User Guide

# AWS Elemental MediaConnect

# AWS Elemental MediaConnect: User Guide

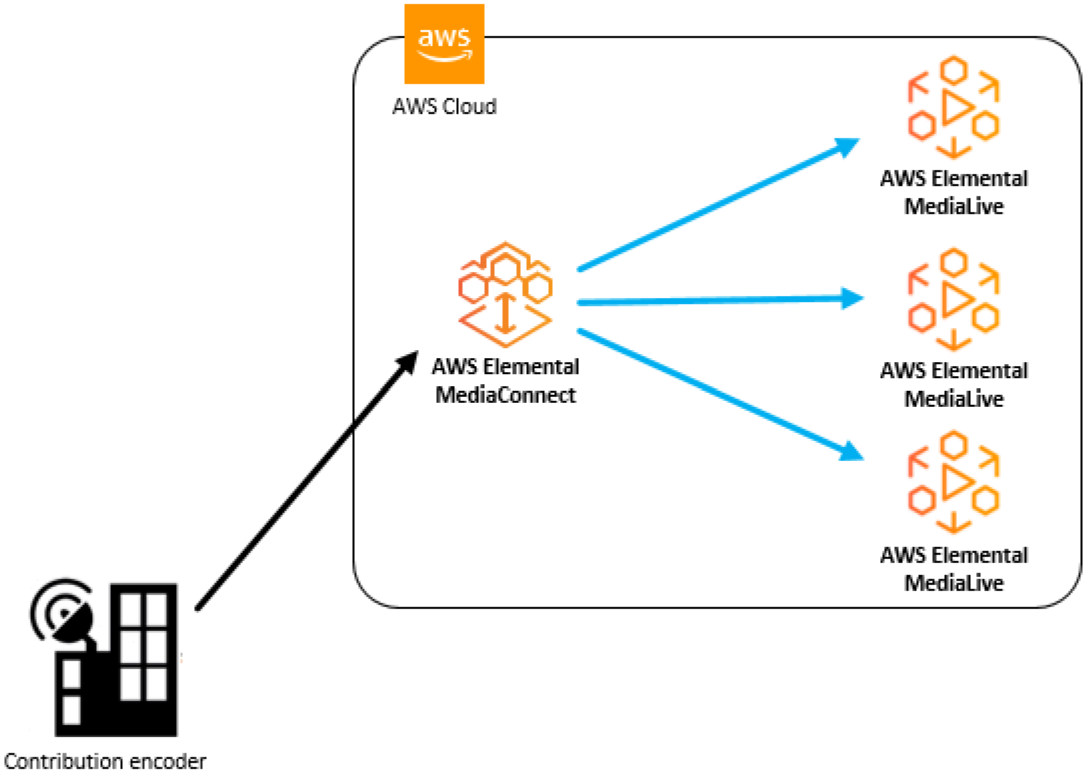
# Table of Contents

# What is AWS Elemental MediaConnect?

AWS Elemental MediaConnect is a service that makes it easy for broadcasters and other premium video providers to reliably ingest live video into the AWS Cloud and distribute it to multiple destinations inside or outside the AWS Cloud. MediaConnect provides the reliability, security, and visibility that you are used to with existing distribution methods, combined with the flexibility and cost-effectiveness that internet-based transmission provides.

For ingest, you send content to AWS Elemental MediaConnect from an on-premises contribution encoder, which encodes your video into a single, high-quality mezzanine file for contribution into the cloud. After the video is in the AWS Cloud, MediaConnect sends it to outputs that you specify, such as a cloud encoder, another MediaConnect flow, or an on-premises destination.

The following illustration shows the basic workflow of how AWS Elemental MediaConnect ingests live video into the cloud and securely distributes it to multiple destinations.



In AWS Elemental MediaConnect, you create a *flow* to establish a transport between a source and one or more outputs. You can also share content with other AWS accounts by creating *entitlements*. This allows the receiving account to create a flow using your content as the source.

With AWS Elemental MediaConnect, you can do the following:

- Ingest live video into the AWS Cloud.
- Distribute live video to multiple destinations inside or outside the AWS Cloud.
- Subscribe to a live video stream that is supplied by another AWS account. (This requires permission from the content originator through an entitlement.)
- Send content from one AWS Region to another.

**Topics**

## MediaConnect concepts and terminology

ARN

An Amazon Resource Name, which is a unique identifier for any AWS resource.

Availability Zone

A specific location where AWS Cloud computing resources are hosted. Availability Zones within an AWS Region are connected to each other with low latency, high throughput, and highly redundant networking. In addition, they are physically separated and isolated from each other. You can choose to create MediaConnect flows in different Availability Zones for redundancy.

AWS Region

A geographic area where one or more Availability Zones are located. Each AWS Region is independent from the other Regions. You can create MediaConnect flows in different Regions to distribute content to receivers in different locations around the world. For more information about AWS Regions and their Availability Zones, see AWS Global Infrastructure.

CDI flow

A MediaConnect flow that transports high-quality content that has been lightly compressed using JPEG XS. The content is demuxed into separate media streams for audio, video, or ancillary data. Each CDI flow can use multiple media streams for the source and multiple media streams for each output. MediaConnect uses AWS Cloud Digital Interface (AWS CDI) network technology to ingest content that adheres to the SMPTE 2110, part 22 transport standard.

Contribution encoder

An encoder that receives a live video feed and encodes the stream into a single, high-quality mezzanine stream for transportation or further processing into an adaptive bitrate (ABR) stream.

Distribution

The result of creating outputs that point to MediaConnect flows in other AWS Regions, for the purpose of delivering content to different geographical locations.

Entitlement

A permission that is granted to allow an AWS account to access the content in a specific MediaConnect flow. The content originator grants an entitlement to a specific AWS account (the subscriber). Once an entitlement is granted, the subscriber can create a flow using the originator's flow as the source. You can only grant entitltements to transport stream flows.

Flow

A connection between one or more video sources and one or more outputs. For each flow, you specify the transport protocol to use, encryption information, and details about the source. MediaConnect returns an ingest endpoint where you can send your live video as a single unicast stream. The service replicates and distributes the video to every output that you specify, whether inside or outside the AWS Cloud. There are two types of flows: transport stream and JPEG XS.

Media stream

A single track or stream of media that contains video, audio, or ancillary data. After you add a media stream to a flow, you can associate it with sources and outputs on that flow, as long as they use the CDI protocol or the ST 2110 JPEG XS protocol. Each source or output can consist of one or many media streams.

Mezzanine stream

A lightly compressed video stream that takes up less space than a full resolution uncompressed stream. The quality of a mezzanine stream is high enough to use as a source for creating final encodes that are delivered to consumer devices.

Offering

A discount that MediaConnect offers in exchange for a commitment to use a certain amount of outbound bandwidth each month. When you purchase an offering, it becomes a reservation.

Originator account

An AWS account that was used to create a flow with at least one entitlement.

Output

The destination where you want MediaConnect to send ingested video. An output can have the same protocol or a different protocol from the source.

Policy

An IAM policy, which is used to manage access in AWS.

Protocol

A set of rules used for file transmission. MediaConnect provides protocol options (such as Zixi, RTP, and RTP-FEC) that implement a quality of service (QoS) layer to enable the service to work with mezzanine-quality live video.

Receiver

The recipient of a stream from MediaConnect. A receiver is any entity, inside or outside of the AWS Cloud, that can receive RTP or Zixi streams. This might be an affiliate, a cloud encoder, or another MediaConnect flow.

Reservation

A commitment to use a specific amount of outbound bandwidth each month over the course of a specified duration. In return, you pay a discounted hourly rate for that bandwidth. When you purchase an offering, it becomes a reservation.

Replication

The result of creating a flow with more than one output. The source is replicated to produce multiple outputs. Replication is useful when you want to distribute your video streams to multiple workflows within your own account or share your content with other AWS accounts.

Resource

An entity in AWS that you can work with. Each AWS resource is assigned an Amazon Resource Name (ARN) that acts as a unique identifier. In MediaConnect, these are the resources and their ARN formats:

- Entitlement: `aws:mediaconnect:`*`region`*`:`*`account-id`*`:entitlement:`*`resourceID`*`:`*`resourceName`*
- Flow: `aws:mediaconnect:`*`region`*`:`*`account-id`*`:flow:`*`resourceID`*`:`*`resourceName`*
- Output: `aws:mediaconnect:`*`region`*`:`*`account-id`*`:output:`*`resourceID`*`:`*`resourceName`*
- Source: `aws:mediaconnect:`*`region`*`:`*`account-id`*`:source:`*`resourceID`*`:`*`resourceName`*

Sharing

Allowing another AWS account to access the content of your flow. To share your content, you (the originator) grant an entitlement to another AWS account (the subscriber).

Source

External video content that includes configuration information (encryption and source type) and a network address. Each flow has at least one source. A standard source comes from a source other than another MediaConnect flow, such as an on-premises encoder. An entitled source comes from an MediaConnect flow that is owned by another AWS account and has granted an entitlement to your account.

Subscriber account

An AWS account that been granted access to content from an AWS Elemental MediaConnect flow that is owned by another AWS account (the originator account). This permission is granted when the originator sets up an entitlement for the subscriber. The entitlement permits the subscriber to create a flow that uses the originator's content as the source.

Transport stream flow

A MediaConnect flow that transports compressed content. Audio, video, and ancillary data must be combined, or *muxed*, into a single stream. The quality is high enough to use as a source for creating final encodes that are delivered to consumer devices. You can add outputs to indicate where you want the content to be sent and how you want it transported. You can also grant entitlements to allow another AWS account to access the content.

VPC interface

A connection between a flow and a virtual private cloud (VPC) that was created using the Amazon Virtual Private Cloud (Amazon VPC) service.

Whitelisting

Allowing a block of Classless Inter-Domain Routing (CIDR) IP addresses to serve as a source to your MediaConnect flow.

## Related services

- **AWS CloudTrail** is a service that lets you monitor the calls made to the CloudTrail API for your account, including calls made by the AWS Management Console, AWS CLI, and other services. For more information, see the AWS CloudTrail User Guide.
- **Amazon CloudWatch** is a monitoring service for AWS Cloud resources and the applications that you run on AWS. Use CloudWatch Events to track changes in the status of flows in AWS Elemental MediaConnect. For more information, see the Amazon CloudWatch documentation.
- **AWS Identity and Access Management (IAM)** is a web service that helps you securely control access to AWS resources for your users. Use IAM to control who can use your AWS resources (authentication) and what resources users can use in which ways (authorization). For more information, see Setting up.
- **AWS Elemental MediaLive** is a video service that allows easy and reliable creation of live outputs for broadcast and streaming delivery. For more information, see the AWS Elemental MediaLive User Guide.

## Accessing MediaConnect

You can access AWS Elemental MediaConnect using any of the following methods:

- **AWS Management Console** – The procedures throughout this guide explain how to use the AWS Management Console to perform tasks for MediaConnect. To access MediaConnect using the console:

  ```
  https://<region>.console.aws.amazon.com/mediaconnect/home
  ```

- **AWS Command Line Interface** – For more information, see the AWS Command Line Interface User Guide. To access MediaConnect using the CLI endpoint:

  ```
  aws mediaconnect
  ```

- **AWS Elemental MediaConnect API** – For information about API actions and about how to make API requests, see the AWS Elemental MediaConnect API Reference. To access MediaConnect using the REST API endpoint:

  ```
  https://mediaconnect.<region>.amazonaws.com
  ```
- **AWS SDKs** – If you're using a programming language that AWS provides an SDK for, you can use an SDK to access AWS Elemental MediaConnect. SDKs simplify authentication, integrate easily with your development environment, and provide easy access to MediaConnect commands. For more information, see Tools for Amazon Web Services.
- **AWS Tools for Windows PowerShell** – For more information, see the AWS Tools for Windows PowerShell User Guide.

## Pricing for MediaConnect

As with other AWS products, there are no contracts or minimum commitments for using MediaConnect.

For transport stream flows, you are charged a per hour fee when the flow is running, and a per GB fee for output delivered to the internet. You are also charged a per GB fee for input or output data within the same Region. In general, higher bitrate flows accrue higher charges per hour.

For CDI flows, you are charged a per hour fee when the flow is running, and a per hour fee for each output delivered to any destination. Running flow rates and per output rates change according to the size of the video. SD outputs are less expensive than HD outputs, which are less expensive than UHD outputs.

For more information on both types of flows, see AWS Elemental MediaConnect Pricing.

## Regions and endpoints for MediaConnect

To reduce data latency in your applications, AWS Elemental MediaConnect offers a regional endpoint to make your requests:

```
https://mediaconnect.<region>.amazonaws.com
```

To view the complete list of AWS Regions where MediaConnect is available, see AWS Elemental MediaConnect endpoints and quotas in the AWS General Reference.

# AWS Elemental MediaConnect use cases

This section provides simplified business use cases to help you understand different ways that you can implement AWS Elemental MediaConnect to deliver content to the AWS Cloud and beyond. The use cases in this section are described in general terms, without the mechanics of how you would use the MediaConnect API to achieve the results that you want.

Your MediaConnect implementation is dependent on your use case:

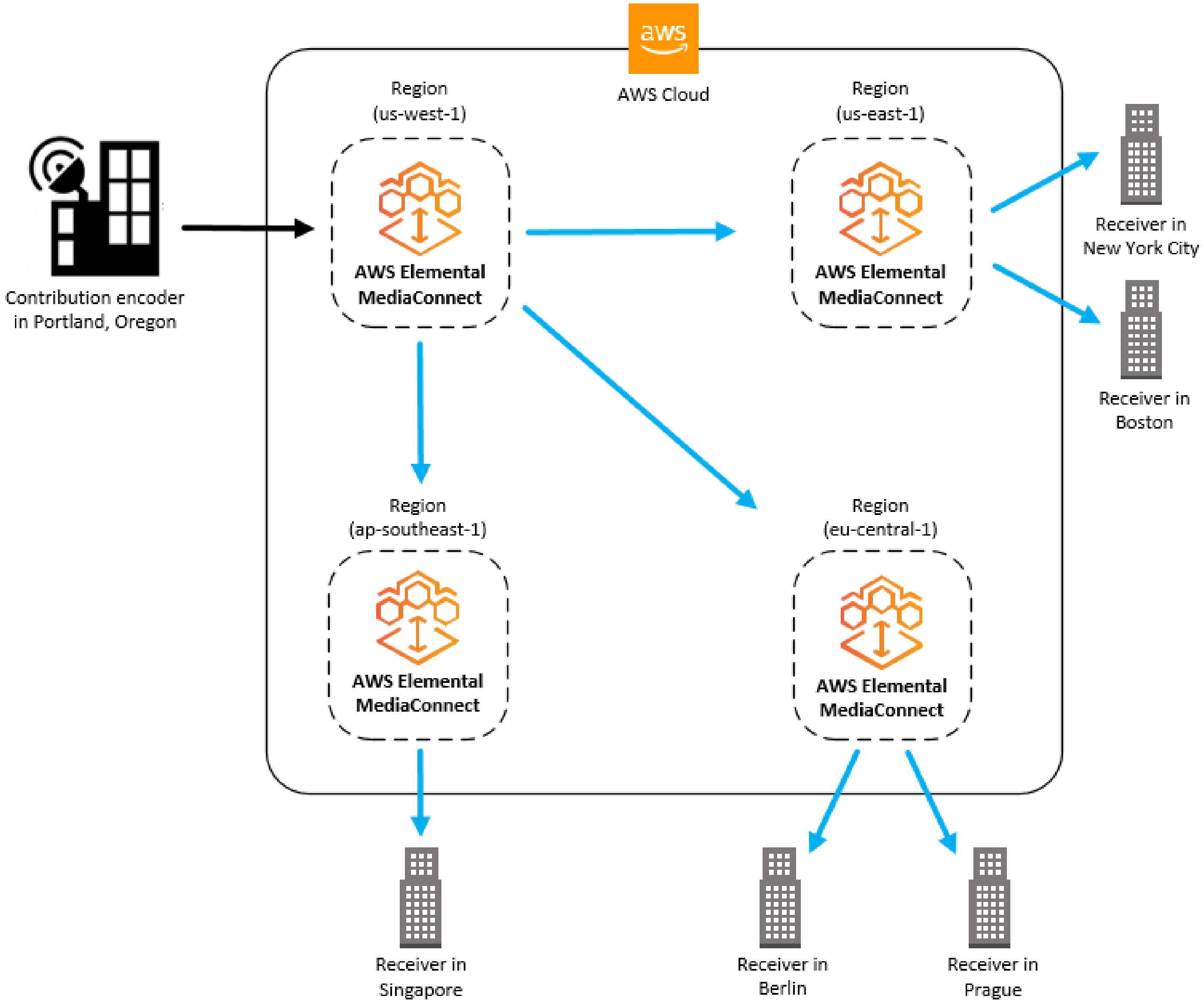
- For **contribution**, use MediaConnect to ingest content from an on-premises encoder into the AWS Cloud. Depending on the type of content you are ingesting, you can create a transport stream flow or a CDI flow.
- For **distribution**, use MediaConnect to deliver content to different geographical areas.
- For **entitlements**, use MediaConnect to share your content with other AWS accounts.
- For **replication and monitoring**, use MediaConnect to distribute video to multiple destinations and enable the monitoring of multiple video signals in real time.

**Topics**

## Use case: distribution

You can use AWS Elemental MediaConnect to distribute your content to different geographical locations. For example, suppose that your on-premises contribution encoder is located in Portland, Oregon and your receivers are located around the world. (A receiver is any entity that will receive content from your flow. This could be an encoder in the cloud, an on-premises encoder at your recipient facility, or another MediaConnect flow.) You set up your initial MediaConnect flow in the us-west-1 Region, which is the closest physical AWS Region to your encoder. After your content is in the AWS Cloud, you send it to other MediaConnect flows located in Regions that are closer to your receivers.

The following illustration shows an on-premises contribution encoder located in Portland, Oregon that uploads content to MediaConnect in the AWS Cloud. The flow has three outputs that send content to others flows in different AWS Regions. These secondary flows are closer to the receivers, which are located in various cities around the world.

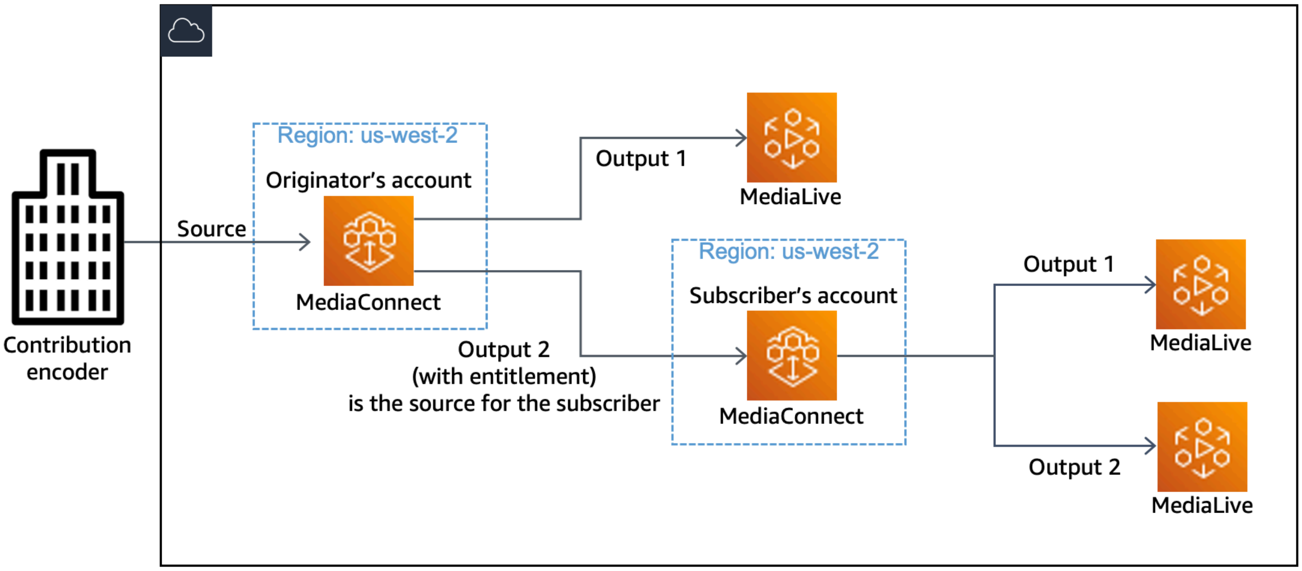

aws
AWS Cloud
Contribution encoder in Portland, Oregon
Region (us-west-1)
AWS Elemental MediaConnect
Region (us-east-1)
AWS Elemental MediaConnect
Receiver in New York City
Receiver in Boston
Region (ap-southeast-1)
AWS Elemental MediaConnect
Region (eu-central-1)
AWS Elemental MediaConnect
Receiver in Singapore
Receiver in Berlin
Receiver in Prague

## Use case: entitlements

Entitlements allow one AWS account holder to share content in a transport stream flow with other AWS account holders. For example, a sports company wants to share a flow (Baseball-Game) with a local TV station. A sports broadcaster (the originator) creates an entitlement on the Baseball-Game flow to allow access for the local TV station (the subscriber). The local TV station creates an AWS Elemental MediaConnect flow using an output from the Baseball-Game flow as the source.

The subscriber must set up their flow in MediaConnect in the same Region as the originator's flow.

This following illustration shows how to share content in a transport stream flow with another AWS subscriber. The output of the originator's flow can be used as the source of the subscriber's flow.

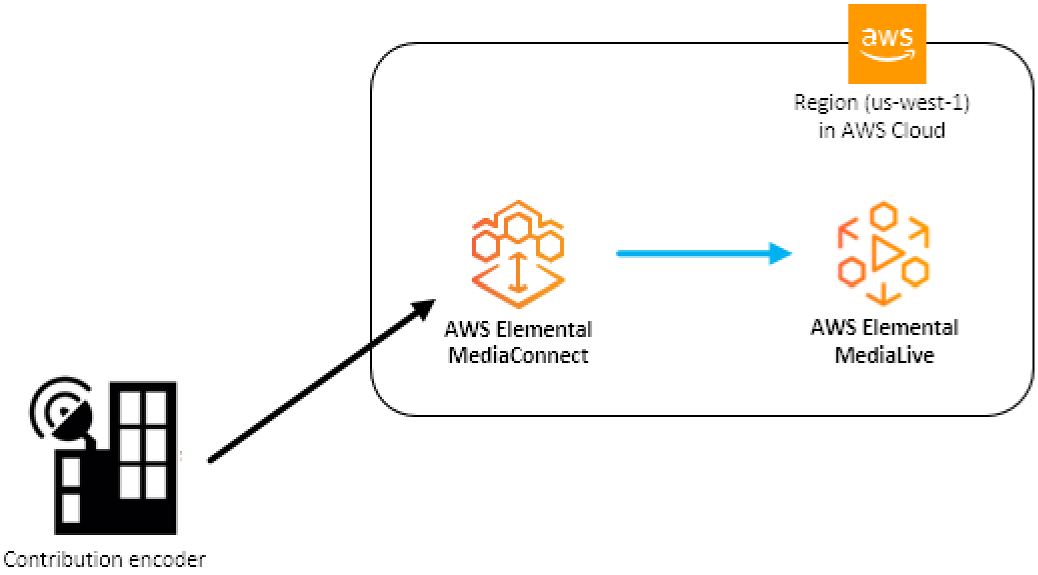


## Use case: contribution for transport stream flows

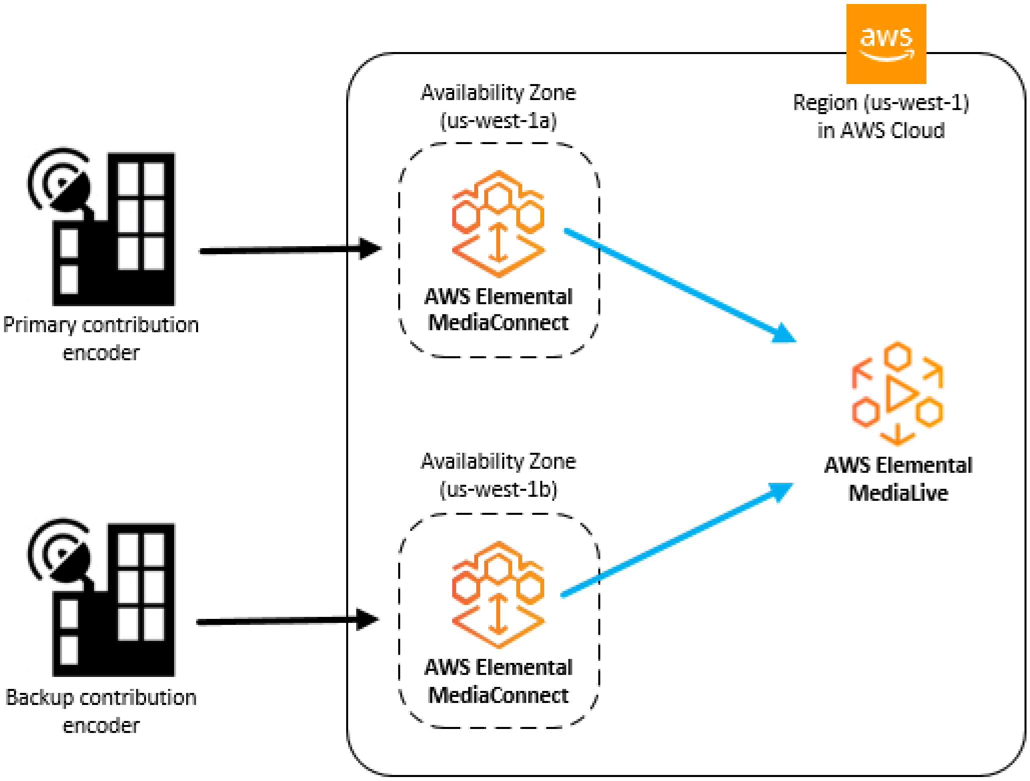
You can use AWS Elemental MediaConnect to ingest your content from an on-premises contribution encoder into the AWS Cloud. The source for your MediaConnect flow comes from your on-premises contribution encoder, and the output points to your encoder in the cloud, such as AWS Elemental MediaLive. If your source content is uncompressed, you can use a CDI workflow.

For redundancy, you can set up your flow to have two outputs that point to your cloud encoder. Another setup for redundancy includes two on-premises contribution encoders—a primary and a backup—that each send content to a different MediaConnect flow. The output from each flow then points to the same cloud encoder.

The following illustration shows an on-premises contribution encoder that uploads content to MediaConnect in the AWS Cloud. The flow output points to an MediaLive channel.



The following illustration shows two on-premises contribution encoders, a primary and a backup, that upload the same content to MediaConnect in the AWS Cloud. There are two flows, each with one output. Both outputs point to a single MediaLive channel.


aws
Region (us-west-1) in AWS Cloud
Availability Zone (us-west-1a)
AWS Elemental MediaConnect
Availability Zone (us-west-1b)
AWS Elemental MediaConnect
AWS Elemental MediaLive
Primary contribution encoder
Backup contribution encoder

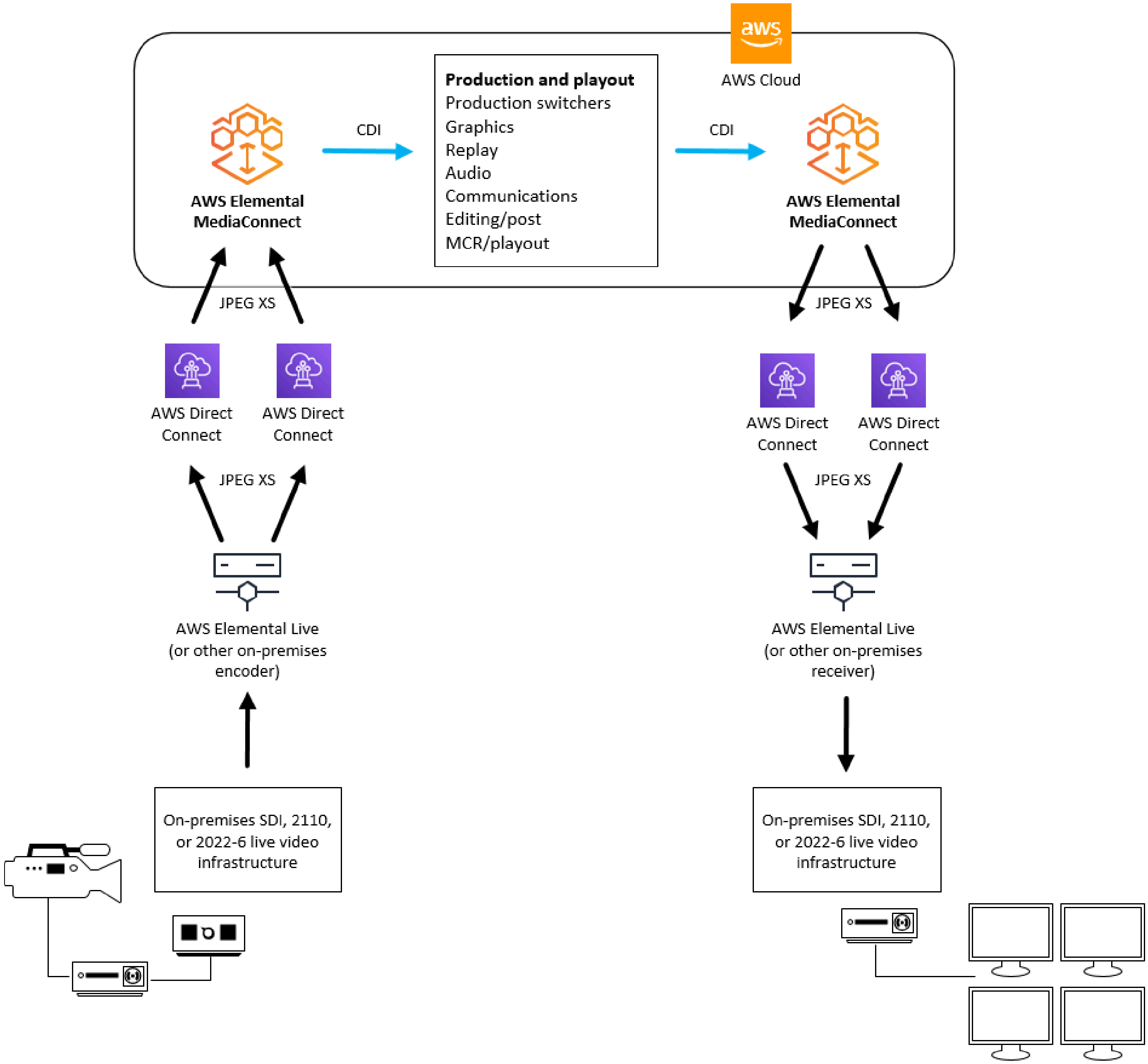
# Use case: Contribution for CDI flows

With AWS Elemental MediaConnect and AWS Direct Connect, you can bridge your on-premises live video network (SDI, 2022-6, or 2110) to your VPC live video network (CDI). MediaConnect uses the JPEG XS codec to reduce your AWS Direct Connect network bandwidth significantly. MediaConnect supports SMPTE 2110 standard (parts 22, 30, and 40) for video, audio, and metadata transfer. MediaConnect converts the content to CDI streams that are ready to be consumed by other services in the cloud, such as AWS Elemental MediaLive. When your cloud VPC content is ready to be distributed back to on-premises networks, you can use MediaConnect to convert the CDI streams back to the SMPTE 2110 standard (parts 22, 30, and 40) for transport.

For redundancy, when you transport content between your on-premises configuration and the AWS Cloud, set up two connections in AWS Direct Connect. Be sure to configure the AWS Elemental Live appliance with settings to match the MediaConnect flows. For more information about configuring the appliance, see SMPTE 2110 inputs and outputs in the *AWS Elemental Live User Guide*.

> **Note**
>
> Because CDI outputs don't support inter-Availability Zone transfers, use ST 2110 JPEG XS outputs if you want to send content to a different Availability Zone.

The following illustration shows a workflow that creates a bridge between your on-premises live video infrastructure and the AWS Cloud.

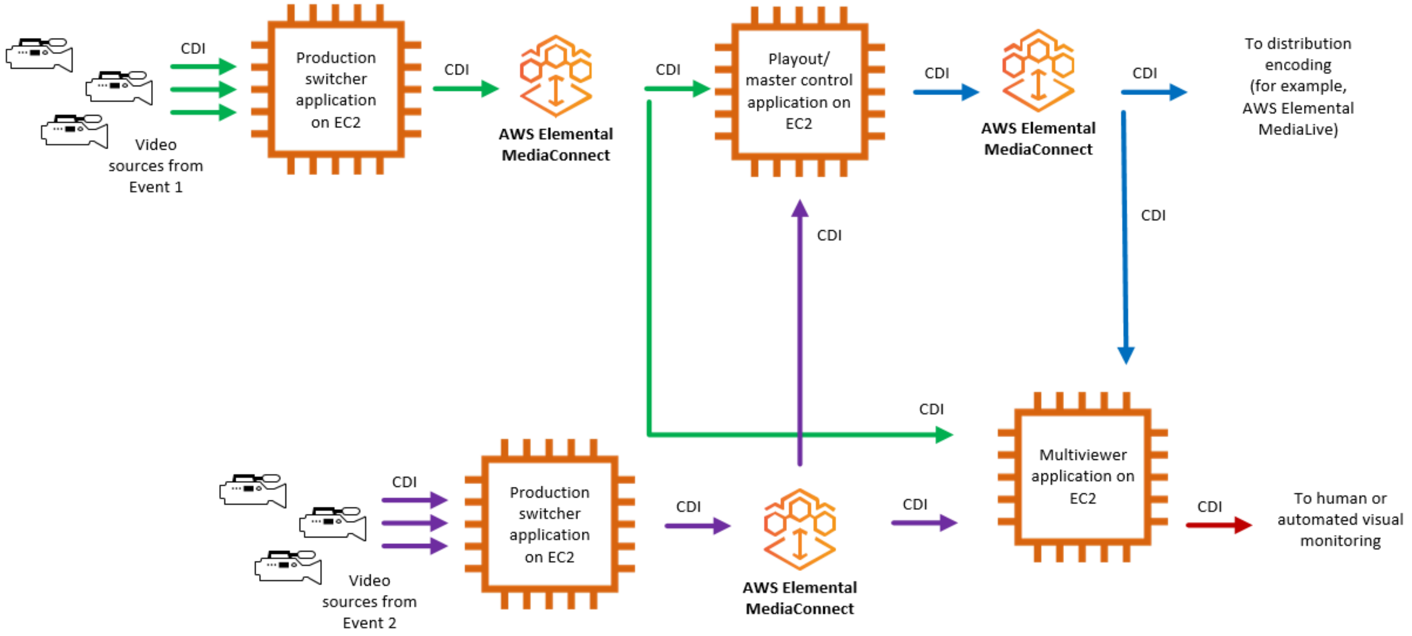

AWS Cloud
AWS Elemental MediaConnect
CDI
Production and playout
Production switchers
Graphics
Replay
Audio
Communications
Editing/post
MCR/playout
CDI
AWS Elemental MediaConnect
JPEG XS
AWS Direct Connect
AWS Direct Connect
JPEG XS
AWS Elemental Live (or other on-premises encoder)
On-premises SDI, 2110, or 2022-6 live video infrastructure
JPEG XS
AWS Direct Connect
AWS Direct Connect
JPEG XS
AWS Elemental Live (or other on-premises receiver)
On-premises SDI, 2110, or 2022-6 live video infrastructure

# Use case: replication and monitoring for CDI flows

You can use AWS Elemental MediaConnect to replicate and distribute video to multiple destinations and monitor the multiple video signals in real time.

For example, you can switch between multiple live events that are happening at different venues to create a single output broadcast. Using a MediaConnect CDI workflow, you can take the outputs from multiple production switchers and send those to a master control switcher and a multiviewer application. You can use another CDI flow to send the final output to the distribution encoder (for example, AWS Elemental MediaLive), and also to the multiviewer application. The production team receives the output from the multiviewer, which enables them to monitor the multiple video signals in real time.

The following illustration shows how you can use MediaConnect CDI workflows to replicate and distribute video to multiple destinations. You can create a single output broadcast from video content coming from multiple events, and also send the output from multiple signals for monitoring in real time.


CDI
Video sources from Event 1
Production switcher application on EC2
CDI
AWS Elemental MediaConnect
CDI
Playout/ master control application on EC2
CDI
AWS Elemental MediaConnect
CDI
To distribution encoding (for example, AWS Elemental MediaLive)
CDI
CDI
CDI
CDI
Video sources from Event 2
Production switcher application on EC2
CDI
AWS Elemental MediaConnect
CDI
Multiviewer application on EC2
CDI
To human or automated visual monitoring

# Setting up AWS Elemental MediaConnect

Before you start using AWS Elemental MediaConnect, you must sign up for AWS (if you don't already have an AWS account) and create IAM users and roles to allow access to MediaConnect. This includes creating an IAM role for yourself. If you want to use encryption to protect your content, you also must store your encryption keys in AWS Secrets Manager, and then give MediaConnect permission to obtain the keys from your Secrets Manager account.

This section guides you through the steps required to configure users and roles to access AWS Elemental MediaConnect. For background and additional information about identity and access management for MediaConnect, see the section called “Identity and access management”.

**Topics**

## Sign Up for AWS

### Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

**To sign up for an AWS account**

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

   When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to an administrative user, and use only the root user to perform tasks that require root user access.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing **My Account**.

## Create an administrative user

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

### Secure your AWS account root user

1. Sign in to the AWS Management Console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   For help signing in by using root user, see Signing in as the root user in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

   For instructions, see Enable a virtual MFA device for your AWS account root user (console) in the *IAM User Guide*.

### Create an administrative user

1. Enable IAM Identity Center.

   For instructions, see Enabling AWS IAM Identity Center in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to an administrative user.

   For a tutorial about using the IAM Identity Center directory as your identity source, see Configure user access with the default IAM Identity Center directory in the *AWS IAM Identity Center User Guide*.

### Sign in as the administrative user

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

For help signing in using an IAM Identity Center user, see Signing in to the AWS access portal in the *AWS Sign-In User Guide*.

# Create non-admin roles

Users in the Administrators group for an account have access to all AWS services and resources in that account. Granting direct access to all AWS resources goes against the best practice of applying the least privileged permissions to a user. This section describes how you can create roles with permissions that are limited to AWS Elemental MediaConnect. This section also describes how your users can assume that role to grant secure and temporary credentials.

**Topics**

## Step 1: Create a non-admin policy

Create two policies for AWS Elemental MediaConnect: one to provide read/write access and one to provide read-only access. Perform these steps one time only for each policy. Later, you will attach these policies to roles. Those roles can then be temporarily assumed by users to grant access to MediaConnect.

**To create policies**

1. Use your AWS account ID or account alias, and the credentials for your admin user, to sign in to the IAM console.
2. In the navigation pane of the console, choose **Policies**.
3. On the **Policies** page, create a policy named `MediaConnectAllAccess` that allows all actions on all resources in AWS Elemental MediaConnect:

   a. Choose **Create policy**.
   b. Choose the **JSON** tab and paste the following policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "mediaconnect:*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "ec2:DescribeAvailabilityZones"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
         {
            "Action": [
                "cloudwatch:GetMetricData"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
           "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "mediaconnect.amazonaws.com"
                }
            }
        }
    ]
}
```

This policy allows all actions on all resources in AWS Elemental MediaConnect.

c. Choose **Next: Tags**.

d. Choose **Next: Review**.

e. On the **Review and create** page, for **Policy name**, enter **`MediaConnectAllAccess`**, and then choose **Create policy**.

4. On the **Policies** page, create a read-only policy named `MediaConnectReadOnlyAccess` for AWS Elemental MediaConnect:

   a. Choose **Create policy**.

   b. Choose the **JSON** tab and paste the following policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "mediaconnect:List*",
                "mediaconnect:Describe*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "ec2:DescribeAvailabilityZones"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
         {
            "Action": [
                "cloudwatch:GetMetricData"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
           "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "mediaconnect.amazonaws.com"
                }
            }
        }
    ]
} .
```

c. Choose **Next: Tags**.

d. Choose **Next: Review**.

e. On the **Review and create** page, for **Policy name**, enter **`MediaConnectReadOnlyAccess`**, and then choose **Create policy**.

## Step 2: Create non-admin roles

You can create a role for each policy and users can assume that role, rather than attaching individual policies to each user. Using the following procedure, create two roles: one for the **MediaConnectAllAccess** policy and one for the **MediaConnectReadOnlyAccess** policy.

**To create roles**

1. In the navigation pane of the IAM console, choose **Roles**.
2. On the **Roles** page, create an administrator role using the `MediaConnectAllAccess` policy:

   a. Choose **Create role**.

   b. In the **Select trusted entity** section, select **AWS account**.

   c. In the **An AWS account** section, select the account with the users that will be assuming this role.

      i. If a third-party will be accessing this role, it is a best practice to select **Require external ID**. For more information about external IDs, visit: Using an external ID for third-party access in the *IAM User Guide*.

      ii. It is a best practice to require multi-factor authentication (MFA). You can select the checkbox next to **Require MFA**. For more information about MFA, visit: Multi-factor authentication (MFA) in the *IAM User Guide*.

   d. Choose **Next** to move to the **Add permissions** section.

   e. In the **Permissions policy** section, choose the **MediaConnectAllAccess** policy that you created in the procedure in Step 3a: Create a Policy.

   f. Verify that the correct policies are added to this group, and then choose **Next**.

   g. In the **Name, review and create** section, name the role `MediaConnectAdmins`. (Optional) Add a description for the role. Select **Create role**.

3. On the **Roles** page, create an administrator role using the `MediaConnectReadOnlyAccess` policy:

    a. Choose **Create role**.

    b. In the **Select trusted entity** section, select **AWS account**.

    c. In the **An AWS account** section, select the account with the users that will be assuming this role.

        i. If a third-party will be accessing this role, it is a best practice to select **Require external ID**. For more information about external IDs, visit: Using an external ID for third-party access in the *IAM User Guide*.

        ii. It is a best practice to require multi-factor authentication (MFA). You can select the checkbox next to **Require MFA**. For more information about MFA, visit: Multi-factor authentication (MFA) in the *IAM User Guide*.

    d. Choose **Next** to move to the **Add permissions** section.

    e. In the **Permissions policy** section, choose the **MediaConnectReadOnlyAccess** policy that you created in the procedure in Step 3a: Create a Policy.

    f. Verify that the correct policies are added to this group, and then choose **Next**.

    g. In the **Name, review and create** section, name the role `MediaConnectReaders`. (Optional) Add a description for the role. Select **Create role**.

## Step 3: Assume the role

After creating a policy and attaching that policy to a role, your users will need to assume that role to be granted secure and temporary access to MediaConnect.

View the following resources for learning about granting permissions for users to assume the role and how users can switch to the role from the console or AWS CLI.

- Granting a user permissions to switch roles: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_permissions-to-switch.html
- Switching roles (console): https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-console.html
- Switching roles (AWS CLI): https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-cli.html

# (Optional) Set up encryption

You can protect your content from unauthorized use through encryption. If your source is encrypted, AWS Elemental MediaConnect can decrypt it. In addition, the service can encrypt outputs and entitlements. AWS Elemental MediaConnect offers two options for encrypting content: static key and Secure Packager and Encoder Key Exchange (SPEKE). The steps to set up encryption depend on the type of encryption that you choose. For more information, see the following:

- Setting up static key encryption using AWS Elemental MediaConnect
- Setting up SPEKE encryption using AWS Elemental MediaConnect

# (Optional) Install the AWS CLI

To use the AWS CLI with AWS Elemental MediaConnect, install the latest AWS CLI version. For information about installing the AWS CLI or upgrading it to the latest version, see Installing the AWS Command Line Interface in the *AWS Command Line Interface User Guide*.

# Getting started with AWS Elemental MediaConnect

This Getting Started tutorial shows you how to use AWS Elemental MediaConnect to create and share flows. The tutorial is based on a scenario where you want to do all of the following:

- Ingest a live video stream of an awards show that is taking place in New York City.
- Distribute your video to an affiliate in Boston who does not have an AWS account, and wants content sent to their on-premises encoder.
- Share your video with an affiliate in Philadelphia who wants to use their AWS account to distribute the video to their three local stations.

**Topics**

## Prerequisites

Before you can use AWS Elemental MediaConnect, you need an AWS account and the appropriate permissions to access, view, and edit MediaConnect components. Complete the steps in Setting up AWS Elemental MediaConnect, and then return to this tutorial.

