from the DynamoDB stack.
SettingsTableName	*<Requires input>*	The name of the DynamoDB `Settings` table from the DynamoDB stack.
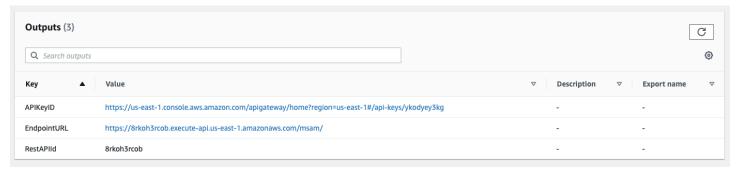
7. Choose **Next**.

8. On the **Configure stack options** page, choose **Next**.

9. 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.

10. 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 two minutes.

Outputs

After the stack is deployed, select the **Outputs** tab and copy the **EndpointURL** link to a notepad.

Outputs (3)			
Key ▲	Value	Description ▽	Export name ▽
APIKeyID	https://us-east-1.console.aws.amazon.com/apigateway/home?region=us-east-1#/api-keys/ykodyey3kg	-	-
EndpointURL	https://8rkoh3rcob.execute-api.us-east-1.amazonaws.com/msam/	-	-
RestAPIId	8rkoh3rcob	-	-

Core API and periodic tasks stack Outputs tab

Template 4: CloudWatch event handler

This template creates a stack for receiving events from media services and CloudWatch alarm state changes. This includes MediaLive pipeline alerts and other service state changes. Create a stack for this template in every Region you will be creating and monitoring media services resources using events, alerts, and alarms.

1. Sign in to the AWS Management Console and select the button to launch the `msam-events-release.template` AWS CloudFormation template.

   Launch solution

   You can also [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 the stack.

5. Choose **Next.**

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
EventsIAMRoleARN	*<Requires input>*	Provide the ARN for the Events IAM role created by the IAM resources template.
ItemTTL	*<Requires input>*	Specify the time-to-live in seconds for all recorded events.
AlarmsTableName	*<Requires input>*	The name of the DynamoDB `Alarms` table from the DynamoDB stack.
CloudWatchEventsTableName	*<Requires input>*	The name of the DynamoDB `CloudWatch Events` table from the DynamoDB stack.
ContentTableName	*<Requires input>*	The name of the DynamoDB `Content` table from the DynamoDB stack.
EventsTableName	*<Requires input>*	The name of the DynamoDB `Events` table from the DynamoDB stack.
EventsTableRegion	*<Requires input>*	The AWS Region name where the DynamoDB stack was installed.

7. Choose **Next**.

8. On the **Configure stack options** page, choose **Next**.

9. 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.

10. 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 two minutes.

Template 5: Browser application

This template installs a copy of the solution's browser application into an Amazon S3 bucket. Files added by the CloudFormation template have their ACL set to `public-read` to allow access over HTTPS.

> ⓘ **Note**
>
> If you prefer not to host the application in an Amazon S3 bucket, refer to [Raw web content](#).

1. Sign in to the AWS Management Console and select the button to launch the `msam-browser-app-release.template` AWS CloudFormation template.

Launch solution

You can also [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 the stack.

5. Choose **Next.**

6. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.
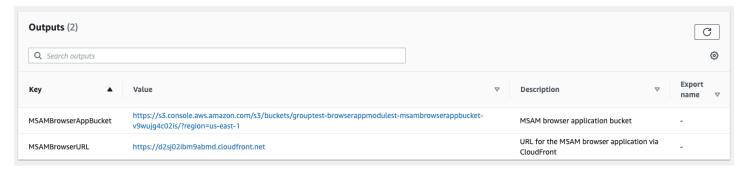
Parameter	Default	Description
WebIAMRoleARN	*<Requires input>*	Provide the ARN for the Web IAM role created by the IAM resources template.

7. Choose **Next**.

8. On the **Configure stack options** page, choose **Next**.

9. 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.

10. 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.

Outputs

After the stack is deployed, select the **Outputs** tab and copy the **MSAMBrowserURL** link to a notepad. Use this CloudFront distribution root to start the application on your workstation.

Outputs (2)

Key	Value	Description	Export name
MSAMBrowserAppBucket	https://s3.console.aws.amazon.com/s3/buckets/grouptest-browserappmodulest-msambrowserappbucket-v9wujg4c02is/?region=us-east-1	MSAM browser application bucket	-
MSAMBrowserURL	https://d2sj02ibm9abmd.cloudfront.net	URL for the MSAM browser application via CloudFront	-

Browser application stack Outputs tab

Optional parameters

The solution's browser URL can be the root of the CloudFront distribution URL or with `index.html` added. These two URLs are equivalent:

`https://d1c8z4f93zrlmx.cloudfront.net/`

`https://d1c8z4f93zrlmx.cloudfront.net/index.html`

The following parameters can be used with the second form of the URL with index.html to customize the start-up of the browser tool.

1. **diagram** - The name of a diagram to show right after start-up.
2. **endpoint** - The endpoint URL to use for the connection.
3. **key** - The API key to use with the chosen endpoint URL.

