
Media Services Application Mapper Implementation Guide



Media Services Application Mapper: Implementation Guide

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

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

Table of Contents

Home	1
Overview	2
Cost	2
Architecture overview	3
Solution components	5
Browser application	5
Tiles	5
Diagrams	5
Solution microservices	6
API Handler microservice	6
Inventory discovery microservice	6
CloudWatch Events microservices	6
Custom resources	7
Amazon DynamoDB	7
Read and write capacity	7
Security	8
Multiple users and access control	8
Design considerations	9
Customization	9
AWS CloudFormation templates	10
Automated deployment	12
Deployment overview	12
Step 1. Launch the root template	12
Step 2. Post-configuration tasks	13
Obtain the browser and endpoint URLs	13
Obtain the API Gateway API key	13
Configure the browser application	13
Using the browser application	15
Working with tiles	15
Create a tile	15
Update tile contents	16
Working with diagrams	16
Add a diagram	16
Adding inventory	16
Connections	17
Custom connections	17
Diagram layout	18
User-defined nodes	18
Monitoring and alarms	18
Menus	18
Troubleshooting	19
Additional resources	20
Source code	21
Uninstall the solution	22
Using the AWS Management Console	22
Using AWS Command Line Interface	22
Revisions	23
Contributors	24
Notices	25

Browser application to visualize relationships between your media services

AWS Implementation Guide

Publication date: *December 2018 (last update (p. 23): May 2021)*

This implementation guide discusses architectural considerations and configuration steps for deploying the Media Services Application Mapper in the Amazon Web Services (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.

Overview

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 to [Amazon CloudWatch](#), it can be a challenge to identify the source of the problem.

Media Services Application Mapper 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.

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 March 2021, the cost for running this solution with default settings for about five channels in the US East (N. Virginia) Region is approximately **\$30 per month**. This amount can vary due to data storage charges for larger resource inventories, monitoring multiple AWS Regions, or monitoring at higher-than-normal 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 (per month)
API Gateway	~ 5,000,000 requests	\$15.00
Amazon DynamoDB	~ 5,500,000 reads/writes	\$6.00
AWS Lambda	~ 750,000 seconds	\$9.00
CloudFront		No cost (free tier)
Amazon S3		No cost (free tier)