## Step 1: Access AWS Elemental MediaConnect

After you set up your AWS account and create IAM roles, you sign in to the console for AWS Elemental MediaConnect.

**To access AWS Elemental MediaConnect**

- Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.

## Step 2: Create a flow

First, you create an AWS Elemental MediaConnect flow to ingest your video from your on-premises encoder into the AWS Cloud. For the purposes of this tutorial, we use the following details:

- Flow name: AwardsNYCShow
- Source name: AwardsNYCSource
- Source protocol: Zixi push
- Zixi stream ID: ZixiAwardsNYCFeed
- CIDR block sending the content: 10.24.34.0/23
- Source encryption: None

**To create a flow**

1. On the **Flows** page, choose **Create flow**.
2. In the **Details** section, for **Name**, enter **`AwardsNYCShow`**.
3. For **Availability Zone**, choose **Any**.
4. In the **Source** section, for **Source type** select **Standard source**.
5. For **Name**, enter **`AwardsNYCSource`**.
6. For **Protocol**, choose **Zixi push**. AWS Elemental MediaConnect will populate the value of the ingest port.
7. For **Stream ID**, enter **`ZixiAwardsNYCFeed`**.
8. For **Allowlist CIDR**, enter **`10.24.34.0/23`**.
9. Choose **Create flow**.

## Step 3: Add an output

To send content to your affiliate in Boston, you must add an output to your flow. This output will send your video to your Boston affiliate's on-premises encoder. For the purposes of this tutorial, we use the following details:

- Output name: AwardsNYCOutput
- Output protocol: Zixi push
- Zixi stream ID: ZixiAwardsOutput

- IP address of the Boston affiliate's on-premises encoder: 198.51.100.11
- Output encryption: None

**To add an output**

1. On the **Flows** page, choose the **`AwardsNYCShow`** flow.
2. Choose the **Outputs** tab.
3. Choose **Add output**.
4. For **Name**, enter **`AwardsNYCOutput`**.
5. For **Output type**, select **Standard output**.
6. For **Protocol**, choose **Zixi push**.
7. For **Stream ID**, enter **`ZixiAwardsOutput`**.
8. For **Destination IP address**, enter **`198.51.100.11`**.
9. For **Port**, enter **`1024`**.
10. Choose **Add output**.

## Step 4: Grant an entitlement

You must grant an entitlement to allow your Philadelphia affiliate to use your content as the source for their AWS Elemental MediaConnect flow. For purposes of this tutorial, we use the following details:

- Entitlement name: PhillyTeam
- Philadelphia affiliate's AWS account ID: 222233334444
- Output encryption: None

**To grant an entitlement**

1. Choose the **Entitlements** tab.
2. Choose **Grant entitlement**.
3. For **Name**, enter **`PhillyTeam`**.
4. For **Subscriber**, enter **`222233334444`**.
5. Choose **Grant entitlement**.

## Step 5: Share details with your affiliates

Now that you've created your AWS Elemental MediaConnect flow with an output for your Boston affiliate and an entitlement for your Philadelphia affiliate, you need to communicate details about the flow.

Your Boston affiliate will receive the flow on their on-premises encoder. The details of where to send your video stream were provided by your Boston affiliate, and you don't need to provide any other information. After you start your flow, the content will be sent to the IP address that you specified when you created the flow.

Your Philadelphia affiliate must create their own AWS Elemental MediaConnect flow, using your flow as the source. You must provide the following information to your Philadelphia affiliate:

- Entitlement ARN: You can find this value on the **Entitlement** tab of the **AwardsNYCShow** flow details page.
- Region: This is the AWS Region that you created the **AwardsNYCShow** flow in.

## Step 6: Clean up

To avoid extraneous charges, be sure to delete all unnecessary flows. You must stop the flow before it can be deleted.

**To stop your flow**

1. On the **Flows** page, choose the **`AwardsNYCShow`** flow.

   The details page for the **AwardsNYCShow** flow appears.
2. Choose **Stop**.

**To delete your flow**

1. On the **AwardsNYCShow** flow details page, choose **Delete**.

   A confirmation message appears.
2. Choose **Delete flow**.

# Flows in AWS Elemental MediaConnect

A flow is a transport between a source and one or more destinations. When you create a flow, you specify the source, a name, and an Availability Zone. After you create a flow, you can add outputs to indicate where you want your content to be sent and how you want it transported.

MediaConnect supports two types of flows:

- **Transport stream flows** transport compressed content that is muxed (audio, video, and ancillary data are combined) into a single stream. The quality is high enough to use as a source for creating final encodes that are delivered to consumer devices. You can add outputs to indicate where you want the content to be sent and how you want it transported.

  You can grant an entitlement to share the content with another AWS account. A user of the subscriber account can then create a new MediaConnect flow using your flow as the source. When this happens, the service generates an output on your flow to represent the stream that feeds the subscriber's flow.

  It is important to manage the number of outputs and entitlements on the flow. Each transport stream flow can only have 50 outputs. Although you can grant up to 50 entitlements on a flow, each of those entitlements will generate an output. For example, you create a flow named **`BasketballGame`** and you add 40 outputs that send content to on-premises encoders. You also grant 30 entitlements to share your content with other AWS accounts. When your subscribers create flows using **`BasketballGame`** as their source, the service generates new outputs for each of those subscribers. After the first 10 subscribers create flows, your **`BasketballGame`** flow reaches its maximum number of outputs (40 for the original outputs that you created and another 10 that the service created for the subscribing flows). When the 11th subscriber tries to create a flow using **`BasketballGame`** as a source, the service returns an error.
- **CDI flows** transport high-quality uncompressed or lightly compressed content into and out of the AWS Cloud. You can configure a CDI flow to use JPEG XS to transport lightly compressed content. The content is demuxed into separate media streams for audio, video, or ancillary data. Each CDI flow can use multiple media streams for the source and multiple media streams for each output. MediaConnect uses AWS Cloud Digital Interface (AWS CDI) network technology to transport content that adheres to the SMPTE 2110, part 22 transport standard.

**Topics**

- Viewing a list of flows
- Viewing the details of a flow
- Starting a flow
- Stopping a flow
- Updating a flow
- Managing tags on a flow
- Deleting a flow

# Creating a flow

A flow is a connection between one or more sources and one or more outputs or entitlements.

The method that you use to create a flow is dependent on the type of flow that you want to create and the type of content in the source:

- Transport stream flow with a standard source – Uses content from any source that is not a VPC source or an entitled source.
- Transport stream flow with an entitled source – Uses content that is owned by another AWS account that has granted an entitlement to your account.
- Transport stream flow with a VPC source – Uses compressed content that comes from a VPC that you configure.
- CDI flow – Uses uncompressed content that comes from a VPC that you configure.

> **Note**
>
> If you want to create a transport stream flow that uses redundant sources for failover, create the flow with one of the sources. After the flow is created, add the other source. Because MediaConnect treats both sources as the primary source, it doesn't matter which one you specify when you first create the flow. If your flow uses an entitled source, you can't add a second source. For redundancy with CDI workflows, create two separate flows.

## Creating a transport stream flow that uses a standard source

Transport stream flows transport compressed content that is muxed into a single stream.

A flow uses a *standard* source when the content comes from anywhere other than a VPC (VPC source) or another AWS account (entitled source).

⚠ **Important**

If the source of your flow requires encryption, set up encryption before you begin this procedure.

## Create a transport stream flow that uses a standard source (console)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose **Create flow**.
3. In the **Details** section, for **Name**, specify a name for your flow. This name will become part of the ARN for this flow.

   ⓘ **Note**

   MediaConnect allows you to create multiple flows with the same name. However, we encourage you to use unique flow names within an AWS Region to help with organization. After you create a flow, you can't change the name.
4. For **Availability Zone**, choose an Availability Zone for your flow. Use this option when you are setting up redundant flows. Otherwise, you can leave this as **Any**. If you leave the default, the service will randomly assign an Availability Zone within the current AWS Region, or if your source comes from a VPC, the service will assign the Availability Zone of the VPC subnet to the flow.
5. Determine which protocol your source uses.

   ⓘ **Note**

   If you want to specify redundant sources for failover, create the flow with one of the sources. After the flow is created, update it to activate failover on the source, and add the second source to the flow. Because MediaConnect treats both sources as the primary source, it doesn't matter which one you specify when you first create the flow.
6. For specific instructions based on your source type and protocol, choose one of the following tabs:

RIST

1. In the **Source** section, for **Source type**, choose **Standard source**.
2. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console.
3. For **Protocol**, choose **RIST**.
4. For **Ingest port**, specify the port that the flow will listen on for incoming content.

   > **Note**
   >
   > The RIST protocol requires one additional port for error correction. To accommodate this requirement, MediaConnect reserves the port that is +1 from the port that you specify. For example, if you specify port 4000 for the output, the service assigns ports 4000 and 4001.

5. For **Allowlist CIDR**, specify a range of IP addresses that are allowed to contribute content to your source. Format the IP addresses as a Classless Inter-Domain Routing (CIDR) block, for example, 10.24.34.0/23. For more information about CIDR notation, see RFC 4632.

   > **Important**
   >
   > Specify a CIDR block that is as precise as possible. Include only the IP addresses that you want to contribute content to your flow. If you specify a CIDR block that is too wide, it allows for the possibility of outside parties sending content to your flow.

6. For **Maximum bitrate**, specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.
7. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 1-15,000 ms. If you keep this field blank, the service uses the default value of 2,000 ms.

RTP or RTP-FEC

1. In the **Source** section, for **Source type**, choose **Standard source**.
2. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console. It is not visible to anyone outside of the current AWS account.
3. For **Protocol**, choose **RTP** or **RTP-FEC**.
4. For **Ingest port**, specify the port that the flow will listen on for incoming content.

   **Note**

   The RTP-FEC protocol requires two additional ports for error correction. To accommodate this requirement, MediaConnect reserves the ports that are +2 and +4 from the port that you specify. For example, if you specify port 4000 for the output, the service assigns ports 4000, 4002, and 4004.

5. For **Allowlist CIDR**, specify a range of IP addresses that are allowed to contribute content to your source. Format the IP addresses as a Classless Inter-Domain Routing (CIDR) block, for example, 10.24.34.0/23. For more information about CIDR notation, see RFC 4632.

   **Important**

   Specify a CIDR block that is as precise as possible. Include only the IP addresses that you want to contribute content to your flow. If you specify a CIDR block that is too wide, it allows for the possibility of outside parties sending content to your flow.

6. For **Maximum bitrate**, specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.

SRT listener

1. In the **Source** section, for **Source type**, choose **Standard source**.

2. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console. It is not visible to anyone outside of the current AWS account.
3. For **Protocol**, choose **SRT listener**.
4. For **Source description**, enter a description that will remind you later where this source is from. This might be the company name or notes about the setup.
5. For **Allowlist CIDR block**, specify a range of IP addresses that are allowed to contribute content to your source. Format the IP addresses as a Classless Inter-Domain Routing (CIDR) block, for example, 10.24.34.0/23. For more information about CIDR notation, see RFC 4632.

   > ⚠ **Important**
   >
   > Specify a CIDR block that is as precise as possible. Include only the IP addresses that you want to contribute content to your flow. If you specify a CIDR block that is too wide, it allows for the possibility of outside parties sending content to your flow.

6. For **Inbound port**, specify the port that the flow listens on for incoming content.
7. For **Source listener address**, enter the address MediaConnect will use for the SRT connection. The address can be an IP address or a domain name.
8. For **Source description**, enter a description that will remind you later where this source is from. This might be the company name or notes about the setup.
9. For **Maximum bitrate**, specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.
10. For **Minimum latency**, specify the minimum size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 100–15,000 ms. If you keep this field blank, MediaConnect uses the default value of 2,000 ms.
11. If the source is encrypted, choose **Activate** in the **Decryption** section and do the following:

    a. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.

b. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.

SRT caller

1. In the **Source** section, for **Source type**, choose **Standard source**.
2. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console. It is not visible to anyone outside of the current AWS account.
3. For **Protocol**, choose **SRT caller**.
4. For **Source description**, enter a description that will remind you later where this source is from. This might be the company name or notes about the setup.
5. For **Source listener address**, enter the address MediaConnect will use for the SRT connection. The address can be an IP address or a domain name.
6. For **Source listener port**, enter the port MediaConnect will use for the SRT connection.
7. For **Maximum bitrate** (optional), specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.
8. For **Minimum latency**, specify the minimum size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 100–15,000 ms. If you keep this field blank, MediaConnect uses the default value of 2,000 ms.
9. For **Stream ID** (optional), enter an identifier for the stream. This identifier can be used to communicate information about the stream.
10. If the source is encrypted, choose **Activate** in the **Decryption** section and do the following:

    a. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.

    b. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.

### Zixi push

1. In the **Source** section, for **Source type**, choose **Standard source**.
2. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console. It is not visible to anyone outside of the current AWS account.
3. For **Protocol**, choose **Zixi push**.

   > **Note**
   >
   > MediaConnect assigns the inbound port for Zixi push sources at the time of creation. A port number of 2088 will be assigned automatically.

4. For **Allowlist CIDR**, specify a range of IP addresses that are allowed to contribute content to your source. Format the IP addresses as a Classless Inter-Domain Routing (CIDR) block, for example, 10.24.34.0/23. For more information about CIDR notation, see RFC 4632.

   > **Important**
   >
   > Specify a CIDR block that is as precise as possible. Include only the IP addresses that you want to contribute content to your flow. If you specify a CIDR block that is too wide, it allows for the possibility of outside parties sending content to your flow.

5. For **Stream ID**, specify the stream ID set in the Zixi feeder.

   > **Important**
   >
   > If you leave this field blank, the service uses the source name as the stream ID. Because the stream ID must match the value set in the Zixi feeder, you need to specify the stream ID if it is not exactly the same as the source name.

6. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less

room for error correction. You can choose a value between 0 and 60,000 ms. If you keep this field blank, the service uses the default value of 6,000 ms.

7. If the source is encrypted, choose **Activate** in the **Decryption** section and do the following:
   a. For **Decryption type**, choose **Static key**.
   b. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.
   c. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.
   d. For **Decryption algorithm**, choose the type of encryption that was used to encrypt the source.

### Zixi push for AWS Elemental Link UHD device

To use an AWS Elemental Link device as a source for MediaConnect, you must create a Zixi push flow using the following procedure. After creating the Zixi push flow, you must configure the AWS Elemental Link device using MediaLive. See the following MediaLive setup instructions to complete the process after you have created the flow: Using a device in a flow in the *MediaLive User Guide*. Ensure you have access to both MediaConnect and MediaLive to complete these steps.

1. In the **Source** section, for **Source type**, choose **Standard source**.
2. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console. It is not visible to anyone outside of the current AWS account.
3. For **Protocol**, choose **Zixi push**.

> **Note**
>
> MediaConnect assigns the inbound port for Zixi push sources at the time of creation. A port number of 2088 will be assigned automatically.

4. For **Allowlist CIDR block**, specify a range of IP addresses that are allowed to contribute content to your source. Format the IP addresses as a Classless Inter-Domain Routing (CIDR) block, for example, 10.24.34.0/23. For more information about CIDR notation, see RFC 4632.

> ⚠ **Important**
>
> If you know the range of public IP addresses that your Link device uses to connect to the internet, enter that CIDR block. Note that this is not the same as the IP address of the AWS Elemental Link device. If you cannot obtain this information, it is possible to configure the CIDR block to be open to all possible IP addresses by using 0.0.0.0/0. Typically, it is not best practice to assign a CIDR block that is open to the entire internet (0.0.0.0/0). However, if this method must be used, the data being transferred is encrypted using AES-128 encryption.

5. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value between 0 and 60,000 ms. If you keep this field blank, the service uses the default value of 6,000 ms. The **Maximum latency** value should match the **Latency** value configured on the AWS Elemental Link device. For information on configuring the Link device's latency, see: Configuring the device in the *AWS Elemental MediaLive User Guide*
6. For **Decryption**, choose **Activate** and do the following:
   a. For **Decryption type**, choose **Static key**.
   b. For **Decryption algorithm**, choose **AES-128**. AWS Elemental Link requires AES-128, do not select another algorithm.
   c. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.
   d. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.

Fujitsu-QoS

1. In the **Source** section, for **Source type**, choose **Standard source**.
2. For **Inbound port**, specify the port that the flow listens on for incoming content.
3. For **Source description**, enter a description that will remind you later where this source is from. This might be the company name or notes about the setup.

   4. For **Sender IP address**, specify the sender's IP address that you want the flow to make connection with. The flow communicates with the specified IP address to initiate connection with the sender.
   5. For **Sender control port**, specify the port that the flow uses to send outbound requests to initiate connection with the sender.
   6. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 300–2,000 ms. If you keep this field blank, MediaConnect uses the default value of 2,000 ms.

7. At the bottom of the page, choose **Create flow**.

   > **Note**
   >
   > The flow doesn't start automatically. You must start the flow manually.

8. Add outputs to specify where you want MediaConnect to send the content, or grant entitlements to allow users of other AWS accounts to subscribe to your content.

## Create a transport stream flow that uses a standard source (AWS CLI)

1. Create a JSON file that contains the details of the flow that you want to create.

   The following example shows the structure for the contents of the file:

```
{
  "Name": "AwardsShow",
  "Outputs": [
    {
      "Destination": "198.51.100.5",
      "Description": "RTP output",
      "Name": "RTPOutput",
      "Protocol": "rtp",
      "Port": 5020
    }
  ],
  "Source": {
    "Name": "AwardsShowSource",
    "Protocol": "rtp-fec",
    "AllowlistCidr": "10.24.34.0/23"
  }
}
```

2. In the AWS CLI, use the `create-flow` command:

   ```
   aws mediaconnect create-flow --cli-input-json file://rtp.json --profile PMprofile
   ```

   The following example shows the return value:

   ```
   {
     "Flow": {
       "EgressIp": "203.0.113.0",
       "AvailabilityZone": "us-east-1d",
       "Name": "AwardsShow",
       "Status": "STANDBY",
       "FlowArn": "arn:aws:mediaconnect:us-
   east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
       "Source": {
               "SourceArn": "arn:aws:mediaconnect:us-
   east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:AwardsShowSource",

               "Name": "AwardsShowSource",
               "IngestPort": 5000,
               "AllowlistCidr": "10.24.34.0/23",
               "IngestIp": "198.51.100.15",
               "Transport": {
                   "Protocol": "rtp-fec",
                   "MaxBitrate": 80000000
               }
           },
           "Entitlements": [],
           "Outputs": [
               {
                   "Port": 5020,
                   "Name": "AwardsShowOutput",
                   "OutputArn": "arn:aws:mediaconnect:us-
   east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:AwardsShowOutput",

                   "Description": "RTP-FEC Output",
                   "Destination": "198.51.100.5",
                   "Transport": {
                       "Protocol": "rtp",
   ```

```
                        "SmoothingLatency": 0
                    }
                }
            ]
        }
}
```

## Creating a transport stream flow that uses an entitled source

Transport stream flows transport compressed content that is muxed into a single stream. An entitled source is content that comes from another AWS account.

### Create a transport stream flow that uses an entitled source (console)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose **Create flow**.
3. In the **Details** section, for **Name**, specify a name for your flow. This name will become part of the ARN for this flow.

   > **Note**
   >
   > MediaConnect allows you to create multiple flows with the same name. However, we encourage you to use unique flow names within an AWS Region to help with organization. After you create a flow, you can't change the name.

4. For **Availability Zone**, choose an Availability Zone for your flow. Use this option when you are setting up redundant flows. Otherwise, you can leave this as **Any**. If you leave the default, the service will randomly assign an Availability Zone within the current AWS Region , or if your source comes from a VPC, the service will assign the Availability Zone of the VPC subnet to the flow.

   > **Note**
   >
   > If your source comes from your VPC, the Availability Zone of your flow must match that of your VPC subnet. We recommend that you leave this as **Any** and let the service ensure that the Availability Zone is set correctly.

5. In the **Source** section, for **Source type** choose **Entitled source**.

6. For **Entitlement ARN**, choose the appropriate entitlement. This list includes all entitlements that have been granted to you.

   > **Tip**
   >
   > You can click in this field and start entering the entitlement name. MediaConnect will filter the list to include only entitlements with a name that matches what you enter.

7. Choose **Create flow**.

   > **Note**
   >
   > The flow doesn't start automatically. You must start the flow manually.

8. Add outputs to specify where you want MediaConnect to send the content, or grant entitlements to allow users of other AWS accounts to subscribe to your content.

## Creating a transport stream flow that uses a VPC source

Transport stream flows transport compressed content that is muxed into a single stream.

When you create a flow that uses a source from your virtual private cloud (VPC), your content does not go over the public internet. This is useful for security reasons as well as reliability. You set up your VPC and then create a flow that has an interface to that VPC. Alternatively, you can create a flow based on an entitlement that another AWS account granted to allow you to use their content (entitled source) or a standard source.

> **Important**
>
> Before you begin this procedure, make sure that the following steps have been completed:
>
> - In Amazon VPC, set up your VPC and associated security groups. For more information about VPCs, see the Amazon VPC User Guide. For information about configuring security groups to work with your VPC interface, see Security group considerations.
> - In IAM, set up MediaConnect as a trusted service.
> - If the source of your flow requires encryption, set up encryption.

## Create a transport stream flow that uses a VPC source (console)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose **Create flow**.
3. In the **Details** section, for **Name**, specify a name for your flow. This name will become part of the ARN for this flow.

> **Note**
>
> MediaConnect allows you to create multiple flows with the same name. However, we encourage you to use unique flow names within an AWS Region to help with organization. After you create a flow, you can't change the name.

4. For **Availability Zone**, choose **Any** or choose the Availability Zone where your VPC subnet resides. We recommend that you leave this as **Any** and let the service ensure that the Availability Zone is set correctly.
5. In the **Source** section, for **Source type**, choose **VPC source**.
6. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console.
7. Determine which protocol your source uses.

> **Note**
>
> If you want to specify redundant sources for failover, create the flow with one of the sources. After the flow is created, update it to activate failover on the source, and add the second source to the flow. Because MediaConnect treats both sources as the primary source, it doesn't matter which one you specify when you first create the flow.

8. For specific instructions based on your protocol, choose one of the following tabs:

   RIST

   1. For **Protocol**, choose **RIST**.
   2. For **Ingest port**, specify the port that the flow will listen on for incoming content.

> **Note**
>
> The RIST protocol requires one additional port for error correction. To accommodate this requirement, MediaConnect reserves the port that is +1 from the port that you specify. For example, if you specify port 4000 for the output, the service assigns ports 4000 and 4001.

3. For **VPC interface name**, choose the name of the VPC interface that you want to use as the source.
4. For **Maximum bitrate**, specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.
5. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 1-15,000 ms. If you keep this field blank, the service uses the default value of 2,000 ms.

RTP or RTP-FEC

1. For **Protocol**, choose **RTP** or **RTP-FEC**.
2. For **Ingest port**, specify the port that the flow will listen on for incoming content.

> **Note**
>
> The RTP-FEC protocol requires two additional ports for error correction. To accommodate this requirement, MediaConnect reserves the ports that are +2 and +4 from the port that you specify. For example, if you specify port 4000 for the output, the service assigns ports 4000, 4002, and 4004.

3. For **VPC interface name**, choose the name of the VPC interface that you want to use as the source.
4. For **Maximum bitrate**, specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.

SRT listener

1. In the **Source** section, for **Source type**, choose **VPC source**.
2. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console. It is not visible to anyone outside of the current AWS account.
3. For **Protocol**, choose **SRT listener**.
4. For **Source description**, enter a description that will remind you later where this source is from. This might be the company name or notes about the setup.
5. For **VPC interface name**, choose the name of the VPC interface that you want to use as the source.
6. For **Inbound port**, specify the port that the flow listens on for incoming content.
7. For **Maximum bitrate**, specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.
8. For **Minimum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 100 -15,000 ms. If you keep this field blank, the service uses the default value of 2,000 ms.
9. If the source is encrypted, choose **Activate** in the **Decryption** section and do the following:

   a. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.

   b. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.

SRT caller

1. In the **Source** section, for **Source type**, choose **VPC source**.
2. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console. It is not visible to anyone outside of the current AWS account.
3. For **Protocol**, choose **SRT caller**.

4. For **Source description**, enter a description that will remind you later where this source is from. This might be the company name or notes about the setup.
5. For **VPC interface name**, choose the name of the VPC interface that you want to use as the source.
6. For **Source listener port**, enter the port the flow will use to pull the source from.
7. For **Maximum bitrate** (optional), specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.
8. For **Minimum latency**, specify the minimum size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 100–15,000 ms. If you keep this field blank, MediaConnect uses the default value of 2,000 ms.
9. For **Stream ID** (optional), enter an identifier for the stream. This identifier can be used to communicate information about the stream.
10. If the source is encrypted, choose **Activate** in the **Decryption** section and do the following:
    a. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.
    b. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.

Zixi push

1. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console. It is not visible to anyone outside of the current AWS account.
2. For **Protocol**, choose **Zixi push**.

> **Note**
>
> MediaConnect assigns the inbound port for Zixi push VPC sources at the time of creation. A port number 2090–2099 will be assigned automatically.

3. For **VPC interface name**, choose the name of the VPC interface that you want to use as the source.

4. For **Stream ID**, specify the stream ID set in the Zixi feeder.

> ⚠ **Important**
>
> If you leave this field blank, the service uses the source name as the stream ID. Because the stream ID must match the value set in the Zixi feeder, you need to specify the stream ID if it is not exactly the same as the source name.

5. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value between 0 and 60,000 ms. If you keep this field blank, the service uses the default value of 6,000 ms.

6. If the source is encrypted, choose **Activate** in the **Decryption** section and do the following:

   a. For **Decryption type**, choose **Static key**.

   b. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.

   c. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.

   d. For **Decryption algorithm**, choose the type of encryption that was used to encrypt the source.

Fujitsu-QoS

1. For **Protocol**, choose **Fujitsu-QoS**.

2. For **Inbound port**, specify the port that the flow listens on for incoming content.

3. For **VPC interface name**, choose the name of the VPC interface that you want to use as the source.

4. For **Source description**, enter a description that will remind you later where this source is from. This might be the company name or notes about the setup.

5. For **Sender IP address**, specify the sender's IP address that you want the flow to make connection with. The flow communicates with the specified IP address to initiate connection with the sender.
6. For **Sender control port**, specify the port that the flow uses to send outbound requests to initiate connection with the sender.
7. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 300–2,000 ms. If you keep this field blank, MediaConnect uses the default value of 2,000 ms.

9. For each VPC that you want to connect to the flow, do the following:

   1. In the **VPC interface** section, choose **Add VPC interface**.
   2. For **Name**, specify a name for your VPC interface. The name of the VPC interface must be unique within the flow.
   3. For **Role ARN**, specify the Amazon Resource Name (ARN) of the role that you created when you set up MediaConnect as a trusted service.
   4. For **VPC**, choose the ID of the VPC that you want to use.

   > **Note**
   >
   > If you don't see the VPC that you want in the list, verify that the VPC has been set up in Amazon Virtual Private Cloud and that you have IAM permissions to view the VPC.

   5. For **Subnet**, choose the VPC subnet that you want MediaConnect to use to set up your VPC configuration. You must choose at least one and can choose as many as you want.
   6. For **Security groups**, specify the VPC security groups that you want MediaConnect to use to set up your VPC configuration. You must choose at least one security group.

10. At the bottom of the page, choose **Create flow**.

> **Note**
>
> The flow doesn't start automatically. You must start the flow manually.

11. Add outputs to specify where you want MediaConnect to send the content, or grant entitlements to allow users of other AWS accounts to subscribe to your content.

## Creating a CDI flow

A CDI flow transports high-quality uncompressed or lightly compressed content into and out of the AWS Cloud. You can configure a CDI flow to use JPEG XS to transport lightly compressed content. The content is demuxed into separate media streams for audio, video, or ancillary data. Each CDI flow can use multiple media streams for the source and multiple media streams for each output. MediaConnect uses AWS Cloud Digital Interface (AWS CDI) network technology to transport content that adheres to the SMPTE 2110, part 22 transport standard.

CDI flows only support sources from a virtual private cloud (VPC) that you set up using Amazon VPC. You set up your VPC and then create a flow that has an interface to that VPC.

MediaConnect doesn't support two sources on CDI flows. For redundancy with ST 2110 JPEG XS sources, you can specify two inbound VPC interfaces on an individual media stream. For redundancy with CDI sources, create a second flow.

⚠ **Important**

Before you begin this procedure, make sure that the following steps have been completed:

- Review the suggested workflow shown in Contribution for CDI flows.
- In Amazon VPC, set up your VPC and associated security groups. For more information about VPCs, see the Amazon VPC User Guide. For information about configuring security groups to work with your VPC interface, see Security group considerations.
- In IAM, set up MediaConnect as a trusted service.

### Create an AWS CDI flow (console)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose **Create flow**.
3. In the **Details** section, for **Name**, specify a name for your flow. This name will become part of the ARN for this flow.

> **Note**
>
> MediaConnect allows you to create multiple flows with the same name. However, we encourage you to use unique flow names within an AWS Region to help with organization. After you create a flow, you can't change the name.

4. For **Availability Zone**, choose the Availability Zone where your VPC subnet resides.
5. In the **Source** section, for **Source type**, choose **VPC source**.
6. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console.
7. Skip to the **VPC interface** section.
8. For each VPC that you want to connect to the flow, do the following:

   1. Choose **Add VPC interface**.
   2. For **Name**, specify a name for your VPC interface. The name of the VPC interface must be unique within the flow.
   3. For **Type**, choose the type of network adapter that you want MediaConnect to use on this interface. If you want to use this interface for a CDI source or output, you must choose **EFA** as the type.
   4. For **Role ARN**, specify the Amazon Resource Name (ARN) of the role that you created when you set up MediaConnect as a trusted service.
   5. For **VPC**, choose the ID of the VPC that you want to use.

   > **Note**
   >
   > If you don't see the VPC that you want in the list, verify that the VPC has been set up in Amazon Virtual Private Cloud and that you have IAM permissions to view the VPC.

   6. For **Subnet**, choose the VPC subnet that you want MediaConnect to use to set up your VPC configuration. You must choose at least one and can choose as many as you want.
   7. For **Security groups**, specify the VPC security groups that you want MediaConnect to use to set up your VPC configuration. You must choose at least one security group.
9. For each media stream that you want to add to the flow, do the following:

1. In the **Media streams** section, choose **Add media stream**.
2. In the **Name** field, specify a descriptive name that will help you distinguish this media stream from others in the flow.
3. For **Description**, specify a description that will help you remember the use of this media stream.
4. For **Stream ID**, specify a unique identifier for the media stream.

   If the source or any of the outputs uses the CDI protocol, specify the value that is expected by the production and playout systems.

   If the source and all outputs use the ST 2110 JPEG XS protocol, specify a value that is unique to that of other media streams within the flow.
5. Choose **Advanced options** to display the additional options based on your stream type.
6. For specific instructions on the advanced options based on your stream type, choose one of the following tabs:

   Audio

   a. For **Stream type**, choose **Audio**.

   b. For **Media clock rate**, specify the sample rate for the stream. This value is measured in Hz.

   c. For **Language**, specify the language of the audio. This value should be in a format that the receiver recognizes.

   d. For **Channel order**, specify the format of the audio channel.

   e. Choose **Add media stream**.

   Video

   a. For **Stream type**, choose **Video**.

      For many fields, MediaConnect provides a default value that represents the recommended setting. Change the default value if needed.

   b. **Media clock rate** is the sample rate for the stream, and is set to 90000. This value is measured in Hz.

   c. For **Video format**, specify the resolution of the video.

   d. For **Exact framerate**, specify the frame rate of the video. This value should be represented in frames per second.

e. For **Colorimetry**, specify the format that was used for the representation of color in the video.

f. For **Scan mode**, specify the method that was used to scan the incoming video.

- Choose **Interlace** if the incoming video is interlaced (for example, 480i or 1080i).
- Choose **Progressive** if the incoming video is progressive (for example, 720p or 1080p).
- Choose **Progressive segmented frame** if the incoming video is PSF (for example, 1080psf).

g. For **TCS**, specify the transfer characteristic system (TCS) that was used in the video.

h. For **Range**, specify the encoding range of the video.

i. For **PAR**, specify the pixel access ratio (PAR) of the video.

j. Choose **Add media stream**.

Ancillary data

a. For **Stream type**, choose **Ancillary data**.

b. **Media clock rate** is the sample rate for the stream, and is set to 90000. This value is measured in Hz.

c. Choose **Add media stream**.

10. Scroll back up to the **Sources** section.
11. Determine which protocol your source uses.
12. For specific instructions based on your protocol, choose one of the following tabs:

CDI

1. For **Protocol**, choose **CDI**.
2. For **Description**, enter a description that will remind you later where this source is from. This might be the company name or notes about the setup.
3. For **Inbound port**, specify the port that the flow will listen on for incoming content. This value can be anything from 1024 to 65535, with the exception of 2077 and 2088 (those ports are reserved for other protocols).
4. For **VPC interface**, choose the name of the VPC interface that you want to use as the source.
5. For each media stream that you want to use as part of the source, do the following.

a. For **Media stream name**, choose the name of the media stream.

b. For **Encoding name**, accept the default value.

- For ancillary data streams, the encoding name is **`smpte291`**.
- For audio streams, the encoding name is **`pcm`**.
- For video, the encoding name is **`raw`**.

ST 2110 JPEG XS

1. For **Protocol**, choose **ST 2110 JPEG XS**.
2. For **Description**, enter a description that will remind you later where this source is from. This might be the company name or notes about the setup.
3. For **Max sync buffer**, specify the size of the buffer that you want MediaConnect to use to sync incoming source data. This value is measured in milliseconds (ms).
4. For **VPC interface name 1**, choose one of the VPC interfaces that you want to use as a source.
5. For **VPC interface name 2**, choose a second VPC interface that you want to use as a source. There is no priority between VPC interfaces 1 and 2.
6. For each media stream that you want to use as part of the source, do the following.

   a. For **Media stream name**, choose the name of the media stream.

   b. For **Encoding name**, accept the default value.

   - For ancillary data streams, the encoding name is **`smpte291`**.
   - For audio streams, the encoding name is **`pcm`**.
   - For video, the encoding name is **`jxsv`**.

   c. For **Inbound port**, specify the port that the flow will listen on for incoming content. This value can be anything from 1024 to 65535, with the exception of 2077 and 2088 (those ports are reserved for other protocols).

13. At the bottom of the page, choose **Create flow**.

> **Note**
>
> The flow doesn't start automatically. You must start the flow manually.

14. Add outputs to specify where you want MediaConnect to send the content.

## Create an AWS CDI flow (AWS CLI)

To use the AWS CLI to create a flow, you must use the `create-flow` command. To simplify the flow creation, we suggest you use the `create-flow` command with the `--cli-input-json` option. The `--cli-input-json` option requires you to create a JSON file with the necessary settings for your new flow. Step 1 of this procedure provides an example of one possible way configure this JSON file. For more information about the `create-flow` command and the `--cli-input-json` option, see: [AWS CLI Command Reference create-flow]

1. Create a JSON file that contains the details of the flow that you want to create.

   The following example shows the structure for the contents of the file. This example uses a JPEG XS source to create a AWS CDI output with the following attributes:

   - 2 Amazon VPC interfaces, 1 EFA (Elastic Fabric Adapter) and 1 ENA (Elastic Network Adapter)
   - 1 video stream, 1 audio stream, and 1 ancillary data stream

```
{
    "Name": "AwardsShow",

    "MediaStreams": [
        {
            "Attributes": {
                "Fmtp": {
                    "Colorimetry": "BT709",
                    "ExactFramerate": "60000/1001",
                    "Par": "1:1",
                    "Range": "NARROW",
                    "ScanMode": "progressive",
                    "Tcs": "SDR"
                }
            },
            "ClockRate": 90000,
            "MediaStreamId": 0,
            "MediaStreamName": "video-stream",
            "MediaStreamType": "video",
            "VideoFormat": "1080p"
        },
        {
            "Attributes": {
                "Fmtp": {
                    "ChannelOrder": "SMPTE2110.(ST)"
                }
            },
            "ClockRate": 48000,
            "MediaStreamId": 1,
            "MediaStreamName": "audio-stream",
            "MediaStreamType": "audio"
        },
        {
            "ClockRate": 90000,
            "MediaStreamId": 2,
            "MediaStreamName": "anc-stream",
            "MediaStreamType": "ancillary-data"
        }
    ],

    "Outputs": [
        {
            "Name": "cdi-output",
            "Protocol": "cdi",
            "Description": "cdi-output to medialive",
            "Destination": "198.51.100.5",
            "MediaStreamOutputConfigurations": [
                {
                    "EncodingName": "raw",
                    "MediaStreamName": "video-stream"
                },
                {
                    "EncodingName": "pcm",
                    "MediaStreamName": "audio-stream"
                }
            ],
            "Port": 5000,
            "VpcInterfaceAttachment": {
                "VpcInterfaceName": "efa-name"
            }
        }
    ],

    "Source": {
        "Name": "jxs-input",
        "Protocol": "st2110-jpegxs",
        "Description": "jxs-input to cdi-output",
        "MaxSyncBuffer": 100,
        "MediaStreamSourceConfigurations": [
            {
                "EncodingName": "jxsv",
                "InputConfigurations": [
                    {
                        "InputPort": 5011,
                        "Interface": {
                            "Name": "efa-name"
                        }
                    },
                    {
                        "InputPort": 5011,
                        "Interface": {
                            "Name": "ena-name"
                        }
                    }
                ],
                "MediaStreamName": "video-stream"
            },
            {
                "EncodingName": "pcm",
                "InputConfigurations": [
                    {
                        "InputPort": 5001,
                        "Interface": {
                            "Name": "efa-name"
                        }
                    },
                    {
                        "InputPort": 5001,
                        "Interface": {
                            "Name": "ena-name"
                        }
                    }
                ],
                "MediaStreamName": "audio-stream"
            }
        ]
    },

    "VpcInterfaces": [
        {
            "Name": "efa-name",
            "NetworkInterfaceType": "efa",
            "RoleArn": "arn:aws:iam::111122223333:role/MediaConnectAccessRole",
            "SecurityGroupIds": [
                "sg-1234567890abcdef0"
            ],
            "SubnetId": "subnet-abcdef01234567890"
        },
        {
            "Name": "ena-name",
            "NetworkInterfaceType": "ena",
            "RoleArn": "arn:aws:iam::111122223333:role/MediaConnectAccessRole",
            "SecurityGroupIds": [
                "sg-1234567890abcdef0"
            ],
            "SubnetId": "subnet-abcdef01234567890"
        }
    ]
}
```

2. In the AWS CLI, use the `create-flow` command.

```
aws mediaconnect create-flow --cli-input-json file://filename.json --
profile YourProfile
```

The following example shows the return value:

```
{
    "Flow": {
        "AvailabilityZone": "us-west-2a",
        "Description": "jxs-input to cdi-output",
        "EgressIp": "203.0.113.0",
        "Entitlements": [],
        "FlowArn": "arn:aws:mediaconnect:us-west-2:111122223333:flow:1-
DwtfUlYOUVABAQNR-c94d84ce4215:AwardsShow",
        "MediaStreams": [
            {
                "Attributes": {
                    "Fmtp": {
                        "Colorimetry": "BT709",
                        "ExactFramerate": "60000/1001",
                        "Par": "1:1",
                        "Range": "NARROW",
                        "ScanMode": "progressive",
                        "Tcs": "SDR"
                    }
                },
                "ClockRate": 90000,
                "Fmt": 96,
                "MediaStreamId": 0,
                "MediaStreamName": "video-stream",
                "MediaStreamType": "video",
                "VideoFormat": "1080p"
            },
            {
                "Attributes": {
                    "Fmtp": {
                        "ChannelOrder": "SMPTE2110.(ST)"
                    }
                },
                "ClockRate": 48000,
                "Fmt": 97,
                "MediaStreamId": 1,
                "MediaStreamName": "audio-stream",
                "MediaStreamType": "audio"
            },
            {
                "ClockRate": 90000,
                "Fmt": 98,
                "MediaStreamId": 2,
                "MediaStreamName": "anc-stream",
                "MediaStreamType": "ancillary-data"
            }
        ],
        "Name": "AwardsShow",
        "Outputs": [
            {
                "Description": "cdi-output to medialive",
                "Destination": "198.51.100.5",
                "MediaStreamOutputConfigurations": [
                    {
                        "EncodingName": "raw",
                        "MediaStreamName": "video-stream"
                    },
                    {
                        "EncodingName": "pcm",
                        "MediaStreamName": "audio-stream"
                    }
                ],
                "Name": "cdi-output",
                "OutputArn": "arn:aws:mediaconnect:us-west-2:111122223333:output:1-
DwtfUlYOUVABAQNR-c94d84ce4215:cdi-output",
                "Port": 5000,
                "Transport": {
                    "Protocol": "cdi"
                },
                "VpcInterfaceAttachment": {
                    "VpcInterfaceName": "efa-name"
                }
            }
        ],
        "Source": {
            "Description": "jxs-input to cdi-output",
            "MediaStreamSourceConfigurations": [
                {
                    "EncodingName": "jxs-input",
                    "InputConfigurations": [
                        {
                            "InputIp": "203.0.113.1",
                            "InputPort": 5011,
                            "Interface": {
                                "Name": "efa-name"
                            }
                        },
                        {
                            "InputIp": "203.0.113.2",
                            "InputPort": 5011,
                            "Interface": {
                                "Name": "ena-name"
                            }
                        }
                    ],
                    "MediaStreamName": "video-stream"
                },
                {
                    "EncodingName": "pcm",
                    "InputConfigurations": [
                        {
                            "InputIp": "203.0.113.3",
                            "InputPort": 5001,
                            "Interface": {
                                "Name": "efa-name"
                            }
                        },
                        {
                            "InputIp": "203.0.113.4",
                            "InputPort": 5001,
                            "Interface": {
                                "Name": "ena-name"
                            }
                        }
                    ],
                    "MediaStreamName": "audio-stream"
                }
            ],
            "Name": "jxs-input",
            "SourceArn": "arn:aws:mediaconnect:us-west-2:111122223333:source:1-
DwtfUlYOUVABAQNR-c94d84ce4215:jxs-input",
            "Transport": {
                "MaxSyncBuffer": 100,
                "Protocol": "st2110-jpegxs"
            }
        },
        "Sources": [
            {
                "Description": "jxs-input to cdi-output",
                "MediaStreamSourceConfigurations": [
                    {
                        "EncodingName": "jxsv",
                        "InputConfigurations": [
                            {
                                "InputIp": "203.0.113.173",
                                "InputPort": 5011,
                                "Interface": {
                                    "Name": "efa-name"
                                }
                            },
                            {
                                "InputIp": "203.0.113.114",
                                "InputPort": 5011,
                                "Interface": {
                                    "Name": "ena-name"
                                }
                            }
                        ],
                        "MediaStreamName": "video-stream"
                    },
                    {
                        "EncodingName": "pcm",
                        "InputConfigurations": [
                            {
                                "InputIp": "203.0.113.173",
                                "InputPort": 5001,
                                "Interface": {
                                    "Name": "efa-name"
                                }
                            },
                            {
                                "InputIp": "203.0.113.114",
                                "InputPort": 5001,
                                "Interface": {
                                    "Name": "ena-name"
                                }
                            }
                        ],
                        "MediaStreamName": "audio-stream"
                    }
                ],
                "Name": "jxs-input",
                "SourceArn": "arn:aws:mediaconnect:us-west-2:111122223333:source:1-
DwtfUlYOUVABAQNR-c94d84ce4215:jxs-input",
                "Transport": {
                    "MaxSyncBuffer": 100,
                    "Protocol": "st2110-jpegxs"
                }
            }
        ],
        "Status": "STANDBY",
        "VpcInterfaces": [
            {
                "Name": "efa-name",
                "NetworkInterfaceIds": [
                    "eni-0ae6ca9ea6673a2a7"
                ],
                "NetworkInterfaceType": "efa",
                "RoleArn": "arn:aws:iam::111122223333:role/MediaConnectAccessRole",
                "SecurityGroupIds": [
                    "sg-1234567890abcdef0"
                ],
                "SubnetId": "subnet-abcdef01234567890"
            },
            {
                "Name": "ena-name",
                "NetworkInterfaceIds": [
                    "eni-0cbabcf978eeb00a2"
                ],
                "NetworkInterfaceType": "ena",
                "RoleArn": "arn:aws:iam::111122223333:role/MediaConnectAccessRole",
                "SecurityGroupIds": [
                    "sg-1234567890abcdef0"
                ],
                "SubnetId": "subnet-abcdef01234567890"
            }
        ]
    }
}
```

# Viewing a list of flows

You can view a list of your AWS Elemental MediaConnect flows in a specific AWS Region.