Examples

Show a default diagram named Livestream on start-up:

`https://d1c8z4f93zrlmx.cloudfront.net/index.html?diagram=Livestream`

Automatically connect to an endpoint with an API key:

`https://d1c8z4f93zrlmx.cloudfront.net/index.html?endpoint=https://`
`oplfnxzh7l.execute-api.us-east-1.amazonaws.com/msam/`
`&key=69ZSAV3tBX7YYfh1XTcsq2fLcE7Z0ETY4JXclqJJ`

> ⚠️ **Important**
>
> Use these parameters only for secure or demonstration environments. Anyone using a URL with endpoint and key parameters can connect and use the solution and share the URL with more users.

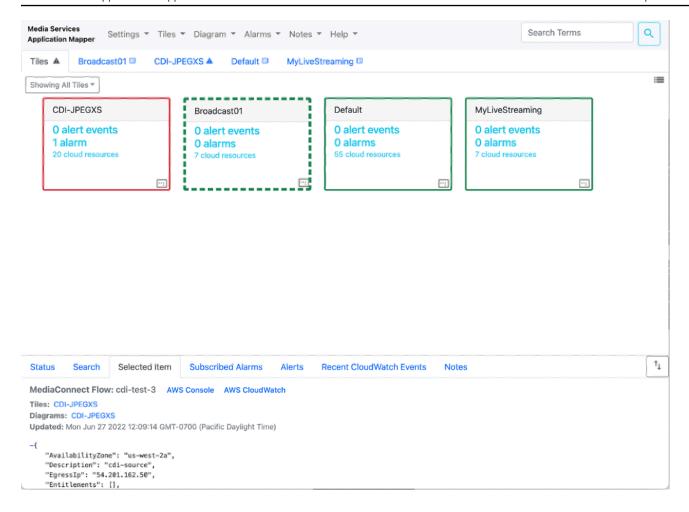# Using the browser application

When you first launch the web application, the solution creates a default tile. To visualize your cloud resources, add new tiles or create diagrams. You can create new tiles and diagrams from the inventory dialog, search results or the contents of existing diagrams and tiles.

Working with tiles

A tile represents all of the cloud resources that make up a streaming channel, system or other related collection of resources that should be monitored together in your environment. Any resource, such as an Amazon S3 bucket or CloudFront distribution, can belong to more than one tile. The alerts and alarms of a tile's resources are aggregated and communicated by visual indications on the tile.

Tiles are sorted by the total number of alarms and alerts, and then by tile name. Tiles are periodically updated to visually reflect the state of subscribed alarms and alerts. The following image of the tile tab shows several tiles with one indicating an alarm and a selected tile (**Broadcast01**) with details in the compartment.

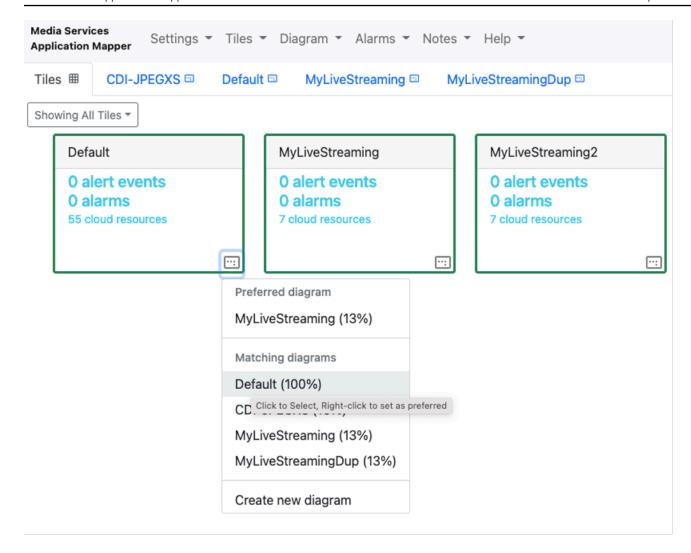**Example channel tiles with details compartment**

Tiles include a dropdown list that shows matching diagrams. Each matching diagram shows how much of a tile's content the diagram contains, indicated by a percentage.

Set a tile's preferred diagram

By default, a tile has no set preferred diagram. To set a tile's preferred diagram:

1. Select the icon at the bottom right of the tile to open the dropdown list.

2. Right-click one of the tile's matching diagrams.

3. Choose **Proceed** to confirm the selection.

Set preferred diagram for a tile

Create a tile

There are several ways to create a tile: adding a new tile, creating a copy from an existing tile, and creating a tile from a diagram, with all its contents.

Add a new tile

1. From any tab, choose **Tiles**, and then **Add New Tile**.
2. Enter a name for your tile and choose **Proceed**.

Create a copy of a tile

1. On the **Tiles** tab, drag an existing tile to the white background, choose **Proceed**.

2. Enter a name for your tile and choose **Proceed**.

Create a tile with the diagram's contents

1. On the **Tiles** tab, drag the tab of an existing diagram to the tile tab white background, choose **Proceed**.
2. Enter a name for your tile and choose **Proceed**.

> ⓘ **Note**
>
> Tiles automatically update every few seconds to show the latest status, and every few minutes (check **Advanced Settings**) to update the inventory of tiles.