AWS service	Quantity	Cost (per month)
Total per month		\$30.00

Architecture overview

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

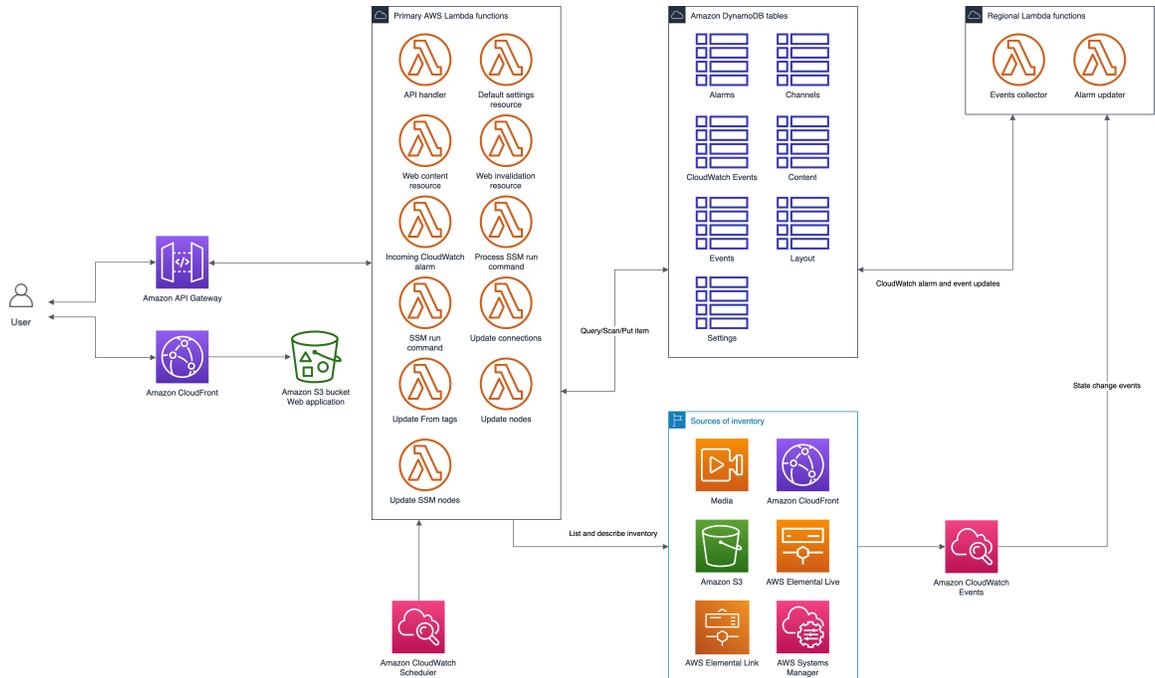


Figure 1: Media Services Application Mapper architecture on AWS

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

- An API Gateway and an API handler [AWS Lambda](#) function that is used by the browser application and custom software to work with the [Amazon DynamoDB](#) tables.
- Ten Lambda functions invoked periodically by CloudWatch scheduled rules or event patterns that are used to scan for media services resources and receive notifications of changes in a resources' status.
- Three Lambda functions that are custom resources used during the installation of the solution.
- Seven DynamoDB tables used to store the solution's back-end and front-end configuration and state.
- A CloudFront distribution connected to an Amazon S3 bucket as an origin to serve the Media Services Application Mapper browser application.

The *Sources of inventory* box in Figure 1 shows the 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 AWS Elemental Live encoders.

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.

Solution components

Browser application

The browser application visualizes the relationships between your inventory. Configure your environment with either the tile view or 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 significant workflow. 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.

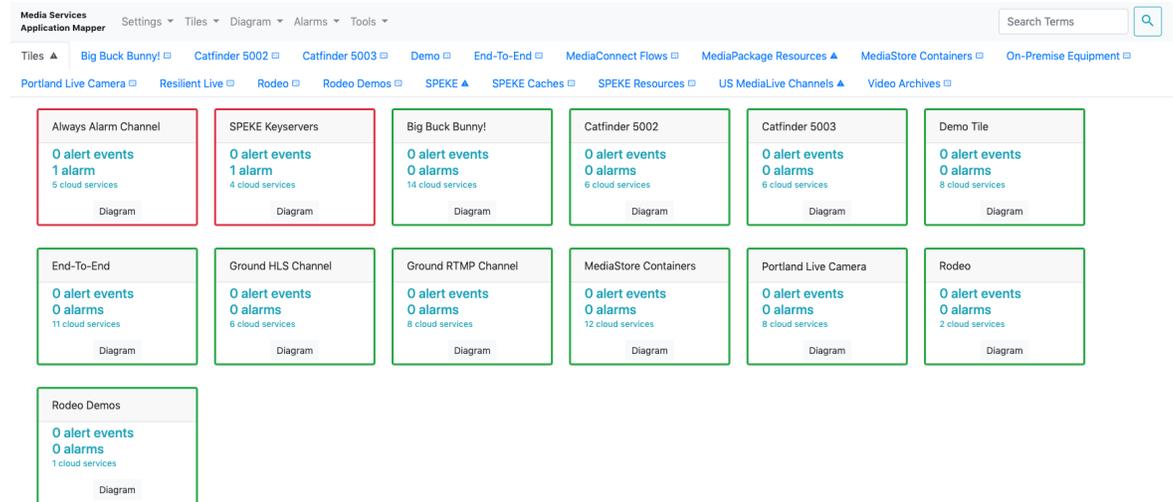


Figure 2: Example channel tiles

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.

Diagrams

The application visualizes your resources and connections as workflow diagrams that are assembled by the user. The browser application includes features to navigate, find, and build diagrams of related cloud resources.

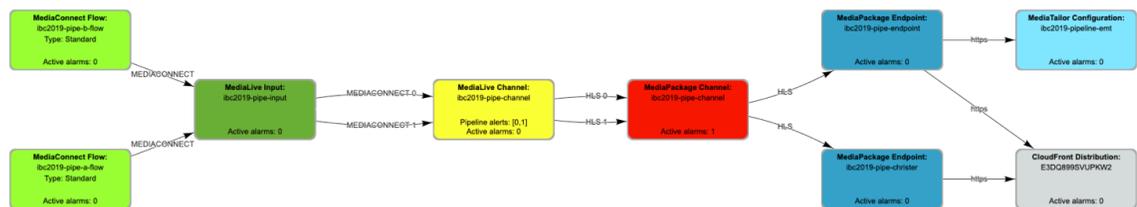


Figure 3: Example workflow diagram

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.

For additional details about tiles and diagrams, refer to [Using the browser application \(p. 15\)](#).

Solution microservices

The Media Services Application Mapper 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.

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 provisions several DynamoDB tables during initial deployment: `Alarms`, `Channels`, `CloudWatchEvents`, `Content`, `Events`, `Layout`, 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
- `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*.

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.

Design considerations

Customization

The Media Services Application Mapper 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 template.