### To view a list of flows (console)

- Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.

  The **Flows** page appears, listing all the flows that are associated with your account.

### To view a list of flows (AWS CLI)

- In the AWS CLI, use the `list-flows` command:

  ```
  aws mediaconnect list-flows --profile PMprofile
  ```

  The following example shows the return value:

  ```
  {
    "Flows": [
      {
        "AvailabilityZone": "us-west-2a",
        "Description": "Temporary listed flow description",
  ```

```
      "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame",
      "Name": "BasketballGame",
      "SourceType": "OWNED",
      "Status": "STOPPING"
    },
    {
      "AvailabilityZone": "us-west-2d",
      "Description": "Temporary listed flow description",
      "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:2-3aBC45dEF67hiJ8k-2AbC34DE5fGa6:AwardsShow",
      "Name": "AwardsShow",
      "SourceType": "OWNED",
      "Status": "STANDBY"
    }
  ]
}
```

# Viewing the details of a flow

You can view a flow's details, such as ARN, Availability Zone, status, source, entitlements, and outputs.

**To view the details of a flow (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that you want to view.

   The details page for that flow appears. This page is divided into the following tabs:

   - The **Source** tab shows details about the source for this flow, including an indication of whether the flow is connected to the source.
   - The **Outputs** tab shows details for each output that you created for this flow.
   - The **Entitlements** tab shows any entitlements that you have granted on this flow.
   - The **VPC interfaces** tab shows a list of connections that this flow has with virtual private clouds (VPCs) based on the Amazon Virtual Private Cloud (Amazon VPC) service.
   - The **Media streams** tab shows a list of media streams that have been created on this flow. Each media stream represents a different component of a video such as video, audio, ancillary data.

- The **Alerts** tab shows a log of active alerts on this flow.

### To view the details of a flow (AWS CLI)

- In the AWS CLI, use the `describe-flow` command:

```
aws mediaconnect describe-flow --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

The following example shows the return value:

```
{
    "Flow": {
        "EgressIp": "54.201.4.39",
        "AvailabilityZone": "us-east-1b",
        "Status": "ACTIVE",
        "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
        "Entitlements": [
            {
                "EntitlementArn": "arn:aws:mediaconnect:us-
east-1:111122223333:entitlement:1-AaBb11CcDd22EeFf-34DE5fG12AbC:MyEntitlement",
                "Description": "Assign to this account",
                "Name": "MyEntitlement",
                "Subscribers": [
                    "444455556666"
                ]
            }
        ],
        "Description": "NYC awards show",
        "Name": "AwardsShow",
        "Outputs": [
            {
                "Port": 2355,
                "Name": "NYC",
                "Transport": {
                    "SmoothingLatency": 0,
                    "Protocol": "rtp-fec"
                },
                "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",
                "Destination": "192.0.2.0"
            },
            {
                "Port": 3025,
                "Name": "LA",
                "Transport": {
                    "SmoothingLatency": 0,
                    "Protocol": "rtp-fec"
                },
                "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-987655dEF67hiJ89-c34de5fG678h:LA",
                "Destination": "192.0.2.0"
            }
        ],
        "Source": {
            "IngestIp": "54.201.4.39",
            "SourceArn": "arn:aws:mediaconnect:us-
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource",
            "Transport": {
                "MaxBitrate": 80000000,
                "Protocol": "rtp"
            },
            "IngestPort": 1069,
            "Description": "Saturday night show",
            "Name": "ShowSource",
            "WhitelistCidr": "10.24.34.0/23"
        }
    }
}
```

## Starting a flow

After you create a flow, you must start the flow. You can also stop and restart a flow at any time.

**To start a flow (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that you want to start.

   The details page for that flow appears.
3. Choose **Start**.

### To start a flow (AWS CLI)

- In the AWS CLI, use the `start-flow` command:

  ```
  aws mediaconnect start-flow --flow-arn arn:aws:mediaconnect:us-
  east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame --
  profile PMprofile
  ```

  The following example shows the return value:

  ```
  {
    "FlowArn": "arn:aws:mediaconnect:us-
  east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame",
    "Status": "STARTING"
  }
  ```

## Stopping a flow

When you stop an active flow, it immediately becomes unavailable to customers who are accessing the output directly from your AWS Elemental MediaConnect flow or through an entitlement. If you want to delete an active flow, you must stop the flow first before you can delete it.

### To stop a flow (console)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that you want to stop.

   The details page for that flow appears.
3. Choose **Stop**.

   The status of the flow changes to **Standby**. The flow stops immediately and is no longer viewable to customers who are accessing the output directly from your MediaConnect flow or through an entitlement.

### To stop a flow (AWS CLI)

- In the AWS CLI, use the `stop-flow` command:

```
aws mediaconnect stop-flow --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame --
profile PMprofile
```

The following example shows the return value:

```
{
  "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame",
  "Status": "STOPPING"
}
```

# Updating a flow

You can change a flow's source, entitlements, and outputs even if the flow is running. However, you can't change the flow's name, ARN, or Availability Zone. For more information, see the following topics:

- Managing tags on a flow
- Updating the source
- Updating outputs
- Updating media streams
- Updating entitlements
- Adding a VPC interface to a flow

# Managing tags on a flow

You can use tags to help you track the billing and organization for your AWS Elemental MediaConnect flows, sources, outputs, and entitlements. These are the same tags that AWS Billing and Cost Management provides for organizing your AWS bill. For more information about using tags for cost allocation, see Use Cost Allocation Tags for Custom Billing Reports in the *AWS Billing User Guide*.

### To add tags to a flow (console)

1. Open the MediaConnect console at [https://console.aws.amazon.com/mediaconnect/](https://console.aws.amazon.com/mediaconnect/).
2. On the **Flows** page, choose the name of the flow that you want to add tags to.

   The details page for that flow appears.
3. In the **Details** section, choose **Manage tags**.
4. Choose **Manage tags**, and then choose **Add tag**.
5. For each tag that you want to add, do the following:

   a. Enter a key and a value. For example, your key can be **`sports`** and your value can be **`golf`**.

   b. Choose **Add tag**.
6. Choose **Update**.

### To edit tags on a flow (console)

1. Open the MediaConnect console at [https://console.aws.amazon.com/mediaconnect/](https://console.aws.amazon.com/mediaconnect/).
2. On the **Flows** page, choose the name of the flow that has the tags you want to edit.

   The details page for that flow appears.
3. In the **Details** section, choose **Manage tags**.
4. Choose **Manage tags**.
5. Update the tags, as needed.
6. Choose **Update**.

### To remove tags from a flow (console)

1. Open the MediaConnect console at [https://console.aws.amazon.com/mediaconnect/](https://console.aws.amazon.com/mediaconnect/).
2. On the **Flows** page, choose the name of the flow that you want to add tags to.

   The details page for that flow appears.
3. In the **Details** section, choose **Manage tags**.
4. Choose **Manage tags**.
5. Choose **Remove tag** next to each tag that you want to delete.

6. Choose **Update**.

# Deleting a flow

When you delete an active flow, it immediately becomes unavailable to customers who are accessing the output directly from your AWS Elemental MediaConnect flow or through an entitlement. After you delete a flow, you can't recover it.

If the flow is active, you must stop the flow before you can delete it.

**To delete a flow (console)**

1. Open the MediaConnect console at [https://console.aws.amazon.com/mediaconnect/](https://console.aws.amazon.com/mediaconnect/).
2. On the **Flows** page, choose the name of the flow that you want to delete.

   The details page for that flow appears.
3. Review the **Status** field to verify that the flow is in **Standby** mode.
4. If the flow status is **Active**, choose **Stop**.
5. Choose **Delete**.

   A confirmation message appears.
6. Choose **Delete flow**.

   The flow is no longer viewable to customers who are accessing the output directly from your MediaConnect flow or through an entitlement. It might take up to five minutes for the flow to be deleted entirely.

**To delete a flow (AWS CLI)**

- In the AWS CLI, use the `delete-flow` command:

  ```
  aws mediaconnect delete-flow --flow-arn arn:aws:mediaconnect:us-
  east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame --
  profile PMprofile
  ```

  The following example shows the return value:

  ```
  {
  ```

```
  "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame",
  "Status": "DELETING"
}
```

# Sources in AWS Elemental MediaConnect

A source in MediaConnect can be anything that provides a live video feed, such as the following:

- An on-premises encoder
- Another AWS Elemental MediaConnect flow
- An AWS Elemental MediaLive output
- A playout system (cloud-based or on-premises)

For a list of supported protocols that you can use for your source, see Protocols.

From the MediaConnect console, you can view Amazon CloudWatch metrics to monitor the source health of an active flow.

**Topics**

## Adding a source to an existing flow

For transport stream flows, you can add a second source for failover. Both sources on the flow must use the same protocol. (However, you can have one source that uses RTP and the other that uses RTP-FEC.) For more information about source failover, see Source failover.

The method you use to add a second source to a flow is dependent on the type of source that you want to use:

- Standard source – Uses content from any source that is not a VPC source or an entitled source.
- VPC source – Uses content that comes from a VPC that you configure.

MediaConnect doesn't support two sources on entitled flows or on CDI flows. For redundancy with ST 2110 JPEG XS sources, you can specify two inbound VPC interfaces on an individual media stream. For redundancy with CDI sources, create a second flow.

From the MediaConnect console, you can view Amazon CloudWatch metrics to monitor the source health of an active flow.

## Adding a standard source to an existing flow

You can add a second source to an existing flow for failover. Both sources on the flow must use the same protocol. (However, you can have one source that uses RTP and the other that uses RTP-FEC.) For more information about source failover, see Source failover.

**To add a standard source to an existing flow (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that you want to update.
3. Choose the **Source** tab.
4. In the **Source failover configuration** section, choose **Edit**.
5. In the **Edit source failover configuration** window, make sure that **Failover** is set to **Active**.

> **Note**
>
> If you activate failover on a flow that is running, you might encounter a brief interruption in the flow output.

6. In the **Failover mode** drop-down menu, select the mode to use with your source protocol. For a list of the modes supported by each protocol, see Failover support for source protocols
7. For **Recovery window**, specify the size of the buffer (delay) that you want the service to maintain. A larger buffer means a longer delay in transmitting the stream, but more room for error correction. A smaller buffer means a shorter delay, but less room for error correction. You can choose a value from 100–15000 ms. If you keep this field blank, MediaConnect uses the default value of 200 ms.
8. Choose **Update**.
9. In the **Sources** section, choose **Add**.
10. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console.

11. For **Source type**, choose **Standard source**.
12. Determine which protocol your source uses.

> **Note**
>
> All sources on a flow must use the same protocol. However, you can have one source that uses RTP and the other that uses RTP-FEC.

13. For specific instructions based on your protocol, choose one of the following tabs:

    RIST

    1. For **Protocol**, choose **RIST**.
    2. For **Inbound port**, specify the port that the flow listens on for incoming content.

       > **Note**
       >
       > The RIST protocol requires one additional port for error correction. To accommodate this requirement, MediaConnect reserves the port that is +1 from the port that you specify. For example, if you specify port 4000 for the output, the service assigns ports 4000 and 4001.

    3. For **Allowlist CIDR**, specify a range of IP addresses that are allowed to contribute content to your source. Format the IP addresses as a Classless Inter-Domain Routing (CIDR) block, for example, 10.24.34.0/23. For more information about CIDR notation, see RFC 4632.

       > **Important**
       >
       > Specify a CIDR block that is as precise as possible. Include only the IP addresses that you want to contribute content to your flow. If you specify a CIDR block that is too wide, it allows for the possibility of outside parties sending content to your flow.

    4. For **Maximum bitrate**, specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.
    5. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but

more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 1–15,000 ms. If you keep this field blank, MediaConnect uses the default value of 2,000 ms.

RTP or RTP-FEC

1. For **Protocol**, choose **RTP** or **RTP-FEC**.
2. For **Inbound port**, specify the port that the flow listens on for incoming content.

   > **Note**
   >
   > The RTP-FEC protocol requires two additional ports for error correction. To accommodate this requirement, MediaConnect reserves the ports that are +2 and +4 from the port that you specify. For example, if you specify port 4000 for the output, the service assigns ports 4000, 4002, and 4004.

3. For **Allowlist CIDR**, specify a range of IP addresses that are allowed to contribute content to your source. Format the IP addresses as a Classless Inter-Domain Routing (CIDR) block, for example, 10.24.34.0/23. For more information about CIDR notation, see RFC 4632.

   > **Important**
   >
   > Specify a CIDR block that is as precise as possible. Include only the IP addresses that you want to contribute content to your flow. If you specify a CIDR block that is too wide, it allows for the possibility of outside parties sending content to your flow.

4. For **Maximum bitrate**, specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.

SRT listener

1. For **Protocol**, choose **SRT listener**.
2. For **Source description**, enter a description that will remind you later where this source is from. This might be the company name or notes about the setup.

3. For **Allowlist CIDR block**, specify a range of IP addresses that are allowed to contribute content to your source. Format the IP addresses as a Classless Inter-Domain Routing (CIDR) block, for example, 10.24.34.0/23. For more information about CIDR notation, see RFC 4632.

   > ⚠ **Important**
   >
   > Specify a CIDR block that is as precise as possible. Include only the IP addresses that you want to contribute content to your flow. If you specify a CIDR block that is too wide, it allows for the possibility of outside parties sending content to your flow.

4. For **Inbound port**, specify the port that the flow listens on for incoming content.
5. For **Source listener address**, enter the address MediaConnect will use for the SRT connection. The address can be an IP address or a domain name.
6. For **Maximum bitrate** (optional), specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.
7. For **Minimum latency**, specify the minimum size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 100–15,000 ms. If you keep this field blank, MediaConnect uses the default value of 2,000 ms.
8. If the source is encrypted, choose **Enable** in the **Decryption** section and do the following:

   a. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.

   b. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.

SRT caller

1. For **Protocol**, choose **SRT caller**.
2. For **Source description**, enter a description that will remind you later where this source is from. This might be the company name or notes about the setup.

3. For **Source listener address**, enter the address MediaConnect will use for the SRT connection. The address can be an IP address or a domain name.
4. For **Source listener port**, enter the port MediaConnect will use for the SRT connection.
5. For **Maximum bitrate** (optional), specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.
6. For **Minimum latency**, specify the minimum size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 100–15,000 ms. If you keep this field blank, MediaConnect uses the default value of 2,000 ms.
7. For **Stream ID** (optional), enter an identifier for the stream. This identifier can be used to communicate information about the stream.
8. If the source is encrypted, choose **Enable** in the **Decryption** section and do the following:

    a. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.

    b. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.

Zixi push

1. For **Protocol**, choose **Zixi push**.

    AWS Elemental MediaConnect populates the value of the inbound port.

2. For **Allowlist CIDR**, specify a range of IP addresses that are allowed to contribute content to your source. Format the IP addresses as a Classless Inter-Domain Routing (CIDR) block, for example, 10.24.34.0/23. For more information about CIDR notation, see RFC 4632.

    > ⚠ **Important**
    >
    > Specify a CIDR block that is as precise as possible. Include only the IP addresses that you want to contribute content to your flow. If you specify a CIDR block that

is too wide, it allows for the possibility of outside parties sending content to your flow.

3. For **Stream ID**, specify the stream ID set in the Zixi feeder.

   > ⚠ **Important**
   >
   > The stream ID must match the value set in the Zixi feeder. If you leave this field blank, MediaConnect uses the source name as the stream ID. If the stream ID is not exactly the same as the source name, you must manually enter the stream ID.

4. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 0–60,000 ms. If you keep this field blank, the service uses the default value of 6,000 ms.
5. If the source is encrypted, choose **Enable** in the **Decryption** section and do the following:

   a. For **Decryption type**, choose **Static key**.

   b. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.

   c. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.

   d. For **Decryption algorithm**, choose the type of encryption that was used to encrypt the source.

### Zixi push for AWS Elemental Link UHD device

After creating the additional Zixi push source, you must configure the AWS Elemental Link device using MediaLive. See the following MediaLive setup instructions to complete the process after you have created the source: Using a device in a flow in the *MediaLive User Guide*. Ensure you have access to both MediaConnect and MediaLive to complete these steps.

> **Note**
>
> Zixi push for AWS Elemental Link UHD devices only supports failover mode. Merge mode is not supported.

1. For **Protocol**, choose **Zixi push**.

   AWS Elemental MediaConnect populates the value of the inbound port.

2. For **Allowlist CIDR**, specify a range of IP addresses that are allowed to contribute content to your source. Format the IP addresses as a Classless Inter-Domain Routing (CIDR) block, for example, 10.24.34.0/23. For more information about CIDR notation, see RFC 4632.

   > **Important**
   >
   > If you know the range of public IP addresses that your Link device uses to connect to the internet, enter that CIDR block. Note that this is not the same as the IP address of the AWS Elemental Link device. If you cannot obtain this information, it is possible to configure the CIDR block to be open to all possible IP addresses by using 0.0.0.0/0. Typically, it is not best practice to assign a CIDR block that is open to the entire internet (0.0.0.0/0). However, if this method must be used, the data being transferred is encrypted using AES-128 encryption.

3. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value between 0 and 60,000 ms. If you keep this field blank, the service uses the default value of 6,000 ms. The **Maximum latency** value should match the **Latency** value configured on the AWS Elemental Link device. For information on configuring the Link device's latency, see: Configuring the device in the *AWS Elemental MediaLive User Guide*

4. For **Decryption**, choose **Activate** and do the following:

   a. For **Decryption type**, choose **Static key**.

   b. For **Decryption algorithm**, choose **AES-128**. AWS Elemental Link requires AES-128, do not select another algorithm.

   c. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.

   d. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.

14. Choose **Save**.

## Adding a VPC source to an existing flow

You can add a second source to an existing transport stream flow for failover. Both sources on the flow must be binary identical (come from the same encoder) and they must use the same protocol. (However, you can have one source that uses RTP and the other that uses RTP-FEC.) For more information about source failover, see Source failover.

> ⚠ **Important**
>
> Before you begin this procedure, make sure that the following steps have been completed:
>
> - In Amazon VPC, set up your VPC and associated security groups. For more information about VPCs, see the Amazon VPC User Guide. For information about configuring security groups to work with your VPC interface, see Security group considerations.
> - In IAM, set up MediaConnect as a trusted service.
> - If the source of your flow requires encryption, set up encryption.

MediaConnect doesn't support two sources on CDI flows. For redundancy with ST 2110 JPEG XS sources, you can specify two inbound VPC interfaces on an individual media stream. For redundancy with CDI sources, create a second flow.

**To add a VPC source to an existing flow (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that you want to update.
3. Choose the **Source** tab.
4. In the **Source failover configuration** section, choose **Edit**.
5. In the **Edit source failover configuration** window, make sure that **Failover** is set to **Enabled**.

> **Note**
>
> If you enable failover on a flow that is running, you might encounter a brief interruption in the flow output.

6. For **Recovery window**, specify the size of the buffer (delay) that you want the service to maintain. A larger buffer means a longer delay in transmitting the stream, but more room for error correction. A smaller buffer means a shorter delay, but less room for error correction. You can choose a value from 100–15000 ms. If you keep this field blank, MediaConnect uses the default value of 200 ms.
7. Choose **Update**.
8. In the **Sources** section, choose **Add source**.
9. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console.
10. For **Source type**, choose **VPC source**.
11. Determine which protocol your source uses.

    > **Note**
    >
    > All sources on a flow must use the same protocol. However, you can have one source that uses RTP and the other that uses RTP-FEC.

12. For specific instructions based on your protocol, choose one of the following tabs:

    RIST

    1. For **Protocol**, choose **RIST**.
    2. For **Inbound port**, specify the port that the flow listens on for incoming content.

       > **Note**
       >
       > The RIST protocol requires one additional port for error correction. To accommodate this requirement, MediaConnect reserves the port that is +1 from the port that you specify. For example, if you specify port 4000 for the output, the service assigns ports 4000 and 4001.

3. For **VPC interface name**, choose the name of the VPC interface that you want to use as the source.
4. For **Maximum bitrate**, specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.
5. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 1–15,000 ms. If you keep this field blank, MediaConnect uses the default value of 2,000 ms.

RTP or RTP-FEC

1. For **Protocol**, choose **RTP** or **RTP-FEC**.
2. For **Inbound port**, specify the port that the flow listens on for incoming content.

   > **Note**
   >
   > The RTP-FEC protocol requires two additional ports for error correction. To accommodate this requirement, MediaConnect reserves the ports that are +2 and +4 from the port that you specify. For example, if you specify port 4000 for the output, the service assigns ports 4000, 4002, and 4004.

3. For **VPC interface name**, choose the name of the VPC interface that you want to use as the source.
4. For **Maximum bitrate**, specify the maximum expected bitrate (in bits per second) for the flow. We recommend that you specify a value that is twice the actual bitrate.

Zixi push

1. For **Protocol**, choose **Zixi push**.

   AWS Elemental MediaConnect populates the value of the inbound port.
2. For **VPC interface name**, choose the name of the VPC interface that you want to use as the source.
3. For **Stream ID**, specify the stream ID set in the Zixi feeder.

> ⚠ **Important**
>
> The stream ID must match the value set in the Zixi feeder. If you leave this field blank, MediaConnect uses the source name as the stream ID. If the stream ID is not exactly the same as the source name, you must manually enter the stream ID.

4. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 0–60,000 ms. If you keep this field blank, the service uses the default value of 6,000 ms.
5. If the source is encrypted, choose **Enable** in the **Decryption** section and do the following:
   a. For **Decryption type**, choose **Static key**.
   b. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.
   c. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.
   d. For **Decryption algorithm**, choose the type of encryption that was used to encrypt the source.

13. Choose **Save**.

# Updating the source of a flow

You can update the source of an existing flow, even when the flow is currently running.

**To update the source of an existing flow (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that you want to update.
3. Choose the **Source** tab.
4. Choose the source that you want to update.
5. Choose **Update**.
6. Make the appropriate changes, and then choose **Update source**.

**To update the source of an existing flow (AWS CLI)**

- In the AWS CLI, use the **update-flow-source** command:

  ```
  aws mediaconnect update-flow-source --flow-arn arn:aws:mediaconnect:us-
  east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow --source-
  arn arn:aws:mediaconnect:us-east-1:111122223333:source:2-3aBC45dEF67hiJ89-
  c34de5fG678h:AwardsShowSource --allowlist-cidr 10.24.34.0/24 --profile PMprofile
  ```

  The following example shows the return value:

## Source failover

Source failover is a setup that involves two redundant sources for a transport stream flow. This redundancy helps to minimize disruption to your video stream. To use source failover, you specify two sources for the flow, then choose one of two options for the failover mode: *Merge* or *Failover*.

- Merge mode combines the sources into a single stream, allowing a graceful recovery from any single-source loss. If you set the failover mode to *Merge*, you can set the recovery window, which is the size of the buffer (delay) that you want MediaConnect to maintain. A larger recovery window means a longer delay in transmitting the stream, but more room for error correction. A smaller recovery window means a shorter delay, but less room for error correction. Sources used this way need to be *binary identical*, which means that they need to have originated from the same encoder. MediaConnect must also receive content from the two sources at the same time. Additionally, if the sources use RTP protocol, they must have RTP headers with aligned sequence numbers and they must also comply with the SMPTE ST 2022-7 standard.

  > **Note**
  >
  > SMPTE ST 2022-7 is a standard developed by the Society of Motion Picture and Television Engineers (SMPTE) group. The ST 2022-7 standard defines a method that replaces missing packets with packets in an identical, redundant stream. This type of failover requires a small latency buffer in your workflow to allow time for MediaConnect to recover packets from the two streams.

- Failover mode allows switching between a primary and a backup stream. This switching facilitates an easy transition to a more reliable stream. If you set the failover mode to *Failover*, you can specify a source as the primary source. The second source serves as the backup. If you don’t specify a primary source, MediaConnect treats both sources with equal priority, and switches to the available source as needed.

MediaConnect uses the two modes of failover in the following ways:

- In the *Merge* mode, MediaConnect uses content from both sources. The flow randomly selects one of the sources to start with. If that source is missing a packet, the flow pulls the missing packet from the other source. For example, if the flow is using source A and packet 123 is missing, MediaConnect pulls in packet 123 from source B and continues using source A. In this mode, the two sources are binary identical/ST 2022-7 compliant.
- In the *Failover* mode, if you don't specify a primary source, MediaConnect randomly uses one of the sources to provide content for the flow. If MediaConnect does not receive data from the source for 500 milliseconds, the flow switches to the other source, and can continue switching back and forth between sources as needed. If you do specify a primary source, MediaConnect uses that source to provide content for the flow. The flow switches to the other source if the primary source does not send data for 500 milliseconds, and switches back to the primary source as soon as data returns.

**Note**

MediaConnect doesn't support source failover on CDI flows or on entitlement flows. For more information about creating redundancy with CDI flows, visit: Creating a CDI flow. Additionally, you cannot add a second source to an existing flow for failover if you are using the Zixi pull or Fujitsu-QoS protocols.

## Failover support for source protocols

The following table describes which source protocols support failover.

| Protocol | Does this protocol support source failover? | How many sources can be added? | Supported failover modes |
|---|---|---|---|
| RIST | Yes | 2 | Merge or failover |

| Protocol | Does this protocol support source failover? | How many sources can be added? | Supported failover modes |
|---|---|---|---|
| RTP | Yes | 2 | Merge or failover |
| RTP-FEC | Yes | 2 | Merge or failover |
| SRT listener | Yes | 2 | Failover only |
| SRT caller | Yes | 2 | Failover only |
| Zixi pull | No | None - Zixi pull cannot be used as a source. | Source failover is not supported |
| Zixi push | Yes | 2 | Merge or failover |
| Zixi push for AWS Elemental Link UHD | Yes | 2 | Failover only |
| Fujitsu-QoS | No | 1 | Source failover is not supported |
| CDI | No | 1 | Source failover is not supported |
| ST 2110 JPEG XS | No | 1 | Source failover is not supported |
| Entitlement flows | No | 1 | Source failover is not supported |

## Managing tags on a source

You can use tags to help you track the billing and organization for your AWS Elemental MediaConnect flows, sources, outputs, and entitlements. These are the same tags that AWS Billing and Cost Management provides for organizing your AWS bill. For more information about using tags for cost allocation, see Use Cost Allocation Tags for Custom Billing Reports in the *AWS Billing User Guide*.

**To add tags to a source (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the source that you want to add tags to.
3. Choose the **Sources** tab.

   A list of sources for that flow appears.
4. Choose the source that you want to add tags to.
5. Choose **Manage tags**.
6. Choose **Manage tags** again, and then choose **Add tag**.
7. For each tag that you want to add, do the following:

   a. Enter a key and a value. For example, your key can be **`sports`** and your value can be **`golf`**.

   b. Choose **Add tag**.
8. Choose **Update**.

**To edit tags on a source (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the source that you want to edit tags for.
3. Choose the **Sources** tab.

   A list of sources for that flow appears.
4. Choose the source that you want to edit tags for.
5. Choose **Manage tags**.
6. Choose **Manage tags** again.
7. Update the tags, as needed.
8. Choose **Update**.

**To remove tags from a source (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the source that you want to remove tags from.
3. Choose the **Sources** tab.

   A list of sources for that flow appears.
4. Choose the source that you want to remove tags from.
5. Choose **Manage tags**.
6. Choose **Manage tags** again.
7. Choose **Remove tag** next to each tag that you want to delete.
8. Choose **Update**.

## Removing a source from a flow

If a flow has more than one source, you can remove one of the sources even when the flow is currently running.

**To remove a source from a flow (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow.
3. Choose the **Source** tab.
4. Choose the source that you want to remove.
5. Choose **Remove**.

## Source ports

Each source on a flow must use a different port (for exceptions to this, see the note). Some protocols require additional ports for error correction. For sources that use these protocols, AWS Elemental MediaConnect automatically reserves the additional ports that are needed. All MediaConnect protocols use UDP ports. The following table lists which additional ports, if any, the service reserves.

⚠ **Important**

There is an exception to the port requirements for sources that use the Zixi protocol. For standard Zixi sources, all sources use port 2088. For VPC Zixi sources, the sources will use an inbound port range of 2090-2099. The 2090-2099 port range is reserved exclusively for Zixi VPC sources and cannot be used by another source protocol. The VPC Zixi source port is assigned by MediaConnect when the source is created.

| Protocol | Ports needed | Ports required |
|---|---|---|
| CDI | Port | The port that you specify. This is the only port needed for the source. |
| RIST | Port and port+1 | The port that you specify, plus one additional port. MediaConnect automatic ally reserves a port that is +1 from the port that you specified.<br>For example, if you specify port 3000 for this output, the service also reserves port 3001. |
| RTP | Port | The port that you specify. This is the only port needed for the output. |
| RTP-FEC | Port, port+2, and port+4 | The port that you specify, plus two additional ports. MediaConnect automatically reserves ports that are +2 and +4 from the port that you specified. |

| Protocol | Ports needed | Ports required |
|---|---|---|
| | | For example, if you specify port 2000 for this output, the service also reserves ports 2002 and 2004 for error correction. |
| SRT listener | Port | The port that you specify. This is the only port needed for the source. |
| SRT caller | Port | The port that you specify. This is the only port needed for the source. |
| Fujitsu-QoS | Port and port+1 | The port that you specify, plus one additional port. MediaConnect automatically reserves a port that is +1 from the port that you specified. |
| ST 2110 JPEG XS | Port | The port that you specify. This is the only port needed for the source. |

| Protocol | Ports needed | Ports required |
|---|---|---|
| Zixi push | Port | **For standard sources**: MediaConnect automatically uses port 2088.<br>**For VPC sources**: MediaConnect automatically assigns a port in the range of 2090-2099 when the source is created. The 2090-2099 port range is reserved exclusively for Zixi VPC sources and cannot be used by another source protocol. |

# Outputs in MediaConnect

Outputs are the different destinations where you want MediaConnect to send the content of your flow. You can add and remove outputs at any time, even when the flow is active. These outputs are sent to the IP address that you specify. This option is useful if you intend to send your content to an on-premises encoder.

For transport stream flows, you can grant an entitlement to share your content with another AWS account (subscriber account). When the subscriber creates a flow using your content as the source, AWS Elemental MediaConnect generates an output on your flow.

> **Note**
>
> If you disable an entitlement after the subscriber creates a flow based on that entitlement, the associated output remains on your flow. This output continues to counts toward your maximum number of outputs. To delete an output that's associated with an entitlement, revoke the entitlement.

**Topics**

## Adding outputs to a flow

For transport stream flows, you can add up to 50 outputs. However, for optimal performance, follow the guidance offered in Best practices. Every output must have a name, a protocol, an IP address, and a port.

**Note**

If you intend to set up an entitlement for an output, don't create the output. Instead, grant an entitlement. When the subscriber creates a flow using your content as the source, the service creates an output on your flow.

The method you use to add an output to a flow is dependent on the type of output that you want to add:

- Standard output (transport stream flow) – Sends compressed content to any destination that is not a virtual private cloud (VPC) that you configured using Amazon Virtual Private Cloud.
- VPC output (transport stream flow) – Sends compressed content to a VPC that you configured using Amazon Virtual Private Cloud.
- VPC output (CDI flow) – Sends uncompressed content to a VPC that you configured using Amazon Virtual Private Cloud.

## Adding standard outputs to a flow

For transport stream flows, you can add up to 50 outputs. However, for optimal performance, follow the guidance offered in Best practices. A standard output goes to any destination that is not part of a virtual private cloud (VPC) that you created using Amazon Virtual Private Cloud.

**Note**

CDI flows don't support standard outputs.

**To add a standard output to a flow (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that you want to add an output to.

   The details page for that flow appears.
3. Choose the **Outputs** tab.
4. Choose **Add output**.

5. For **Name**, specify a name for your output. This value is an identifier that is visible only on the AWS Elemental MediaConnect console and is not visible to the end user.
6. For **Output type**, choose **Standard output**.
7. For **Description**, enter a description that will remind you later where this output is going. This might be the company name or notes about the setup.
8. Determine which protocol you want to use for the output.
9. For specific instructions based on the protocol that you want to use, choose one of the following tabs:

   RIST

   1. For **Protocol**, choose **RIST**.
   2. For **IP address**, choose the IP address where you want to send the output.
   3. For **Port**, choose the port that you want to use when the content is distributed to this output. For more information about ports, see Output destinations.

      > **Note**
      >
      > The RIST protocol requires one additional port for error correction. To accommodate this requirement, AWS Elemental MediaConnect reserves the port that is +1 from the port that you specify. For example, if you specify port 4000 for the output, the service assigns ports 4000 and 4001.

   4. For **Smoothing latency**, specify the additional delay that you want to use with output smoothing. We recommend that you specify a value of 0 ms to disable smoothing. However, if the receiver can't process the stream properly, specify a value between 100 and 1,000 ms. This way, AWS Elemental MediaConnect attempts to correct jitter from the flow source. If you keep this field blank, the service uses the default value of 0 ms.

   RTP or RTP-FEC

   1. For **Protocol**, choose **RTP** or **RTP-FEC**.
   2. For **IP address**, choose the IP address where you want to send the output.
   3. For **Port**, choose the port that you want to use when the content is distributed to this output. For more information about ports, see Output destinations.

> **Note**
>
> The RTP-FEC protocol requires two additional ports for error correction. To accommodate this requirement, AWS Elemental MediaConnect reserves the ports that are +2 and +4 from the port that you specify. For example, if you specify port 4000 for the output, the service assigns ports 4000, 4002, and 4004.

4. For **Smoothing latency**, specify the additional delay that you want to use with output smoothing. We recommend that you specify a value of 0 ms to disable smoothing. However, if the receiver can't process the stream properly, specify a value between 100 and 1,000 ms. This way, AWS Elemental MediaConnect attempts to correct jitter from the flow source. If you keep this field blank, the service uses the default value of 0 ms.

SRT listener

1. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console. It is not visible to anyone outside of the current AWS account.
2. For **Protocol**, choose **SRT listener**.
3. For **Minimum latency**, specify the minimum size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 100–15,000 ms. If you keep this field blank, MediaConnect uses the default value of 2,000 ms.
4. For **CIDR allow list**, specify a range of IP addresses that are allowed to view content from your output. Format the IP addresses as a Classless Inter-Domain Routing (CIDR) block, for example, 10.24.34.0/23. For more information about CIDR notation, see RFC 4632.

> **Important**
>
> Specify a CIDR block that is as precise as possible. Include only the IP addresses that you want to contribute content to your flow. If you specify a CIDR block that is too wide, it allows for the possibility of outside parties sending content to your flow.

5. For **Port**, choose the port that you want to use when the content is distributed to this output. For more information about ports, see Output destinations.
6. If you want to encrypt the video as it is sent to this output, do the following:
   a. In the **Encryption** section, choose **Enable**.
   b. **Encryption type** will not be selectable. **srt-password** is the only available encryption for this protocol.
   c. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.
   d. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the SRT password.

SRT caller

1. For **Protocol**, choose **SRT-caller**.
2. For **Minimum latency**, specify the minimum size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 100–15,000 ms. If you keep this field blank, MediaConnect uses the default value of 2,000 ms.
3. For **Destination IP address**, enter the IP address or domain of the output's destination.
4. For **Port**, choose the port that you want to use when the content is distributed to this output. For more information about ports, see Output destinations.
5. If you want to encrypt the video as it is sent to this output, do the following:
   a. In the **Encryption** section, choose **Enable**.
   b. **Encryption type** will not be selectable. **Srt-password** is the only available encryption for this protocol.
   c. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.
   d. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the SRT password.

Fujitsu-QoS

1. For **Protocol**, choose **Fujitsu-QoS**.

2. For **Port**, choose the port on which to exchange control packets with the receiver. For more information about ports, see Output destinations.
3. For **CIDR allow list**, specify a range of IP addresses that are allowed to view content from your output. Format the IP addresses as a Classless Inter-Domain Routing (CIDR) block, for example, 10.24.34.0/23. For more information about CIDR notation, see RFC 4632.

> ⚠ **Important**
>
> Specify a CIDR block that is as precise as possible. Include only the IP addresses that you want to contribute content to your flow. If you specify a CIDR block that is too wide, it allows for the possibility of outside parties sending content to your flow.

Zixi pull

1. For **Protocol**, choose **Zixi pull**.
2. For **Stream ID**, enter the **Stream** value that was configured when you added the input on the Zixi receiver. In the Zixi receiver, this value is found in the **Stream parameters** section.

> ⚠ **Important**
>
> If you keep this field blank, the service uses the output name as the stream ID. Because the stream ID must match the value that is set in the Zixi receiver, you must specify the stream ID if it is not exactly the same as the output name.

3. For **Remote ID**, enter the **ID** value that is assigned to the Zixi receiver. In the Zixi receiver, this value is located in the **General** settings menu and is labelled **ID**. The **ID** value can also be found on the Zixi receiver **Status** page.
4. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value between 0 and 60,000 ms. If you keep this field blank, the service uses the latency that is set in the receiver.

5. For **CIDR allow list**, specify a range of IP addresses that are allowed to retrieve content from your source. Format the IP addresses as a Classless Inter-Domain Routing (CIDR) block, for example, 10.24.34.0/23. For more information about CIDR notation, see RFC 4632.

> **Tip**
>
> To specify an additional CIDR block, choose **Add**. You can specify up to three CIDR blocks.

6. If you want to encrypt the video as it is sent to this output, do the following:
   a. In the **Encryption** section, choose **Enable**.
   b. For **Encryption type**, choose **Static key**.
   c. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.
   d. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.
   e. For **Encryption algorithm**, choose the type of encryption that you want to use to encrypt the source.

Zixi push

1. For **Protocol**, choose **Zixi push**.
2. For **IP address**, choose the IP address where you want to send the output.
3. For **Port**, choose the port that you want to use when the content is distributed to this output. For more information about ports, see Output destinations.
4. For **Stream ID**, enter the stream ID that is set in the Zixi receiver.

> **Important**
>
> If you keep this field blank, the service uses the output name as the stream ID. Because the stream ID must match the value set in the Zixi receiver, you must specify the stream ID if it is not exactly the same as the output name.

5. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but

more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value between 0 and 60,000 ms. If you keep this field blank, the service uses the default value of 6,000 ms.

6. If you want to encrypt the video as it is sent to this output, do the following:

   a. In the **Encryption** section, choose **Enable**.

   b. For **Encryption type**, choose **Static key**.

   c. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.

   d. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.

   e. For **Encryption algorithm**, choose the type of encryption that you want to use to encrypt the source.

10. Choose **Add output**.

#### To add an output to a flow (AWS CLI)

1. Create a JSON file that contains the details of the output that you want to add to the flow.

   The following example shows the structure for the contents of the file:

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame",
    "Outputs": [
        {
            "Description": "RTP-FEC Output",
            "Destination": "192.0.2.12",
            "Name": "RTPOutput",
            "Port": 5020,
            "Protocol": "rtp-fec",
            "SmoothingLatency": 100
        }
    ]
}
```

2. In the AWS CLI, use the `add-flow-output` command:

```
aws mediaconnect add-flow-outputs --flow-arn "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame" --cli-
input-json file://addFlowOutput.txt --region us-west-2
```

The following example shows the return value:

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame",
    "Outputs": [
        {
            "Name": "RTPOutput",
            "Port": 5020,
            "Transport": {
                "SmoothingLatency": 100,
                "Protocol": "rtp-fec"
            },
            "Destination": "192.0.2.12",
            "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:RTPOutput",
            "Description": "RTP-FEC Output"
        }
    ]
}
```

## Adding VPC outputs to a flow

A VPC output goes to a virtual private cloud (VPC) that you created using Amazon Virtual Private Cloud.

For transport stream flows, you can add outputs (up to 50) even if the flow is active. For CDI flows, you can add outputs (up to 10) only if the flow is in standby mode. For optimal performance, follow the guidance offered in Best practices.

**To add a VPC output to a flow (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that you want to add an output to.

The details page for that flow appears.

3. Choose the **Outputs** tab.
4. Choose **Add output**.
5. For **Name**, specify a name for your output. This value is an identifier that is visible only on the AWS Elemental MediaConnect console and is not visible to the end user.
6. For **Output type**, choose **VPC output**.
7. For **Protocol**, choose the appropriate protocol.
8. For **Description**, enter a description that will remind you later where this output is going. This might be the company name or notes about the setup.
9. Determine which protocol you want to use for the output. The protocol options are dependent on the flow type.

   - For transport stream flows, the protocol options are: RTP, RTP-FEC, RIST, SRT, and Zixi.
   - For CDI flows, the protocol options are: CDI and ST 2110 JPEG XS.

10. For specific instructions based on the protocol that you want to use, choose one of the following tabs:

    RIST

    1. For **Protocol**, choose **RIST**.
    2. For **IP address**, choose the IP address where you want to send the output.
    3. For **Port**, choose the port that you want to use when the content is distributed to this output. For more information about ports, see Output destinations.

       > **Note**
       >
       > The RIST protocol requires one additional port for error correction. To accommodate this requirement, AWS Elemental MediaConnect reserves the port that is +1 from the port that you specify. For example, if you specify port 4000 for the output, the service assigns ports 4000 and 4001.

    4. For **Smoothing latency**, specify the additional delay that you want to use with output smoothing. We recommend that you specify a value of 0 ms to disable smoothing. However, if the receiver can't process the stream properly, specify a value between 100

and 1,000 ms. This way, AWS Elemental MediaConnect attempts to correct jitter from the flow source. If you keep this field blank, the service uses the default value of 0 ms.

5. For **Output to VPC**, choose the name of the VPC interface that you want to send your output to.

RTP or RTP-FEC

1. For **Protocol**, choose **RTP** or **RTP-FEC**.

> **Note**
>
> RTP and RTP-FEC outputs are compliant with the SMPTE 2022-7 standard. If your downstream receiver supports 2022-7 source merging, RTP and RTP-FEC outputs will be compatible.

2. For **IP address**, choose the IP address where you want to send the output.
3. For **Port**, choose the port that you want to use when the content is distributed to this output. For more information about ports, see Output destinations.

> **Note**
>
> The RTP-FEC protocol requires two additional ports for error correction. To accommodate this requirement, AWS Elemental MediaConnect reserves the ports that are +2 and +4 from the port that you specify. For example, if you specify port 4000 for the output, the service assigns ports 4000, 4002, and 4004.

4. For **Smoothing latency**, specify the additional delay that you want to use with output smoothing. We recommend that you specify a value of 0 ms to disable smoothing. However, if the receiver can't process the stream properly, specify a value between 100 and 1,000 ms. This way, AWS Elemental MediaConnect attempts to correct jitter from the flow source. If you keep this field blank, the service uses the default value of 0 ms.
5. For **Output to VPC**, choose the name of the VPC interface that you want to send your output to.

SRT listener

1. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console. It is not visible to anyone outside of the current AWS account.
2. For **Output type**, select **VPC output**.
3. For **Protocol**, choose **SRT listener**.
4. For **Description**, enter a description that can help you distinguish one output from another. This might be the company name or notes about the setup.
5. For **Minimum latency**, specify the minimum size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 100–15,000 ms. If you keep this field blank, MediaConnect uses the default value of 2,000 ms.
6. For **Port**, choose the port that you want to use when the content is distributed to this output. For more information about ports, see Output destinations.
7. For **Output to VPC**, choose the name of the VPC interface that you want to send your output to.
8. If you want to encrypt the video as it is sent to this output, do the following:
   a. In the **Encryption** section, choose **Enable**.
   b. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.
   c. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the SRT password.

SRT caller

1. For **Name**, specify a name for your source. This value is an identifier that is visible only on the MediaConnect console. It is not visible to anyone outside of the current AWS account.
2. For **Output type**, select **VPC output**.
3. For **Protocol**, choose **SRT caller**.

4. For **Description**, enter a description that can help you distinguish one output from another. This might be the company name or notes about the setup.
5. For **Minimum latency**, specify the minimum size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value from 100–15,000 ms. If you keep this field blank, MediaConnect uses the default value of 2,000 ms.
6. For **Destination IP address**, enter the IP address or domain of the output's destination.
7. For **Port**, choose the port that you want to use when the content is distributed to this output. For more information about ports, see Output destinations.
8. For **Output to VPC**, choose the name of the VPC interface that you want to send your output to.
9. If you want to encrypt the video as it is sent to this output, do the following:
   a. In the **Encryption** section, choose **Enable**.
   b. **Encryption type** will not be selectable. **Srt-password** is the only available encryption for this protocol.
   c. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.
   d. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the SRT password.

Zixi push

1. For **Protocol**, choose **Zixi push**.
2. For **IP address**, choose the IP address where you want to send the output.
3. For **Port**, choose the port that you want to use when the content is distributed to this output. For more information about ports, see Output destinations.
4. For **Stream ID**, enter the stream ID that is set in the Zixi receiver.

> ⚠ **Important**
>
> If you keep this field blank, the service uses the output name as the stream ID. Because the stream ID must match the value set in the Zixi receiver, you must specify the stream ID if it is not exactly the same as the output name.

5. For **Maximum latency**, specify the size of the buffer (delay) that you want the service to maintain. A higher latency value means a longer delay in transmitting the stream, but more room for error correction. A lower latency value means a shorter delay, but less room for error correction. You can choose a value between 0 and 60,000 ms. If you keep this field blank, the service uses the default value of 6,000 ms.
6. For **Output to VPC**, choose the name of the VPC interface that you want to send your output to.
7. If you want to encrypt the video as it is sent to this output, do the following:
   a. In the **Encryption** section, choose **Enable**.
   b. For **Encryption type**, choose **Static key**.
   c. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.
   d. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.
   e. For **Encryption algorithm**, choose the type of encryption that you want to use to encrypt the source.

Fujitsu-QoS

1. For **Protocol**, choose **Fujitsu-QoS**.
2. For **Port**, choose the port on which to exchange control packets with the receiver. For more information about ports, see Output destinations.
3. For **Output to VPC**, choose the name of the VPC interface that you want to send your output to.

CDI

1. For **Protocol**, choose **CDI**.
2. For **IP address**, choose the IP address where you want to send the output.
3. For **Port**, choose the port that you want to use when the content is distributed to this output. For more information about ports, see Output destinations.
4. For **VPC interface**, choose the name of the VPC interface that you want to send your output to.
5. For each media stream that you want to send as part of the output, do the following:

   a. For **Media stream name**, choose the name of the media stream. You can only add the media streams that the source on your flow uses.

   b. For **Encoding name**, confirm the default value, which is pre-selected based on the media stream type.

   c. For **FMT**, specify the format type number (sometimes referred to as *RTP payload type*) of the media stream. This value should be in a format that the receiver recognizes.

ST 2110 JPEG XS

1. For **Protocol**, choose **ST 2110 JPEG XS**.
2. For **VPC interface 1**, choose one of the VPC interfaces that you want to send content to and then choose the specific IP address where you want to send the output.
3. For **VPC interface 2**, choose a second VPC interface that you want to send content to and then choose the specific IP address where you want to send the output. There is no priority between VPC interfaces 1 and 2.
4. For each media stream that you want to send as part of the output, do the following:

   a. For **Media stream name**, choose the name of the media stream. You can only add the media streams that the source on your flow uses.

   b. For **Encoding name**, choose the format that was used to encode the data.

   - For ancillary data streams, set the encoding name to **`smpte291`**.
   - For audio streams, set the encoding name to **`pcm`**.
   - For video, set the encoding name to **`jxsv`**.

   c. For **Port**, choose the port that you want to use when the content is distributed to this output. For more information about ports, see Output destinations.

   d. For **Encoder profile**, choose a setting for the compression. This property only applies if the source uses the CDI protocol.

   e. For **Compression factor**, specify a value that you want the service to use when calculating the compression for the output. Valid values are floating point numbers in the range of 3.0 to 10.0, inclusive The bitrate of the output is calculated as follows:

   Output bitrate = (1 / compressionFactor) * (source bitrate)

   This property only applies if the source uses the CDI protocol.
5. Choose **Add output**.

# Viewing a list of outputs of a flow

You can view a list of a flow's outputs, along with the setup that is associated with each output. This list includes outputs that you added, as well as outputs that AWS Elemental MediaConnect added when subscribers create flows based on entitlements that you granted.

### To view a list of outputs on an existing flow (console)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that you want to view.

   The details page for that flow appears.
3. Choose the **Outputs** tab.

   A list of outputs for that flow appears.

### To view a list of outputs on an existing flow (AWS CLI)

- In the AWS CLI, use the `describe-flow` command:

  ```
  aws mediaconnect describe-flow --flow-arn "arn:aws:mediaconnect:us-
  east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame" --
  region us-east-1 --profile PMprofile
  ```

  The return value shows the details of the entire flow, including all the outputs. The following example shows the return value:

  ```
  {
    "Flow": {
      "AvailabilityZone": "us-east-1d",
      "Entitlements": [],
      "FlowArn": "arn:aws:mediaconnect:us-
  east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame",
      "Name": "BasketballGame",
      "Outputs": [
        {
          "Address": "192.0.2.12",
          "Description": "RTP-FEC Output",
          "Name": "NYCOutput",
  ```

```
        "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYCOutput",
        "Port": 5020,
        "Protocol": "rtp-fec"
      },
      {
        "Address": "198.51.100.8",
        "Description": "RTP Output",
        "Name": "DCOutput",
        "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-987655dEF67hiJ89-c34de5fG678h:DCOutput",
        "Port": 5110,
        "Protocol": "rtp"
      }
    ],
    "Source": {
      "IngestIp": "195.51.100.21",
      "IngestPort": 5010,
      "Name": "BasketballGameSource",
      "Protocol": "rtp-fec",
      "SourceArn": "arn:aws:mediaconnect:us-
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:BasketballGameSource",
      "AllowlistCidr": "10.24.34.0/23"
    },
    "Status": "STANDBY"
  }
}
```

# Updating outputs on a flow

You can update outputs on a flow, even when the flow is active.

### To update an output on a flow (console)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the output that you want to update.
3. Choose the **Outputs** tab.

   A list of outputs for that flow appears.
4. Choose the output that you want to update.

5. Choose **Update**.
6. Make the appropriate changes, and then choose **Save**.

**To update a flow output (AWS CLI)**

- In the AWS CLI, use the `update-flow-output` command:

```
aws mediaconnect update-flow-output --flow-arn "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame" --
output-arn "arn:aws:mediaconnect:us-east-1:111122223333:output:2-3aBC45dEF67hiJ89-
c34de5fG678h:NYCfeed" --port 5040 --region us-east-1 --profile PMprofile
```

The following example shows the return value:

```
{
  "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame",
  "Output": {
    "Address": "192.0.2.12",
    "Encryption": {
      "Algorithm": "aes256",
      "KeyType": "static-key",
      "RoleArn": "arn:aws:iam::111122223333:role/AllowMediaConnect",
      "SecretArn": "arn:aws:secretsmanager:us-west-2:111122223333:secret:SECRETID"
    },
    "Name": "Output1",
    "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:Output1",
    "Port": 5040,
    "Protocol": "rtp-fec"
  }
}
```

# Managing tags on an output

You can use tags to help you track the billing and organization for your AWS Elemental MediaConnect flows, sources, outputs, and entitlements. These are the same tags that AWS Billing and Cost Management provides for organizing your AWS bill. For more information about using tags for cost allocation, see Use Cost Allocation Tags for Custom Billing Reports in the *AWS Billing User Guide*.

**To add tags to an output (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the output that you want to add tags to.
3. Choose the **Outputs** tab.

   A list of outputs for that flow appears.
4. Choose the output that you want to add tags to.
5. Choose **Manage tags**.
6. Choose **Manage tags** again, and then choose **Add tag**.
7. For each tag that you want to add, do the following:

   a. Enter a key and a value. For example, your key can be **`sports`** and your value can be **`golf`**.

   b. Choose **Add tag**.
8. Choose **Update**.

**To edit tags on an output (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the output that you want to edit tags for.
3. Choose the **Outputs** tab.

   A list of outputs for that flow appears.
4. Choose the output that you want to edit tags for.
5. In the **Details** section, choose **Manage tags**.
6. Choose **Manage tags** again.
7. Update the tags, as needed.
8. Choose **Update**.

### To remove tags from an output (console)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the output that you want to remove tags from.
3. Choose the **Outputs** tab.

   A list of outputs for that flow appears.
4. Choose the output that you want to remove tags from.
5. In the **Details** section, choose **Manage tags**.
6. Choose **Manage tags** again.
7. Choose **Remove tag** next to each tag that you want to delete.
8. Choose **Update**.

# Removing outputs from a flow

You can remove outputs that you added to the flow. If AWS Elemental MediaConnect generated the output as the result of an entitlement, you must revoke the entitlement.

### To remove an output from a flow (console)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the output that you want to remove.

   The details page for that flow appears.
3. Choose the **Outputs** tab.
4. Choose the output, and then choose **Remove**.

### To remove an output from a flow (AWS CLI)

- In the AWS CLI, use the `remove-flow-output` command:

```
aws mediaconnect remove-flow-output --flow-arn "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame" --
```

```
output-arn "arn:aws:mediaconnect:us-east-1:111122223333:output:2-3aBC45dEF67hiJ89-
c34de5fG678h:Output1" --region us-west-2
```

The following example shows the return value:

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame",
    "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:Output1"
}
```

# Output destinations

Each output on a flow must be sent to a different destination. The parameters that define the destination depend on the protocol, but every protocol uses a compound identifier for the destination. For example, multiple outputs can point to the same destination IP address, as long as none of their ports overlap. Likewise, multiple outputs can point to the same stream ID as long as their remote IDs are different. The following table lists how each protocol defines the destination.

> **Note**
>
> Some protocols require additional ports for error correction. For outputs that use these protocols, AWS Elemental MediaConnect automatically reserves the additional ports. The protocol defines specifically which ports must be reserved. For example, some protocols require port+2 and port+4 for error correction. If you specify port 5000 for the output, the service assigns ports 5000, 5002, and 5004.

| Protocol | Destination definition | Ports required |
|---|---|---|
| CDI | Ports for each media stream | The ports that you specify for each media stream. These are the only ports needed for the output. |

| Protocol | Destination definition | Ports required |
|---|---|---|
| RIST | IP address, port, and port+1 | The port that you specify, plus one additional port. The service automatically reserves a port that is +1 from the port that you specified.<br>For example, if you specify port 3000 for this output, the service also reserves port 3001. |
| RTP | IP address and port | The port that you specify. This is the only port needed for the output. |
| RTP-FEC | IP address, port, port+2, and port+4 | The port that you specify, plus two additional ports. The service automatically reserves ports that are +2 and +4 from the port that you specified.<br>For example, if you specify port 2000 for this output, the service also reserves ports 2002 and 2004 for error correction. |
| SRT listener | CIDR allow list and port | The port that you specify. This is the only port needed for the output. |
| SRT caller | IP address and port | The port that you specify. This is the only port needed for the output. |

| Protocol | Destination definition | Ports required |
|---|---|---|
| Fujistu-QoS | CIDR allow list and port | The port that you specify. This is the only port needed for the output. |
| ST 2110 JPEG XS | Ports for each media stream | The ports that you specify for each media stream. These are the only ports needed for the output. |
| Zixi pull | Stream ID, remote ID, and CIDR allow list | The service automatically uses port 2077 for these outputs. |
| Zixi push | IP address, stream ID, and port | The port that you specify is the only port needed for the output. |

# Determining an output's IP address

For flows that use listener protocols (such as Zixi pull or SRT listener), the receiver requires the IP address of the output to establish a connection with the flow.

**To determine an output's IP address**

1. On the **Flows** page, choose the name of the flow that you want to view.
2. For specific instructions based on how content is sent to your output, choose one of the following tabs:

   Public internet

   1. In the **Details** section, note the **Public outbound IP address**. This is the IP address that the receiver needs.

   Private internet

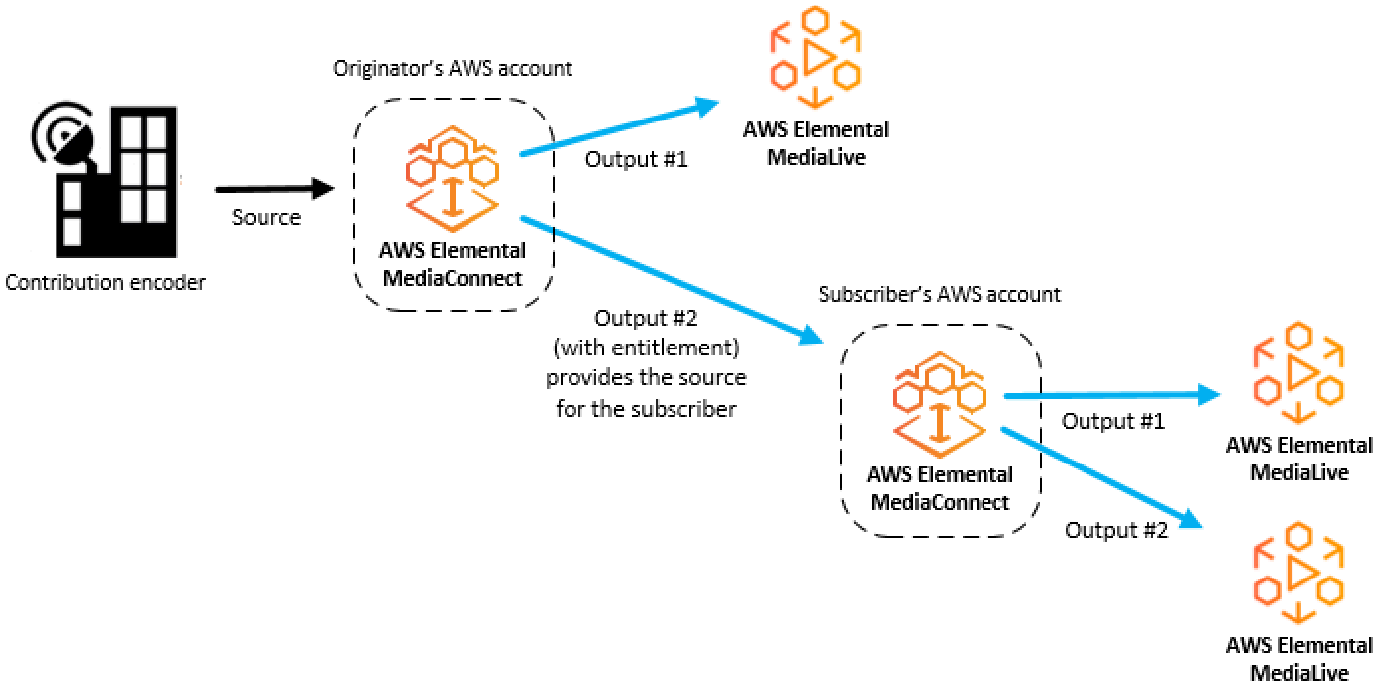
   1. Choose the **Outputs** tab, and then locate the output that you want to view.

2. Under **Listener address** for that output, note the IP address. This is the IP address that the receiver needs.

# Entitlements in AWS Elemental MediaConnect

Content originators can grant entitlements to share their content with other AWS accounts (subscriber accounts). Subscribers can then set up their own AWS Elemental MediaConnect flows using the originator's flow as their source. The following illustration shows this process.

> **Note**
>
> You can only grant entitlements on transport stream flows. MediaConnect doesn't support entitlements on CDI flows.



## Topics

- Sharing content with other AWS accounts
- Subscribing to content provided by another AWS account

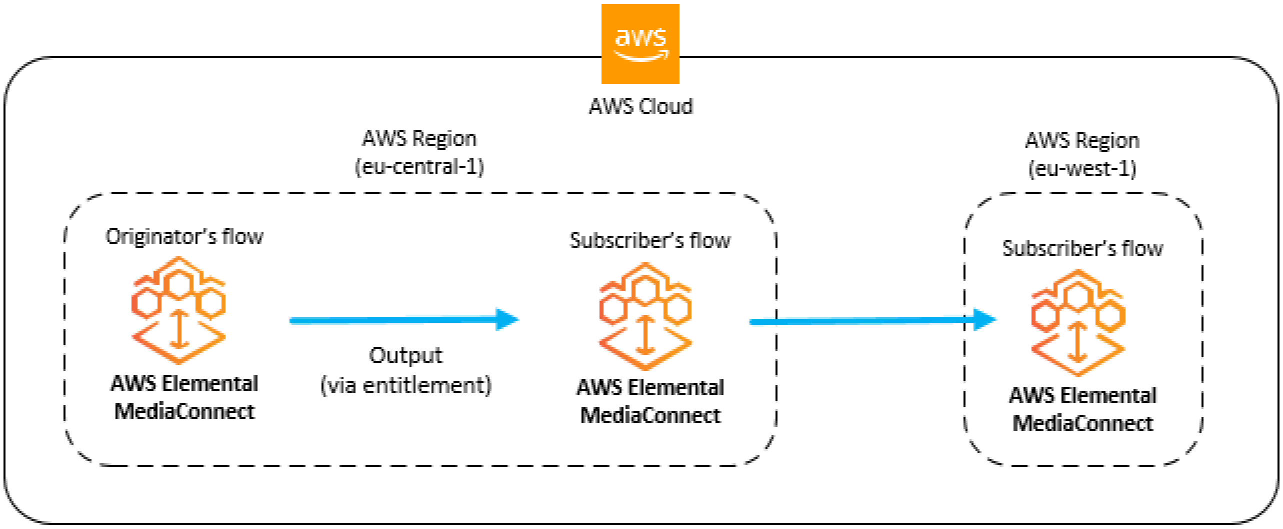
# Sharing content with other AWS accounts

You can grant an entitlement to share the content in your AWS Elemental MediaConnect flow with another AWS account (subscriber account). When the subscriber sets up a flow based on the entitlement, the service generates an output on your flow to represent the stream from your flow to the subscriber's flow. This output is counted as part of the 50 maximum outputs that you can have on your flow.

You can grant, update, and revoke entitlements at any time, even on an active flow. If you want to stop streaming content to the subscriber's flow on a temporary basis, you can disable the entitlement. Later, you can enable the entitlement when you're ready to allow content to stream to the subscriber's flow again. You can also specify the percentage of the entitlement data transfer fee that you want the subscriber to be responsible for.

> **Note**
>
> If you grant an entitlement and later disable it (to temporarily stop streaming content to the subscriber's flow), the entitlement remains associated with your flow and counts toward your maximum number of entitlements. However, if you revoke the entitlement (to permanently stop streaming content to the subscriber's flow), the entitlement is removed from your flow and no longer counts toward the maximum number of entitlements.

After you grant an entitlement, you provide information about the entitlement (name, AWS Region, and encryption details) to the subscriber. The subscriber uses this information to create a MediaConnect flow that uses your flow as the source. The subscriber's flow must be in the same AWS Region as your flow. If the subscriber wants a flow in a different Region, they must create a second flow in the new Region. The following illustration shows this process.



**Note**

You can only grant entitlements on transport stream flows. MediaConnect doesn't support entitlements on CDI flows.

**Topics**

## Granting an entitlement on a flow

You can grant an entitlement to an existing flow to share your content with another AWS account (the subscriber account). The subscriber creates an AWS Elemental MediaConnect flow in the same AWS Region, using your flow as the source. When this happens, the service generates an output on your flow to represent the video stream from your flow to the subscriber's flow.

The subscriber can use an entitlement only once.

### Prerequisites

Before you can grant an entitlement, you must do the following:

- Obtain the subscriber's AWS account number.
- If you want to encrypt the video as it is sent from your flow to the subscriber's flow, set up encryption using static key encryption or Secure Packager and Encoder Key Exchange (SPEKE).

**To grant an entitlement on a flow (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that you want to grant an entitlement on.

   The details page for that flow appears.
3. Choose the **Entitlements** tab.
4. Choose **Grant entitlement**.

   The **Grant entitlement** page appears.
5. For **Name**, specify a name for the entitlement that will help you and the subscriber differentiate this flow from other flows. The name also becomes part of the entitlement ARN, which is visible to the subscriber.
6. For **Subscriber account ID**, specify the subscriber's 12-digit AWS account ID. Don't include hyphens in the ID.
7. For **Description**, specify a description that will help you identify this entitlement later. The description is visible only on the AWS Elemental MediaConnect console for your account.
8. For **Data transfer subscriber fee percent**, specify the percentage of the entitlement data transfer fee that you want the subscriber to be responsible for. AWS bills your account for the remainder. For example, if you specify **15**, AWS bills the subscriber's account for 15% of the entitlement data transfer fee and your account for the remaining 85%.

   > **Note**
   >
   > Even if you specify that the subscriber is responsible for a portion or all of the entitlement data transfer fee, the subscriber will not incur fees until they create and start a flow that is based on this entitlement.

9. For **Entitlement status**, specify whether you want the entitlement enabled or disabled. If the entitlement is enabled, the subscriber can create a flow based on the entitlement and start streaming content right away. If the entitlement is disabled, the subscriber must wait for you to enable it before content can stream from your flow to their flow.

10. If you want to encrypt the video as it is sent from your flow to the subscriber's flow, choose one of the following tabs:

    Static key encryption

    1. In the **Encryption** section, choose **Enable**.
    2. For **Encryption type**, choose **Static key**.
    3. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.
    4. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.
    5. For **Encryption algorithm**, choose the type of encryption that you want to use to encrypt the source.

    SPEKE encryption

    1. In the **Encryption** section, choose **Enable**.
    2. For **Encryption type**, choose **SPEKE**.
    3. For **Encryption algorithm**, choose the type of encryption that you want to use to encrypt the source.
    4. For **Role ARN**, enter the Amazon Resource Name (ARN) of the IAM role that provides you access to send your requests through API Gateway. You created this role when you set up encryption.

       The following example shows a role ARN:

       ```
       arn:aws:iam::111122223333:role/SpekeAccess
       ```

    5. For **Resource ID**, enter an identifier for the content. The service sends this to the key server to identify the current endpoint. How unique you make this depends on how fine-grained you want access controls to be. The resource ID is also known as the content ID.

       The following example shows a resource ID:

```
    MovieNight20171126093045
    ```

    6. For **Device ID**, enter the value of one of the devices that you configured with your conditional access (CA) platform key provider.
    7. For **URL**, enter the URL of the API Gateway proxy that you set up to talk to your key server. The API Gateway proxy must reside in the same AWS Region as MediaConnect.

       The following example shows a URL.

       ```
       https://1wm2dx1f33.execute-api.us-west-2.amazonaws.com/SpekeSample/
       copyProtection
       ```

    8. (Optional) For **Constant initialization vector** enter a 128-bit, 16-byte hex value represented by a 32-character string, to be used with the key for encrypting content.

11. At the bottom of the page, choose **Grant entitlement**.
12. On the **Entitlements** tab, locate the new entitlement in the list.
13. Make a note of the entitlement ARN.
14. Provide the following information to the subscriber:

    - The entitlement ARN
    - The AWS Region that you created the flow in
    - The encryption key and algorithm if you set up encryption on the entitlement
    - The percentage of the entitlement data transfer fee that the subscriber is responsible for

> **Note**
>
> MediaConnect suppresses null packets in an effort to optimize the data connection between the content originator's flow and the subscriber's flow. This can result in a fluctuating bitrate on the subscriber's flow, or a difference between the bitrate of the content originator's flow and the subscriber’s flow. We recommend that you monitor source health as a combination of `SourceBitRate` and other metrics such as `SourceContinuityCounter` and `SourceNotRecoveredPackets`.

### To grant an entitlement on a flow (AWS CLI)

1. Create a JSON file that contains the details of the entitlements that you want to grant.

   The following example shows the structure for the contents of the file:

```
[
  {
    "Description": "For AnyCompany",
    "Encryption": [
      {
        "Algorithm": "aes128",
        "KeyType": "static-key",
        "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
        "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1"
      }
    ],
    "Name": "AnyCompany_Entitlement",
    "Subscribers": [
      "444455556666",
      "123456789012"
    ]
  },
  {
    "Description": "For Example Corp",
    "Name": "ExampleCorp",
    "Subscribers": [
      "777788889999"
    ]
  }
]
```

2. In the AWS CLI, use the `grant-flow-entitlements` command:

```
aws mediaconnect grant-flow-entitlements --entitlements --flow-
arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame  --cli-input-
json file://entitlements.json
```

   The following example shows the return value:

```
{
    "Entitlements": [
        {
            "Name": "AnyCompany_Entitlement",
            "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement",
            "Subscribers": [
                "444455556666", "123456789012"
            ],
            "Description": "For AnyCompany",
            "Encryption": {
                "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1",
                "Algorithm": "aes128",
                "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
                "KeyType": "static-key"
            }
        },
        {
            "Name": "ExampleCorp",
            "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-3333cccc4444dddd-1111aaaa2222:ExampleCorp",
            "Subscribers": [
                "777788889999"
            ],
            "Description": "For Example Corp"
        }
    ],
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame"
}
```

## Updating an entitlement

After an entitlement has been created, you can still update the description, status, and subscribers. If you change the subscriber account ID, the content becomes unavailable to the original subscriber account. If the original subscriber already created a flow that used the entitlement as a source, the associated output is removed from your flow.

### To update an entitlement (console)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the entitlement that you want to update.

   The details page for that flow appears.
3. Choose the **Entitlements** tab.
4. Choose the entitlement that you want to update.
5. Choose **Update**.
6. Make the appropriate changes, and then choose **Save**.

### To update an entitlement on a flow (AWS CLI)

- In the AWS CLI, use the `update-flow-entitlement` command:

```
aws mediaconnect update-flow-entitlement --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame --
entitlement-arn arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement
 --description 'For AnyCompany Affiliate' --subscribers 444455556666",
 "123456789012
```

  The following example shows the return value:

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
    "Entitlement": {
        "Name": "AnyCompany_Entitlement",
        "Description": "For AnyCompany Affiliate",
        "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement",
        "Encryption": {
            "KeyType": "static-key",
            "Algorithm": "aes128",
            "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
            "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1"
```

```
        },
        "Subscribers": [
            "444455556666", "123456789012"
        ]
    }
}
```

## Managing tags on an entitlement

You can use tags to help you track the billing and organization for your AWS Elemental MediaConnect flows, sources, outputs, and entitlements. These are the same tags that AWS Billing and Cost Management provides for organizing your AWS bill. For more information about using tags for cost allocation, see Use Cost Allocation Tags for Custom Billing Reports in the *AWS Billing User Guide*.

**To add tags to an entitlement (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the entitlement that you want to add tags to.
3. Choose the **Entitlements** tab.

   A list of entitlements for that flow appears.
4. Choose the entitlement that you want to add tags to.
5. Choose **Manage tags**.
6. Choose **Manage tags**, and then choose **Add tag**.
7. For each tag that you want to add, do the following:

   a. Enter a key and a value. For example, your key can be **`sports`** and your value can be **`golf`**.

   b. Choose **Add tag**.
8. Choose **Update**.

**To edit tags on an entitlement (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.

2. On the **Flows** page, choose the name of the flow that is associated with the entitlement that you want to edit tags for.
3. Choose the **Entitlements** tab.

   A list of entitlements for that flow appears.
4. Choose the entitlement that you want to edit tags for.
5. In the **Details** section, choose **Manage tags**.
6. Choose **Manage tags**.
7. Update the tags, as needed.
8. Choose **Update**.

**To remove tags from an entitlement (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the entitlement that you want to remove tags from.
3. Choose the **Entitlements** tab.

   A list of entitlements for that flow appears.
4. Choose the entitlement that you want to remove tags from.
5. In the **Details** section, choose **Manage tags**.
6. Choose **Manage tags**.
7. Choose **Remove tag** next to each tag that you want to delete.
8. Choose **Update**.

## Revoking an entitlement

After you revoke an entitlement, the content becomes unavailable to the subscriber account permanently. The entitlement and the associated output are removed from your flow. If you revoke an entitlement and later decide you need to grant that entitlement again, the subscriber's flow must be manually restarted. The subscriber's flow will not start automatically after the entitlement has been granted.

If you want to stop streaming content to the subscriber’s flow temporarily, disable the entitlement instead.

### To revoke an entitlement (console)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the entitlement that you want to revoke.

   The details page for that flow appears.
3. Choose the **Entitlements** tab.
4. Choose the entitlement that you want to revoke.
5. Choose **Revoke**.

### To revoke an entitlement on a flow (AWS CLI)

- In the AWS CLI, use the `revoke-flow-entitlement` command:

  ```
  aws mediaconnect revoke-flow-entitlement --flow-arn arn:aws:mediaconnect:us-
  east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame --
  entitlement-arn arn:aws:mediaconnect:us-
  west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement
  ```

  The following example shows the return value:

  ```
  {
      "FlowArn": "arn:aws:mediaconnect:us-
  east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
      "EntitlementArn": "arn:aws:mediaconnect:us-
  west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement"
  }
  ```

## Disabling an entitlement temporarily

When you disable an entitlement, the content becomes unavailable to the subscriber account immediately. However, the entitlement and the associated output remain on your flow. These resources continue to count toward your quota for outputs and entitlements. Later, you can enable the entitlement to re-instate access.

If you want to stop streaming content to the subscriber's flow permanently, revoke the entitlement instead. That action removes the entitlement and the associated output from your flow.

**To disable an entitlement (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the entitlement that you want to disable.

   The details page for that flow appears.
3. Choose the **Entitlements** tab.
4. Choose the entitlement that you want to disable.
5. Choose **Disable**.

## Enabling an entitlement that has been temporarily disabled

If an entitlement has been disabled, you can enable it to start streaming content to the subscriber’s flow again.

> **Note**
>
> If the entitlement was revoked, you can't enable it. You must grant a new entitlement.

**To enable an entitlement (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that is associated with the entitlement that you want to enable.

   The details page for that flow appears.
3. Choose the **Entitlements** tab.
4. Choose the entitlement that you want to enable.
5. Choose **Enable**.

# Subscribing to content provided by another AWS account

When another AWS account (originator account) grants an entitlement to your AWS account (subscriber account), you can create a flow that uses the originator's content as your source. To subscribe to content provided by another AWS account, you create a flow based on the entitlement granted to you. You must set up your flow in the same AWS Region as the originator's flow.

You can use an entitlement only once.

**Note**

MediaConnect suppresses null packets in an effort to optimize the data connection between the content originator's flow and the subscriber's flow. This can result in a fluctuating bitrate on the subscriber's flow, or a difference between the bitrate of the content originator's flow and the subscriber’s flow. We recommend that you monitor source health as a combination of `SourceBitRate` and other metrics such as `SourceContinuityCounter` and `SourceNotRecoveredPackets`.

### Prerequisites

Before you can create your flow, you must do the following:

- Obtain the following information from the content originator:
  - The entitlement ARN
  - The AWS Region that the originator created the flow in
  - The encryption key and algorithm if the originator set up encryption on the entitlement
- If the entitlement is encrypted using static key encryption, store the encryption key in AWS Secrets Manager before you begin this procedure. (If the content is encrypted using SPEKE, you don't need to do anything to configure encryption.)

### To create a flow based on an entitlement (console)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. Verify that you are logged in to the same AWS Region that the originator's flow is in.
3. On the **Flows** page, choose **Create flow**.
4. In the **Details** section, for **Name**, specify a name for your flow.
5. For **Availability Zone**, choose an Availability Zone for your flow. This does not need to match the Availability Zone of the originator's flow.
6. In the **Source** section, for **Source type**, choose **Entitled source**.

7. For **Entitlement ARN**, choose the appropriate entitlement. This list includes all entitlements that have been granted to you.

   > **Tip**
   >
   > You can click in this field and start typing the entitlement name. AWS Elemental MediaConnect will filter the list to include only entitlements with a name that matches what you type.

   > **Note**
   >
   > The percentage of the entitlement data transfer fee that you are responsible for is listed next to each entitlement. This value is set by the content originator.

8. If the originator set up encryption on the entitlement, choose **Enable** in the **Decryption** section and do the following:

   a. For **Decryption type**, choose **Static key**.

   b. For **Role ARN**, specify the ARN of the role that you created when you set up encryption.

   c. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key.

   d. For **Decryption algorithm**, choose the type of encryption that the originator provided.

9. At the bottom of the page, choose **Create flow**.

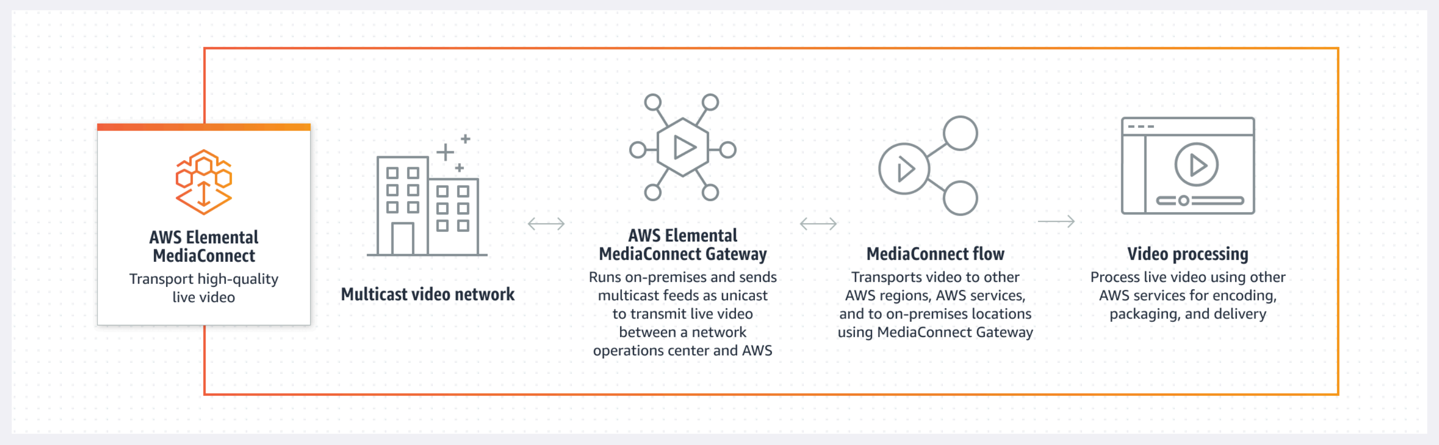
   > **Note**
   >
   > The flow does not start automatically. You must start the flow manually.

10. Add outputs to specify where you want AWS Elemental MediaConnect to send the content, or grant entitlements to allow users of other AWS accounts to subscribe to your content.

# AWS Elemental MediaConnect Gateway

*AWS Elemental MediaConnect Gateway* is a feature of MediaConnect that deploys on-premises resources for transporting live video to and from the AWS Cloud. MediaConnect Gateway allows you to contribute live video to the AWS Cloud from on-premises hardware, as well as distribute live video from the AWS Cloud to your local data center.

The following graphic depicts a workflow where AWS Elemental MediaConnect Gateway runs on-premises and sends multicast feeds as unicast. This process transmits live video between the on-premises operations center and the AWS Cloud. From there, AWS Elemental MediaConnect Gateway distributes that same content to a different on-premises location.


AWS Elemental MediaConnect
Transport high-quality live video
Multicast video network
AWS Elemental MediaConnect Gateway
Runs on-premises and sends multicast feeds as unicast to transmit live video between a network operations center and AWS
MediaConnect flow
Transports video to other AWS regions, AWS services, and to on-premises locations using MediaConnect Gateway
Video processing
Process live video using other AWS services for encoding, packaging, and delivery

This section will cover the following topics:

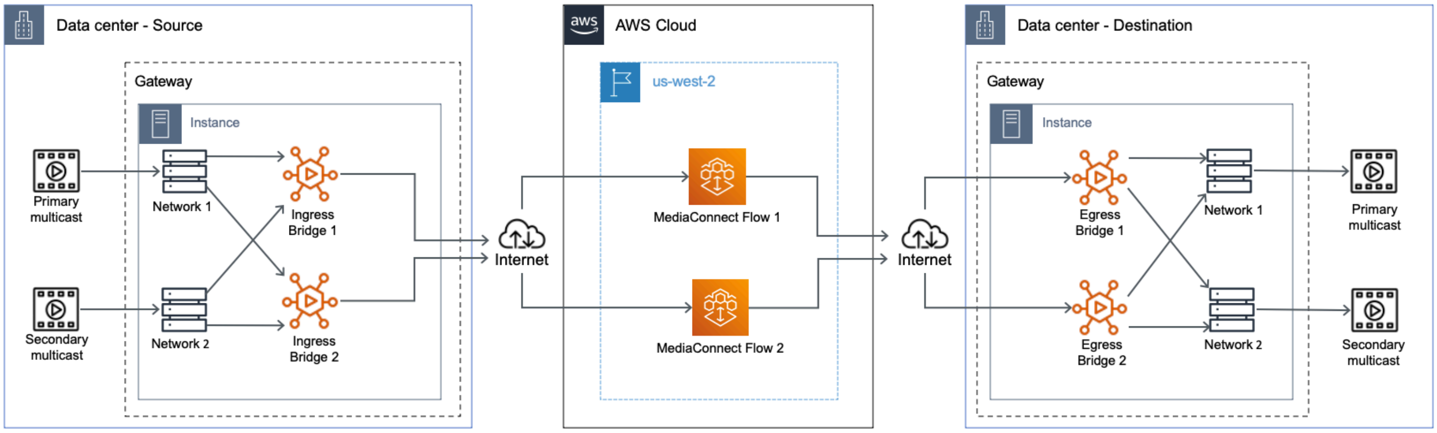
- Prerequisites: on-premises system information and other considerations for using MediaConnect Gateway.
- Components of MediaConnect Gateway: an explanation of MediaConnect Gateway and its components.
- Creating a gateway: step-by-step instructions for building your gateway and its components.

# Components of MediaConnect Gateway

AWS Elemental MediaConnect Gateway is made up of four major components: *gateways*, *networks*, *instances*, and *bridges*. Each of these components are explained in greater detail in the following sections of this guide. The following describes the basic relationship of these components:

- **Gateways**: A logical grouping of instances and bridges. Each gateway utilizes user-defined IP information for communication between data centers and the AWS Cloud.
- **Networks**: A MediaConnect Gateway network is a collection of IP information that instances and bridges use to communicate on your local data center network. The network information must match the local data center network that you are using to communicate with gateway. Each MediaConnect Gateway may contain a maximum of two networks. All gateways must contain at least one network.
- **Instances**: A compute instance running on equipment in your data center and managed by MediaConnect. This instance is an on-premises implementation of the MediaConnect service and is contained within a gateway. Instances use bridges to communicate between your data center and the AWS Cloud. You create instances by installing software on an on-premises server.
- **Bridges**: A connection between your data center's instances and the AWS Cloud. A bridge can be used to send video from the AWS Cloud to your data center or from your data center to the AWS Cloud.

The following graphic depicts the interactions of each component in a common workflow scenario. In this workflow, multicast from the data center is ingested into a gateway instance and contributed across a bridge to MediaConnect in the AWS Cloud. From the AWS Cloud, the multicast is distributed to a different data center's gateway instance.



## MediaConnect Gateway terminology

The following section provides details about MediaConnect Gateway concepts and terminology.

- **Ingress**: In MediaConnect Gateway, ingress refers to content contributed to the AWS Cloud from an on-premises location. If the content is leaving your location using an ingress bridge, this means its destination is AWS.

- **Egress**: In MediaConnect Gateway, egress refers to content distributed to your on-premises location from the AWS Cloud. If the content is entering your location using an egress bridge, this means its source is AWS.
- **Cloud flow**: A MediaConnect flow that exists in the AWS Cloud. Typically, this will be an existing MediaConnect flow that you might already be using and want to distribute to an on-premises gateway.
- **Flow source**: A source that originates in the AWS Cloud. An egress bridge uses this type of source.
- **Network source**: A source that originates at your on-premises location. An ingress bridge uses this type of source.
- **Flow output**: An output that is delivered to the AWS Cloud. An ingress bridge uses this type of output.
- **Network output**: An output that is delivered to your on-premises location. An egress bridge uses this type of output.

# Prerequisites

Before you can use AWS Elemental MediaConnect Gateway, you need an AWS account and the appropriate permissions to access, view, and edit MediaConnect components. Additionally, you will need physical hardware that complies to the MediaConnect Gateway requirements listed in the following sections.

## Supported operating systems and system architectures

### General information

AWS Elemental MediaConnect Gateway is built on the Amazon Elastic Container Service Anywhere (ECS Anywhere) service. Amazon ECS Anywhere provides support for registering an *external instance*, such as an on-premises server to your AWS infrastructure. Because of this architecture, external instances using MediaConnect Gateway must conform to the Amazon ECS Anywhere requirements and additional requirements specifically for MediaConnect Gateway. The following sections list hardware and operating system (OS) requirements, in addition to MediaConnect Gateway-specific requirements.

The following table contains the default quotas for each MediaConnect Gateway component.

| Component | Default quota | Can this quota be increased? |
|---|---|---|
| Maximum number of gateways for each AWS Region | 3 | Yes |
| Maximum number of instances for each gateway | 20 | No |
| Maximum number of bridges for each gateway | 40 | No |
| Maximum bitrate for each bridge | 100 Mbps | No |

## Supported system architectures

The following table contains the recommended system architectures for your individual gateway instances. The system will determine the maximum number of bridges that can run on the instance. Only x86_64 CPU architectures are supported. MediaConnect Gateway does not support ARM-based CPUs:

| Number of bridges | vCPU cores (2.6 GHz) | vCPU cores (3.0 GHz) | Minimum RAM (GB) | Minimum disk space (GB) |
|---|---|---|---|---|
| 10 | 2 | 2 | 4 | 25 |
| 25 | 6 | 4 | 8 | 25 |
| 40 | 10 | 8 | 16 | 25 |

### CPU references

The CPU architectures are benchmarked using these CPUs:

- 2.6 GHz - Intel E5-2660 v3
- 3.0 GHz - AMD 7302

### Supported operating systems

The following list contains the supported operating systems (OS) and software configurations for your MediaConnect Gateway instances.

#### Recommended operating system

- Ubuntu 20.04

#### Supported operating systems

You can register MediaConnect Gateway instances to other Linux distributions that are supported by Amazon ECS Anywhere. Windows operating systems are not supported by MediaConnect Gateway. See the Amazon ECS user guide for the complete list of supported Linux distributions: Supported operating systems

#### Required software

- Docker - MediaConnect Gateway requires that you install the latest release of Docker. If you are using a Linux distribution other than RHEL, the instance registration script provided by MediaConnect will install Docker for you. Neither Docker or RHEL's open package repositories support installing Docker natively on RHEL. When using RHEL, you must ensure that Docker is installed before you run the instance registration script that's described in this document.

# Networks

A gateway *network* is a collection of IP information that will be used by the instances and bridges to communicate on your local data center network. The gateway network information must match the local data center network that you are using to communicate with the gateway. Each gateway may contain a maximum of two networks. All gateways must contain at least one network.

## Creating or deleting a gateway network

You must create the networks during the initial creation of a new gateway. You can't add or edit a network after the initial creation of the gateway. For more information about the initial creation of a gateway and its networks, see Create a gateway (console).

To delete a network, you must delete the gateway that's associated with it. For more information about deleting a gateway and its networks, see Removing a gateway and its components (console).

# Instances

An *instance* is a compute instance running on equipment in your data center and managed by MediaConnect Gateway. This instance is an on-premises implementation of the MediaConnect service and is contained within a gateway. Instances use bridges to communicate between your data center and the AWS Cloud. Instances are created by installing software on an on-premises server.

## Registering a MediaConnect Gateway instance

You can register an instance by running a custom Linux command on the device that will be hosting the instance. You generate the command by following the instance registration process in the AWS Management Console.

**To register a MediaConnect Gateway instance**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. From the navigation pane, select **Gateways**. In the **Gateways** section, select the gateway you want to register the instance to.
3. On the gateway **Details** page, select the **Instances** tab. Select **Register instance**.
4. On the **Register Gateway instances** page, complete the following steps:
    1. For **Activation key duration**, enter the number of days that the activation key will remain active. After that number of days, the key will no longer work when registering a gateway instance.
    2. For **Number of instances**, enter the number of instances that you want to register to your gateway with this activation key.
    3. For **Instance role**, choose the IAM role to associate with your external instances.
    4. Select **Generate registration command**.
5. A **Linux command** will be displayed. Copy the command. You must run this command on each instance you want to register to this gateway.

    > ⚠ **Important**
    >
    > The bash portion of the script must be run as root. If the command isn't run as root, an error is returned.

6. After a few minutes, the instance will register to the gateway. All instances registered to this gateway will appear in the **Instances** tab.

## Deregistering a gateway instance

You can deregister an instance you no longer want to use within MediaConnect Gateway. By deregistering the instance, it will no longer support bridges and will not be a part of your gateway. If you want to reuse the instance for Amazon ECS Anywhere or as another gateway instance, you will need to follow the additional steps in **Step 6** to prepare the deregistered instance for reuse.