Update tile contents

There are several ways to add inventory to a tile.

- Use the keyword search at the top right of the page and drag results from the bottom of the page to the tile, choose **Proceed**.
- Drag one tile on top of another tile to copy inventory from one tile to the other.
- Drag a diagram tab on top of a tile to copy the inventory from the diagram to the tile.
- From a diagram, select a node and choose **Tiles**, **Add Selected Node to Tile**, select the tiles to update, and then choose **Save**.
- To remove inventory or rename a tile, select a tile and choose **Tiles**, **Show/Edit Contents of Selected Tile**. Make your changes and choose **Save**.

Working with diagrams

Diagrams visualize your cloud resources with connections between resources to indicate data flow between cloud resources.

Add a diagram

1. From any tab, choose **Diagram**, and then **Add New Diagram**.
2. Enter a unique name for your diagram and choose **Proceed**.

Adding inventory

A resource can exist on more than one diagram. Add or remove inventory to and from diagrams to best suit your needs.

There are several ways to populate diagrams with inventory.

If you know the name of the resource

Use the keyword search at the top right of the page and drag the result from the bottom of the page to the diagram.

If you want to select items from the complete inventory

1. From the diagram's tab, choose **Diagram**, and then **Manage Diagram Contents**.
2. Identify and add items to the current diagram. The **Diagram Contents** box lists the downloaded inventory on the left side and the current diagram's contents on the right side. Use the buttons in the middle to add or remove nodes from the current diagram.
3. Choose **Save**.

Automatically expand the diagram

Media Services Application Mapper on AWS can automatically expand the diagram based on connections from existing nodes to nodes not yet on the diagram.

1. Populate the diagram with inventory.
2. Select (or multi-select) the applicable items on the diagram.
3. Choose **Diagram**, and then one of the following menu items to automatically expand the diagram's contents:

- Add Downstream Connected Nodes
- Add Upstream Connected Nodes
- Add All Connected Nodes

The concept of upstream and downstream connections between nodes is related to the direction of the arrowhead on each connection and the implied data flow direction between nodes.

Connections to downstream nodes have arrowheads pointing to the other nodes, indicating data flowing in that direction. Connections to upstream nodes have arrowheads pointing at this node, indicating data flowing from those nodes into this one. Nodes can have zero or more upstream connections coming to it, and zero or more downstream connections going from it to other nodes.

Connections

Visual connections between nodes are automatically added to diagrams when discovered. They are also removed automatically by the tool when either or both nodes supporting the connection are removed from the diagram. The back-end rules are reevaluated periodically against the inventory to show new connections or expire old connections.

Custom connections

You can add custom connections between two nodes that do not have an existing connection (cannot be automatically discovered).

1. Select two nodes on the diagram without an existing connection.

2. Choose the **Create Connection** button (blue arrow) at the right of the navigation bar. The first node selected is designated as the upstream node and the second node selected is designated as the downstream node. The arrow points toward the downstream node.

To edit or delete the custom connection

1. Select a custom connection.

2. Choose the **Edit Connection** (blue pencil) or **Delete Connection** (blue no entry symbol) button at the right of the navigation bar. You can provide a connection label name and set an expiration date using the **Edit Connection** button.

Custom connections display the "user-defined": true attribute in the **Selected Item** compartment at the bottom of the page.

Diagram layout

When you have inventory on the diagram, you can organize the nodes to make the diagram meaningful. Reposition nodes manually by selecting and dragging nodes to a new position.

Navigation

- Fit and magnification buttons are located at the bottom left of the diagram.

- The fit button resizes the entire diagram to the visible page.

- Movement buttons are also located at the bottom left of the diagram.

- Select and drag the background of the diagram to reposition.

- Double-click a node or near a node to zoom in to it.

Selection

- Selection notifications are shown at the far right of the navigation bar.

- Use the Command or Ctrl keys with left-click to select multiple items.

- Select a group of items on the diagram by dragging with the right button pressed.

- Selected nodes can be moved by dragging the nodes with the left button pressed.

Diagram layout tools

Media Services Application Mapper on AWS includes several tools to help organize and layout diagrams.

Layout tools available from the **Diagram** menu:

- **Layout Hierarchical Top-Bottom** - Reorganizes the current diagram top to bottom.

- **Layout Hierarchical Left-Right** - Reorganizes the current diagram left to right.

- **Layout Isolated Nodes** - Lays out nodes without connections. This feature groups the same types of nodes together into a square.