AWS CloudFormation templates

This solution uses AWS CloudFormation to automate the deployment of the Media Services Application Mapper in the AWS Cloud. It includes the following AWS CloudFormation templates, which you can download before deployment:

[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](#)

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](#)

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](#)

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](#)

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](#)

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 the [INSTALL.md](#) file in the GitHub repository.

Automated deployment

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

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

Deployment overview

Use the following steps to deploy this solution on AWS. For detailed instructions, follow the links for each step.

[Step 1. Launch the root template \(p. 12\)](#)

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

[Step 2. Configure the browser application \(p. 13\)](#)

- Launch the browser application.
- Enter values for the **Endpoint URL** and **API Key** fields.

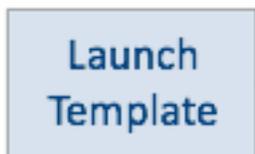
Step 1. Launch the root template

This automated AWS CloudFormation template deploys the Media Services Application Mapper solution in the AWS Cloud.

Note

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

1. Sign in to the AWS Management Console and select the button to launch the `aws-media-services-application-mapper-release.template` AWS CloudFormation template.



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

2. The template launches in the US East (N. Virginia) Region by default. To launch the solution in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.

4. On the **Specify stack details** page, assign a name to your solution stack 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. Post-configuration 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 requires an API key to authorize access.

1. Sign in to the [Amazon API Gateway console](#).
2. Under **APIs**, select the **msam** API created by this solution.
3. In the left navigation pane, choose **API Keys**.
4. Select the API key.
5. 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**.

6. 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 **Do Not Remember** to toggle to **Remember** and to store the endpoint and API key for recall after the browser is closed. Select **Do Not 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.

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 also create a tile from the contents of an existing diagram.

Working with tiles

A tile represents all of the cloud resources that make up a streaming channel, or some type of system or solution. Any resource can be a member of more than one tile, such as an Amazon S3 bucket or CloudFront distribution. 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. Diagram nodes are periodically updated to visually reflect the state of subscribed alarms and alerts. Tiles reflect the aggregate status of all the cloud resources associated with the tile. The following image of a tile tab shows several tiles indicating an alarm, a selected tile (Portland Live Camera) with details in the compartment.

The screenshot displays the Media Services Application Mapper interface. At the top, there is a navigation bar with 'Settings', 'Tiles', 'Diagram', 'Alarms', and 'Tools' menus, along with a search box. Below this is a list of tiles, including 'Big Buck Bunny!', 'Catfinder 5002', 'Catfinder 5003', 'Demo', 'End-To-End', 'MediaConnect Flows', 'MediaPackage Resources', 'MediaStore Containers', and 'On-Premise Equipment'. A sub-menu shows 'Portland Live Camera' selected. The main area contains a grid of tiles, each showing '0 alert events' and '0 alarms' (except for 'Always Alarm Channel' which has 1 alarm and 5 cloud services). The 'Portland Live Camera' tile is highlighted with a dashed green border. Below the grid is a details compartment for the selected tile, showing the channel name 'Portland Live Camera' and a JSON object with metadata such as 'Id', 'ARN', 'Status', 'LastModifiedTime', 'DomainName', 'Aliases', 'Origins', 'OriginGroups', 'DefaultCacheBehavior', 'CacheBehaviors', 'CustomErrorResponses', and 'Comment'.

Figure 4: Example channel tiles with details compartment

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.

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 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 one upstream connection coming to it, and one or more downstream connections going from it to other nodes.

Connections

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.

For additional navigation information and diagram layout tools, refer to the [USAGE.md](#) file on the Git Hub Repository.

User-defined nodes

This solution allows you to define your own nodes. For details, refer to the [REST_API.md](#) file on the Git Hub Repository.

Monitoring and alarms

For more information about monitoring cloud resources and to subscribe to various alarms, refer to the [USAGE.md](#) file on the Git Hub Repository.

Menus

For details about all of the options listed in the web application's menus, refer to the [USAGE.md](#) file on the Git Hub Repository.

Troubleshooting

Issue: The CloudFormation template taking longer than expected.

Solution: 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.

Solution: 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?

Solution: 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 **Never Cache Regions** field. Choose **Save**.

Additional resources

AWS services

- [AWS CloudFormation](#)
- [AWS Lambda](#)
- [Amazon API Gateway](#)
- [Amazon CloudFront](#)
- [Amazon Simple Storage Service](#)
- [Amazon DynamoDB](#)

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.

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 AWS CloudFormation stack, which deletes the other stacks and all of the resources created by the template.

Using the AWS Management Console

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

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 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 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 in the GitHub repository.

Contributors

- Tom Gilman
- Jim Thario
- Joan Morgan

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 is licensed under the terms of the of the Apache License Version 2.0 available at <https://www.apache.org/licenses/LICENSE-2.0>.