**To deregister a gateway instance**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. From the navigation pane, select **Gateways**. In the **Gateways** section, select the gateway that contains the instance you want to deregister.
3. On the gateway **Details** page, select the **Instances** tab. Select the **Instance ID** of the instance you want to deregister.
4. Select **Deregister**.
5. Confirm the deregistration of the instance by selecting **Deregister instance**.
6. Repeat the previous steps for any additional instances you need to deregister.

**To reuse a gateway instance (optional)**

If you want to reuse the instance for Amazon ECS Anywhere or as another gateway instance, you will need to complete the following steps.

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. From the navigation pane, select **Gateways**. In the **Gateways** section, select the gateway that contains the instance you want to reuse.
3. On the gateway **Details** page, select the **Instances** tab. Locate the **Instance ID** of the instance you want to reuse.
4. Make sure that the **Instance state** is **Deregistered** for the instance you want to reuse.
5. From a computer with the access to do so, connect to the instance using SSH.
6. Run the following commands, in order.

```
sudo docker stop $(docker ps -f "name=MediaConnectGatewayAgent" -q); \
sudo docker stop ecs-agent; \
sudo systemctl stop ecs amazon-ssm-agent; \
sudo yum remove -y amazon-ecs-init amazon-ssm-agent;  `# or apt or snap as needed`
 \
sudo rm /var/lib/ecs /etc/ecs /var/lib/amazon/ssm /var/log/ecs /var/log/amazon/ssm
 -rf; \
sudo docker rm -f ecs-agent ssm-agent; \
sudo docker container rm -f $(docker ps -a -f "name=MediaConnectGatewayAgent" -q);
 \
sudo docker volume rm -f ecsdata docker run; \
sudo pkill -f -KILL network_bootstra[p]; \
sudo pkill -KILL mcproxy;
```

For more information about deleting a MediaConnect Gateway and its networks, see: Removing a gateway and its components (console)

# Bridges

A *bridge* is a connection between your data center's instances and the AWS Cloud. Depending on the selected bridge type, a bridge can be used to send content from the AWS Cloud to your data center or from your data center to the AWS Cloud.

## Bridge types

AWS Elemental MediaConnect Gateway supports two types of bridges. Each bridge type serves a different purpose and determines if you will be contributing content to the AWS Cloud or distributing content to a physical location. The following are the two types of bridges and their different functions:

**Ingress bridge**: A ground-to-cloud bridge. On an ingress bridge, the content originates at your premises and is delivered to the AWS Cloud.

**Egress bridge**: A cloud-to-ground bridge. On an egress bridge, the content comes from an existing MediaConnect flow and is delivered to your premises.

## Bridge sources

Each bridge requires you to create a minimum of one source. The source is the content that will be ingested by the MediaConnect Gateway. The origin of the source content will be different depending on the bridge type you select. If you create multiple bridge sources, you can enhance the resiliency of your bridge by activating failover during the creation process. The following are the two types of sources:

- **Ingress bridge source**: For an ingress bridge, the content originates at your premises and is delivered to the cloud. When creating an ingress bridge source, you will need to select the protocol (RTP, RTP-FEC, or UDP) and enter the multicast IP address and port of the content originating in your premises.
- **Egress bridge source**: For an egress bridge, the content originates as an existing MediaConnect flow and is delivered to your premises. When creating an egress bridge source, you will need to select the MediaConnect flow that you would like to send to your premises. You don't need to select the protocol. The source will use the same protocol as the existing flow.

### Bridge source failover

If you create multiple bridge sources, you can enhance the resiliency of your bridge by activating failover during the creation process. The failover configuration determines how AWS Elemental MediaConnect Gateway behaves in the event of source input loss. The bridge type will determine which of the two failover modes are available. The following are the two failover modes:

- **Failover**: This mode allows switching between a primary and a backup source. You can specify a source as the primary source. The second source serves as the backup. The service switches to the backup source if the primary source fails, and switches back to the primary source as soon as it is reliable.
- **Merge**: This mode combines the sources into a single stream, allowing a graceful recovery from any single-source loss. In merge mode, if a source is missing a packet the service pulls the missing packet from the other source.

## Bridge outputs

Each bridge requires you to create a minimum of one output. The following are the two types of outputs:

- **Ingress bridge output**: For an ingress bridge, the content originates at your premises and is delivered to the cloud. You do not need to configure outputs for ingress bridge types. When you create a MediaConnect flow using the ingress bridge as a source, the output is automatically created when the flow is started.

- **Egress bridge output**: For an egress bridge, the content originates as an existing MediaConnect flow and is delivered to your premises. When you create an egress bridge output, you will need to configure the IP and protocol information that will be delivered to your premises. Egress bridge outputs support RTP, RTP-FEC, and UDP protocols.

## Creating a MediaConnect Gateway bridge

After you have registered at least one instance to your Gateway, you can create a bridge. The process for creating a bridge will vary depending on the bridge type you select in step 4.

**To create an ingress bridge**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. From the navigation pane, select **Gateways**. In the **Gateways** section, select the gateway you want to create the bridge on.
3. On the gateway **Details** page, select the **Bridges** tab. Select **Create bridge**.
4. On the **Create bridge** page, complete the following steps in the **Details** section:

    1. Enter a **Name** for the bridge.
    2. For **Bridge type**, select **Ingress bridge**.
    3. Enter the **Maximum bitrate** for the content you will transport over the bridge.
    4. Enter the **Maximum outputs** for the bridge.
5. Next, complete the following steps in the **Sources** section. The source of an ingress bridge is multicast content that originates at your premises. To create a source:

    1. Enter a **Name** for the bridge source.
    2. Select a **Network**. This is a network you created during the gateway setup process.
    3. Select the **Protocol** of the source content.
    4. Enter the **Multicast IP** and the **Port** of the source.
6. If you add more than one source, you can setup failover in the **Failover configuration** section.

    a. Select the **Failover mode**: **Failover** or **Merge**

    b. If you select **Failover** as the mode, select one of the sources you configured in step 5 to be the **Primary source**.
7. Select **Create bridge**.

8. After the bridge is created, you can start the bridge by selecting **Start** on the bridge's **Details** page.

**To create an Egress bridge**

1. Open the MediaConnect console at [https://console.aws.amazon.com/mediaconnect/](https://console.aws.amazon.com/mediaconnect/).
2. From the navigation pane, select **Gateways**. In the **Gateways** section, select the gateway you want to create the bridge on.
3. On the gateway **Details** page, select the **Bridges** tab. Select **Create bridge**.
4. On the **Create bridge** page, complete the following steps in the **Details** section:

   1. Enter a **Name** for the bridge.
   2. For **Bridge type**, select **Egress bridge**.
   3. Enter the **Maximum bitrate** for the content you will transport over the bridge.
5. Next, complete the following steps in the **Sources** section:

   1. Enter a **Name** for the bridge source. For an Egress bridge, the source is the content coming from a MediaConnect flow and delivered to your premises.
   2. Select a **Network**. This is a network you created during the gateway setup process.
   3. Select the **Flow ARN**. This is the ARN of the MediaConnect flow you will use as a source.
   4. If this flow uses a **VPC interface**, select it.
6. If you add more than one source, you can setup failover in the **Failover configuration** section.

   a. When you select an egress bridge, the only available **Failover mode** is **Failover**. **Merge** cannot be selected.

   b. Select one of the sources you configured in step 5 to be the **Primary source**.
7. The final section in egress bridge creation is **Outputs**. Complete the following steps.

   1. Enter a **Name** for the bridge output.
   2. Select a **Network**. This is a network you created during the gateway setup process.
   3. Select the transport **Protocol** you want to use for the output.
   4. Enter an **IP address** for the output. This must be an IP that is compatible with your local network.
   5. Enter the **Port** for the output. This must be a port that is compatible with your local network.

6. Enter a **TTL** (time-to-live) for the output.

8. Select **Create bridge**.

9. After the bridge is created, you can start the bridge by selecting **Start** on the bridge's **Details** page.

# Creating a gateway (console)

Setup begins with creating the gateway. This can be done in the MediaConnect console, programmatically using the MediaConnect API, or by using AWS CloudFormation. After a MediaConnect Gateway and its Networks are created, you can begin registering instances to that MediaConnect Gateway and creating Bridges on those instances.

**Topics**

## Create a gateway (console)

Your first step is to create the gateway and a network. The gateway is a logical grouping of instances and bridges. Each gateway utilizes user-defined IP information for communication between data centers and the AWS Cloud.

**To create a gateway**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. From the navigation pane, select **Gateways**. In the **Gateways** section, choose **Create gateway**.
3. On the **Create gateway** page, enter a **Name** for your gateway. This name can't be modified later.
4. For the **Egress CIDR blocks**: Enter a CIDR block for the egress of your gateway. These IP addresses should be in the form of a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. This CIDR block represents a range of IP addresses that are allowed to contribute content or initiate output requests for flows communicating with this gateway.

> ⚠ **Important**
>
> Don't use 0.0.0.0/0 for the **Egress CIDR blocks**. This will open the gateway to the public.

5. In the **Networks** section, enter a name for your first network. A gateway may contain a maximum of two networks. Each network name must be unique for this gateway.
6. Enter a **CIDR block** for this network. To complete the creation of the gateway, choose the **Create Gateway** button.

## Register an instance (console)

After you create a gateway, you can register instances to that gateway. An instance is a computing resource that runs on equipment in your data center and managed by MediaConnect. This instance is an on-premises implementation of the MediaConnect service and is contained within a gateway. Instances use bridges to communicate between your data center and the AWS cloud. Instances are created by installing software on an on-premises server.

**To register an instance**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. From the navigation pane, select **Gateways**. In the **Gateways** section, select the gateway you want to register the instance to.
3. On the gateway's **Details** page, select the **Instances** tab.
4. On the **Instances** tab, choose **Register instance**.
5. On the **Register Gateway instances** page, complete the following steps:
   1. For **Activation key duration**, enter the number of days that the activation key will remain active. After that number of days, the key will no longer work when registering a gateway instance.
   2. For **Number of instances**, enter the number of instances that you want to register to your gateway with the activation key.
   3. For **Instance role**, choose the AWS Identity and Access Management (IAM) role to associate with your external instances.
   4. Choose **Generate registration command**.

6. A **Linux command** will be displayed. Copy the command. You must run this command on each instance you want to register to this gateway.

   > ⚠ **Important**
   >
   > The bash portion of the script must be run as root. If the command isn't run as root, an error is returned.

7. After a few minutes, the instance will register to the gateway. All instances registered to this gateway will appear in the **Instances** tab.

## Create a bridge (console)

After you have registered at least one instance to your gateway, you can create a bridge. The process for creating a bridge will vary depending on the bridge type you select.

**To create an ingress bridge**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. From the navigation pane, select **Gateways**. In the **Gateways** section, select the gateway you want to create the bridge on.
3. From the gateway's **Details** page, select the **Bridges** tab.
4. From the **Bridges** tab, select **Create bridge**.
5. From the **Create bridge** page, complete the following steps in the **Details** section:
   1. Enter a **Name** for the bridge.
   2. Select a **Bridge type** of **Ingress bridge**.
   3. Enter the **Maximum bitrate** for the content you will transport over the bridge.
   4. Enter the **Maximum outputs** for the bridge.
6. Next, complete the following steps in the **Sources** section. The source of an ingress bridge is multicast content that originates at your premises:
   1. Enter a **Name** for the bridge source.
   2. Select a **Network**. This is a network you created during the gateway setup process.
   3. Select the **Protocol** for this source.
   4. Enter the **Multicast IP** and the **Port** of the source.

7. If you add more than one source, you can setup failover using the **Failover configuration** section.

    a. Select the **Failover mode**: **Failover** or **Merge**

    b. Optional - If you select **Failover** as the mode, you may select one of the sources you previously configured to be the **Primary source**. If you don't select a **Primary source**, MediaConnect will select one at random.

8. To complete the bridge creation, choose **Create bridge**.

9. After the bridge is created, you can start the bridge by selecting **Start** on the bridge's **Details** page.

#### To create an Egress bridge

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.

2. From the navigation pane, select **Gateways**. In the **Gateways** section, select the gateway you want to create the bridge on.

3. On the gateway's **Details** page, select the **Bridges** tab. Select **Create bridge**.

4. On the **Create bridge** page, complete the following steps in the **Details** section:

    1. Enter a **Name** for the bridge.
    2. Select a **Bridge type** of **Egress bridge**.
    3. Enter the **Maximum bitrate** for the content you will transport over the bridge.

5. Next, complete the following steps in the **Sources** section:

    1. Enter a **Name** for the bridge source. For an Egress bridge, the source is the content coming from a MediaConnect flow and delivered to your premises.
    2. Select a **Network**. This is a network you created during the gateway setup process.
    3. Select the **Flow ARN**. This is the ARN of the MediaConnect flow that you will use as a source.
    4. If this flow uses a **VPC interface**, select it.

6. If you add more than one source, you can setup failover using the **Failover configuration** section.

    a. When you select an egress bridge, the only available **Failover mode** is **Failover**. **Merge** cannot be selected.

   b. Optional - Select one of the sources that you previously created to be the **Primary source**. If you don't select a **Primary source**, MediaConnect will select one at random.

7. The final section in egress bridge creation is **Outputs**. Complete the following steps.

   1. Enter a **Name** for the bridge output.
   2. Select a **Network**. This is a network that you created during the MediaConnect Gateway setup process.
   3. Select a transport **Protocol** for the output.
   4. Enter an **IP address** for the output. This must be an IP that is compatible with your local network.
   5. Enter the **Port** for the output. This must be a port that is compatible with your local network.
   6. Enter a **TTL** (time-to-live) for the output.

8. Select **Create bridge**.
9. After the bridge is created, you can start the bridge by selecting **Start** on the bridge details page.

## Removing a gateway and its components (console)

To remove a gateway, you must first remove all of its components, such as its networks, instances, and bridges. The following is the process for removing a gateway and its components.

**To remove a gateway**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. From the navigation pane, select **Gateways**. In the **Gateways** section, select the gateway you want to delete.
3. On the MediaConnect Gateway details page, select the **Bridges** tab. Complete the following steps to delete the bridges:

   1. Select the bridge you want to delete.
   2. If the bridge has been started, select **Stop**.
   3. When the bridge is stopped, select **Delete**.
   4. Confirm the deletion of the bridge by selecting **Delete bridge**.
   5. Repeat these steps for any additional bridges you need to delete.

4. Return to the gateway's **Details** page, select the **Instances** tab. Complete the following steps to delete the instances:

    1. Select the instance you want to delete.
    2. Select **Deregister**.
    3. Confirm the deregistration of the instance by selecting **Deregister instance**.
    4. Repeat these steps for any additional instances you need to deregister.

    > **Note**
    >
    > **OPTIONAL**: If you want to reuse the instance for Amazon ECS Anywhere or as another gateway instance, you will need to complete the following steps. If not, continue with Step 5.

    a. Make sure that the **Instance state** is **Deregistered** for the instance you want to reuse.

    b. From a computer with the access to do so, connect to the instance using SSH.

    c. Run the following commands, in order:

```
sudo docker stop $(sudo docker ps -f "name=MediaConnectGatewayAgent" -q); \
sudo docker stop ecs-agent; \
sudo systemctl stop ecs amazon-ssm-agent; \
sudo yum remove -y amazon-ecs-init amazon-ssm-agent;  `# or apt or snap as
 needed` \
sudo rm /var/lib/ecs /etc/ecs /var/lib/amazon/ssm /var/log/ecs /var/log/amazon/
ssm -rf; \
sudo docker rm -f ecs-agent ssm-agent; \
sudo docker container rm -f $(sudo docker ps -a -f
 "name=MediaConnectGatewayAgent" -q); \
sudo docker volume rm -f ecsdata docker run; \
sudo pkill -f -KILL network_bootstra[p]; \
sudo pkill -KILL mcproxy;
```

5. After successfully deleting all bridges and deregistering all instances associated with the gateway, you may delete the gateway. Deleting the gateway will delete all networks created under that gateway.

    1. From the navigation pane, select **Gateways**.
    2. In the **Gateways** section, select the gateway that you want to delete to view that gateway's **Details** page.
    3. Choose the **Delete** button.
    4. Confirm the deletion of the gateway by choosing **Delete gateway**.

# Creating a gateway (AWS CLI)

To create the gateway using the AWS CLI, see the following instructions.

**Topics**

## Create a gateway (AWS CLI)

The gateway is a logical grouping of instances and bridges. Each gateway utilizes user-defined IP information for communication between data centers and the AWS Cloud.

Before creating a gateway using the AWS CLI, you will need the name, egress CIDR IP information, and network information of the gateway you want to create. Store this information in a JSON file on the computer that runs the AWS CLI. The JSON file should be named `gateway.json`. The following example shows the correct sections and formatting for the JSON file.

```
{
    "Name": "gateway",
    "EgressCidrBlocks": [
        "10.20.30.0/24"
    ],
    "Networks": [
        {
            "Name": "blue",
            "CidrBlock": "172.31.48.0/20",
        }
    ]
}
```

**To create a gateway using the AWS CLI**

1. Enter the following command into the AWS CLI interface. Replace the `<yourprofile>` and `<region>` values with your desired profile and AWS Region.

   ```
   aws --profile <yourprofile> --region <region> mediaconnect create-gateway
         --cli-input-json file://gateway.json
   ```

2. The AWS CLI will return a response like the following example.

   ```
       "Gateway": {
           "EgressCidrBlocks": [
               "10.20.30.0/24"
           ],
           "GatewayArn": "arn:aws:mediaconnect:us-
   west-2:111122223333:gateway:1-23aBC45dEF67hiJ8-12AbC34DE5fG:gateway",
           "GatewayState": "CREATING",
           "Name": "gateway",
           "Networks": [
               {
                   "CidrBlock": "172.31.48.0/20",
                   "Name": "blue"
               }
           ]
   ```

```
    }
}
```

3. The MediaConnect Gateway has been created.

## Register an instance (AWS CLI)

Once you have created a gateway, you can register instances to that gateway. An instance is a computing resource running on equipment in your data center and is managed by MediaConnect. This instance is an on-premises implementation of the MediaConnect service and is contained within a gateway. Instances use bridges to communicate between your data center and the AWS Cloud. Instances are created by installing software on an on-premises server.

Registering an instance using the AWS CLI is not currently supported. Follow the console instructions in Register an instance (console) to register the instance using the AWS console.

## Create a bridge (AWS CLI)

After you have registered at least one instance to your gateway component, you can create a bridge. The bridge is the connection between the instances and the AWS Cloud.

Before creating a bridge using the AWS CLI, you will need to collect the details of the bridge you want to create. These details will be stored in a JSON file on the computer running the AWS CLI. The JSON file should be named `bridge.json`. The following example shows the correct sections and formatting for the JSON file.

```
{
    "Name": "bridge",
    "PlacementArn": "arn:aws:mediaconnect:us-west-2:111122223333:gateway:1-23aBC45dEF67hiJ8-12AbC34DE5fG:gateway",
    "EgressGatewayBridge": {
        "MaxBitrate": 100000000
    },
    "SourceFailoverConfig": {
        "FailoverMode": "FAILOVER",
        "State": "ACTIVE"
    },
    "Sources": [
        {
            "FlowSource": {
                "Name": "Source0",
                "FlowArn": "arn:aws:mediaconnect:us-west-2:111122223333:flow:1-UAECXlABCQJeVwMB-95ec11ac6059:gatewayFlow",
                "NetworkName": "blue"
            }
        },
        {
            "FlowSource": {
                "Name": "Source1",
                "FlowArn": "arn:aws:mediaconnect:us-west-2:111122223333:flow:1-ECRZVGADYMGtPGTM-c1iPQ5FNL7Qn:gatewayFlow",
                "NetworkName": "blue",
                "FlowVpcInterfaceAttachment": {
                    "VpcInterfaceName": "VPCIF"
                }
            }
        }
    ],
    "Outputs": [
        {
            "NetworkOutput": {
                "Name": "Output0",
                "NetworkName": "blue",
                "IpAddress": "225.1.2.3",
                "Port": 5010,
                "Protocol": "rtp-fec",
                "Ttl": 8
            }
        },
        {
            "NetworkOutput": {
                "Name": "Output1",
                "NetworkName": "blue",
                "IpAddress": "225.1.2.4",
                "Port": 6010,
                "Protocol": "rtp",
                "Ttl": 250
            }
        }
    ]
}
```

### To create a Bridge using the AWS CLI

1. Enter the following command into the AWS CLI interface. Replace the `<yourprofile>` and `<region>` values with your desired profile and AWS Region.

   ```
   aws --profile <yourprofile> --region <region> mediaconnect create-bridge
         --cli-input-json file://bridge.json
   ```

2. The AWS CLI will return a response like the following example.

```
{
    "Bridge": {
        "BridgeArn": "arn:aws:mediaconnect:us-west-2:111122223333:bridge:1-
GLxlBRLrHzzvpwyb-1dd820
66b207:bridge",
        "BridgeMessages": [],
        "BridgeState": "STANDBY",
        "EgressGatewayBridge": {
            "MaxBitrate": 100000000
        },
        "Name": "bridge",
        "Outputs": [
            {
                "NetworkOutput": {
                    "IpAddress": "225.1.2.3",
                    "Name": "Output0",
                    "NetworkName": "blue",
                    "Port": 5010,
                    "Protocol": "rtp-fec",
                    "Ttl": 8
                }
            },
            {
                "NetworkOutput": {
                    "IpAddress": "225.1.2.4",
                    "Name": "Output1",
                    "NetworkName": "blue",
                    "Port": 6010,
                    "Protocol": "rtp",
                    "Ttl": 250
                }
            }
        ],
        "PlacementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:gateway:1-23aBC45dEF67hiJ8-12AbC34DE5fG:gateway",
        "SourceFailoverConfig": {
            "FailoverMode": "FAILOVER",
            "State": "ENABLED"
        },
        "Sources": [
            {
                "FlowSource": {
                    "FlowArn": "arn:aws:mediaconnect:us-west-2:111122223333:flow:1-
UAECXlABCQJeVwMB-95ec11ac6059:gatewayFlow",
                    "Name": "Source0",
                    "NetworkName": "blue"
                }
            },
            {
                "FlowSource": {
                    "FlowArn": "arn:aws:mediaconnect:us-west-2:111122223333:flow:1-
ECRZVGADYMGtPGTM-c1iPQ5FNL7Qn:gatewayFlow",
                    "Name": "Source1",
                    "NetworkName": "blue",
                    "FlowVpcInterfaceAttachment": {
                        "VpcInterfaceName": "VPCIF"
                    }
                }
            }
        ]
    }
}
```

3. The Bridge has been created.

## Removing a gateway and its components (AWS CLI)

To remove a gateway, you must first remove all of its components, such as its networks, instances, and bridges. The following is the process for removing a gateway and its components by using the AWS Command Line Interface (AWS CLI).

**To remove a gateway using the AWS CLI**

1. Delete the bridges by running the following command.

```
aws --profile <Profile> --region <Region> mediaconnect delete-bridge --bridge-
arn <BridgeArn>
```

2. Deregister the instances by running the following command.

```
aws --profile <Profile> --region <Region> mediaconnect deregister-gateway-instance
 --gateway-instance-arn <GatewayArn>
```

**Note**

**OPTIONAL**: If you want to reuse the instance for Amazon ECS Anywhere or as another AWS Elemental MediaConnect Gateway instance, you will need to complete the following steps. If not, continue with Step 3.

a. Make sure that the `InstanceState` is `DEREGISTERED` for the instance you want to reuse. You can verify using the `describe-gateway-instance` command shown in the following example.

```
aws --profile <Profile> --region <Region> mediaconnect describe-gateway-
instance
      --gateway-instance-arn <GatewayInstanceArn>
```

b. From a computer with the access to do so, connect to the instance using SSH.

c. Run the following commands, in order.

```
sudo docker stop $(sudo docker ps -f "name=MediaConnectGatewayAgent" -q); \
sudo docker stop ecs-agent; \
sudo systemctl stop ecs amazon-ssm-agent; \
sudo yum remove -y amazon-ecs-init amazon-ssm-agent;  `# or apt or snap as
 needed` \
sudo rm /var/lib/ecs /etc/ecs /var/lib/amazon/ssm /var/log/ecs /var/log/amazon/
ssm -rf; \
sudo docker rm -f ecs-agent ssm-agent; \
sudo docker container rm -f $(sudo docker ps -a -f
 "name=MediaConnectGatewayAgent" -q); \
sudo docker volume rm -f ecsdata docker run; \
sudo pkill -f -KILL network_bootstra[p]; \
sudo pkill -KILL mcproxy;
```

3. Delete the gateway. This will delete all networks associated with the gateway.

```
aws --profile <Profile> --region <Region> mediaconnect delete-gateway --gateway-
arn <GatewayArn>
```

# VPC interfaces

A virtual private cloud (VPC) based on the Amazon Virtual Private Cloud service is your private, logically isolated network in the AWS Cloud. You can set up a VPC interface to establish a connection between your AWS Elemental MediaConnect flow and your VPC.

For more information, see the following sections.

- Creating a transport stream flow that uses a VPC source
- Adding a VPC interface to a flow
- Removing a VPC interface from a flow
- Adding a VPC source to an existing flow
- Adding VPC outputs to a flow
- Security group considerations for VPC interfaces

## Adding a VPC interface to a flow

To avoid streaming your content over the public internet, you can add a VPC interface to your MediaConnect flow. You can add up to two VPC interfaces to each flow.

> ⚠ **Important**
>
> Before you begin this procedure, make sure that the following steps have been completed:
>
> - In Amazon VPC, set up your VPC and associated security groups. For more information about VPCs, see the Amazon VPC User Guide. For information about configuring security groups to work with your VPC interface, see Security group considerations.
> - In IAM, set up MediaConnect as a trusted service.

**To add a VPC interface to a flow (console)**

1. On the **Flows** page, choose the name of the flow that you want to update.
2. Choose the **VPC interfaces** tab.
3. Choose **Add VPC interface**.

4. For **Name**, specify a name for your VPC interface. The name of the VPC interface must be unique within the flow.

5. For **Network interface type**, specify the type of network adapter that you want MediaConnect to use on this interface. If you don't set this value, it defaults to **ENA**.

   > **Note**
   >
   > You can add only one EFA VPC interface, and up to two ENA VPC interfaces to a flow.

6. For **Role ARN**, specify the Amazon Resource Name (ARN) of the role that you created when you set up MediaConnect as a trusted service.

7. For **VPC**, choose the ID of the VPC that you want to use.

8. For **Subnet**, choose the VPC subnet that you want MediaConnect to use to set up your VPC configuration. The subnet must reside in the same Availability Zone as the flow.

9. For **Security groups**, specify the VPC security groups that you want MediaConnect to use to set up your VPC configuration. You must choose at least one security group.

# Removing a VPC interface from a flow

You can remove a VPC interface from your flow if it isn't used as a source for the flow. The flow must also be in **Standby**.

> **Note**
>
> If the flow has an error, you must resolve the error before you complete this procedure.

### To remove a VPC interface from a flow (console)

1. On the **Flows** page, choose the name of the flow that is associated with the VPC interface that you want to remove.

2. Choose **Stop**.

   The status of the flow changes to **Standby**. The flow stops immediately and is no longer viewable to customers who are accessing the output directly from your flow or through an entitlement.

3. Choose the **VPC interfaces** tab.
4. Choose the VPC interface that you want to remove, and then choose **Remove**.

# Security group considerations for VPC interfaces

When you set up a virtual private cloud (VPC) in Amazon Virtual Private Cloud, you create security groups that control inbound and outbound traffic. Then, when you create a VPC interface in AWS Elemental MediaConnect, you specify the security groups that you want MediaConnect to use when it sends and receives content from your VPC.

To ensure that content can flow between your VPC and MediaConnect, adhere to the following guidelines:

| Make sure that the VPC interface has a security group with... | Additional information |
|---|---|
| An inbound rule that allows the private IP address of the resource within the VPC that is sending content. | **Zixi sources:** When you create a VPC source using Zixi protocol, the inbound port is automatically assigned by MediaConnect. The assigned port will be in the range of 2090-2099 and assigned at the time of source creation. You should create the Zixi VPC source first and note the assigned port. After you have the assigned port informati on, you can configure your security groups. |
| An outbound rule that allows all outbound traffic. By default, all security groups include this rule. As long as you haven't deleted that rule from the security group, you don't need to create a new one. | On the resource that receives traffic from your flow, you also need to set up a security group with an inbound rule that allows the private IP of the network interface ID that is associated with the VPC interface. (In MediaConnect, you can look at the flow details to find the network interface ID. Then in EC2, you view details about the network interface to obtain the IP address.) |
| An inbound rule and an outbound rule that meet the requirements listed above. | You can use one security group that has both rules or two security groups (one for each rule). |

| Make sure that the VPC interface has a security group with... | Additional information |
|---|---|
| | For CDI flows, the security group specified for the VPC interfaces must be self referential. Verify that the security group used has the same security group ID added to both inbound and outbound rules. |

For more information about security groups, see the Amazon VPC User Guide.

# Media streams in AWS Elemental MediaConnect

A media stream is an essential component in a CDI flow, which you can use to ingest content into and transport content within the AWS Cloud via the SMPTE 2110, part 22 transport standard. Each media stream represents a single track or stream of media that contains video, audio, or ancillary data.

You define a media stream as part of the flow. Then, you can associate it with a source and multiple outputs on that flow. The source and outputs must use the CDI protocol or the ST 2110 JPEG XS protocol, and can consist of one or many media streams.

The type of media stream that you create is based on the output that you are receiving from or sending to an on-premises device, such as AWS Elemental Live.

**Note**

You use media streams only for CDI flows that have ST 2110 with JPEG XS as their input and output protocol. If you have configured your flows to use CDI as the input and output protocol, you don't need media streams.

| AWS Elemental Live output | MediaConnect media stream type |
|---|---|
| SMPTE 2110-20: Uncompressed video | (Not supported) |
| SMPTE 2110-22: Compressed video with JPEG XS | Video |
| SMPTE 2110-30: PCM audio | Audio |
| SMPTE 2110-31: Dolby audio (AC3, EAC3) | (Not supported) |
| SMPTE 2110-40: Ancillary data | Ancillary data |

For illustrations of CDI workflows, see Contribution for CDI flows and CDI replication and monitoring.

**Topics**

- Adding a media stream to a flow
- Updating a media stream
- Removing a media stream

## Adding a media stream to a flow

Before you can associate a media stream with a source or an output, you need to add it to the flow. After you add a media stream to a flow, you can associate it with a source and then with outputs.

> **Note**
>
> You can only associate a media stream with an output if it has already been associated with a source on the flow.

**To add a media stream to a flow**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow that you want to add the media stream to.
3. Choose the **Media streams** tab.
4. Choose **Add media stream**.
5. In the **Name** field, specify a descriptive name that will help you distinguish this media stream from others in the flow.
6. For **Description**, specify a description that will help you remember the use of this media stream.
7. For **Stream ID**, specify a unique identifier for the media stream.

   If the source or any of the outputs uses the CDI protocol, specify the value that is expected by the production and playout systems.

   If the source and all outputs use the ST 2110 JPEG XS protocol, specify a value that is unique to that of other media streams within the flow.
8. Choose **Advanced options** to display the additional options based on your stream type.
9. For specific instructions on the advanced options based on your stream type, choose one of the following tabs:

Audio

1. For **Stream type**, choose **Audio**.
2. For **Media clock rate**, specify the sample rate for the stream. This value is measured in Hz.
3. For **Language**, specify the language of the audio. This value should be in a format that the receiver recognizes.
4. For **Channel order**, specify the format of the audio channel.
5. Choose **Add media stream**.

Video

1. For **Stream type**, choose **Video**.

   For many fields, MediaConnect provides a default value that represents the recommended setting. Change the default value if needed.
2. **Media clock rate** is the sample rate for the stream, and is set to 90000. This value is measured in Hz.
3. For **Video format**, specify the resolution of the video.
4. For **Exact framerate**, specify the frame rate of the video. This value should be represented in frames per second.
5. For **Colorimetry**, specify the format that was used for the representation of color in the video.
6. For **Scan mode**, specify the method that was used to scan the incoming video.
   - Choose **Interlace** if the incoming video is interlaced (for example, 480i or 1080i).
   - Choose **Progressive** if the incoming video is progressive (for example, 720p or 1080p).
   - Choose **Progressive segmented frame** if the incoming video is PSF (for example, 1080psf).
7. For **TCS**, specify the transfer characteristic system (TCS) that was used in the video.
8. For **Range**, specify the encoding range of the video.
9. For **PAR**, specify the pixel access ratio (PAR) of the video.
10. Choose **Add media stream**.

Ancillary data

1. For **Stream type**, choose **Ancillary data**.
2. **Media clock rate** is the sample rate for the stream, and is set to 90000. This value is measured in Hz.
3. Choose **Add media stream**.

# Updating a media stream

You can update media streams even if the flow is running. However, if the media stream is associated with a source or any outputs, you can't update its type.

### To update a media stream on a flow

1. Open the MediaConnect console at [https://console.aws.amazon.com/mediaconnect/](https://console.aws.amazon.com/mediaconnect/).
2. On the **Flows** page, choose the name of the flow that is associated with the media stream that you want to update.
3. Choose the **Media streams** tab.

   A list of media streams for that flow appears.
4. Choose the media stream that you want to update.
5. Choose **Update**.
6. Make the appropriate changes, and then choose **Save**.

# Removing a media stream

You can remove a media stream from a flow if the flow is not active and if the media stream is not associated with a source or any outputs.

### To remove a media stream from a flow

1. Open the MediaConnect console at [https://console.aws.amazon.com/mediaconnect/](https://console.aws.amazon.com/mediaconnect/).
2. On the **Flows** page, choose the name of the flow that is associated with the media stream that you want to remove.

The details page for that flow appears.

3. Choose the **Media streams** tab.
4. Choose the media stream, and then choose **Remove**.

# Reservations for AWS Elemental MediaConnect

Reservations provide you with significant savings on your AWS Elemental MediaConnect costs compared to on-demand pricing.

A *reservation* is a commitment to use a specific amount of outbound bandwidth each month over the course of a specified duration. In return, you pay a discounted hourly rate for that bandwidth. The reservation is allocated and billed on a monthly basis through the duration of the reservation.

The discounted rate applies to outbound bandwidth from all of the MediaConnect flows in your account up to the amount of bandwidth specified in the reservation.

*Outbound bandwidth* refers to data that is transferred from a MediaConnect flow to a location or endpoint outside of the AWS Cloud. It does not include data transferred *in* to your MediaConnect flow, nor does it include data transferred from a MediaConnect flow to any location within the AWS Cloud.

For information on charges for reservations, see the MediaConnect price list.

## How billing works

Reserved outbound bandwidth is billed hourly. For each billing cycle, AWS charges your account for outbound bandwidth at the discounted rate, as specified in your reservation. If your account uses more outbound bandwidth than is covered in the reservation, the overage is charged at on-demand rates. If your account used less bandwidth, AWS charges you for the amount of outbound bandwidth that's specified in the reservation. Unused bandwidth is not carried over to the next month.

## Viewing reservations

On the console, you can view the reservations that you have purchased.

**To view a list of reservations (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. In the navigation pane, choose **Reservations**.

   A list appears, showing all reservations that you have purchased.

# Offerings

*Offerings* are discounts that MediaConnect offers in exchange for a commitment to use a certain amount of outbound bandwidth each month. The components of a MediaConnect offering are:

- Duration
- Outbound bandwidth
- Price (billed hourly)

When you purchase an offering, you specify the start date and time. The resulting resource is called a *reservation* because you are "reserving" a certain amount of outbound bandwidth for a period of time.

*Outbound bandwidth* refers to data that is transferred from a MediaConnect flow to a location or endpoint outside of the AWS Cloud. It does not include data transferred *in* to your MediaConnect flow, nor does it include data transferred from a MediaConnect flow to any location within the AWS Cloud.

## Viewing offerings

On the console, you can view the offerings that are available in the current AWS Region.

**To view a list of offerings (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. In the navigation pane, choose **Offerings**.

   A list appears, showing all offerings that are available in the current Region.

## Purchasing an offering

If your account doesn't already have an active reservation, you can purchase an offering to create a new reservation.

**To purchase an offering (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.

2. In the navigation pane, choose **Offerings**.

   A list appears, showing all offerings that are available in the current Region.

   > **Note**
   >
   > If you have an active reservation, you can't purchase another offering.

3. Choose the reservation that you want to purchase, and choose **Purchase**.

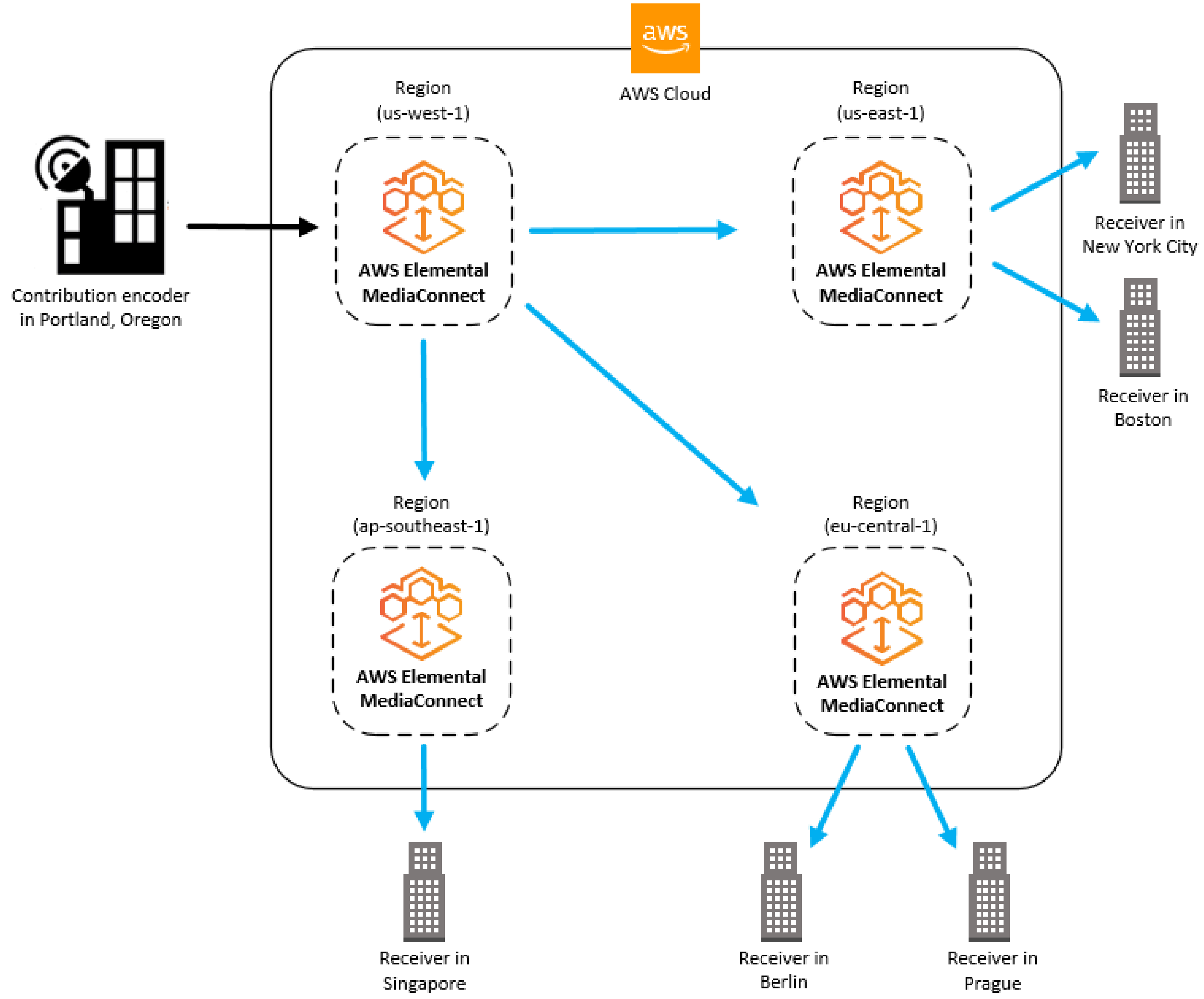
   The **Enter reservation details** page appears.
4. In the **Name** field, enter a name for the reservation. Reservation names must be unique within your account, including expired reservations.
5. For **Start date**, click the calendar icon and choose the date that you want the reservation to begin. You can choose a date as early as the first day of the current month and as recent as today.
6. In the **Start time** field, enter the time of day that you want the reservation to begin. If your start date occurs in the past, you can choose any time of day. If your start date occurs today, you can choose any time up to and including the current time.
7. Choose **Next**.

   The **Review and purchase** page appears.
8. Review the details of the reservation. If you need to make changes to the reservation name or start, choose **Previous** and make the changes. If you need to choose a different offering, choose **Cancel** and start over.
9. Choose **Purchase**.

# Distributing content using AWS Elemental MediaConnect

You can use AWS Elemental MediaConnect to distribute content to different geographical locations. For example, suppose that your source is an on-premises contribution encoder that is located in Portland, Oregon and you want to distribute your content to locations around the world. You set up your initial AWS Elemental MediaConnect flow in the `us-west-1` Region, which is the closest physical AWS Region to your encoder. After your content is in the AWS Cloud, you send it to other MediaConnect flows located in Regions that are closer to your receivers.

The following illustration shows an on-premises contribution encoder located in Portland, Oregon that uploads content to AWS Elemental MediaConnect in the AWS Cloud. The flow has three outputs that send content to others flows in different AWS Regions. These secondary flows are closer to the receivers, which are located in various cities around the world.

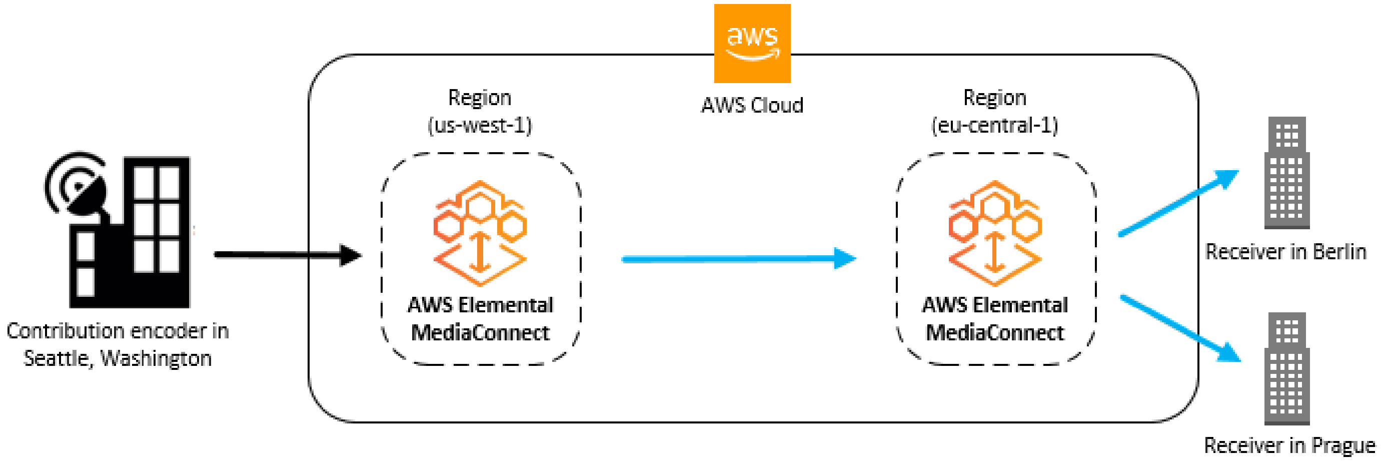


**Topics**

- Distributing content across Regions
- Distributing content to AWS Elemental MediaLive
- Distributing content from an AWS Elemental MediaLive Multiplex

## Distributing content across Regions

You can set up two AWS Elemental MediaConnect flows to distribute content from one AWS Region to another. In this scenario, you create one flow in the Region that is closest to your contribution encoder and a second flow in the Region that is closest to your receiver. The following illustration shows this process.



This topic assumes that you already know how to create a flow and add outputs to a flow.

**To distribute content across Regions (console)**

1. In the AWS Region that is closest to your source, create a flow. (We'll refer to this as flow A.)
2. Review the **Details** page for flow A to determine its egress IP address.
3. In the AWS Region that is closest to your destination, create a second flow (flow B) with the following details:
   - Source type: Choose **Standard source**.
   - Protocol: Choose **Zixi push**.
   - Inbound port: Selecting Zixi push as the protocol will automatically set this port to **`2088`**.
   - Allowlist CIDR block: Enter a CIDR value that includes the egress IP of flow A.
4. Review the **Details** page, **Source** tab for flow B to determine its ingest IP address.
5. In flow A, create an output with the following details:
   - Protocol: Choose **Zixi push**.
   - IP address: Enter the ingest IP address of flow B.
   - Port: Enter **`2088`**.