If your diagram contents don't look right, go to **Settings** > **Advanced Settings** > **Layout Method** and toggle the Layout Method between **Directed** and **Hub Size** to change how the diagram is organized. Some diagrams automatically layout better with one method over the other. One method prioritizes direction and the other prioritizes nodes with the most connections.
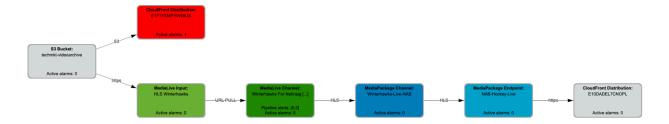
You can align sets of nodes vertically or horizontally using these tools from the **Diagram** menu:
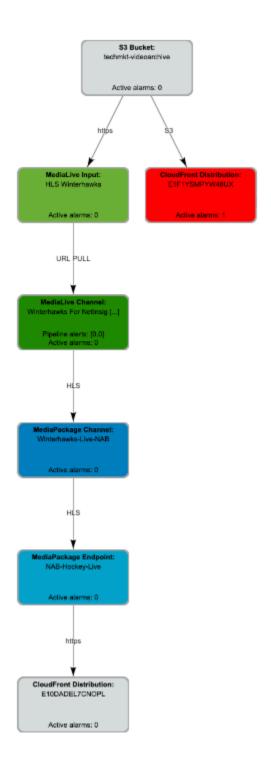
- **Align Selected Vertically**

- **Align Selected Horizontally**

Select all the nodes you want to align, select **Diagram**, then choose one of the alignment menu items.

Example diagram with horizontal alignment

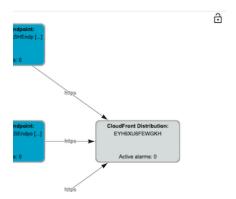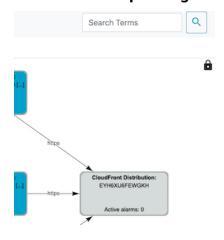## Example diagram with vertical alignment

Locking diagrams from unintended changes

The solution's permissions model allows anyone with access to the tool to create and update diagrams. Lock a diagram from unintended changes by selecting the small lock icon at the top right of the diagram.

Locking a diagram prevents changes to the diagram's contents or layout. Anyone can view the diagram, select nodes, and monitor for state changes. Additionally, anyone can unlock a locked diagram to make changes. This additional step prevents unintended changes.

Unlocked example diagram

Locked example diagram

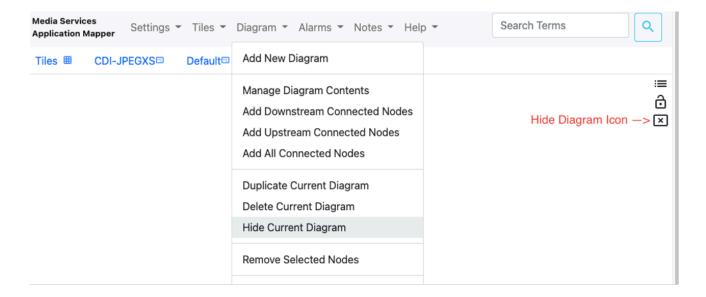# Managing the number of displayed diagrams

The maximum number of diagrams displayed at a time defaults to 10, but you can adjust this number using the **Settings** > **Advanced Settings** > **Maximum Number of Diagrams to Display** menu item. Once the maximum diagrams displayed has been reached, the application automatically hides the least recently viewed diagram when a new diagram is opened.

Hide a currently selected diagram in one of two ways

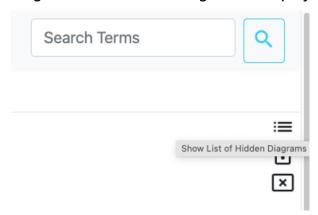1. Select **Diagram** > **Hide Current Diagram** menu item.

   (or)

2. Choose the X icon at the top right corner of the diagram view.

Hide current diagram

Redisplay a hidden diagram in one of two ways

1. Select the small list icon just above the lock icon at the top right corner of either the tile or diagram view. Select a diagram to display.
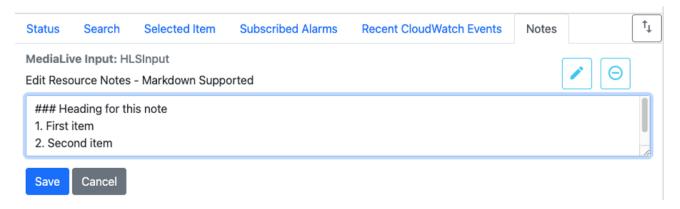

   Redisplay hidden diagram

(or)

2. Choose the **Diagram** tab, then select **Add New Diagram** and enter the name of the hidden
   diagram.

Resource notes

Select any node or connection on your diagram or select a tile in the **Tiles** tab. From the bottom of
the page, choose the **Notes** tab to attach notes to your resource. This solution supports Markdown
format.

Status	Search	Selected Item	Subscribed Alarms	Recent CloudWatch Events	Notes	↑↓

MediaLive Input: HLSInput

Edit Resource Notes – Markdown Supported

```
### Heading for this note
1. First item
2. Second item
```

Save   Cancel

Notes field

View all resource notes using the **Notes** > **Manage Notes** menu item. From the **Manage Notes**
window, you can filter notes by ARN, update time, notes content, and whether the resource
associated with this note still exists. You can also edit or delete individual notes from this view.
Notes are saved in a DynamoDB table and do not expire. To delete all notes, go to **Settings** >
Advanced Settings > **Bulk Delete** > **Notes**.

Data freshness

The browser application loads inventory once while it is starting. It does not automatically show
additions or changes in the cloud inventory. Refresh the browser page to retrieve the latest
inventory from the cloud cache. Diagrams, tiles, and other settings will be restored from the cloud.
Alarm and alert statuses of each resource in the inventory is updated on a periodic interval, which
you can configure using the **Advanced Settings** menu item.

Searching for resources, diagrams, and tiles

Search controls are located at the upper-right of the page and inside the **Search** tab at the bottom of the page. Inventory in-memory, all diagrams, and tiles are searched simultaneously. Search results are displayed in the **Search** tab at the lower part of the page.

Search results show several results:

- Inventory - Matches that may or may not be on a tile or diagram.
- Diagram names - Names of diagrams that match search criteria.
- Diagram contents - Diagrams with inventory that match search criteria.
- Tile names - Names of tiles that match search criteria.
- Tile contents - Tiles with inventory that match search criteria.

You can add search results to an existing diagram or tile by dragging the item from the results compartment onto the diagram or tile.

User-defined nodes

This solution allows you to define your own nodes. For details, refer to the REST_API.md file on the GitHub repository.

Monitoring and alarms

Select any item on your diagram or select a tile in the **Tiles** tab. From the bottom of the page, choose the **Subscribed Alarms** tab or **Recent CloudWatch Events** tab and view alarms and alerts related to the resources contained by the selected item or tile.

CloudWatch events and media services alerts

Install the `msam-events-release.template` CloudFormation template in each AWS Region with your MediaLive channels or multiplexes, or MediaConnect flows. The solution automatically receives events and alerts and indicates them on the diagrams and tiles containing those resources. The visual items representing MediaLive change color as Pipeline alerts are set and cleared.

Media services alerts are represented in yellow or red, depending on the resource's total number of pipelines and number of pipelines currently experiencing a problem. If the number of pipelines

experiencing a problem is less than the total number of pipelines, the node is colored yellow to indicate the resource is still moving data but without full redundancy. If all the pipelines of a resource are experiencing a problem, then the node is colored red to indicate a failure.

CloudWatch alarms

Media Services Application Mapper on AWS can associate any CloudWatch alarm from any AWS Region to any node in its inventory.

To subscribe a resource to an alarm:

> **ⓘ Note**
>
> You must create CloudWatch alarms in advance of performing these steps.

1. Select one or more resources on a diagram.
2. From the **Alarms** menu, select **Subscribe Selected to Alarms**.
3. Use the Region list and column filters to select one or more CloudWatch alarms.
4. Choose **Save**.

   All tiles and diagrams containing this node will reflect the current alarm state.

To view the alarm:

1. Select the diagram node.
2. From the bottom menu, select the **Subscribed Alarms** tab.

The upper area of the **Subscribed Alarms** table contains subscribed alarms and the lower area contains MediaLive alerts. Use the control at the far right of the table to remove an alarm from a cloud resource. The diagram item shows a state change of any subscribed alarms.

CloudWatch metrics

Each of the solution's stacks sends metrics to CloudWatch to track the count of each type of resource known in its inventory. The solution queries its DynamoDB Content table for the count of fourteen different resource types, including MediaLive channels and inputs, MediaPackage

channels and origin endpoints, S3 buckets, and more. The solution sends these metrics to CloudWatch once per hour. Metrics are stored under the **MSAM** namespace in CloudWatch, with dimensions for the resource type and stack name. The stack name is a dimension in determining the resource counts since different stacks installed in the same account might be tracking different Regions.

You can use these metrics to set alarms to track inventory sizes in your account. For example, you could create an alarm for the following scenarios:

- The number of MediaLive channels is not equal to the number of MediaPackage channels.
- The number of buckets exceeds a threshold.
- There is a sudden change in the number of resources of a specific type (anomaly alarm).
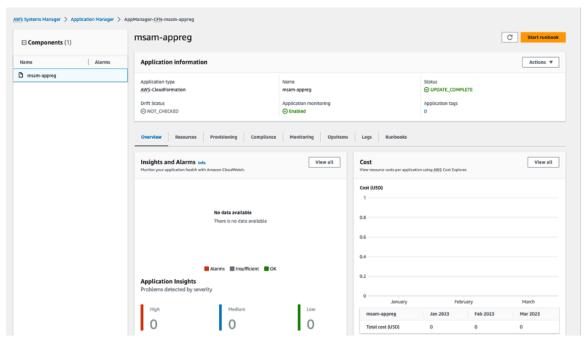
Monitoring this solution with Service Catalog AppRegistry

The Media Services Application Mapper on AWS solution includes a Service Catalog AppRegistry resource to register the CloudFormation template and underlying resources as an application in both AWS 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, such as deployment status, CloudWatch alarms, resource configurations, and operational issues.