# Distributing content to AWS Elemental MediaLive

If you plan to distribute the contents of your AWS Elemental MediaConnect flow to AWS Elemental MediaLive, remember the following:

- For each video stream, create two flows in the same AWS Region, and in the same Availability Zones (such as `us-east-1a`). For example, if you are creating two MediaLive inputs using MediaConnect flows, the first flow of input 1 needs to be in the same Availability Zone as the first flow of input 2. These redundant flows will serve as the primary and backup inputs for the MediaLive channel.
- Create the MediaLive channel in the same AWS Region as the AWS Elemental MediaConnect flows.
- Set up permissions that allow MediaLive to communicate with AWS Elemental MediaConnect. This process consists of the following procedures:
  1. Create a policy that allows MediaLive to submit a request to AWS Elemental MediaConnect (see Create a MediaLive Policy).
  2. Assign that policy to a role for MediaLive (see Create a Role for MediaLive). You will need the Amazon Resource Name (ARN) for this role when you specify AWS Elemental MediaConnect flows as inputs to a MediaLive channel.
- Create your AWS Elemental MediaConnect and MediaLive resources in this order:
  1. Set up permissions.
  2. Create the AWS Elemental MediaConnect flows.
  3. Make a note of the flow ARNs.
  4. Create the inputs on the MediaLive channel. (You can create the MediaLive channel whenever you want. Just be sure to create the inputs for that channel after you create the flows.)

# Distributing content from an AWS Elemental MediaLive Multiplex

An AWS Elemental MediaLive multiplex creates a UDP transport stream (TS) that carries multiple programs, also known as a multi-program transport stream (MPTS). When you create a multiplex, MediaLive automatically grants an entitlement in MediaConnect for your account. Create a flow based on that entitlement and distribute the content from that flow.

**To distribute content from a MediaLive multiplex (console)**

1. In MediaLive, create a multiplex.

   MediaLive creates a MediaConnect entitlement that uses the multiplex as the source. The name of the entitlement includes `multiplex` and the name you chose for the multiplex.
2. In MediaConnect, create a flow based on the new entitlement.
3. Add outputs to distribute the content.

# Protocols in AWS Elemental MediaConnect

AWS Elemental MediaConnect supports different protocols for incoming (source) and outgoing (output) live video streams depending on the type of flow you use.

For transport stream flows, which transport compressed content that is muxed (audio, video, and ancillary data are combined) into a single stream, you use the following protocols:

- **Reliable Internet Stream Transport (RIST) (Simple profile only)** is a highly available, low-latency protocol that is suitable for long-distance applications. MediaConnect doesn't support encryption for sources or outputs that use the RIST protocol.
- **Real-Time Transport Protocol (RTP)** has wide applicability and takes less bandwidth than RTP-FEC. MediaConnect doesn't support encryption for sources or outputs that use the RTP protocol.
- **Real-Time Transport Protocol with Forward Error Correction (RTP-FEC)** has wide applicability and forward error correction (FEC) to self-heal any corruption and packet loss. Using this protocol takes more bandwidth than RTP without FEC. AWS Elemental MediaConnect doesn't support encryption for sources or outputs that use the RTP-FEC protocol.
- **Secure Reliable Transport (SRT)** is a highly available, low-latency protocol that is suitable for long-distance applications.
  - **SRT listener** is a pull-based implementation of the SRT protocol. SRT listener can be used as a Source or Output. SRT listener must communicate with an SRT caller.
  - **SRT caller** is a push-based implementation of the SRT protocol. SRT caller can be used as a Source or Output. SRT caller must communicate with an SRT listener.
- **Zixi** is a highly available protocol suitable for most applications, especially use cases that involve longer distances. If your encoder is not capable of using Zixi, you can use the Zixi feeder/receiver software that was created specifically for use with MediaConnect. You can access this software on the Zixi website, where you will be asked to provide your information before you can download the software. If you set up multiple flows for distribution, we recommend that you use Zixi as the protocol to send content between flows. MediaConnect supports two Zixi protocol options:
  - **Zixi pull** uses the Zixi protocol to send content to a receiver or an integrated receiver decoder (IRD) that is behind a firewall. Additionally, you can use this option when you need network address translation (NAT) to route the traffic from MediaConnect to the receiver.

  - **Zixi push** uses the Zixi protocol to send content to a receiver that has a static, publicly addressable IP address. Use this option when the receiver is not behind a firewall or NAT-based router.
  - **Zixi push for AWS Elemental Link** uses the Zixi push protocol to connect an AWS Elemental Link UHD device with a MediaConnect flow.
- **Fujitsu-QoS** is a low-latency, high throughput proprietary protocol from Fujitsu that enables transport from Fujitsu devices into MediaConnect, and from MediaConnect into Fujitsu devices. MediaConnect does not support source failover if you use the Fujitsu protocol.

For CDI flows, which transport high-quality content that has been lightly compressed using JPEG XS, you use the following protocols:

- **AWS Cloud Digital Interface (AWS CDI)** is a technology that allows you to transport high-quality uncompressed video inside the AWS Cloud, with high reliability and network latency as low as 8 milliseconds.
- **ST 2110 JPEG XS** is a low-latency protocol that can be used on streams with minimal compression.

## Protocol support for sources and outputs

The following table describes what protocols can be used for sources, outputs, or both.

**Transport stream protocols**

| Protocol | Can this be used as a Source? | Can this be used as an Output? |
|---|---|---|
| RIST | Yes | Yes |
| RTP | Yes | Yes |
| RTP-FEC | Yes | Yes |
| SRT listener | Yes | Yes |
| SRT caller | Yes | Yes |
| Zixi pull | No | Yes |

| Protocol | Can this be used as a Source? | Can this be used as an Output? |
|---|---|---|
| Zixi push | Yes | Yes |
| Fujitsu-QoS | Yes | Yes |

**CDI protocols**

| Protocol | Can this be used as a Source? | Can this be used as an Output? |
|---|---|---|
| CDI | Yes | Yes |
| ST 2110 JPEG XS | Yes | Yes |

# Color support for CDI protocols

MediaConnect CDI flows support multiple configurations of color space, bit depth, and chroma sampling for each protocol. The following table describes the configurations supported by each CDI protocol.

> **Note**
>
> MediaLive does not currently support RGB color space for CDI inputs. If you will be ouputting a CDI flow from MediaConnect to MediaLive, ensure that you use YCbCr color space.

**CDI color support**

| Protocol | Supported color configurations |
|---|---|
| CDI | • YCbCr 10-bit 4:2:2<br>• RGB 10-bit 4:4:4<br>• RGB 12-bit 4:4:4 |
| ST 2110 JPEG XS | • YCbCr 10-bit 4:2:2 |

| Protocol | Supported color configurations |
|---|---|
| | • RGB 10-bit 4:4:4<br>• RGB 12-bit 4:4:4 |

# Security in AWS Elemental MediaConnect

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS Elemental MediaConnect, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS Elemental MediaConnect. The following topics show you how to configure AWS Elemental MediaConnect to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS Elemental MediaConnect resources.

**Topics**

# Data protection for AWS Elemental MediaConnect

You can protect your data using tools that are provided by AWS. AWS Elemental MediaConnect can decrypt your incoming video (source) and encrypt your outgoing video (outputs and entitlements).

You have three options for encrypting content in transit:

- **Static key encryption:** You can use this option to encrypt sources, outputs, and entitlements. You store your encryption key in AWS Secrets Manager, and then you give MediaConnect permission to obtain the encryption key from Secrets Manager.

  Advantages: You have full control over storage of the encryption key for your account. The key is stored in AWS Secrets Manager, where you can access it any time.

  Challenges: All parties (the owners of the source, the flow, any outputs, and any entitlements) need the encryption key. If the content is shared using an entitlement, both the originator and the subscriber must store the encryption key in AWS Secrets Manager. If the encryption key changes, you must notify all parties of the new key.
- **Secure Packager and Encoder Key Exchange (SPEKE):** You can use this option to encrypt content that is sent through an entitlement. You partner with a conditional access (CA) platform key provider who manages and provides encryption keys. Then you give Amazon API Gateway permission to act as a proxy between the CA platform key provider and your AWS account.

  Advantages: The content originator has full control over access to the encryption key. As the content originator, you partner with your CA platform key provider who manages the encryption key, but you don't handle the key itself and you don't share it with any other parties. Depending on the capabilities of your key provider, this option allows you to assign time limitations to an encryption key or revoke the key entirely. The subscriber doesn't need to set up encryption. This information is automatically provided through the entitlement.

  Challenges: You must work with a third party (the key provider).
- **Secure Reliable Transport (SRT) password encryption:** You can use this option to encrypt sources and outputs when using SRT protocols. SRT protocols are highly available, low-latency protocols that are suitable for long-distance applications. You store your encryption password in AWS Secrets Manager, and then you give MediaConnect permission to obtain the encryption password from Secrets Manager.

Advantages: Uses 128/256 bit AES for encryption and decryption. SRT protocols use error correction to minimize packet loss. You have full control over storage of the encryption password. The password is stored in AWS Secrets Manager, where you can access it any time.

Challenges: Only usable with SRT protocols. MediaConnect does not support source failover if you use an SRT protocol.

> **Note**
>
> Encryption is supported only for entitlements, for sources that use the Zixi or SRT protocols, and for outputs that use the Zixi or SRT protocols.

**Topics**

- Static key encryption in AWS Elemental MediaConnect
- SPEKE encryption in AWS Elemental MediaConnect
- SRT password encryption in AWS Elemental MediaConnect
- Internetwork traffic privacy

## Static key encryption in AWS Elemental MediaConnect

You can use static key encryption to protect your sources, outputs, and entitlements. You store your encryption key in AWS Secrets Manager, and then you give MediaConnect permission to obtain the encryption key from Secrets Manager.

**Topics**

- Key management for static key encryption
- Setting up static key encryption using AWS Elemental MediaConnect

### Key management for static key encryption

In AWS Elemental MediaConnect, you can use static key encryption to secure content in sources, outputs, and entitlements. To use this method, you store an encryption key as a *secret* in AWS Secrets Manager, and you give AWS Elemental MediaConnect permission to access the secret.

Secrets Manager keeps your encryption key secure, allowing it be accessed only by entities that you specify in an AWS Identity and Access Management (IAM) policy.

With static key encryption, all participants (the owner of the source, the flow, and any outputs or entitlements) need the encryption key. If the content is shared using an entitlement, both AWS account owners must store the encryption key in AWS Secrets Manager.

For more information, see Setting up static key encryption.

## Setting up static key encryption using AWS Elemental MediaConnect

Before you can create a flow with an encrypted source or an output or entitlement that uses static key encryption, you must perform the following steps:

**Step 1** – Store your encryption key as a secret in AWS Secrets Manager.

**Step 2** – Create an IAM policy that allows AWS Elemental MediaConnect to read the secret that you stored in AWS Secrets Manager.

**Step 3** – Create an IAM role and attach the policy that you created in step 2. Next, set up AWS Elemental MediaConnect as a trusted entity that is allowed to assume this role and make requests on behalf of your account.

> **Note**
>
> MediaConnect supports encryption only for entitlements, and for sources and outputs that use the Zixi and SRT protocols. Your stored key in Secrets Manager for the Zixi protocol is a static key in a hexadecimal format. SRT uses a passkey for encryption.

### Step 1: Store your encryption key in AWS Secrets Manager

To use static key encryption to encrypt your AWS Elemental MediaConnect content, you must use AWS Secrets Manager to create a secret that stores the encryption key. You must create the secret, and the resource (source, output, or entitlement) that uses the secret in the same AWS account. You can’t share secrets across accounts.

**Note**

If you use two flows to distribute video from one AWS Region to another, you must create two secrets (one secret in each Region).

**To store an encryption key in Secrets Manager**

1. Obtain the encryption key from the entity that manages the source.
2. Sign in to the AWS Secrets Manager console at https://console.aws.amazon.com/secretsmanager/.
3. On the **Store a new secret** page, for **Select secret type**, choose **Other type of secrets**.
4. For **Key/value pairs**, choose **Plaintext**.
5. Clear any text in the box and replace it with only the **value** of the encryption key. For hexadecimal keys, check the length of the key to ensure that it matches the length specified for the encryption type. For example, an AES-256 encryption key must have 64 digits, because each digit is 4 bits in size.
6. For **Select the encryption key**, keep the default set to **DefaultEncryptionKey**.
7. Choose **Next**.
8. For **Secret name**, specify a name for your secret that will help you identify it later. For example, **`2018-12-01_baseball-game-source`**.
9. Choose **Next**.
10. For **Configure automatic rotation** section, choose **Disable automatic rotation**.
11. Choose **Next**, and then choose **Store**.

    The details page for your new secret appears, showing information such as the secret ARN.
12. Make a note of the secret ARN from Secrets Manager. You will need this information in the next procedure.

### Step 2: Create an IAM policy to allow AWS Elemental MediaConnect to access your secret

In step 1, you created a secret and stored it in AWS Secrets Manager. In this step, you create an IAM policy that allows AWS Elemental MediaConnect to read the secret that you stored.

### To create an IAM policy that allows MediaConnect to access your secret

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane of the IAM console, choose **Policies**.
3. Choose **Create policy**, and then choose the **JSON** tab.
4. Enter a policy that uses the following format:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetResourcePolicy",
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:ListSecretVersionIds"
      ],
      "Resource": [
        "arn:aws:secretsmanager:us-west-2:111122223333:secret:aes256-7g8H9i"
      ]
    }
  ]
}
```

   In the `Resource` section, each line represents the ARN of a different secret that you created. For more examples, see IAM policy examples for secrets in AWS Secrets Manager.

5. Choose **Review policy**.
6. For **Name**, enter a name for your policy such as **`SecretsManagerForMediaConnect`**.
7. Choose **Create policy**.

### Step 3: Create an IAM role with a trusted relationship

In step 2, you created an IAM policy that allows read access to the secret that you stored in AWS Secrets Manager. In this step, you create an IAM role and assign the policy to that role. Then you define AWS Elemental MediaConnect as a trusted entity that can assume the role. This allows MediaConnect to have read access to your secret.

**To create a role with a trusted relationship**

1. In the navigation pane of the IAM console, choose **Roles**.
2. On the **Role** page, choose **Create role**.
3. On the **Create role** page, for **Select type of trusted entity**, choose **AWS service** (the default).
4. For **Choose the service that will use this role**, choose **EC2**.

   You choose EC2 because AWS Elemental MediaConnect is not currently included in this list. Choosing EC2 lets you create a role. In a later step, you change this role to include MediaConnect instead of EC2.
5. Choose **Next: Permissions**.
6. For **Attach permissions policies**, enter the name of the policy that you created in step 2, such as **`SecretsManagerForMediaConnect`**.
7. For **SecretsManagerReadWrite**, select the check box, and then choose **Next: Review**.
8. For **Role name**, enter a name. We highly recommend that you don't use the name `MediaConnectAccessRole` because it is reserved. Instead, use a name that includes `MediaConnect` and describes this role's purpose, such as **`MediaConnect-ASM`**.
9. For **Role description**, replace the default text with a description that will help you remember the purpose of this role. For example, **`Allows MediaConnect to view secrets stored in AWS Secrets Manager.`**
10. Choose **Create role**.
11. In the confirmation message that appears across the top of your page, choose the name of the role that you just created.
12. Choose **Trust relationships**, and then choose **Edit trust policy**.
13. in the **Edit trust policy** window, make the following changes to the JSON:

    - For **Service**, change `ec2.amazonaws.com` to `mediaconnect.amazonaws.com`
    - For added security, define specific conditions for the trust policy. This will limit MediaConnect to only using resources in your account. You do this by using a global condition such as the **Account ID**, the **flow ARN**, or both. See the following example of the conditional trust policy. For more information about the security benefits of the global conditions, see Cross-service confused deputy prevention.

> **Note**
>
> The following example uses both the **Account ID** and **flow ARN** conditions. Your policy will look different if you do not use both conditions. If you don't know the full ARN of the flow or if you are specifying multiple flows, use the `aws:SourceArn` global context condition key with wildcard characters (*) for the unknown portions of the ARN. For example, `arn:aws:mediaconnect:*:111122223333:*`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "mediaconnect.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:mediaconnect:us-
west-2:111122223333:flow:*:flow-name"
                }
            }
        }
    ]
}
```

14. Choose **Update Trust Policy**.
15. On the **Summary** page, make a note of the value for **Role ARN**. It looks like this: `arn:aws:iam::111122223333:role/MediaConnectASM`.

## SPEKE encryption in AWS Elemental MediaConnect

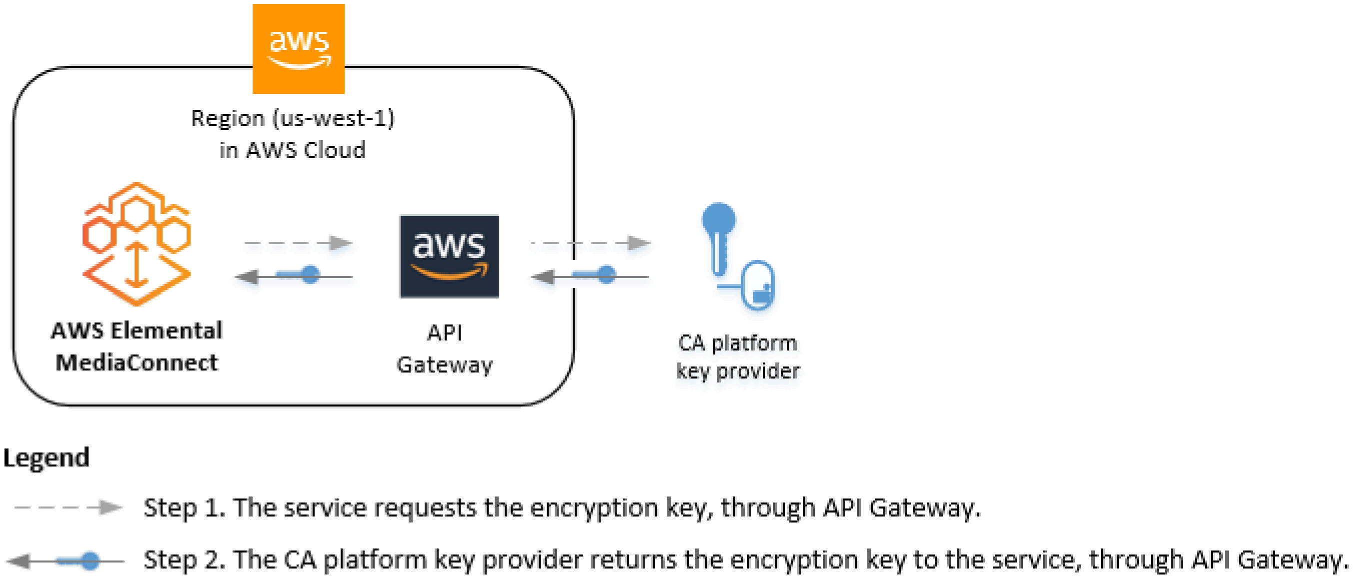
You can use Secure Packager and Encoder Key Exchange (SPEKE) with AWS Elemental MediaConnect to encrypt an entitlement. This gives you, as the content originator, full control of permissions for this content. This usage is a customization of the SPEKE cloud-based architecture described in the SPEKE documentation.

**Topics**

## Key management for SPEKE

With a SPEKE implementation, a conditional access (CA) system provides keys to AWS Elemental MediaConnect for content encryption and decryption. API Gateway acts as a proxy for the communication between the service and the CA platform key provider. Each AWS Elemental MediaConnect flow must reside in the same AWS Region as its API Gateway proxy.

The following illustration shows how AWS Elemental MediaConnect obtains the encryption or decryption key using SPEKE. In the originator's flow, the service obtains the encryption key and uses it to encrypt the content before sending it through the entitlement. In the subscriber's flow, the service obtains the decryption key when the content is received from the entitlement.


aws
Region (us-west-1)
in AWS Cloud
AWS Elemental
MediaConnect
aws
API
Gateway
CA platform
key provider
Legend
Step 1. The service requests the encryption key, through API Gateway.
Step 2. The CA platform key provider returns the encryption key to the service, through API Gateway.

These are the main services and components:

- **AWS Elemental MediaConnect** – Provides and controls the encryption setup for the flow. AWS Elemental MediaConnect obtains the encryption keys from the CA platform key provider through

Amazon API Gateway. Using the encryption keys, AWS Elemental MediaConnect encrypts the content (for the originator's flow) or decrypts the content (for the subscriber's flow).

- **API Gateway** – Manages customer-trusted roles and proxy communication between the encryptor and the key provider. API Gateway provides logging capabilities and lets customers control their relationships with the encryptor and with the CA platform. The API Gateway must reside in the same AWS Region as the encryptor.
- **CA platform key provider** – Provides encryption and decryption keys to AWS Elemental MediaConnect through a SPEKE-compliant API.

For more information, see Setting up SPEKE encryption.

## Setting up SPEKE encryption using AWS Elemental MediaConnect

Before you can grant an entitlement that uses SPEKE encryption, you must perform the following steps:

**Step 1.** – Get on board with a conditional access (CA) platform key provider who will manage your encryption key. During this process, you create an API in Amazon API Gateway that sends requests on behalf of AWS Elemental MediaConnect to the key provider.

**Step 2** – Create an IAM policy that allows the API that you created in step 1 to act as a proxy to make requests to the key provider.

**Step 3.** – Create an IAM role and attach the policy that you created in step 2. Next, set up AWS Elemental MediaConnect as a trusted entity that is allowed to assume this role and access the API Gateway endpoint on your behalf.

### Step 1: Get on board with a CA provider

To use SPEKE with AWS Elemental MediaConnect, you must have a CA platform key provider. The following AWS partners provide conditional access (CA) solutions for the MediaConnect customization of SPEKE:

- Verimatrix

If you are a content originator, contact your CA platform key provider for assistance with the onboarding process. With the help of your CA platform key provider, you manage who gets access to which content.

During the onboarding process, make a note of the following:

- **ARN of the POST method request** – The Amazon Resource Name (ARN) that AWS assigns to the request that you create in API Gateway.
- **Constant initialization vector (optional)** – A 128-bit, 16-byte hex value represented by a 32-character string, to be used with the key for encrypting content.
- **Device ID** – A unique identifier for each device that you configure with the key provider. Each device represents a different recipient for your content.
- **Resource ID** – A unique identifier that you create for each piece of content that you configure with the key provider.
- **URL** – The URL assigned by AWS for the API that you create in Amazon API Gateway.

You need these values later, when you configure the entitlement in MediaConnect.

### Step 2: Create an IAM policy to allow API Gateway to act as your proxy

In step 1, you worked with a CA platform key provider who manages your encryption key. In this step, you create an IAM policy that allows API Gateway to make requests on your behalf. API Gateway acts as a proxy for communication between your account and the key provider.

#### To create an IAM policy for an API Gateway proxy

1. In the navigation pane of the IAM console, choose **Policies**.
2. Choose **Create policy**, and then choose the **JSON** tab.
3. Enter a policy that uses the following format:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "execute-api:Invoke"
      ],
      "Resource": [
        "arn:aws:execute-api:us-west-2:111122223333:1abcdefghi/*/POST/*"
      ]
    }
  ]
}
```

In the `Resource` section, replace the sample Amazon Resource Name (ARN) with the ARN of the `POST` method request that you created in API Gateway with the CA platform key provider.

4. Choose **Review policy**.
5. For **Name**, enter **`APIGateway-Proxy-Access`**.
6. Choose **Create policy**.

### Step 3: Create an IAM role with a trusted relationship

In step 2, you created an **APIGateway-Proxy-Access** policy that allows API Gateway to act as a proxy and make requests on your behalf. In this step, you create an IAM role and attach the following permissions:

- The **APIGateway-Proxy-Access** policy allows Amazon API Gateway to act as a proxy on your behalf so that it can make requests between your account and the CA platform key provider. This is the policy you created in step 1.
- A **trust relationship** policy allows AWS Elemental MediaConnect to assume the role on your behalf. You will create this policy as part of the following procedure.

#### To create an IAM role with a trusted relationship

1. In the navigation pane of the IAM console, choose **Roles**.
2. On the **Role** page, choose **Create role**.
3. On the **Create role** page, for **Select type of trusted entity**, choose **AWS service** (the default).
4. For **Choose the service that will use this role**, choose **EC2**.

   You choose EC2 because AWS Elemental MediaConnect is not currently included in this list. Choosing EC2 lets you create a role. In a later step, you change this role to include MediaConnect instead of EC2.
5. Choose **Next: Permissions**.
6. For **Filter policies**, choose **Customer managed**.
7. Select the check box next to **APIGateway-Proxy-Access**, and then choose **Next: Tags**.
8. Enter tag values (optional), and then choose **Next: Review**.
9. For **Role name**, enter a name such as **`SpekeAccess`**.

10. For **Role description**, replace the default text with a description that will help you remember the purpose of this role. For example, **`Allows AWS Elemental MediaConnect to talk to API Gateway on my behalf.`**
11. Choose **Create role**.
12. In the confirmation message that appears across the top of your page, choose the name of the role that you just created.
13. Choose **Trust relationships**, and then choose **Edit Trust Relationship**.
14. For **Policy Document**, change the policy to look like this:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "mediaconnect.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

15. Choose **Update Trust Policy**.
16. On the **Summary** page, make a note of the value for **Role ARN**. It looks like this: `arn:aws:iam::111122223333:role/SpekeAccess`.

## SRT password encryption in AWS Elemental MediaConnect

You can use the Secure Reliable Transport (SRT) password encryption option to encrypt sources and outputs when using the SRT protocols. SRT protocols are a highly available, low-latency protocol suitable for long-distance applications. You store your encryption password in AWS Secrets Manager, and then you give MediaConnect permission to obtain the encryption password from Secrets Manager.

**Topics**

## Password management for SRT password encryption

In AWS Elemental MediaConnect, you can use SRT password encryption to secure content in sources and outputs. To use this method, you store an SRT password as a *secret* in AWS Secrets Manager, and you give AWS Elemental MediaConnect permission to access the secret. Secrets Manager keeps your password secure, allowing it be accessed only by entities that you specify in an AWS Identity and Access Management (IAM) policy.

With SRT password encryption, all participants (the owner of the source, the flow, and any outputs) need the SRT password.

For more information, see Setting up SRT password encryption.

## Setting up SRT password encryption using AWS Elemental MediaConnect

Before you can create a flow with an encrypted source or an output that uses SRT password encryption, you must perform the following steps:

**Step 1** – Store your SRT password as a secret in AWS Secrets Manager.

**Step 2** – Create an IAM policy that allows AWS Elemental MediaConnect to read the secret that you stored in AWS Secrets Manager.

**Step 3** – Create an IAM role and attach the policy that you created in step 2. Next, set up AWS Elemental MediaConnect as a trusted entity that is allowed to assume this role and make requests on behalf of your account.

### Step 1: Store your encryption password in AWS Secrets Manager

To use SRT password encryption to encrypt your AWS Elemental MediaConnect content, you must use AWS Secrets Manager to create a secret that stores the password. You must create the secret, and the resource (source or output) that uses the secret in the same AWS account. You can’t share secrets across accounts.

> **Note**
>
> If you use two flows to distribute video from one AWS Region to another, you must create two secrets (one secret in each Region).

If you are creating a new SRT password to encrypt an output, we recommend the following password policy:

- Minimum password length of 10 characters and a maximum length of 80 characters
- Minimum of three of the following mix of character types: uppercase, lowercase, numbers, and `! @ # $ % ^ & * ( ) _ + - = [ ] { } | '` symbols
- Not be identical to your AWS account name or email address

**To store a password in Secrets Manager**

1. Sign in to the AWS Secrets Manager console at https://console.aws.amazon.com/secretsmanager/.
2. On the **Store a new secret** page, for **Select secret type**, choose **Other type of secrets**.
3. For **Key/value pairs**, choose **Plaintext**.
4. Clear any text in the box and replace it with only the **value** of the SRT password.
5. For **Encryption key**, keep the default set to **aws/secretsmanager**.
6. Choose **Next**.
7. For **Secret name**, specify a name for your secret that will help you identify it later. For example, **`2018-12-01_baseball-game-source`**.
8. Choose **Next**.
9. For the **Configure automatic rotation** section, leave **Automatic rotation** off.
10. Choose **Next**, and then choose **Store**. On the next screen, select the name of the secret you created.

    The details page for your new secret appears, showing information such as the secret ARN.
11. Make a note of the secret ARN from Secrets Manager. You will need this information in the next procedure.

### Step 2: Create an IAM policy to allow AWS Elemental MediaConnect to access your secret

In step 1, you created a secret and stored it in AWS Secrets Manager. In this step, you create an IAM policy that allows AWS Elemental MediaConnect to read the secret that you stored.

### To create an IAM policy that allows MediaConnect to access your secret

1. Open the IAM console at [https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. In the navigation pane of the IAM console, choose **Policies**.
3. Choose **Create policy**, and then choose the **JSON** tab.
4. Enter a policy that uses the following format:

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": [
           "secretsmanager:GetResourcePolicy",
           "secretsmanager:GetSecretValue",
           "secretsmanager:DescribeSecret",
           "secretsmanager:ListSecretVersionIds"
         ],
         "Resource": [
           "arn:aws:secretsmanager:us-west-2:111122223333:secret:aes256-7g8H9i"
         ]
       }
     ]
   }
   ```

   In the `Resource` section, each line represents the ARN of a different secret that you created. Enter the secret ARN from the previous procedure. Choose **Next: Tags**.
5. Choose **Next: Review**.
6. For **Name**, enter a name for your policy such as **`SecretsManagerForMediaConnect`**.
7. Choose **Create policy**.

### Step 3: Create an IAM role with a trusted relationship

In step 2, you created an IAM policy that allows read access to the secret that you stored in AWS Secrets Manager. In this step, you create an IAM role and assign the policy to that role. Then you define AWS Elemental MediaConnect as a trusted entity that can assume the role. This allows MediaConnect to have read access to your secret.

### To create a role with a trusted relationship

1. In the navigation pane of the IAM console, choose **Roles**.
2. On the **Role** page, choose **Create role**.
3. On the **Create role** page, for **Select type of trusted entity**, choose **AWS service** (the default).
4. For **Choose the service that will use this role**, choose **EC2**.

   You choose EC2 because AWS Elemental MediaConnect is not currently included in this list. Choosing EC2 lets you create a role. In a later step, you change this role to include MediaConnect instead of EC2.
5. Choose **Next: Permissions**.
6. For **Attach permissions policies**, enter the name of the policy that you created in step 2, such as **`SecretsManagerForMediaConnect`**.
7. For **SecretsManagerForMediaConnect**, select the check box, and then choose **Next**.
8. For **Role name**, enter a name. We highly recommend that you don't use the name `MediaConnectAccessRole` because it is reserved. Instead, use a name that includes `MediaConnect` and describes this role's purpose, such as **`MediaConnect-ASM`**.
9. For **Role description**, replace the default text with a description that will help you remember the purpose of this role. For example, **`Allows MediaConnect to view secrets stored in AWS Secrets Manager.`**
10. Choose **Create role**.
11. In the confirmation message that appears across the top of your page, choose the name of the role that you just created.
12. Choose **Trust relationships**, and then choose **Edit trust policy**.
13. For **Edit trust policy**, change `ec2.amazonaws.com` to `mediaconnect.amazonaws.com`.

    The policy document should now look like this:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "mediaconnect.amazonaws.com"
      },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

14. Choose **Update policy**.
15. On the **Summary** page, make a note of the value for **Role ARN**. It looks like this: `arn:aws:iam::111122223333:role/MediaConnectASM`.

## Internetwork traffic privacy

**To route traffic directly between MediaConnect and your corporate network via a virtual private cloud (VPC)**

1. Set up a private connection between your Amazon VPC and your corporate network. Set up either an IPsec VPN connection over the internet or a private physical connection using AWS Direct Connect connection. AWS Direct Connect enables you to establish a private virtual interface from your on-premises network directly to your Amazon VPC, providing you with a private, high-bandwidth network connection between your network and your VPC. With multiple virtual interfaces, you can establish private connectivity to multiple VPCs while maintaining network isolation. For more information, see What is AWS Site-to-Site VPN? and What is AWS Direct Connect?
2. Create a flow that uses a VPC *source*. During this process, you add a VPC *interface* to your flow to establish the initial connection between your VPC and your flow. You also specify that same VPC interface as the source for the new flow.

> **Note**
>
> If your flow already exists, you can update the flow to add a VPC interface and then add another source that uses that VPC interface.

# Identity and access management for AWS Elemental MediaConnect

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use MediaConnect resources. IAM is an AWS service that you can use with no additional charge.

## Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in MediaConnect.

**Service user –** If you use the MediaConnect service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more MediaConnect features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in MediaConnect, see Troubleshooting AWS Elemental MediaConnect identity and access.

**Service administrator –** If you're in charge of MediaConnect resources at your company, you probably have full access to MediaConnect. It's your job to determine which MediaConnect features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with MediaConnect, see How AWS Elemental MediaConnect works with IAM.

**IAM administrator –** If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to MediaConnect. To view example MediaConnect identity-based policies that you can use in IAM, see AWS Elemental MediaConnect identity-based policy examples.

## Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see Signing AWS API requests in the *IAM User Guide*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center User Guide* and Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

## AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

## IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the *IAM User Guide*.

## IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an

  action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

  - **Service role –** A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role –** A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2 –** You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

## Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

## Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing between managed policies and inline policies in the *IAM User Guide*.

## Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to

any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the *AWS Organizations User Guide*.

- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

### Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

## Learn more

For more information about identity and access management for MediaConnect, continue to the following pages:

- How MediaConnect works with IAM
- Identity-based policy examples
- Resource-based policy examples
- Policy examples for secrets in AWS Secrets Manager
- Troubleshooting

## How AWS Elemental MediaConnect works with IAM

Before you use IAM to manage access to MediaConnect, you should understand what IAM features are available to use with MediaConnect. To get a high-level view of how MediaConnect and other AWS services work with IAM, see AWS Services That Work with IAM in the *IAM User Guide*.

**Topics**

- Authorization based on MediaConnect tags
- MediaConnect IAM roles

## MediaConnect identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. MediaConnect supports specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see IAM JSON Policy Elements Reference in the *IAM User Guide*.

### Actions

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

Policy actions in MediaConnect use the following prefix before the action: `mediaconnect:`. For example, to grant someone permission to view a list of entitlements with the MediaConnect `ListEntitlements` API operation, you include the `mediaconnect:ListEntitlements` action in their policy. Policy statements must include either an `Action` or `NotAction` element. MediaConnect defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": [
      "mediaconnect:action1",
      "mediaconnect:action2"
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `List`, include the following action:

```
"Action": "mediaconnect:List*"
```

To see a list of MediaConnect actions, see Actions Defined by AWS Elemental MediaConnect in the *IAM User Guide*.

**Resources**

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

MediaConnect has the following ARNs:

```
arn:${Partition}:mediaconnect:${Region}:${Account}:entitlement:${resourceID}:
${resourceName}
arn:${Partition}:mediaconnect:${Region}:${Account}:flow:${resourceID}:${resourceName}
arn:${Partition}:mediaconnect:${Region}:${Account}:output:${resourceID}:${resourceName}
arn:${Partition}:mediaconnect:${Region}:${Account}:source:${resourceID}:${resourceName}
```

For more information about the format of ARNs, see Amazon Resource Names (ARNs) and AWS Service Namespaces.

For example, to specify the `1-23aBC45dEF67hiJ8-12AbC34DE5fG` flow in your statement, use the following ARN:

```
"Resource": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame"
```

To specify all flows that belong to a specific account, use the wildcard (*):

```
"Resource": "arn:aws:mediaconnect:us-east-1:111122223333:flow:*"
```

Some MediaConnect actions, such as those for creating resources, can't be performed on a specific resource. In those cases, you must use the wildcard (*).

```
"Resource": "*"
```

Many MediaConnect API actions involve multiple resources. For example, `RemoveFlowOutput` removes an output from a particular flow, so an IAM user must have permissions for the flow and the output. To specify multiple resources in a single statement, separate the ARNs with commas.

```
"Resource": [
      "resource1",
      "resource2"
```

To see a list of MediaConnect resource types and their ARNs, see Resources Defined by AWS Elemental MediaConnect in the *IAM User Guide*. To learn with which actions you can specify the ARN of each resource, see Actions Defined by AWS Elemental MediaConnect.

**Condition keys**

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

### Examples

To view examples of MediaConnect identity-based policies, see AWS Elemental MediaConnect identity-based policy examples.

## MediaConnect resource-based policies

AWS Elemental MediaConnect does not support resource-based policies.

## Authorization based on MediaConnect tags

AWS Elemental MediaConnect does not support tagging resources or controlling access based on tags.

## MediaConnect IAM roles

An IAM role is an entity within your AWS account that has specific permissions.

### Using temporary credentials with MediaConnect

You can use temporary credentials to sign in with federation, assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as AssumeRole or GetFederationToken.

MediaConnect supports using temporary credentials.

### Service-linked roles

Service-linked roles allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

MediaConnect does not support service-linked roles.

### Service roles

This feature allows a service to assume a service role on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service roles appear in your IAM account and are owned by the account. This means that an IAM administrator can change the permissions for this role. However, doing so might break the functionality of the service.

MediaConnect does not support service roles.

## AWS Elemental MediaConnect identity-based policy examples

By default, IAM users and roles don't have permission to create or modify MediaConnect resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see Creating Policies on the JSON Tab in the *IAM User Guide*.

### Policy best practices

Identity-based policies determine whether someone can create, access, or delete MediaConnect resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides

more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see IAM Access Analyzer policy validation in the *IAM User Guide*.

- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Configuring MFA-protected API access in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

## Using the MediaConnect console

To access the AWS Elemental MediaConnect console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the MediaConnect resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (IAM users or roles) with that policy.

To ensure that those entities can still use the MediaConnect console, also attach the following AWS managed policy to the entities. For more information, see Adding Permissions to a User in the *IAM User Guide*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "mediaconnect:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcs",
    "ec2:CreateNetworkInterface",
    "ec2:CreateNetworkInterfacePermission",
    "ec2:DeleteNetworkInterface",
    "ec2:DeleteNetworkInterfacePermission"
   ],
   "Effect": "Allow",
   "Resource": "*"
  },
  {
   "Action": [
    "cloudwatch:GetMetricData"
   ],
   "Effect": "Allow",
   "Resource": "*"
  },
  {
      "Action": [
          "iam:PassRole"
      ],
      "Effect": "Allow",
     "Resource": "*",
      "Condition": {
          "StringLike": {
              "iam:PassedToService": "mediaconnect.amazonaws.com"
       }
     }
   }
 ]
}
```

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that you're trying to perform.

## Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
  "Version": "2012-10-17",
  "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

## AWS Elemental MediaConnect resource-based policy examples

To access the AWS Elemental MediaConnect console, you must have a minimum set of permissions that allows you to list and view details about the MediaConnect resources in your AWS account. The IAM policies in this section show examples of policies that allow specific actions on resources in AWS Elemental MediaConnect.

### Allow read access to all resources in AWS Elemental MediaConnect

To access the AWS Elemental MediaConnect console, you must have a policy that defines which actions you are allowed to take on MediaConnect resources in your AWS account. The IAM policy below provides the following permissions:

- The section for the `mediaconnect:List*` and `mediaconnect:Describe*` actions allow read-only access to all resources that you create in AWS Elemental MediaConnect.
- The section for the `ec2:DescribeAvailabilityZones` action allows the service to obtain information about which Availability Zone the flow is in. This portion of the policy is required.
- The section for the `cloudwatch:GetMetricData` action allows the service to obtain metrics from Amazon CloudWatch. This portion of the policy is required.
- The section for the `iam:PassRole` action allows IAM to *pass* a role to AWS Elemental MediaConnect the service to communicate with IAM in order to assume a role on behalf of the service. This allows the service to assume the role later and perform actions on your behalf. This portion of the policy is required.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "mediaconnect:List*",
                "mediaconnect:Describe*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "ec2:DescribeAvailabilityZones"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
         {
            "Action": [
                "cloudwatch:GetMetricData"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
           "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "mediaconnect.amazonaws.com"
                }
            }
        }
    ]
}
```

## Allow all actions on all AWS Elemental MediaConnect resources

Every user of AWS Elemental MediaConnect must have a policy that defines permissions on AWS Elemental MediaConnect resources. The IAM policy below provides the following permissions:

- The section for the `mediaconnect:*` action allows all actions on all resources that you create in AWS Elemental MediaConnect.
- The section for the `ec2:DescribeAvailabilityZones` action allows the service to obtain information about which Availability Zone the flow is in. This portion of the policy is required.
- The section for the `cloudwatch:GetMetricData` action allows the service to obtain metrics from Amazon CloudWatch. This portion of the policy is required.
- The section for the `iam:PassRole` action allows IAM to *pass* a role to AWS Elemental MediaConnect the service to communicate with IAM in order to assume a role on behalf of the service. This allows the service to assume the role later and perform actions on your behalf. This portion of the policy is required.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "mediaconnect:*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "ec2:DescribeAvailabilityZones"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
         {
            "Action": [
                "cloudwatch:GetMetricData"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
           "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "mediaconnect.amazonaws.com"
                }
            }
        }
    ]
}
```

## Allow AWS Elemental MediaConnect to create and manage network interfaces in your VPC

This example IAM policy allows AWS Elemental MediaConnect to create and manage network interfaces in your VPC so that content can flow from your VPC to MediaConnect. If you want to connect your VPC to your flow, you must set up this policy.

- The section for the `ec2:` actions allows MediaConnect to create, read, update, and delete network interfaces in your VPC. This portion of the policy is required.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:describeNetworkInterfaces",
                "ec2:describeSecurityGroups",
                "ec2:describeSubnets",
                "ec2:createNetworkInterface",
                "ec2:createNetworkInterfacePermission",
                "ec2:deleteNetworkInterface",
                "ec2:deleteNetworkInterfacePermission"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

# IAM policy examples for secrets in AWS Secrets Manager

During setup, you create an IAM policy that you assign to AWS Elemental MediaConnect. This policy allows MediaConnect to read secrets that you have stored in AWS Secrets Manager. The settings for this policy are entirely up to you. The policy can range from most restrictive (allowing access to only specific secrets) to least restrictive (allowing access to any secret that you create using this AWS account). We recommend using the most restrictive policy as a best practice. However, the examples in this section show you how to set up policies with different levels of restriction. Because MediaConnect needs only read access to secrets, all the examples in this section show only the actions necessary to read the values that you store.

**Topics**

## Allow read access to specific secrets in AWS Secrets Manager

The following IAM policy allows read access to specific resources (secrets) that you create in AWS Secrets Manager.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-west-2:111122223333:secret:aes128-1a2b3c",
                "arn:aws:secretsmanager:us-west-2:111122223333:secret:aes192-4D5e6F",
                "arn:aws:secretsmanager:us-west-2:111122223333:secret:aes256-7g8H9i"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "secretsmanager:ListSecrets",
            "Resource": "*"
        }
    ]
}
```

## Allow read access to all secrets created in a specific Region in AWS Secrets Manager

The following IAM policy allows read access to all secrets that you create in a specific AWS Region in AWS Secrets Manager. This policy applies to resources that you have created already and all resources that you create in the future in the specified Region.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": "arn:aws:secretsmanager:us-west-2:111122223333:secret:*"
        },
        {
            "Effect": "Allow",
            "Action": "secretsmanager:ListSecrets",
            "Resource": "*"
        }
    ]
}
```

### Allow read access to all resources in AWS Secrets Manager

The following IAM policy allows read access to all resources that you create in AWS Secrets Manager. This policy applies to resources that you have created already and all resources that you create in the future.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds",
                "secretsmanager:ListSecrets"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

## AWS managed policies for AWS Elemental MediaConnect

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

## AWS managed policy: MediaConnectGatewayInstanceRolePolicy

You can attach the `MediaConnectGatewayInstanceRolePolicy` policy to your IAM identities.

This policy grants permission to register MediaConnect Gateway Instances to a MediaConnect Gateway. This policy will be attached to a role. The entity assuming the role will have the ability to register instances to the gateway.

### Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "MediaConnectGateway",
   "Effect": "Allow",
   "Action": [
    "mediaconnect:DiscoverGatewayPollEndpoint",
    "mediaconnect:PollGateway",
    "mediaconnect:SubmitGatewayStateChange"
   ],
   "Resource": "*"
  }
 ]
}
```

## AWS managed policy: AWSMediaConnectServicePolicy

You can’t attach AWSMediaConnectServicePolicy to your IAM entities. This policy is attached to a service-linked role that allows MediaConnect to perform actions on your behalf. For more information, visit Using service-linked roles.