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

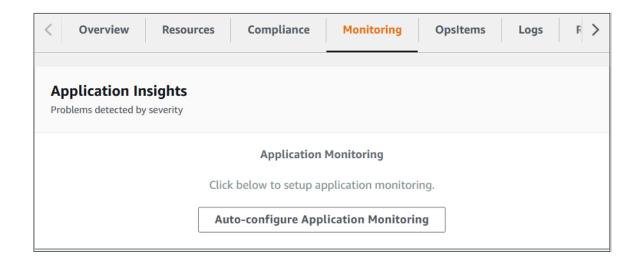**Media Services Application Mapper on AWS stack in Application Manager**

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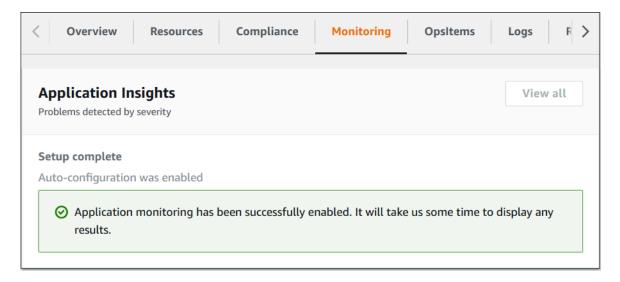
Activate CloudWatch Application Insights

1. Open 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**.

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

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 left navigation menu, 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 a cost allocation tag 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:

To activate cost allocation tags:

1. Sign in to the [AWS Billing and Cost Management console](#).

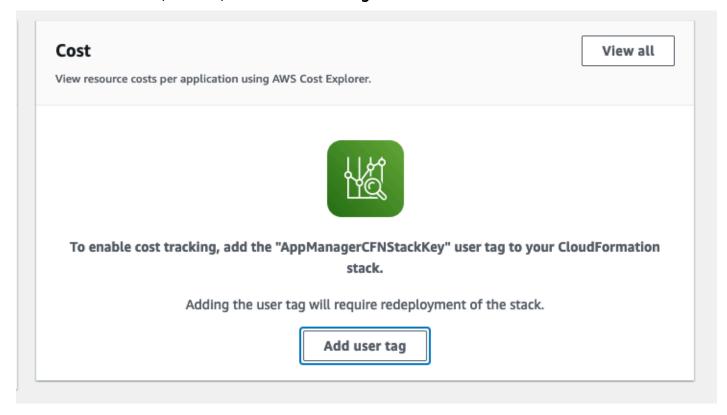2. In the navigation pane, select **Cost Allocation Tags**.

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**.

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

If you have previously deployed the solution, follow this procedure to update the Media Services Application Mapper on AWS CloudFormation stack to get the latest version of the solution's framework.

1. Sign in to the AWS CloudFormation console, select your existing Media Services Application Mapper on AWS CloudFormation stack, and select **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 root template for details about the parameters.

5. On the **Configure stack options** page, choose **Next**.

6. On the **Review** page, review and confirm the settings. Be sure to check the boxes acknowledging that the template might create AWS Identity and Access Management (IAM) resources.

7. Choose **View change set** and verify the changes.

8. 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 UPDATE_COMPLETE status in approximately 10 minutes.

Troubleshooting

This section provides known issue resolution when deploying the solution.

Issue: The CloudFormation template is taking longer than expected to deploy the stack.

Resolution

This solution installs a CloudFront distribution with the web application while the Amazon S3 bucket that hosts the content remains private. The first time installing the web application can take as long as 15-25 minutes for the CloudFront distribution to complete the deployment process. On rare occasions, it might take longer for CloudFront to complete its initial deployment.

Issue: I get a DNS redirect after navigating to the browser endpoint.

Resolution

You might receive DNS redirects or access denied from CloudFront for a short time if you install into a Region other than US East (N. Virginia). This is normal and might take from several minutes to an hour for DNS changes to propagate throughout the Amazon network. For more details, refer to [Temporary request redirection](#) in the *Amazon S3 User Guide*.

Issue: My installation is in US East (N. Virginia), but I need media services inventory from other Regions too. Do I install a copy of the solution in every Region?

Resolution

A single installation can collect inventory for all Regions within a single AWS account. By default, this solution only collects inventory from the Region in which it was installed. To collect inventory from more Regions, use the browser application. Choose **Settings**, **Advanced Settings**, and update the list of Regions in the **Inventory Regions** field. Choose **Save**.

Uninstall the solution

You can uninstall the Media Services Application Mapper solution from the AWS Management Console or by using the AWS Command Line Interface. To uninstall the solution, you must delete the solution's root CloudFormation stack, which deletes the other stacks and all of the resources created by the template.

> ⓘ **Note**
>
> If you used separate template to install the solution, you must delete those stacks separately.

Using the AWS Management Console

1. Sign in to the AWS CloudFormation console.

2. On the **Stacks** page, select this solution's root installation stack.

3. Choose **Delete**.

Using AWS Command Line Interface

Verify that the AWS Command Line Interface (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 command.

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

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

CloudWatch event handler stack

Delete the CloudWatch event handler stack from any other Regions that were installed after you installed the main solution.

Deleting separate stacks

If you used separate templates to install the solution, delete the stacks in the opposite order they were installed.

The order in which to delete the stacks is as follows:

1. Browser application stack

2. CloudWatch event handler stack

3. Core API and periodic tasks stack

4. DynamoDB tables stack

5. IAM resources stack

> ⚠️ **Important**
>
> You must remove the DynamoDB tables stack after the Core API and periodic tasks and CloudWatch event handler stacks. These stacks run software that access the tables. Removing the tables before removing the running software has the potential to create excessive error rates in your account which might result in service usage throttling.