This policy is attached to the **AWSServiceRoleForMediaConnect** service-linked role. This policy allows the service-linked role to manage Amazon ECS resources on your behalf. AWS Elemental MediaConnect Gateway uses Amazon ECS as the foundation for the on-premises implementation of AWS Elemental MediaConnect Gateway and MediaConnect must have the ability to create, update, and delete Amazon ECS resources as needed.

### Permissions details

This policy includes the following permissions.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "ecs:UpdateService",
    "ecs:DeleteService",
    "ecs:CreateService",
    "ecs:DescribeServices",
    "ecs:PutAttributes",
    "ecs:DeleteAttributes",
    "ecs:RunTask",
    "ecs:ListTasks",
    "ecs:StartTask",
    "ecs:StopTask",
    "ecs:DescribeTasks",
    "ecs:DescribeContainerInstances",
    "ecs:UpdateContainerInstancesState"
   ],
   "Resource": "*",
   "Condition": {
    "ArnLike": {
     "ecs:cluster": "arn:aws:ecs:*:*:cluster/MediaConnectGateway"
    }
   }
  },
  {
```

## MediaConnect updates to AWS managed policies

View details about updates to AWS managed policies for MediaConnect since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the MediaConnect document history page.

| Change | Description | Date |
|---|---|---|
| The MediaConnect managed policy **MediaConnectGatewayInstanceRolePolicy** has been added. | This policy grants permission to register MediaConnect Gateway Instances to a MediaConnect Gateway. | APRIL 12, 2023 |
| The MediaConnect managed policy **AWSMediaConnectServicePolicy** has been added. | This policy is used by a service-link role and grants permissions to access AWS services and resources used by MediaConnect. | APRIL 12, 2023 |
| MediaConnect started tracking changes | MediaConnect started tracking changes for its AWS managed policies. | APRIL 12, 2023 |

# Using service-linked roles for MediaConnect

AWS Elemental MediaConnect uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to MediaConnect. Service-linked roles are predefined by MediaConnect and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up MediaConnect easier because you don’t have to manually add the necessary permissions. MediaConnect defines the permissions of its service-linked roles, and unless defined otherwise, only MediaConnect can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your MediaConnect resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-linked roles** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

## Service-linked role permissions for MediaConnect

MediaConnect uses the service-linked role named **AWSServiceRoleForMediaConnect** – The default Service-Linked Role that enables access to AWS Services and Resources used or managed by MediaConnect..

The AWSServiceRoleForMediaConnect service-linked role trusts the following services to assume the role:

- `MediaConnect`

The role permissions policy named MediaConnectServiceRolePolicy allows MediaConnect to complete the following actions on the specified resources:

- Action: `ecs:CreateCluster, ecs:RegisterTaskDefinition, ecs:DescribeTaskDefinition, ecs:ListAttributes, ecs:UpdateContainerInstancesState, ecs:DeregisterContainerInstance` on Resource `arn:aws:ecs:*:*:*`

- Action: `ecs:UpdateCluster, ecs:UpdateClusterSettings, ecs:DescribeClusters` on Resource`arn:aws:ecs:*:*:cluster/MediaConnect`
- Action: `ecs:CreateService, ecs:UpdateService, ecs:RunTask, ecs:StartTask, ecs:StopTask, ecs:ExecuteCommand, ecs:PutAttributes, ecs:DeleteAttributes, ecs:DescribeServices, ecs:DescribeTasks, ecs:ListTasks` on Resource `arn:aws:ecs:*:*:*` with the Condition of `StringLike: {ecs:Cluster: arn:aws:ecs:*:*:cluster/MediaConnect}`

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*.

## Creating a service-linked role for MediaConnect

You don't need to manually create a service-linked role. When you create an associated MediaConnect resource in the AWS Management Console, the AWS CLI, or the AWS API, MediaConnect creates the service-linked role for you.

> ⚠ **Important**
>
> This service-linked role can appear in your account if you completed an action in another service that uses the features supported by this role. Also, if you were using the MediaConnect service before January 1, 2023, when it began supporting service-linked roles, then MediaConnect created the AWSServiceRoleForMediaConnect role in your account. To learn more, see A new role appeared in my IAM account.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you create an associated MediaConnect resource, MediaConnect creates the service-linked role for you again.

You can also use the IAM console to create a service-linked role with the **MediaConnect** use case. In the AWS CLI or the AWS API, create a service-linked role with the `MediaConnect` service name. For more information, see Creating a service-linked role in the *IAM User Guide*. If you delete this service-linked role, you can use this same process to create the role again.

## Editing a service-linked role for MediaConnect

MediaConnect does not allow you to edit the AWSServiceRoleForMediaConnect service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a service-linked role in the *IAM User Guide*.

## Deleting a service-linked role for MediaConnect

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

> **Note**
>
> If the MediaConnect service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

**To delete MediaConnect resources used by the AWSServiceRoleForMediaConnect**

1. Delete all Bridges in all Gateways.
2. De-register all Instances in all Gateways.
3. Delete all Gateways.

**To manually delete the service-linked role using IAM**

Use the IAM console, the AWS CLI, or the AWS API to delete the AWSServiceRoleForMediaConnect service-linked role. For more information, see Deleting a service-linked role in the *IAM User Guide*.

## Supported regions for MediaConnect service-linked roles

MediaConnect supports using service-linked roles in all of the regions where the service is available. For more information, see MediaConnect regions and endpoints.

# Setting up AWS Elemental MediaConnect as a trusted service

You can use AWS Identity and Access Management (IAM) to control which AWS resources can be accessed by which users and applications. This includes setting up permissions to allow AWS

Elemental MediaConnect to communicate with other services on behalf of your account. To set up AWS Elemental MediaConnect as a trusted entity, you must perform the following steps:

**Step 1.** – Create an IAM policy that governs which actions you want to allow.

**Step 2** – Create an IAM role with a trusted relationship, and attach the policy that you created in the previous step.

## Step 1: Create an IAM policy to allow specific actions

In this step, you create an IAM policy that governs which actions you want to allow.

**To create the IAM policy**

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Policies**.
3. Choose **Create policy**, and then choose the **JSON** tab.
4. Enter a policy that uses the JSON format. For examples, see the following:
   - Policy example for connecting to your VPC
   - Policy examples for secrets in AWS Secrets Manager
5. Choose **Review policy**.
6. For **Name**, enter a name for your policy.
7. Choose **Create policy**.

## Step 2: Create an IAM role with a trusted relationship

In step 1, you created an IAM policy that governs which actions you want to allow. In this step, you create an IAM role and assign the policy to that role. Then you define AWS Elemental MediaConnect as a trusted entity that can assume the role.

**To create a role with a trusted relationship**

1. In the navigation pane of the IAM console, choose **Roles**.
2. On the **Role** page, choose **Create role**.
3. On the **Create role** page, for **Select type of trusted entity**, choose **AWS service** (the default).
4. For **Choose the service that will use this role**, choose **EC2**.

You choose EC2 because MediaConnect is not currently included in this list. Choosing EC2 lets you create a role. In a later step, you change this role to include MediaConnect instead of EC2.

5. Choose **Next: Permissions**.
6. For **Attach permissions policies**, enter the name of the policy that you created in step 1.
7. Select the check box next to the name of the policy, and then choose **Next: Tags**.
8. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM Entities in the *IAM User Guide*.
9. Choose **Next: Review**.
10. For **Role name**, enter a name. The name `MediaConnectAccessRole` is reserved, so you can't use it. Instead, use a name that includes `MediaConnect` and describes this role's purpose.
11. For **Role description**, replace the default text with a description that will help you remember the purpose of this role.
12. Choose **Create role**.
13. In the confirmation message that appears across the top of your page, choose the name of the role that you just created by selecting **View role**.
14. Choose **Trust relationships** tab, and then choose **Edit trust policy**.
15. in the **Edit trust policy** window, make the following changes to the JSON:

    - For **Service**, change `ec2.amazonaws.com` to `mediaconnect.amazonaws.com`
    - For added security, define specific conditions for the trust policy. This will limit MediaConnect to only using resources in your account. You do this by using a global condition such as the **Account ID**, the **flow ARN**, or both. See the following example of the conditional trust policy. For more information about the security benefits of the global conditions, see Cross-service confused deputy prevention.

    > **Note**
    >
    > The following example uses both the **Account ID** and **flow ARN** conditions. Your policy will look different if you do not use both conditions. If you don't know the full ARN of the flow or if you are specifying multiple flows, use the `aws:SourceArn` global context condition key with wildcard characters (*) for the unknown portions of the ARN. For example, `arn:aws:mediaconnect:*:`*`111122223333`*`:*`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "mediaconnect.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:mediaconnect:us-
west-2:111122223333:flow:*:flow-name"
                }
            }
        }
    ]
}
```

16. Choose **Update policy**.
17. On the **Summary** page, make a note of the value for **Role ARN**. It looks like this: `arn:aws:iam::111122223333:role/MediaConnectASM`.

## Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the *calling service*) calls another service (the *called service*). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the `aws:SourceArn` of the flow and `aws:SourceAccount` global condition context keys in resource policies to limit the permissions that AWS Elemental MediaConnect gives another service to the resource. Use the flow's aws:SourceArn if you want only one resource to be associated with the cross-service access. Use aws:SourceAccount if you want to allow any resource in that account to be associated with the cross-service use.

The most effective way to protect against the confused deputy problem is to use the aws:SourceArn global condition context key with the full ARN of the flow. If you don't know the full ARN of the flow or if you are specifying multiple flows, use the aws:SourceArn global context condition key with wildcard characters (*) for the unknown portions of the ARN. For example, arn:aws:mediaconnect:*:*111122223333*:*.

The following example shows how you can use the aws:SourceArn and aws:SourceAccount global condition context keys in MediaConnect to prevent the confused deputy problem.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "mediaconnect.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:mediaconnect:us-
west-2:111122223333:flow:1-ABCDEFGHJxyzMNoP-a1234bc12345:flow-name"
                }
            }
        }
    ]
}
```

## Troubleshooting AWS Elemental MediaConnect identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with MediaConnect and IAM.

### Topics

- I am not authorized to perform an action in MediaConnect
- I want to allow people outside of my AWS account to access my MediaConnect resources

## I am not authorized to perform an action in MediaConnect

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password.

The following example error occurs when the `mateojackson` user tries to use the console to view details about a flow but does not have `mediaconnect:DescribeFlow` permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 mediaconnect:DescribeFlow on resource: myExampleFlow
```

In this case, Mateo asks his administrator to update his policies to allow him to access the `myExampleFlow` resource using the `mediaconnect:DescribeFlow` action.

## I want to allow people outside of my AWS account to access my MediaConnect resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether MediaConnect supports these features, see How AWS Elemental MediaConnect works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

# Logging and monitoring

This section provides an overview of the options for logging and monitoring in AWS Elemental MediaConnect for security purposes. For more information about logging and monitoring in MediaConnect see *Monitoring and tagging*.

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Elemental MediaConnect and your AWS solutions. You should collect monitoring data from all of the parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your MediaConnect resources and responding to potential incidents:

## Amazon CloudWatch alarms

Using CloudWatch alarms, you watch a single metric over a time period that you specify. If the metric exceeds a given threshold, a notification is sent to an Amazon SNS topic or AWS Auto Scaling policy. CloudWatch alarms do not invoke actions because they are in a particular state. Rather, the state must have changed and been maintained for a specified number of periods. For more information, see Monitoring with CloudWatch metrics.

## AWS CloudTrail logs

CloudTrail provides a record of actions taken by a user, role, or an AWS service in AWS Elemental MediaConnect. Using the information collected by CloudTrail, you can determine the request that was made to MediaConnect, the IP address from which the request was made, who made the request, when it was made, and additional details. For more information, see Logging API calls with AWS CloudTrail.

## AWS Trusted Advisor

Trusted Advisor draws upon best practices learned from serving hundreds of thousands of AWS customers. Trusted Advisor inspects your AWS environment and then makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps. All AWS customers have access to five Trusted Advisor checks. Customers with a Business or Enterprise support plan can view all Trusted Advisor checks.

For more information, see AWS Trusted Advisor.

# Compliance validation for AWS Elemental MediaConnect

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- Architecting for HIPAA Security and Compliance on Amazon Web Services – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

> **Note**
>
> Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.

- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your

compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.

- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

# Resilience in AWS Elemental MediaConnect

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

# Infrastructure security in AWS Elemental MediaConnect

As a managed service, AWS Elemental MediaConnect is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well‐Architected Framework*.

You use AWS published API calls to access MediaConnect through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

# MediaConnect interface VPC endpoints (AWS PrivateLink)

You can use an interface VPC endpoint to keep all MediaConnect API request traffic between your VPC and MediaConnect in the Amazon network, thus improving the security of your VPC. Interface VPC endpoints don't need an internet gateway, a NAT device, or a virtual private gateway. The VPC endpoints are powered by AWS PrivateLink, a technology that you can use to privately access MediaConnect APIs with private IP addresses.

For more information about AWS PrivateLink and VPC endpoints, see VPC endpoints in the *Amazon VPC User Guide*.

## Considerations for MediaConnect VPC endpoints

Before you set up an interface endpoint for MediaConnect, be sure to review Interface endpoint properties and limitations in the *Amazon VPC User Guide*, and be aware of the following considerations:

- VPC endpoints currently don't support cross-Region requests. Ensure that you create your endpoint in the same Region where you plan to interact with MediaConnect.
- VPC endpoints only support Amazon-provided DNS through Amazon Route 53. If you want to use your own DNS, you can use conditional DNS forwarding. For more information, see DHCP Options Sets in the *Amazon VPC User Guide*.
- The security group attached to the VPC endpoint must allow incoming connections on port 443 from the private subnet of the VPC.

## Creating the VPC Endpoints for MediaConnect

You can create an interface endpoint for MediaConnect using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). Follow the procedure outlined in Creating an interface endpoint in the *Amazon VPC User Guide*.

## Controlling Access to VPC Endpoints for MediaConnect

You can control access to MediaConnect by attaching an endpoint policy to your VPC endpoint. The policy specifies the following information:

- The principal that can perform actions.
- The actions that can be performed.

- The resources on which actions can be performed.

For more information, see Controlling access to services with VPC endpoints in the *Amazon VPC User Guide*.

### Example: VPC endpoint policy for actions

The following is an example of an endpoint policy for MediaConnect. When attached to an endpoint, this policy grants access to the listed MediaConnect actions for all principals on all resources.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "mediaconnect:action-1",
            "mediaconnect:action-2",
            "mediaconnect:action-3"
         ],
         "Resource":"*"
      }
   ]
}
```

# Monitoring and tagging in AWS Elemental MediaConnect

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Elemental MediaConnect and your other AWS solutions. AWS provides the following monitoring tools to watch MediaConnect, report when something is wrong, and take automatic actions when appropriate:

- *MediaConnect flow source monitoring* displays detailed information about a source stream and its program media. You can view status messages about the stream as well as details about the program video, audio, and other data. For more information, see the Monitoring source streams section of this guide.
- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.
- *Amazon CloudWatch Events* delivers a near real-time stream of system events that describe changes in AWS resources. CloudWatch Events enables automated event-driven computing, as you can write rules that watch for certain events and trigger automated actions in other AWS services when these events happen. For more information, see the Amazon CloudWatch Events User Guide.
- *Amazon CloudWatch* monitors your AWS resources and the applications that you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track the number of dropped and unrecovered packets on your AWS Elemental MediaConnect flows and automatically notify you when those values exceed a certain number. For more information, see the Amazon CloudWatch User Guide.

# Monitoring AWS Elemental MediaConnect source metadata

MediaConnect source metadata monitoring displays information about the transport stream and its program media. You can view status messages about the flow's source as well as details about the program's video, audio, and other data. Source metadata monitoring can be used with the MediaConnect console, API, AWS CLI, or SDK. For more information about the API, see: `DescribeFlowSourceMetadata` in the *MediaConnect API Reference.*

**Note**

If you are using more than one source for your flow, source metadata is only displayed for the source currently used by the flow.

## Source metadata details

The following sections provide details about the type of information displayed by source metadata monitoring.

### Alerts and messages

The **active alerts** section of the **Source metadata** console tab and the **messages** section of the `DescribeFlowSourceMetadata` API/CLI response can contain status messages with more information about the transport stream. If MediaConnect detects an issue or cannot retrieve the source stream metadata, an associated status message will be displayed.

### Programs

The **programs** section contains information about the individual programs contained in the transport stream. This section contains the following fields:

| Field | Details |
|---|---|
| Program number | The program number of this program. |
| Program PID | The program Packet Identifier (PID). |
| PCR PID | The Program Clock Reference (PCR) PID of this program. |
| Program name | The name of this program. |
| Streams | The nested sections contain info about the video, audio, and data stream types. |

## Streams

The **streams** section is nested within each individual transport stream program. This section contains the following fields:

| Field | Details |
|---|---|
| Stream type | The type of content that this stream contains. This value can be video, audio, data, or unknown. |
| Codec | The codec of the stream. This value will vary depending on the type of stream. For example, a video stream type might display a H264 value while an audio stream type displays AAC. |
| PID | The Packet Identifier (PID) of the stream. |

## Sample source metadata API/CLI response

The following is a sample response from the `DescribeFlowSourceMetadata` API/CLI. In this example, there are two programs. Each program has one video stream and two audio streams. The first program has a SCTE-35 data stream. The second program has an invalid stream type on PID 139. The invalid stream type is indicated by the status code in the messages section and the Unknown stream type in the program section.

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame",
    "TransportMediaInfo": {
        "Messages": [
            {
                "StatusCode": "InvalidType",
                "Message": "We could not determine the type of pid 139 on program 2"
            }
        ],
        "Programs": [
          {
                "ProgramNumber": 1,
                "ProgramPid": 16,
                "PcrPid": 56,
                "ProgramName": "Basketball HD",
                "Streams": [
                    {
                        "StreamType": "Video",
                        "Codec": "H264",
                        "Pid": 126
                    },
                    {
                        "StreamType": "Audio",
                        "Codec": "AAC",
                        "Pid": 127
                    },
                    {
                        "StreamType": "Audio",
                        "Codec": "AAC",
                        "Pid": 128
                    },
                    {
                        "StreamType": "Data",
                        "Codec": "SCTE35",
                        "Pid": 129
                    }
                ]
          },
          {
                "ProgramNumber": 2,
                "ProgramPid": 26,
                "PcrPid": 66,
                "ProgramName": "Basketball SD",
                "Streams": [
                    {
                        "StreamType": "Video",
                        "Codec": "H264",
                        "Pid": 136
                    },
                    {
                        "StreamType": "Audio",
                        "Codec": "AAC",
                        "Pid": 137

                    },
                    {
                        "StreamType": "Audio",
                        "Codec": "AAC",
                        "Pid": 138

                    },
                    {
                        "StreamType": "Unknown",
                        "Codec": "Unknown",
                        "Pid": 139
                    }
                ]
          }
       ]
    },
}
```

## Using source metadata monitoring (console)

You can retrieve the latest source metadata from MediaConnect by using the console.

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. From the **Flows** screen, select the flow you want to inspect.
3. Select the **Source metadata** tab.
4. The **source metadata** tab contains an expandable list of every active alert, program, and stream for the selected flow's source.

## Using source metadata monitoring (AWS CLI)

You can retrieve the latest source metadata from MediaConnect by using the AWS CLI. The following example shows the AWS CLI command and return value for a typical scenario.

1. In the AWS CLI, use the `describe-flow-source-metadata` command with the `--flow-arn` option of the flow you want to inspect.

   ```
   aws mediaconnect describe-flow-source-metadata --flow-arn arn:aws:mediaconnect:us-east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
   ```

2. The return value will contain the media information for the selected flow's source. The following is a generic example of the format of the return value. In this example, there are no messages to display.

    ```
    {
        "FlowArn": "arn:aws:mediaconnect:us-
    east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
        "Messages": [],
        "Timestamp": "2023-12-06T19:57:54Z",
        "TransportMediaInfo": {
            "Programs": [
                {
                    "PcrPid": 1000,
                    "ProgramNumber": 1,
                    "ProgramPid": 2000,
                    "ProgramName": "AwardsShow HD",
                    "Streams": [
                        {
                            "StreamType": "Video",
                            "Codec": "H264",
                            "Pid": 256
                        },
                        {
                            "StreamType": "Audio",
                            "Codec": "AAC",
                            "Pid": 257
                        },
                        {
                            "StreamType": "Data",
                            "Codec": "SCTE35",
                            "Pid": 258
                        }
                    ]
                }
            ]
        }
    }
    ```

# Monitoring AWS Elemental MediaConnect with Amazon CloudWatch metrics

You can monitor AWS Elemental MediaConnect using CloudWatch, which collects raw data and processes it into readable, near real-time metrics. These metrics are kept for 15 months, so that you can access historical information and gain a better perspective on how your web application or service is performing. Most MediaConnect metrics can be accessed in periods as short as one second. You can also set alarms that watch for certain thresholds, and send notifications or take actions when those thresholds are met. For more information, see the Amazon CloudWatch User Guide.

You can view CloudWatch metrics for your flows directly on the MediaConnect console. On the console, you can view these metrics in periods as short as one second or as long as 30 minutes.

> ⓘ **Note**
>
> MediaConnect Gateway metrics are not available in high resolution periods (one second). You must select a period of at least one minute.

## Definition of a metric

AWS Elemental MediaConnect collects data that is the basis for metrics. It collects these *datapoints* every second and sends them immediately to Amazon CloudWatch. You can use CloudWatch to generate *metrics* for these datapoints.

A metric is a collection of datapoints that has had an aggregation (a *statistic*) applied and that has a *period* and a *time range*. For example, you can request the Dropped packets metric as an average (the statistic) for a 1 minute period over 10 minutes (the time range). This result of this request is 10 metrics (because the range divided by the period is 10).

### Period

Most MediaConnect metrics have a *high resolution period*, which means that the minimum period is one second. MediaConnect Gateway metrics are the only metrics not available in a high resolution period.

### Time range

Each period has a *maximum time range*. For example, if you specify 1 day as the time range, you won't be able to retrieve metrics with a 10 second period.

| Period | Maximum time range |
|---|---|
| 1 second | The last 3 hours |
| 5 seconds | |
| 10 seconds | |
| 30 seconds | |
| 60 seconds | The last 360 hours (15 days) |
| 300 seconds (5 minutes) | The last 1512 hours (63 days) |
| 900 seconds (15 minutes) | |
| 3600 seconds (1 hour) or longer | The last 455 days (15 months) |

Periods don't have a *minimum time range*. But there is a point where the statistic you apply becomes meaningless if you have a low period. For example, assume that you set the period to one second. This means that CloudWatch retrieves one datapoint. You can't obtain an average, a minimum or a maximum on one datapoint. However, this doesn't mean that the metric is meaningless. Instead, the metric is for the raw datapoint, with no statistic.

### Maximum storage time

Metrics are available for the last 15 months. Make sure that you specify a period that allows the time range that you want.

## Viewing metrics

You can view some metrics in the MediaConnect console. You can view all metrics in the CloudWatch console. You can also retrieve metrics using the CLI, the REST API, or any AWS SDK.

On the CloudWatch console, the minimum refresh rate for metrics is 30 seconds.

### To view metrics on the MediaConnect console

You can view some metrics in the MediaConnect console. You can view the current metrics, going back from 1 hour to 1 week. (To view other metrics or to view historical metrics, you must use the CloudWatch console.)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. In the navigation pane, choose **Flows**. On the **Flows** page, choose the flow you want. The **Details** page appears.
3. Choose the **Health** tab. The metrics that MediaConnect supports on this tab appears.
4. Choose the period and time range. For example, **Past 1 day (5 min period)**.

### To view metrics using the CloudWatch console

On the CloudWatch console you can view all MediaConnect metrics for any range of time — the current metrics or historical metrics. There is a charge to view metrics on the CloudWatch console.

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**, then choose **All metrics**. In the bottom half of the page, the **Browse** tab shows cards with names.

   No cards appear if you are completely new to AWS, and you haven't performed an action that creates metrics in any service.
3. Select the card that is named **AWS/MediaConnect**.

   This card appears only if you have started at least one flow in the last 15 months in the AWS Region that is currently selected for CloudWatch. This card won't appear if have never started a MediaConnect flow. In this case, come back to this procedure after you have created and started a flow.

   (A card named **MediaConnect** might appear in the custom namespace section of the page. This card is for the old namespace for MediaConnect metrics. The two namespaces became duplicates of each other in September of 2022, so there is no advantage to choosing this card. Always choose **AWS/MediaConnect**.)
4. The **Browse** tab in the bottom half of the page now shows dimensions. Choose a metric dimension. For example, choose **Flow ARN**.

The **Browse** tab now shows a table with one column that shows the chosen dimension (for example, Flow ARN) and one column that shows all the metrics. You can sort the table.

5. Select one or more rows. As soon as you select a row, it appears in the graph in the top half of the page.
6. In the bottom half of the page, choose the **Graphed metrics** tab.
7. On the choices on the right of the tab, specify the **Statistic** and the **Period**.

   When you choose the period, the graph refreshes to show the maximum time range for that period. If the graph is now empty on the left, you can adjust the timeline in the choices at the top right of the graph. Choose a lower number so that the full space is filled up. For example, change **1w** to **1d**.

**To view metrics using the AWS CLI**

- At a command prompt, use the following command:

```
aws cloudwatch list-metrics --namespace "AWS/MediaConnect"
```

## AWS Elemental MediaConnect metrics to monitor flow health

AWS Elemental MediaConnect sends metrics to CloudWatch. You can review specific metrics to evaluate the health of your flow. If the flow is unhealthy, these metrics can help you determine where the issue originates. For details about each metric, see the tables in this section.

For information about source metrics, see Metrics to monitor source health.

> **Note**
>
> Metrics tracked by MediaConnect adhere to the standard as defined by the TR 101 290 spec.

**Topics**

- TR 101 290 Priority 2 metrics
- Maintenance metrics

## Flow metrics

The following table lists network metrics that AWS Elemental MediaConnect sends to CloudWatch.

| Metric | Description |
|---|---|
| `ARQRecovered` | The number of dropped packets that were recovered by automatic repeat request (ARQ). This metric doesn't apply to flows that receive content from an entitlement or to flows that have multiple sources. For flows that have multiple sources, use the SourceARQRecovered metric to view data for each source.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `ARQRequests` | The number of retransmitted packets that were requested through automatic repeat request (ARQ) and received. This metric doesn't apply to flows that receive content from an entitlement or to flows that have multiple sources. For flows that have multiple sources, use the SourceARQRequests metric to view data for each source.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `BitRate` | The bitrate of the incoming (source) video. |

| Metric | Description |
|---|---|
| | Units: bits per second (b/s)<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `Connected` | The status of the source. A value of 1 indicates that the source is connected and a value of 0 (zero) indicates that the source is disconnected. This metric applies only to sources that use the Zixi, SRT, Fujitsu, or RIST protocols.<br>Units: None<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `Disconnections` | The number of times that the source status changed from connected to disconnected.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `DroppedPackets` | The number of packets that were lost during transit. This value is measured before any error correction takes place.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `FECPackets` | The number of packets that were transmitted using forward error correction (FEC) and received. This metric applies only to flows that have one source that uses the RTP-FEC, Zixi, or Fujitsu protocols. It doesn't apply to flows that receive content from an entitlement or to flows that have multiple sources. For flows that have multiple sources, use the SourceFECPackets metric to view data for each source.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `FECRecovered` | The number of packets that were transmitted using forward error correction (FEC), lost during transit, and recovered. This metric applies only to flows that have one source that uses the RTP-FEC, Zixi, or Fujitsu protocols. It doesn't apply to flows that receive content from an entitlement or to flows that have multiple sources. For flows that have multiple sources, use the SourceFECRecovered metric to view data for each source.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `MergeActive` | The merge status of all sources on the flow. A value of 1 indicates that all sources are merged. A value of 0 (zero) indicates that at least one source is not actively merged with 2022-7.<br>Units: None<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `MergeLatency` | The maximum value for SourceMergeLatency.<br>Units: Milliseconds<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `NotRecoveredPackets` | The number of packets that were lost during transit and were not recovered by error correction.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `OverflowPackets` | The number of packets that were lost in transit because the video required more buffer than was available. This metric doesn't apply to flows that receive content from an entitlement or to flows that have multiple sources.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `PacketLossPercent` | The percentage of packets that were lost during transit, even if they were recovered.<br>Units: Percent<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `RecoveredPackets` | The number of packets that were lost during transit, but recovered.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `RoundTripTime` | The amount of time it takes for the source to send a signal and receive an acknowledgment from AWS Elemental MediaConnect. This metric doesn't apply to flows that receive content from an entitlement or to flows that have multiple sources. For flows that have multiple sources, use the SourceRoundTripTime metric to view data for each source.<br>Units: Milliseconds<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `TotalPackets` | The total number of packets that were received.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `FailoverSwitches` | The total number of times the flow switches back and forth between sources when using the *Failover* mode for source failover. |

## TR 101 290 Priority 1 metrics

The following table lists TR 101 290 Priority 1 metrics that AWS Elemental MediaConnect sends to CloudWatch.

| Metric | Description |
|---|---|
| `ContinuityCounter` | The number of times that a continuity error occurred. This error indicates an incorrect packet order or lost packets.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `PATError` | The number of times that a program association table (PAT) error occurred. This error indicates that the PAT is missing. The PAT lists the programs that are available in a transport stream (TS) and points to the program map tables (PMTs). The decoder needs the PAT to do its job.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `PIDError` | The number of times that a packet identifier (PID) error occurred. This error indicates that a PID is missing its associated data stream. The PIDs are identifiers that provide the location of the video, audio, and data streams. This error can occur after the transport stream has been multiplexed and then remultiplexed.<br>Units: Count |

| Metric | Description |
|---|---|
| | Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `PMTError` | The number of times that a program map table (PMT) error occurred. This error happens when the PMT is not received at least every 500 milliseconds (ms). Each PMT contains a list of PIDs, which help decoders reassemble data. The decoder needs the PMTs to do its job.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `TSByteError` | The number of times that a transport stream byte error occurred. This error indicates that the sync byte did not appear after the prescribed number of bytes.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `TSSyncLoss` | The number of times that a TS sync loss error occurred. This error happens after two or more consecutive TS byte errors.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |

## TR 101 290 Priority 2 metrics

The following table lists TR 101 290 Priority 2 metrics that AWS Elemental MediaConnect sends to CloudWatch.

| Metric | Description |
|---|---|
| `CATError` | The number of times that a conditional access table (CAT) error occurred. This error indicates that the CAT is not present. The CAT tells the integrate d receiver decoder (IRD) where to find management messages for the conditional access (CA) systems that are in use.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `CRCError` | The number of times that a cyclic redundancy check (CRC) error occurred. This error happens when a CRC determines that data is corrupted.<br>Units: Count |

| Metric | Description |
|---|---|
| | Valid dimensions:<br>- Flow ARN<br>- Availability Zone<br>- All flows |
| `PCRAccuracyError` | The number of times that a program clock register (PCR) accuracy error occurred. This error happens when the value of the transmitted PCR differs from what is expected by more than 500 nanoseconds (ns). When a stream is encoded, the encoder assigns periodic PCR values of the encoder's program clock. The decoder relies on these values to ensure that the stream is kept in sync.<br>Units: Count<br>Valid dimensions:<br>- Flow ARN<br>- Availability Zone<br>- All flows |
| `PCRError` | The number of times that a PCR error occurred. This error happens when PCR values are not sent frequently enough. The service relies on consistent, frequent PCRs to reset the local 27 MHz system clock. Although the error occurs when the interval exceeds 100 milliseconds (ms), best practices dictate that PCRs should be received at least every 40 ms.<br>Units: Count<br>Valid dimensions:<br>- Flow ARN<br>- Availability Zone<br>- All flows |

| Metric | Description |
|---|---|
| `PTSError` | The number of times that a presentation timestamp (PTS) error occurred. This error happens when a presentation timestamp (PTS) is not received at least every 700 ms. This can occur if the PTS is sent less frequently or not at all. The most common cause of this error is when the transport stream (TS) is scrambled.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `TransportError` | The number of times that a primary transport error occurred. This error indicates that the TS packet is unusable. When this error occurs, ignore all other TR 101 290 errors for this packet.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |

## Maintenance metrics

The following table lists flow maintenance metrics that AWS Elemental MediaConnect sends to CloudWatch.

| Metric | Description |
|---|---|
| `MaintenanceScheduled` | Maintenance is scheduled for the flow.<br>Units: Count |

| Metric | Description |
|---|---|
| | Valid dimensions:<br>- Flow ARN<br>- All flows |
| `MaintenanceRescheduled` | MediaConnect is unable to perform maintenance at the previously scheduled date and time. A new date and time has been automatically assigned by MediaConnect for this flow's maintenance.<br>Units: Count<br>Valid dimensions:<br>- Flow ARN<br>- All flows |
| `MaintenanceCanceled` | Maintenance for this flow is cancelled by MediaConnect.<br>Units: Count<br>Valid dimensions:<br>- Flow ARN<br>- All flows |
| `MaintenanceStarted` | Maintenance has started and is currently in progress for this flow.<br>Units: Count<br>Valid dimensions:<br>- Flow ARN<br>- All flows |

| Metric | Description |
|---|---|
| `MaintenanceSucceeded` | Maintenance completed successfully for this flow.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• All flows |
| `MaintenanceFailed` | Maintenance did not complete successfully for this flow.<br>Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• All flows |

## AWS Elemental MediaConnect metrics to monitor source health

AWS Elemental MediaConnect sends metrics to CloudWatch. You can review specific metrics to evaluate the health of the source of your flow. If the flow is unhealthy, these metrics can help you determine whether the issue originates with the source. For details about each metric, see the tables in this section.

For information about flow metrics, see Metrics to monitor flow health.

> **Note**
>
> Metrics tracked by MediaConnect adhere to the standard as defined by the TR 101 290 spec.

**Topics**

- Source metrics
- TR 101 290 Priority 1 metrics

- TR 101 290 Priority 2 metrics

## Source metrics

The following table lists source metrics that AWS Elemental MediaConnect sends to CloudWatch.

| Metric | Description |
|---|---|
| `SourceARQRecovered` | The number of dropped packets that were recovered by automatic repeat request (ARQ). This metric applies to sources that use the RIST, Zixi, SRT or Fujitsu-QoS protocol. It doesn't apply to flows that receive content from an entitlement.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourceARQRequests` | The number of retransmitted packets that were requested through automatic repeat request (ARQ) and received. This metric applies to sources that use the RIST, Zixi, SRT or Fujitsu-QoS protocol. It doesn't apply to flows that receive content from an entitlement.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourceBitRate` | The bitrate of the incoming (source) video. |

| Metric | Description |
|---|---|
| | Units: bits per second (b/s)<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows<br>**Note**<br>MediaConnect suppresses null packets in an effort to optimize the data connection between the content originator's flow and the subscriber's flow. This can result in a fluctuating bitrate on the subscriber's flow, or a difference between the bitrate of the content originator's flow and the subscriber’s flow. We recommend that you monitor source health as a combination of `SourceBitRate` and other metrics such as `SourceContinuityCounter` and `SourceNotRecoveredPackets` . |
| `SourceConnected` | The status of the source. A value of 1 indicates that the source is connected and a value of 0 (zero) indicates that the source is disconnected. This metric applies only to sources that use the Zixi, SRT, or Fujitsu-QoS protocol.<br>Units: None<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `SourceDisconnections` | The number of times that the source status changed from connected to disconnected.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourceDroppedPackets` | The number of packets that were lost during transit. This value is measured before any error correction takes place.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `SourceFEC Packets` | The number of packets that were transmitted using forward error correction (FEC) and received. This metric applies only to sources that use the RTP-FEC, Zixi, or Fujitsu protocols. It doesn't apply to flows that receive content from an entitlement.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourceFEC Recovered` | The number of packets that were transmitted using forward error correction (FEC), lost during transit, and recovered. This metric applies only to sources that use the RTP-FEC, Zixi, or Fujitsu protocols. It doesn't apply to flows that receive content from an entitlement.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `SourceMergeActive` | An indication of the status of the source with respect to other sources. This metric is useful when the flow has multiple sources for failover and you are using the Merge failover mode. A value of 1 indicates that the flow has multiple sources and that this source is actively in use, with 2022-7 merge. A value of 0 (zero) indicates that the flow is not using the source to form the stream.<br>Units: None<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourceSelected` | An indication if a source is being used as the input for flow ingest. This metric applies if your flow uses source failover, and the failover mode is set to *Failover*. A value of 1 indicates that the source is being used as the input. A value of 0 (zero) indicates that the flow is being used as the backup stream.<br>Units: None<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| SourceMergeLatency | The amount of time that this source trails the primary source. If this source is the primary source, the value is 0 (zero).<br>Units: Milliseconds<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| SourceMergeStatusWarnMismatch | A status metric warning that the flow is receiving mismatched sources. This means that any dropped packets will not be recovered and will result in poor network reliability. This metric only applies to sources using **merge mode** failover. Merge mode failover requires both sources to be *binary identical*. To be binary identical, the sources must originate from the same encoder. This will ensure the sources can share missing packets, as the packets are identical .<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `SourceMergeStatusWarnSolo` | A status metric warning that the flow is only receiving one source. This means any dropped packets will not be recovered and will result in poor network reliability. This metric only applies to sources using **merge mode** failover.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourceNotRecoveredPackets` | The number of packets that were lost during transit and were not recovered by error correction.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `SourceMissingPackets` | A packet was missing from both source streams, this means the packet could not be recovered. This metric only applies to sources using **merge mode** failover.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourceOverflowPackets` | The number of packets that were lost in transit because the video required more buffer than was available. This metric doesn't apply to flows that receive content from an entitlement or to flows that have multiple sources.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| SourcePacketLossPercent | The percentage of packets that were lost during transit, even if they were recovered.<br>Units: Percent<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| SourceRecoveredPackets | The number of packets that were lost during transit, but recovered.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| SourceRoundTripTime | The amount of time it takes for the source to send a signal and receive an acknowledgment from AWS Elemental MediaConnect. This metric applies to sources that use the RIST, Zixi, SRT or Fujitsu-QoS protocol. It doesn't apply to flows that receive content from an entitlement.<br>Units: Milliseconds<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `SourceTotalPackets` | The total number of packets that were received.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourceTotalBytes` | Total amount of bytes transferred to MediaConnect from the source.<br>Units: Bytes<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourceDroppedPayloads` | Payloads that were lost during transit to MediaConnect from the source. A payload is a frame of video or an audio sample. Payloads can consist of multiple packets. Payload metrics are only applicable when using CDI.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `SourceLatePayloads` | Packets of a payload that arrive outside of the configured **Max sync buffer** time frame. A payload is a frame of video or an audio sample. Payloads can consist of multiple packets. Payload metrics are only applicable when using CDI.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourceTotalPayloads` | Total amount of payloads delivered to MediaConnect from the source. A payload is a frame of video or an audio sample. Payloads can consist of multiple packets. Payload metrics are only applicable when using CDI.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

## TR 101 290 Priority 1 metrics

The following table lists TR 101 290 Priority 1 metrics that AWS Elemental MediaConnect sends to CloudWatch.

| Metric | Description |
|---|---|
| `SourceContinuityCounter` | The number of times that a continuity error occurred. This error indicates an incorrect packet order or lost packets.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourcePATError` | The number of times that a program association table (PAT) error occurred. This error indicates that the PAT is missing. The PAT lists the programs that are available in a transport stream (TS) and points to the program map tables (PMTs). The decoder needs the PAT to do its job.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourcePIDError` | The number of times that a packet identifier (PID) error occurred. This error indicates that a PID is missing its associated data stream. The PIDs are identifiers that provide the location of the video, audio, and data streams. This error can occur after the TS has been multiplexed and then remultipl exed.<br>Units: Count<br>Valid dimensions:<br>• Source ARN |

| Metric | Description |
|---|---|
| | • Flow ARN<br>• Availability Zone<br>• All flows |
| `SourcePMTError` | The number of times that a program map table (PMT) error occurred. This error happens when the PMT is not received at least every 500 milliseconds (ms). Each PMT contains a list of PIDs, which help decoders reassemble data. The decoder needs the PMTs to do its job.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourceTSByteError` | The number of times that a TS byte error occurred. This error indicates that the sync byte did not appear after the prescribed number of bytes.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `SourceTSSyncLoss` | The number of times that a TS sync loss error occurred. This error happens after two or more consecutive TS byte errors.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

## TR 101 290 Priority 2 metrics

The following table lists TR 101 290 Priority 2 metrics that AWS Elemental MediaConnect sends to CloudWatch.

| Metric | Description |
|---|---|
| `SourceCATError` | The number of times that a conditional access table (CAT) error occurred. This error indicates that the CAT is not present. The CAT tells the integrated receiver decoder (IRD) where to find management messages for the conditional access (CA) systems that are in use.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourceCRCError` | The number of times that a cyclic redundancy check (CRC) error occurred. This error happens when a CRC determines that data is corrupted. |

| Metric | Description |
|---|---|
| | Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourcePCRAccuracyError` | The number of times that a program clock register (PCR) accuracy error occurred. This error happens when the value of the transmitted PCR differs from what is expected by more than 500 nanoseconds (ns). When a stream is encoded, the encoder assigns periodic PCR values from the encoder's program clock. The decoder relies on these values to ensure that the stream is kept in sync.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `SourcePCRError` | The number of times that a PCR error occurred. This error happens when PCR values are not sent frequently enough. The service relies on consistent, frequent PCRs to reset the local 27 MHz system clock. Although the error occurs when the interval exceeds 100 milliseconds (ms), best practices dictate that PCRs should be received at least every 40 ms.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `SourcePTSError` | The number of times that a presentation timestamp (PTS) error occurred. This error happens when a presentation timestamp (PTS) is not received at least every 700 ms. This can occur if the PTS is sent less frequently or not at all. The most common cause of this error is when the TS is scrambled.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| SourceTransportError | The number of times that a primary transport error occurred. This error indicates that the TS packet is unusable. When this error occurs, ignore all other TR 101 290 errors for this packet.<br>Units: Count<br>Valid dimensions:<br>• Source ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

## AWS Elemental MediaConnect metrics to monitor output health

AWS Elemental MediaConnect sends metrics to CloudWatch. You can review specific metrics to evaluate the health of the output of your flow.

**Note**

Metrics tracked by MediaConnect adhere to the standard as defined by the TR 101 290 spec.