To delete a stack:

1. Sign in to the [AWS CloudFormation console](#).

2. On the **Stacks** page, select the stack.

3. From the **Actions** menu, select **Delete Stack**

Deleting the Amazon S3 buckets

This solution is configured to retain the solution-created Amazon S3 to maintain history. After uninstalling the solution, you can manually delete this S3 bucket if you do not need to retain the data. Follow these steps to delete the Amazon S3 bucket.

1. Sign in to the [Amazon S3 console](#).

2. Choose **Buckets** from the left navigation pane.

3. Locate the *<stack-name-msam-browserappmod-msambrowserapplogicbuc-xxxxxxxxxxxx>* S3 bucket.

4. Select the S3 bucket and choose **Empty**.

5. Choose **Buckets** and select the S3 bucket again.

6. Choose **Delete**.

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

```
$ aws s3 rb s3:// <stack-name-msam-browserappmod-msambrowserapplogicbuc-xxxxxxxxxxxx>
  --force
```

Developer guide

This section provides the source code for the solution and additional customizations (if applicable).

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.

DynamoDB table migration tool

The Migration Tool is a Python program designed to copy data from one DynamoDB table to another. You can use this tool to copy the contents of each of the solution's DynamoDB tables from one installation to another. There are several reasons you might want to do this:

- Setting up a duplicate system for production and test, with the same diagrams, alarms, and layout for both.
- Moving from an old to new installation of MSAM that cannot be upgraded with the templates.
- Using the Point-in-Time Recovery feature of DynamoDB to restore one or more MSAM tables to a previously known state.

Requirements

- Python 3.x installed and available from the command line
- boto3 package is installed and accessible by the above Python
- AWS CLI profile name that is configured to access the account and DynamoDB tables used by MSAM
- Source and destination DynamoDB tables are located in the same account and same AWS Region

Usage

```
$ python copy_table.py -h
```

```
usage: copy_table.py [-h] --source SOURCE --destination DESTINATION
                     [--region REGION] [--profile PROFILE]

Copy database items from one DynamoDB table to another.

optional arguments:
  -h, --help            show this help message and exit
  --source SOURCE       source table name
  --destination DESTINATION
                        destination table name
  --region REGION       the region where the tables reside (if not provided,
                        default is us-west-2)
  --profile PROFILE     the AWS profile to use (if not provided, default
                        profile is used)
```

Warning

This tool copies all records from one table to another without consideration for overwrite or data loss. It will overwrite existing keys in the target table if the same keys exist in the source table.

Example: Duplicating tables between two stacks

1. Sign in to the AWS CloudFormation console.

2. Find the DynamoDB stack for the old and new stacks.

3. Record the table names for each stack.

4. Copy the tables from the old to new stack, skipping the DynamoDB Content table.

5. Start the new installation of the solution and verify tiles, diagrams, layout, and alarms.

Source table names

```
oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-Alarms-1VM2MNB13D37Y
oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-Channels-OJ059JTCKFVL
oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-CloudWatchEvents-1LA5KI02ULY10
oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-Content-JLLPFGYMS2LW
oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-Events-UB108ABQTFK
oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-Layout-SJVH87QY4VPM
oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-ResourceNotes-1NH97JNNT5ZQ5
oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-Settings-1IZ5B60KHU0IL
```

Destination table names

```
freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-Alarms-EXEGD9CBBRVT
freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-Channels-S7H7V3GU06L8
freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-CloudWatchEvents-18BSF813RQWLU
freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-Content-5ZTZ0BYLED9M
freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-Events-VK1R6QJ3HAL5
freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-Layout-1HQ6C00JC60ZJ
freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-ResourceNotes-QCKYRZF3JAV
freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-Settings-HF5BF72M35KH
```

Commands

> ⓘ **Note**
>
> If you don't have any user defined nodes, you can skip copying the Content table because it
> is populated automatically with the inventory of the selected Regions in the account.

If your source stack is missing a table, such as `CloudWatchEvents`, you can safely skip it.

```
python copy_table.py --source oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-
Alarms-1VM2MNB13D37Y --destination freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-
Alarms-EXEGD9CBBRVT --profile personal --region us-east-1
python copy_table.py --source oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-Channels-
0J059JTCKFVL --destination freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-Channels-
S7H7V3GU06L8 --profile personal --region us-east-1
python copy_table.py --source oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-
CloudWatchEvents-1LA5KI02ULY10 --destination freshinstall-
DynamoDBModuleStack-47U9UHHXPWO0-CloudWatchEvents-18BSF813RQWLU --profile personal --
region us-east-1
python copy_table.py --source oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-Events-
UB108ABQTFK --destination freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-Events-
VK1R6QJ3HAL5 --profile personal --region us-east-1
python copy_table.py --source oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-
Layout-SJVH87QY4VPM --destination freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-
Layout-1HQ6C00JC60ZJ --profile personal --region us-east-1
python copy_table.py --source oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-
ResourceNotes-1NH97JNNT5ZQ5 --destination freshinstall-
DynamoDBModuleStack-47U9UHHXPWO0-ResourceNotes-QCKYRZF3JAV --profile personal --region
 us-east-1
```

```
python copy_table.py --source oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-
Settings-1IZ5B60KHU0IL --destination freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-
Settings-HF5BF72M35KH --profile personal --region us-east-1
```

If you have user-defined nodes, you can copy the `Content` table too:

```
python copy_table.py --source oldinstall-DynamoDBModuleStack-1BL5SHLW7QL8P-
Content-JLLPFGYMS2LW --destination freshinstall-DynamoDBModuleStack-47U9UHHXPWO0-
Content-5ZTZ0BYLED9M
--profile personal --region us-east-1
```

References

This section includes information about an optional feature for collecting unique metrics for this solution 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 derive the scale, load, and storage needs placed on the application. The operational metrics consist of the maximum count of each resource type in the solution's inventory over the past 24 hours. The [CloudWatch metrics](#) generated on an hourly basis are used to generate the daily operational metrics.

The following JSON structure is an example of the metrics sent:

```
{
    "medialive-input": 6,
    "medialive-channel": 6,
    "medialive-multiplex": 0,
    "mediapackage-channel": 7,
    "mediapackage-origin-endpoint": 11,
    "mediastore-container": 0,
    "speke-keyserver": 3,
    "mediaconnect-flow": 1,
    "mediatailor-configuration": 0,
    "ec2-instances": 0,
    "link-devices": 0,
    "ssm-managed-instances": 0,
    "s3": 66,
    "cloudfront-distribution": 7
}
```

AWS owns the data gathered though this survey. Data collection is subject to the [AWS Privacy Policy](#). 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:

```
"Mappings": {
  "Send": {
    "AnonymizedUsage": {
      "Data": "Yes"
    }
  }
}
```

to:

```
"Mappings": {
  "Send": {
    "AnonymizedUsage": {
      "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 root template](#) in the [Deploy the solution](#) section of this guide.

Deactivate the EventBridge rule

If you have already installed the solution stacks, you can turn off anonymized metrics by deactivating the EventBridge rule. You can deactivate and activate metric gathering using this method without updating or changing the installation process.

1. Sign in to the [Amazon EventBridge console](#).

2. Find the EventBridge rule starting with the stack name and `ReportMetrics` in the name.

> ⓘ **Note**
>
> By default, the rule is set to run every 24 hours.

3. Select the rule and choose **Disable**.

Contributors

- Tom Gilman
- Jim Thario
- Joan Morgan
- Colin McCoy
- David Chung
- Eddie Goynes

Revisions

Date	Change
December 2018	Initial release
June 2019	Clarified information about the solution table names and where to find them
January 2020	Fixed Amazon CloudWatch link for EMX (MediaConnect) flows, added inventory and visualization support for AWS Elemental MediaLive multiplex, and made front-end performance improvements (repeated REST calls, expensive loop optimization)
March 2021	Release v1.8.0: Updated to reflect latest architecture and installation process changes. For a detailed description of the changes from version 1.7.5 to version 1.8.0, refer to the CHANGELOG.md file in the GitHub repository.
March 2021	Release v1.9.0: For a detailed description of the changes from version 1.8.0 to version 1.9.0, refer to the CHANGELOG.md file in the GitHub repository.
May 2021	Release v1.9.1: For a detailed description of the changes from version 1.9.0 to version 1.9.1, refer to the CHANGELOG.md file in the GitHub repository.
February 2022	Release v1.10.0: For a detailed descripti on of the changes from version 1.9.1 to version 1.10.0, refer to the CHANGELOG .md file in the GitHub repository. Document enhancements, including updates to Solution

Date	Change
	components, Automated deployment, Using the browser application, and Uninstall the solution sections.
August 2022	Release v1.11.0: Managing diagrams, tile view, and resource notes features. For more information, refer to the CHANGELOG.md file in the GitHub repository.
April 2023	Release v1.11.1: Mitigated impact caused by new default settings for S3 Object Ownership (ACLs disabled) for all new S3 buckets. For more information, refer to the CHANGELOG .md file in the GitHub repository.
May 2023	Release v1.12.0: Added integration with AWS Service Catalog AppRegistry and AWS Systems Manager Application Manager. Upgraded to CDK. Added a security patch. For more information, refer to the the CHANGELOG.md file in the GitHub repository.
August 2023	Release v1.13.0: Updated the build spec to Node.js 18 and updated the JavaScript SDK to v3. Fixed logging print statement for MSAM tags, an error in the MSAM browser when an EC2 instance did not have a tag, and an error in MediaConnect when using a VPC interface . Updated CloudFront distribution to use CDK L2 construct. Added Anonymized Data Cfn Mapping to allow for opting out. For more information, refer to the the CHANGELOG.md file in the GitHub repository.

Date	Change
October 2023	Release v1.13.1: Updated package versions to resolve security vulnerabilities. For more information, refer to the the CHANGELOG.md file in the GitHub repository.
November 2023	Documentation update: Added Confirm cost tags associated with the solution to the Monitoring the solution with AWS Service Catalog AppRegistry section.

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 Services Application Mapper on AWS is licensed under the terms of the of the Apache License Version 2.0 available at The Apache Software Foundation.