**Topics**

- Output metrics for transport stream protocols
- Output metrics for CDI protocols

### Output metrics for transport stream protocols

| Metric | Description |
|---|---|
| Connected Outputs | The number of outputs that are currently connected. This metric applies to outputs that use the Zixi, Fujitsu or SRT protocol. |

| Metric | Description |
|---|---|
| | Units: Count<br>Valid dimensions:<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `OutputConnected` | The status of the output. A value of 1 indicates that the output is connected , and a value of 0 (zero) indicates that the output is disconnected. This metric applies to outputs that use the Zixi or SRT protocol.<br>Units: None<br>Valid dimensions:<br>• Output ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `OutputDisconnections` | The number of times that the output status changed from connected to disconnected. This metric applies to outputs that use the Zixi or SRT protocol.<br>Units: Count<br>Valid dimensions:<br>• Output ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| OutputBitrate | The bitrate of the outgoing (output) video. This metric applies to outputs that use the SRT, or Fujitsu-QoS protocols or output to MediaLive.<br>Units: bits per second (b/s)<br>Valid dimensions:<br>• Output ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows<br>**Note**<br>The OutputBitrate value can vary depending on the selected protocol due to non-payload packets, retransmitted packets, packet headers, and other protocol-specific packets. Due to these factors the bitrate value reported by this metric might vary between outputs. |
| OutputTotalPackets | The total number of packets that were sent to the output. This metric applies to outputs that use the SRT, or Fujitsu-QoS protocols or output to MediaLive.<br>Units: Count<br>Valid dimensions:<br>• Output ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `OutputFEC Packets` | The number of packets that were transmitted using forward error correctio n (FEC) and received. This metric applies to outputs that use the Fujitsu protocol.<br>Units: Count<br>Valid dimensions:<br>• Output ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `OutputARQ Requests` | The number of retransmitted packets that were requested through automatic repeat request (ARQ) and received. This metric applies to outputs that use the SRT protocol or output to MediaLive.<br>Units: Count<br>Valid dimensions:<br>• Output ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `OutputResentPackets` | The number of packets that were retransmitted to the output destination. This metric applies to outputs that use the SRT protocol or output to MediaLive.<br>Units: Count<br>Valid dimensions:<br>• Output ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `OutputRoundTripTime` | The amount of time it takes for the output to send a signal and receive an acknowledgment from the output destination. This metric applies to outputs that use the SRT protocol or output to MediaLive.<br>Units: Milliseconds<br>Valid dimensions:<br>• Output ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `OutputNotRecoveredPackets` | The number of packets that were lost during transit and were not recovered by error correction. This metric applies to outputs to MediaLive.<br>Units: Count<br>Valid dimensions:<br>• Output ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |

## Output metrics for CDI protocols

| Metric | Description |
|---|---|
| `OutputTotalBytes` | Total amount of bytes transferred from MediaConnect to the output. This metric is only applicable when using CDI.<br>Units: Bytes<br>Valid dimensions:<br>• Output ARN<br>• Flow ARN<br>• Availability Zone<br>• All flows |
| `OutputDroppedPayloads` | Payloads that were lost during transit from MediaConnect to the output. A payload is a frame of video or an audio sample. Payloads can consist of multiple packets. Payload metrics are only applicable when using CDI.<br>Units: Count<br>Valid dimensions: |

| Metric | Description |
|---|---|
| | - Output ARN<br>- Flow ARN<br>- Availability Zone<br>- All flows |
| `OutputLatePayloads` | Packets of a payload that arrive at the output outside of MediaConnect's internal buffer. A payload is a frame of video or an audio sample. Payloads can consist of multiple packets. Payload metrics are only applicable when using CDI.<br>Units: Count<br>Valid dimensions:<br>- Output ARN<br>- Flow ARN<br>- Availability Zone<br>- All flows |
| `OutputTotalPayloads` | Total amount of payloads delivered from MediaConnect to the output. A payload is a frame of video or an audio sample. Payloads can consist of multiple packets. Payload metrics are only applicable when using CDI.<br>Units: Count<br>Valid dimensions:<br>- Output ARN<br>- Flow ARN<br>- Availability Zone<br>- All flows |

# AWS Elemental MediaConnect metrics to monitor media health

AWS Elemental MediaConnect sends metrics to CloudWatch. You can review specific metrics to evaluate the health of the media transmitted by MediaConnect. The media health metrics listed below only apply to Transport Stream (TS) flows. For details about each metric, see the table in this section.

## Media metrics

The following table lists media metrics that AWS Elemental MediaConnect sends to CloudWatch.

| Metric | Description |
|---|---|
| `ConsecutiveDrops` | The number of data packets that were dropped in a row during transmission of data to or from MediaConnect.<br>Units: Count<br>Supported protocols:<br>- Zixi<br>Supported statistics:<br>- Maximum<br>- Minimum<br>- Average<br>Valid dimension sets:<br>- Flow ARN<br>- Source ARN<br>- Availability Zone<br>- All flows |
| `ConsecutiveNotRecovered` | The number of data packets that were not recovered in a row. After a data packet is dropped, error correction attempts to recover that packet. This |

| Metric | Description |
|---|---|
| | metric helps to identify extended periods of data packets that were dropped and not recovered.<br>Units: Count<br>Supported protocols:<br>- Zixi<br>Supported statistics:<br>- Maximum<br>- Minimum<br>- Average<br>Valid dimension sets:<br>- Flow ARN<br>- Source ARN<br>- Availability Zone<br>- All flows |

| Metric | Description |
|---|---|
| `Jitter` | The current network jitter, measured in milliseconds. Network jitter is a measurement of changes in latency. An increase in network jitter indicates inconsistency in the latency and can negatively impact quality.<br>Units: milliseconds (ms)<br>Supported protocols:<br>• All Transport Stream (TS) protocols<br>Supported statistics:<br>• Maximum<br>• Minimum<br>• Average<br>Valid dimension sets:<br>• Flow ARN<br>• Source ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `Latency` | The stream latency of the flow or source. Latency is the time it takes for data packets to travel from your source to MediaConnect.<br>Units: milliseconds (ms)<br>Supported protocols:<br>• All Transport Stream (TS) protocols<br>Supported statistics:<br>• Maximum<br>• Minimum<br>• Average<br>Valid dimension sets:<br>• Flow ARN<br>• Source ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| `ConnectionAttempts` | The number of reconnection attempts. If the MediaConnect flow or source loses its connection, it will attempt to reconnect automatically.<br>Units: Count<br>Supported protocols:<br>• Zixi<br>• SRT listener<br>• SRT caller<br>Supported statistics:<br>• Sum<br>Valid dimension sets:<br>• Flow ARN<br>• Source ARN<br>• Availability Zone<br>• All flows |

| Metric | Description |
|---|---|
| SourceUptime | The number of seconds that the source has been active. If the source is disconnected or has a connection timeout, this metric resets to zero.<br>Units: Count<br>Supported protocols:<br>• All Transport Stream (TS) protocols<br>Supported statistics:<br>• Maximum<br>• Minimum<br>• Average<br>Valid dimension sets:<br>• Flow ARN<br>• Source ARN<br>• Availability Zone<br>• All flows |

## AWS Elemental MediaConnect metrics to monitor gateway health

AWS Elemental MediaConnect sends metrics to CloudWatch. You can review specific metrics to evaluate the health of your gateways. If the flow in or out of the gateway is unhealthy, these metrics can help you determine where the issue originates from. For details about each metric, see the tables in this section.

> **Note**
>
> MediaConnect Gateway metrics are not available in high resolution periods (one second). You must select a period of at least one minute.

**Topics**

- Gateway ingress metrics
- Gateway ingress source metrics
- Gateway egress metrics
- Gateway egress source metrics

## Gateway ingress metrics

The following table lists gateway ingress metrics that AWS Elemental MediaConnect sends to CloudWatch.

| Metric | Description |
|---|---|
| `IngressBridgeBitRate` | The bitrate of the ingress bridge's source, after the failover merge. This source originates from your local datacenter.<br>Units: bits per second (bps)<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `IngressBridgeCATError` | The number of times that a conditional access table (CAT) error occurred. This error indicates that the CAT is not present. The CAT tells the integrated receiver decoder (IRD) where to find management messages for the conditional access (CA) systems that are in use.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `IngressBridgeCRCError` | The number of times that a cyclic redundancy check (CRC) error occurred. This error happens when a CRC determines that data is corrupted. |

| Metric | Description |
|---|---|
| | Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `IngressBridgeContinuityCounter` | The number of times that a continuity error occurred. This error indicates an incorrect packet order or lost packets.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `IngressBridgeDroppedPackets` | The number of packets that were lost during transit. This value is measured before any error correction takes place.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `IngressBridgeFailoverSwitches` | The total number of times the bridge switches back and forth between sources when using the *Failover* mode for source failover.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

| Metric | Description |
|---|---|
| `IngressBridgeMergeActive` | The merge status of all sources on the bridge. A value of 1 indicates that all sources are merged. A value of 0 (zero) indicates that at least one source is not actively merged with 2022-7.<br>Units: None<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `IngressBridgeNotRecoveredPackets` | The number of packets that were lost during transit and were not recovered by error correction.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `IngressBridgePATError` | The number of times that a program association table (PAT) error occurred. This error indicates that the PAT is missing. The PAT lists the programs that are available in a transport stream (TS) and points to the program map tables (PMTs). The decoder needs the PAT to do its job.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

| Metric | Description |
|---|---|
| IngressBridgePCRAccuracyError | The number of times that a program clock register (PCR) accuracy error occurred. This error happens when the value of the transmitted PCR differs from what is expected by more than 500 nanoseconds (ns). When a stream is encoded, the encoder assigns periodic PCR values of the encoder's program clock. The decoder relies on these values to ensure that the stream is kept in sync.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| IngressBridgePCRError | The number of times that a PCR error occurred. This error happens when PCR values are not sent frequently enough. The service relies on consisten t, frequent PCRs to reset the local 27 MHz system clock. Although the error occurs when the interval exceeds 100 milliseconds (ms), best practices dictate that PCRs should be received at least every 40 ms.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

| Metric | Description |
|---|---|
| `IngressBridgePIDError` | The number of times that a packet identifier (PID) error occurred. This error indicates that a PID is missing its associated data stream. The PIDs are identifiers that provide the location of the video, audio, and data streams. This error can occur after the transport stream has been multiplexed and then remultiplexed.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `IngressBridgePMTError` | The number of times that a program map table (PMT) error occurred. This error happens when the PMT is not received at least every 500 milliseconds (ms). Each PMT contains a list of PIDs, which help decoders reassemble data. The decoder needs the PMTs to do its job.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `IngressBridgePTSError` | The number of times that a presentation timestamp (PTS) error occurred. This error happens when a presentation timestamp (PTS) is not received at least every 700 ms. This can occur if the PTS is sent less frequently or not at all. The most common cause of this error is when the transport stream (TS) is scrambled.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

| Metric | Description |
|---|---|
| `IngressBridgePacketLossPercent` | The percentage of packets that were lost during transit, even if they were recovered.<br>Units: Percent<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `IngressBridgeRecoveredPackets` | The number of packets that were lost during transit, but recovered.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `IngressBridgeTSByteError` | The number of times that a transport stream byte error occurred. This error indicates that the sync byte did not appear after the prescribed number of bytes.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

| Metric | Description |
|---|---|
| `IngressBridgeTSSyncLoss` | The number of times that a transport stream sync loss error occurred. This error happens after two or more consecutive transport stream byte errors.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `IngressBridgeTotalPackets` | The total number of packets that were received.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `IngressBridgeTransportError` | The number of times that a primary transport error occurred. This error indicates that the transport stream packet is unusable. When this error occurs, ignore all other TR 101 290 errors for this packet.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

## Gateway ingress source metrics

The following table lists gateway ingress source metrics that AWS Elemental MediaConnect sends to CloudWatch.

| Metric | Description |
|---|---|
| `IngressBridgeSourceARQRecovered` | The number of dropped packets that were recovered by automatic repeat request (ARQ). It doesn't apply to flows that receive content from an entitlement.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourceARQRequests` | The number of retransmitted packets that were requested through automatic repeat request (ARQ) and received. It doesn't apply to flows that receive content from an entitlement.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourceBitRate` | The bitrate of the ingress bridge's source, prior to any failover merge. This source originates from your local datacenter.<br>Units: bits per second (bps)<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourceCATError` | The number of times that a conditional access table (CAT) error occurred. This error indicates that the CAT is not present. The CAT tells the integrated receiver decoder (IRD) where to find management messages for the conditional access (CA) systems that are in use.<br>Units: Count |

| Metric | Description |
|---|---|
| | Valid dimension sets:<br>- Bridge ARN, Bridge Source Name<br>- Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourceCRCError` | The number of times that a cyclic redundancy check (CRC) error occurred. This error happens when a CRC determines that data is corrupted.<br>Units: Count<br>Valid dimension sets:<br>- Bridge ARN, Bridge Source Name<br>- Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourceContinuityCounter` | The number of times that a continuity error occurred. This error indicates an incorrect packet order or lost packets.<br>Units: Count<br>Valid dimension sets:<br>- Bridge ARN, Bridge Source Name<br>- Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourceDroppedPackets` | The number of packets that were lost during transit. This value is measured before any error correction takes place.<br>Units: Count<br>Valid dimension sets:<br>- Bridge ARN, Bridge Source Name<br>- Gateway ARN, Instance ID, Network Name |

| Metric | Description |
|---|---|
| `IngressBridgeSourceFECPackets` | The number of packets that were transmitted using forward error correction (FEC) and received. It doesn't apply to flows that receive content from an entitlement.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourceFECRecovered` | The number of packets that were transmitted using forward error correction (FEC), lost during transit, and recovered. It doesn't apply to flows that receive content from an entitlement.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourceMergeActive` | An indication of the status of the source with respect to other sources. This metric is useful when the bridge has multiple sources for failover and you are using the Merge failover mode. A value of 1 indicates that the bridge has multiple sources and that this source is actively in use, with 2022-7 merge. A value of 0 (zero) indicates that the bridge is not using the source to form the stream.<br>Units: None<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |

| Metric | Description |
|---|---|
| `IngressBridgeSourceMergeLatency` | The amount of time that this source trails the primary source. If this source is the primary source, the value is 0 (zero).<br>Units: Milliseconds<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourceNotRecoveredPackets` | The number of packets that were lost during transit and were not recovered by error correction.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourceOverflowPackets` | The number of packets that were lost in transit because the video required more buffer than was available. This metric doesn't apply to flows that receive content from an entitlement or to flows that have multiple sources.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |

| Metric | Description |
|---|---|
| `IngressBridgeSourcePATError` | The number of times that a program association table (PAT) error occurred. This error indicates that the PAT is missing. The PAT lists the programs that are available in a transport stream (TS) and points to the program map tables (PMTs). The decoder needs the PAT to do its job.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourcePCRAccuracyError` | The number of times that a program clock register (PCR) accuracy error occurred. This error happens when the value of the transmitted PCR differs from what is expected by more than 500 nanoseconds (ns). When a stream is encoded, the encoder assigns periodic PCR values from the encoder's program clock. The decoder relies on these values to ensure that the stream is kept in sync.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |

| Metric | Description |
|---|---|
| `IngressBridgeSourcePCRError` | The number of times that a PCR error occurred. This error happens when PCR values are not sent frequently enough. The service relies on consistent, frequent PCRs to reset the local 27 MHz system clock. Although the error occurs when the interval exceeds 100 milliseconds (ms), best practices dictate that PCRs should be received at least every 40 ms.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourcePIDError` | The number of times that a packet identifier (PID) error occurred. This error indicates that a PID is missing its associated data stream. The PIDs are identifiers that provide the location of the video, audio, and data streams. This error can occur after the transport stream has been multiplexed and then remultiplexed.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourcePMTError` | The number of times that a program map table (PMT) error occurred. This error happens when the PMT is not received at least every 500 milliseconds (ms). Each PMT contains a list of PIDs, which help decoders reassemble data. The decoder needs the PMTs to do its job.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |

| Metric | Description |
|---|---|
| `IngressBridgeSourcePTSError` | The number of times that a presentation timestamp (PTS) error occurred. This error happens when a presentation timestamp (PTS) is not received at least every 700 ms. This can occur if the PTS is sent less frequently or not at all. The most common cause of this error is when the TS is scrambled.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourcePacketLossPercent` | The percentage of packets that were lost during transit, even if they were recovered.<br>Units: Percent<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourceRecoveredPackets` | The number of packets that were lost during transit, but recovered.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |

| Metric | Description |
|---|---|
| `IngressBridgeSourceRoundTripTime` | The amount of time it takes for the source to send a signal and receive an acknowledgment from AWS Elemental MediaConnect. It doesn't apply to flows that receive content from an entitlement.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourceTSByteError` | The number of times that a transport stream byte error occurred. This error indicates that the sync byte did not appear after the prescribed number of bytes.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| `IngressBridgeSourceTSSyncLoss` | The number of times that a transport stream sync loss error occurred. This error happens after two or more consecutive transport stream byte errors.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |

| Metric | Description |
|---|---|
| IngressBridgeSourceTotalPackets | The total number of packets that were received.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |
| IngressBridgeSourceTransportError | The number of times that a primary transport error occurred. This error indicates that the transport stream packet is unusable. When this error occurs, ignore all other TR 101 290 errors for this packet.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name<br>• Gateway ARN, Instance ID, Network Name |

## Gateway egress metrics

The following table lists gateway egress metrics that AWS Elemental MediaConnect sends to CloudWatch.

| Metric | Description |
|---|---|
| EgressBridgeBitRate | The bitrate of the egress bridge's source, after the failover merge. This source originates from a MediaConnect flow.<br>Units: bits per second (bps)<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

| Metric | Description |
|---|---|
| `EgressBridgeCATError` | The number of times that a conditional access table (CAT) error occurred. This error indicates that the CAT is not present. The CAT tells the integrated receiver decoder (IRD) where to find management messages for the conditional access (CA) systems that are in use.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `EgressBridgeCRCError` | The number of times that a cyclic redundancy check (CRC) error occurred. This error happens when a CRC determines that data is corrupted.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `EgressBridgeContinuityCounter` | The number of times that a continuity error occurred. This error indicates an incorrect packet order or lost packets.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

| Metric | Description |
|---|---|
| `EgressBridgeDroppedPackets` | The number of packets that were lost during transit. This value is measured before any error correction takes place.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `EgressBridgeFailoverSwitches` | The total number of times the bridge switches back and forth between sources when using the *Failover* mode for source failover.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `EgressBridgeMergeActive` | The merge status of all sources on the bridge. A value of 1 indicates that all sources are merged. A value of 0 (zero) indicates that at least one source is not actively merged with 2022-7.<br>Units: None<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

| Metric | Description |
|---|---|
| `EgressBridgeNotRecoveredPackets` | The number of packets that were lost during transit and were not recovered by error correction.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `EgressBridgePATError` | The number of times that a program association table (PAT) error occurred. This error indicates that the PAT is missing. The PAT lists the programs that are available in a transport stream (TS) and points to the program map tables (PMTs). The decoder needs the PAT to do its job.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `EgressBridgePCRAccuracyError` | The number of times that a program clock register (PCR) accuracy error occurred. This error happens when the value of the transmitted PCR differs from what is expected by more than 500 nanoseconds (ns). When a stream is encoded, the encoder assigns periodic PCR values of the encoder's program clock. The decoder relies on these values to ensure that the stream is kept in sync.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

| Metric | Description |
|---|---|
| `EgressBridgePCRError` | The number of times that a PCR error occurred. This error happens when PCR values are not sent frequently enough. The service relies on consistent, frequent PCRs to reset the local 27 MHz system clock. Although the error occurs when the interval exceeds 100 milliseconds (ms), best practices dictate that PCRs should be received at least every 40 ms.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `EgressBridgePIDError` | The number of times that a packet identifier (PID) error occurred. This error indicates that a PID is missing its associated data stream. The PIDs are identifiers that provide the location of the video, audio, and data streams. This error can occur after the transport stream has been multiplexed and then remultiplexed.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `EgressBridgePMTError` | The number of times that a program map table (PMT) error occurred. This error happens when the PMT is not received at least every 500 milliseconds (ms). Each PMT contains a list of PIDs, which help decoders reassemble data. The decoder needs the PMTs to do its job.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

| Metric | Description |
|---|---|
| `EgressBridgePTSError` | The number of times that a presentation timestamp (PTS) error occurred. This error happens when a presentation timestamp (PTS) is not received at least every 700 ms. This can occur if the PTS is sent less frequently or not at all. The most common cause of this error is when the transport stream (TS) is scrambled.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `EgressBridgePacketLossPercent` | The percentage of packets that were lost during transit, even if they were recovered.<br>Units: Percent<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `EgressBridgeRecoveredPackets` | The number of packets that were lost during transit, but recovered.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

| Metric | Description |
|---|---|
| `EgressBridgeTSByteError` | The number of times that a transport stream byte error occurred. This error indicates that the sync byte did not appear after the prescribed number of bytes.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `EgressBridgeTSSyncLoss` | The number of times that a transport stream sync loss error occurred. This error happens after two or more consecutive transport stream byte errors.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |
| `EgressBridgeTotalPackets` | The total number of packets that were received.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

| Metric | Description |
|---|---|
| `EgressBridgeTransportError` | The number of times that a primary transport error occurred. This error indicates that the transport stream packet is unusable. When this error occurs, ignore all other TR 101 290 errors for this packet.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN<br>• Gateway ARN, Instance ID |

## Gateway egress source metrics

The following table lists gateway egress source metrics that AWS Elemental MediaConnect sends to CloudWatch.

| Metric | Description |
|---|---|
| `EgressBridgeSourceBitRate` | The bitrate of the egress bridge's source, prior to any failover merge. This source originates from a MediaConnect flow.<br>Units: bits per second (bps)<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |
| `EgressBridgeSourceCATError` | The number of times that a conditional access table (CAT) error occurred. This error indicates that the CAT is not present. The CAT tells the integrate d receiver decoder (IRD) where to find management messages for the conditional access (CA) systems that are in use.<br>Units: Count<br>Valid dimension sets: |

| Metric | Description |
|---|---|
| | • Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |
| `EgressBridgeSourceCRCError` | The number of times that a cyclic redundancy check (CRC) error occurred. This error happens when a CRC determines that data is corrupted.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |
| `EgressBridgeSourceContinuityCounter` | The number of times that a continuity error occurred. This error indicates an incorrect packet order or lost packets.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |
| `EgressBridgeSourceDroppedPackets` | The number of packets that were lost during transit. This value is measured before any error correction takes place.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |

| Metric | Description |
|---|---|
| `EgressBridgeSourceMergeActive` | An indication of the status of the source with respect to other sources. This metric is useful when the bridge has multiple sources for failover and you are using the Merge failover mode. A value of 1 indicates that the bridge has multiple sources and that this source is actively in use, with 2022-7 merge. A value of 0 (zero) indicates that the bridge is not using the source to form the stream.<br>Units: None<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |
| `EgressBridgeSourceMergeLatency` | The amount of time that this source trails the primary source. If this source is the primary source, the value is 0 (zero).<br>Units: Milliseconds<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |
| `EgressBridgeSourceNotRecoveredPackets` | The number of packets that were lost during transit and were not recovered by error correction.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |

| Metric | Description |
|---|---|
| EgressBridgeSourcePATError | The number of times that a program association table (PAT) error occurred. This error indicates that the PAT is missing. The PAT lists the programs that are available in a transport stream (TS) and points to the program map tables (PMTs). The decoder needs the PAT to do its job.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |
| EgressBridgeSourcePCRAccuracyError | The number of times that a program clock register (PCR) accuracy error occurred. This error happens when the value of the transmitted PCR differs from what is expected by more than 500 nanoseconds (ns). When a stream is encoded, the encoder assigns periodic PCR values from the encoder's program clock. The decoder relies on these values to ensure that the stream is kept in sync.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |

| Metric | Description |
|---|---|
| `EgressBridgeSourcePCRError` | The number of times that a PCR error occurred. This error happens when PCR values are not sent frequently enough. The service relies on consistent, frequent PCRs to reset the local 27 MHz system clock. Although the error occurs when the interval exceeds 100 milliseconds (ms), best practices dictate that PCRs should be received at least every 40 ms.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |
| `EgressBridgeSourcePIDError` | The number of times that a packet identifier (PID) error occurred. This error indicates that a PID is missing its associated data stream. The PIDs are identifiers that provide the location of the video, audio, and data streams. This error can occur after the transport stream has been multiplexed and then remultiplexed.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |
| `EgressBridgeSourcePMTError` | The number of times that a program map table (PMT) error occurred. This error happens when the PMT is not received at least every 500 milliseconds (ms). Each PMT contains a list of PIDs, which help decoders reassemble data. The decoder needs the PMTs to do its job.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |

| Metric | Description |
|---|---|
| EgressBridgeSourcePTSError | The number of times that a presentation timestamp (PTS) error occurred. This error happens when a presentation timestamp (PTS) is not received at least every 700 ms. This can occur if the PTS is sent less frequently or not at all. The most common cause of this error is when the TS is scrambled.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |
| EgressBridgeSourcePacketLossPercent | The percentage of packets that were lost during transit, even if they were recovered.<br>Units: Percent<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |
| EgressBridgeSourceRecoveredPackets | The number of packets that were lost during transit, but recovered.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |

| Metric | Description |
|---|---|
| `EgressBridgeSourceTSByteError` | The number of times that a transport stream byte error occurred. This error indicates that the sync byte did not appear after the prescribed number of bytes.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |
| `EgressBridgeSourceTSSyncLoss` | The number of times that a transport stream sync loss error occurred. This error happens after two or more consecutive transport stream byte errors.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |
| `EgressBridgeSourceTotalPackets` | The total number of packets that were received.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |

| Metric | Description |
|---|---|
| EgressBridgeSourceTransportError | The number of times that a primary transport error occurred. This error indicates that the transport stream packet is unusable. When this error occurs, ignore all other TR 101 290 errors for this packet.<br>Units: Count<br>Valid dimension sets:<br>• Bridge ARN, Bridge Source Name, Flow ARN<br>• Gateway ARN, Instance ID, Availability Zone |

## Using metrics to troubleshoot

You can monitor the health of your stream by reviewing the metrics that AWS Elemental MediaConnect sends to CloudWatch. In particular, if you encounter a problem on your MediaConnect flow, these metrics can help you isolate the problem. The specific metrics to watch depend on the protocol that your source uses. Review the lists below, which are sorted by source protocol.

**Topics**

### Metrics to watch if your source uses the RIST protocol

If the protocol of your source is RIST, watch the metrics below to evaluate the health of your source.

- ARQRecovered

- `ARQRequests`
- `DroppedPackets`
- `NotRecoveredPackets`
- `OverflowPackets`
- `PacketLossPercent`
- `RecoveredPackets`
- `RoundTripTime`
- `TotalPackets`

## Metrics to watch if your source uses the RTP protocol

If the protocol of your source is RTP, watch the metrics below to evaluate the health of your source.

- `DroppedPackets`
- `OverflowPackets`
- `RoundTripTime`
- `TotalPackets`

## Metrics to watch if your source uses the RTP-FEC protocol

If the protocol of your source is RTP-FEC, watch the metrics below to evaluate the health of your source.

- `DroppedPackets`
- `FECPackets`
- `FECRecovered`
- `NotRecoveredPackets`
- `OverflowPackets`
- `RecoveredPackets`
- `RoundTripTime`
- `TotalPackets`

## Metrics to watch if your source uses the SRT protocol

If the protocol of your source is SRT (listener or caller), watch the metrics below to evaluate the health of your source.

- `ARQRecovered`
- `ARQRequests`
- `DroppedPackets`
- `NotRecoveredPackets`
- `OverflowPackets`
- `RecoveredPackets`
- `RoundTripTime`
- `TotalPackets`

## Metrics to watch if your source uses the Zixi push protocol

If the protocol of your source is Zixi push, watch the metrics below to evaluate the health of your source.

- `ARQRecovered`
- `ARQRequests`
- `DroppedPackets`
- `FECPackets`
- `FECRecovered`
- `NotRecoveredPackets`
- `OverflowPackets`
- `RecoveredPackets`
- `RoundTripTime`
- `TotalPackets`

## Metrics to watch if your source comes from an entitlement

If your source comes from an entitlement that was granted to your account by another AWS account, watch the metrics below to evaluate the health of your source.

- `ARQRecovered`
- `ARQRequests`
- `DroppedPackets`
- `FECPackets`
- `FECRecovered`
- `NotRecoveredPackets`
- `OverflowPackets`
- `RecoveredPackets`
- `RoundTripTime`
- `TotalPackets`

## Metrics to watch if you are using gateways

Watch the metrics below to evaluate the health of your gateway.

### Metrics to watch if you are using a gateway with an ingress bridge

Watch the metrics below to evaluate the health of your gateway's ingress bridge. The recommended ingress bridge troubleshooting metrics are separated by protocol.

- RTP
  - `IngressBridgeTotalPackets`
  - `IngressBridgeDroppedPackets`
  - `IngressBridgeSourceTotalPackets`
  - `IngressBridgeSourceDroppedPackets`
  - `IngressBridgeSourceOverflowPackets`
  - `IngressBridgeSourceRoundTripTime`

- RTP-FEC
  - `IngressBridgeTotalPackets`
  - `IngressBridgeDroppedPackets`
  - `IngressBridgeRecoveredPackets`
  - `IngressBridgeNotRecoveredPackets`

  - `IngressBridgeSourceTotalPackets`
  - `IngressBridgeSourceDroppedPackets`
  - `IngressBridgeSourceRecoveredPackets`
  - `IngressBridgeSourceNotRecoveredPackets`
  - `IngressBridgeSourceOverflowPackets`
  - `IngressBridgeSourceFECPackets`
  - `IngressBridgeSourceFECRecovered`
  - `IngressBridgeSourceRoundTripTime`
- UDP
  - `IngressBridgeTotalPackets`
  - `IngressBridgeSourceTotalPackets`
  - `IngressBridgeSourceOverflowPackets`

### Metrics to watch if you are using a gateway with an egress bridge

Watch the metrics below to evaluate the health of your gateway's egress bridge.

- `EgressBridgeTotalPackets`
- `EgressBridgeDroppedPackets`
- `EgressBridgeRecoveredPackets`
- `EgressBridgeNotRecoveredPackets`
- `EgressBridgeSourceTotalPackets`
- `EgressBridgeSourceDroppedPackets`
- `EgressBridgeSourceRecoveredPackets`
- `EgressBridgeSourceNotRecoveredPackets`

## Monitoring with CloudWatch events

Amazon CloudWatch Events enables you to automate your AWS services and respond automatically to system events such as application availability issues or resource changes. Events from AWS services are delivered to CloudWatch Events in near real time. You can write simple rules to indicate which events are of interest to you, and what automated actions to take when an event matches a rule.

The actions that can be automatically triggered using CloudWatch Events include the following:

- Invoking an AWS Lambda function
- Invoking Amazon EC2 Run Command
- Relaying the event to Amazon Kinesis Data Streams
- Activating an AWS Step Functions state machine
- Notifying an Amazon SNS topic or an Amazon SQS queue

For more information, see the Amazon CloudWatch Events User Guide.

**CloudWatch Events in MediaConnect**

- AWS Elemental MediaConnect flow state change event
- AWS Elemental MediaConnect flow maintenance event
- AWS Elemental MediaConnect flow health event
- AWS Elemental MediaConnect alert event
- AWS Elemental MediaConnect source health event
- AWS Elemental MediaConnect output health event

## AWS Elemental MediaConnect flow state change event

This event is published when a flow's state has changed from or to any of the following states: Standby, Active, Updating, Deleting, Starting, Stopping, or Error.

For information about subscribing to this event, see Amazon CloudWatch.

The following message is an example of this CloudWatch event.

```
{
    "account": "111122223333",
    "detail": {
        "currentStatus": "STARTING",
        "previousStatus": "STANDBY"
    },
    "detail-type": "MediaConnect Flow Status Change",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "region": "us-east-1",
    "resources": ["arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow"],
    "source": "aws.mediaconnect",
    "time": "2022-01-06T00:45:47Z",
    "version": "0"
}
```

## AWS Elemental MediaConnect flow maintenance event

This event is published when a flow's maintenance status has changed, either to or from any of the following states:

- **SCHEDULED** - Maintenance is scheduled for the flow.
- **RESCHEDULED** - MediaConnect is unable to perform maintenance at the previously scheduled date and time. A new date and time has been automatically assigned by MediaConnect for this flow's maintenance.
- **CANCELED** - Maintenance for this flow is cancelled by MediaConnect.
- **INPROGRESS** - Maintenance has started and is currently in progress for this flow.
- **FINISHED** - Maintenance completed successfully for this flow.
- **FAILED** - Maintenance did not complete successfully for this flow.

For information about subscribing to this event, see Amazon CloudWatch.

For information about MediaConnect maintenance, see MediaConnect flow maintenance.

The following message is an example of this CloudWatch event.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "MediaConnect Flow Maintenance",
    "source": "aws.mediaconnect",
    "account": "111122223333",
    "time": "2022-02-14T00:45:47Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1:23aBC45dEF67hiJ8:12AbC34DE5fG:ExampleFlow"
    ],
    "detail": {
        "currentStatus": "FINISHED"
    }
}
```

## AWS Elemental MediaConnect flow health event

AWS Elemental MediaConnect publishes flow health events after a flow health indicator state changes.

MediaConnect publishes this event any time there is a state change to one or more of the following flow health indicators. This event publishes the current and previous state of the flow.

The following are flow health indicators:

- **Source state**
  - Possible states: `connected`, `receiving`, `disconnected`, `idle`
- **Failover switch**
  - Possible states: `true`, `false`
- **TR-101**: TR-101 is an industry standard technical recommendation for the monitoring of transport streams (TS). The following events are only published for TS based protocols.
  - **TS sync loss** is `true` when source payloads do not look like a valid transport stream.
  - **Continuity count error** is `true` when the source finds continuity count errors.
  - **Transport error** is `true` when the TS has the transport indicator set.
  - **PCR error** is `true` when there is a PCR discontinuity or a long gap in PCR packet reception.

For information about subscribing to this event, see Amazon CloudWatch.

The following message is an example of this CloudWatch event.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "MediaConnect Flow Health",
  "source": "aws.mediaconnect",
  "account": "012345678901",
  "time": "2006-01-02T15:04:05Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:mediaconnect:us-
east-1:012345678901:flow:1:AbCdEfGhIjKlMnOp:abcdef123455:ExampleFlow"
  ],
  "detail": {
    "unhealthy": true,
    "current": {
      "failover_switch": false,
      "source_state": "CONNECTED",
      "tr101": {
        "ts_sync_loss": false,
        "continuity_count_error": true,
        "transport_error": true,
        "pcr_error": true
      }
    },
    "previous": {
      "failover_switch": false,
      "source_state": "CONNECTED",
      "tr101": {
        "ts_sync_loss": false,
        "continuity_count_error": false,
        "transport_error": false,
        "pcr_error": false
      }
    }
  }
}
```

## AWS Elemental MediaConnect alert event

MediaConnect publishes an alert event when a resource encounters an error. The event contains an error code and a message that describes the issue. These alerts are visible on the MediaConnect console, or by using the `describe-flow` AWS Command Line Interface (AWS CLI) command. For more information about the `describe-flow` command, see AWS CLI Command Reference.

For information about subscribing to this event, see Amazon CloudWatch.

The following message is an example of this CloudWatch event.

```
{
 "version": "0",
 "id": "01234567-0123-0123-0123-0123456789ab",
 "detail-type": "MediaConnect Alert",
 "source": "aws.mediaconnect",
 "account": "111122223333",
 "time": "2022-01-06T00:45:47Z",
 "region": "us-east-1",
 "resources": [
 "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1:AbCdEfGhIjKlMnOp:abcdef123455:ExampleFlow"
 ],
 "detail": {
 "errored": true,
 "error-code": "AccessDeniedException",
 "error-message": "Permission denied accessing encryption key for output
 Test. Removing output until it is fixed (secret arn:aws:secretsmanager:us-
east-1:111122223333:secret:ExampleSecret, role arn:aws:iam::111122223333:role/
ExampleKey)"
 }
}
```

## AWS Elemental MediaConnect source health event

AWS Elemental MediaConnect publishes source health events after a source health indicator state changes.

MediaConnect publishes this event any time there is a state change to one or more of the following source health indicators. This event publishes the current and previous state of the flow. Note that the source health event lists the affected flow and source in the `resources` section.

The following are source health indicators:

- **Source state**
  - Possible states: `connected`, `receiving`, `disconnected`, `idle`
- **TR-101**: TR-101 is an industry standard technical recommendation for the monitoring of transport streams (TS). The following events are only published for TS based protocols.
  - **TS sync loss** - true when source payloads do not look like a valid transport stream.
  - **Continuity count error** - true when the source finds continuity count errors.
  - **Transport error** - true when the TS has the transport indicator set.
  - **PCR error** - true when there is a PCR discontinuity or a long gap in PCR packet reception.

For information about subscribing to this event, see Amazon CloudWatch.

The following message is an example of this CloudWatch event.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "MediaConnect Source Health",
  "source": "aws.mediaconnect",
  "account": "012345678901",
  "time": "2006-01-02T15:04:05Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:mediaconnect:us-
east-1:012345678901:flow:1:AbCdEfGhIjKlMnOp:abcdef123455:ExampleFlow",
    "arn:aws:mediaconnect:us-
east-1:012345678901:source:1:AbCdEfGhIjKlMnOp:abcdef123455:ExampleSource"
  ],
  "detail": {
    "unhealthy": true,
    "current": {
      "source_state": "CONNECTED",
      "tr101": {
        "ts_sync_loss": false,
        "continuity_count_error": true,
        "transport_error": true,
        "pcr_error": true
      }
    },
    "previous": {
      "source_state": "CONNECTED",
      "tr101": {
        "ts_sync_loss": false,
        "continuity_count_error": false,
        "transport_error": false,
        "pcr_error": false
      }
    }
  }
}
```

## AWS Elemental MediaConnect output health event

AWS Elemental MediaConnect publishes output health events after an output health indicator state changes.

MediaConnect publishes this event any time there is a state change to one or more of the following output health indicators. This event publishes the current and previous state of the flow. Note that the output health event lists the affected flow and output in the `resources` section.

The following are output health indicators:

- **Output state**
  - Possible states: `connected`, `receiving`, `disconnected`, `idle`

For information about subscribing to this event, see Amazon CloudWatch.

The following message is an example of this CloudWatch event.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "MediaConnect Output Health",
  "source": "aws.mediaconnect",
  "account": "012345678901",
  "time": "2006-01-02T15:04:05Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:mediaconnect:us-
east-1:012345678901:flow:1:AbCdEfGhIjKlMnOp:abcdef123455:ExampleFlow",
    "arn:aws:mediaconnect:us-
east-1:012345678901:output:1:AbCdEfGhIjKlMnOp:abcdef123455:ExampleOutput"
  ],
  "detail": {
    "current": {
      "output_state": "CONNECTED"
    },
    "previous": {
      "output_state": "DISCONNECTED"
    }
  }
}
```

# Logging AWS Elemental MediaConnect API calls with AWS CloudTrail

AWS Elemental MediaConnect is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Elemental MediaConnect. CloudTrail captures all API calls for AWS Elemental MediaConnect as events. The calls captured include calls from the AWS Elemental MediaConnect console and code calls to the AWS Elemental MediaConnect API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS Elemental MediaConnect. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to AWS Elemental MediaConnect, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

## AWS Elemental MediaConnect information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS Elemental MediaConnect, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for AWS Elemental MediaConnect, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All AWS Elemental MediaConnect actions are logged by CloudTrail and are documented in the AWS Elemental MediaConnect API Reference. For example, calls to the `CreateFlow`, `StartFlow` and `UpdateFlowOutput` operations generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

## Understanding AWS Elemental MediaConnect log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested operation, the date and time of the operation, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `DescribeFlow` operation:

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "ABCDEFGHIJKL123456789",
    "arn": "arn:aws:sts::111122223333:user/testUser",
    "accountId": "111122223333",
    "accessKeyId": "ABCDE12345EFGHIJKLMN",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-11-16T20:34:51Z",
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "ABCDEFGHIJKL123456789",
        "arn": "arn:aws:iam::111122223333:role/Administrator",
        "accountId": "111122223333",
        "userName": "Administrator",
      },
    },
  },
  "eventTime": "2018-11-16T20:34:52Z",
  "eventSource": "mediaconnect.amazonaws.com",
  "eventName": "DescribeFlow",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.17",
  "userAgent": "aws-cli/1.15.40 Python/3.6.5 Darwin/16.7.0 botocore/1.10.40",
  "requestParameters": {
    "flowArn": "arn%3Aaws%3Amediaconnect%3Aus-west-2%111122223333%3Aflow
%3A1-23aBC45dEF67hiJ8-12AbC34DE5fG%3AAwardsShow",
  },
  "responseElements": {
  },
  "requestID": "1a2b3c4d-1234-5678-1234-1a2b3c4d5e6f",
  "eventID": "987abc65-1a2b-3c4d-5d6e-987abc654def",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333",
}
```

# Monitoring flow and source health

On the AWS Elemental MediaConnect console, you can monitor the health of your flows and their sources.

*Flow* health indicates if your flow is not connected due to an issue with an entitlement or encryption.

*Source* health indicates if your source is connected. If it is, the console shows Amazon CloudWatch metrics that provide the status of the source over a period of time.

**Topics**

- Monitoring a flow's health
- Monitoring a source's health

# Monitoring a flow's health

The **Alerts** tab on the MediaConnect console displays a list of alerts that occurred when you started or stopped the current flow. For the full list of alerts for a flow, see Amazon CloudWatch.

MediaConnect displays the following alerts on the **Alerts** tab:

- Contextual error messages about your flow, called **stream errors**.
- The entitlement that this flow is based on is already in use. This occurs if you create more than one flow based on the same entitlement. If the one of those flows is already running, MediaConnect displays an alert if you try to start the second flow.
- The entitlement that this flow is based on no longer exists. This occurs if the account that granted the entitlement (the content originator) revokes the entitlement.
- The entitlement that this flow is based on does not have an active source. This occurs if the originator's flow is deleted or stopped. When you start your flow based on that entitlement, there is no content coming from the originator's flow.
- The decryption or encryption information for the flow isn't valid. This can happen for a number of reasons. For example, the decryption key doesn't match the type for the specified algorithm. Or, your flow is based on an entitlement that uses SPEKE encryption and MediaConnect can't contact the conditional access (CA) platform key provider.
- Your flow is based on an entitlement, and the content originator's flow already has the maximum number of outputs.

## Stream errors

MediaConnect **Alerts** can also contain contextual errors for the flow's sources and outputs. These are called *stream errors* and follow a specific format.

- Source *`source name`* Stream Error: *`error message`*. Please investigate the flow source.
- Output *`output name`* Stream Error: *`error message`*. Please investigate the flow output.

The error message will provide more context for the issue and you can use it as an indicator of where to begin troubleshooting.

### Example

If you received the following alert on a flow named *NationalBroadcast*:

Source `StudioFeed2` Stream Error: `CDI Configuration Error`. Please investigate the flow source.

This would indicate an error with the inbound CDI at the source. Specifically, your next step should be to verify the settings for the *StudioFeed2* source on the flow named *NationalBroadcast*. You would need to pay special attention to the CDI-specific source settings such as the **inbound port**, the **VPC interface** being used, and the **media streams**.

### Viewing flow alerts

**To view any active alerts (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow.
3. Choose the **Alerts** tab.

   The service displays a list of alerts, if there are any, on the flow.

## Monitoring a source's health

In the AWS Elemental MediaConnect console, you can view Amazon CloudWatch metrics that show the health of the source over a period of time. Source health is reported with the following metrics:

- **Source bitrate** – The bitrate of the incoming video.
- **Total packets received** – The total number of packets that MediaConnect received.

**To monitor the health of a source (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow.
3. Choose the **Source** tab and view the status of your source. This includes:

   - The **Source health** field provides the current status of the source.
     - **Connected** indicates that the flow is connected successfully to its source.
     - **Disconnected** indicates that the flow is not connected to its source. To resolve this issue, verify that the source is actually sending content. Also, check the source settings on the flow such as the allowlist CIDR and the protocol configuration.

  - **The flow is inactive** indicates that the flow has not been started. To resolve this issue, start the flow.
  - **Error** indicates that MediaConnect doesn't have permission to communicate with CloudWatch. To resolve the error, you must sign in to the AWS Management Console as an entity that allows MediaConnect to get metric statistics from CloudWatch. For guidance, see this example.
- The **Source health metrics** section is visible only if your source health is **Connected**. The charts show source bitrate and total packets received over the last hour. You can choose different time periods from the dropdown in the top-right corner of the section.

> **Note**
>
> MediaConnect refreshes data from CloudWatch automatically every 1 minute, 5 minutes, or 30 minutes, depending on the time period that you chose. When the charts refresh, data is 1 minute behind real time.

# Tagging AWS Elemental MediaConnect resources

A *tag* is a custom attribute label that you assign or that AWS assigns to an AWS resource. Each tag has two parts:

- A *tag key* (for example, `CostCenter`, `Environment`, or `Project`). Tag keys are case sensitive.
- An optional field known as a *tag value* (for example, `111122223333` or `Production`). Omitting the tag value is the same as using an empty string. Like tag keys, tag values are case sensitive.

Tags help you do the following:

- Identify and organize your AWS resources. Many AWS services support tagging, so you can assign the same tag to resources from different services to indicate that the resources are related. For example, you could assign the same tag to an AWS Elemental MediaConnect flow that you assign to an AWS Elemental MediaLive channel output.
- Track your AWS costs. You activate these tags on the AWS Billing and Cost Management dashboard. AWS uses the tags to categorize your costs and deliver a monthly cost allocation report to you. For more information, see Use Cost Allocation Tags in the *AWS Billing User Guide*.

The following sections provide more information about tags for AWS Elemental MediaConnect.

**Topics**

## Supported resources in AWS Elemental MediaConnect

The following resources in AWS Elemental MediaConnect support tagging:

- Flows
- Sources
- Outputs
- Entitlements

For information about adding and managing tags, see Managing tags.

AWS Elemental MediaConnect doesn't support the tag-based access control feature of AWS Identity and Access Management (IAM).

## Tag naming and usage conventions

The following basic naming and usage conventions apply to using tags with AWS Elemental MediaConnect resources:

- Each resource can have a maximum of 50 tags.
- For each resource, each tag key must be unique, and each tag key can have only one value.
- The maximum tag key length is 128 Unicode characters in UTF-8.
- The maximum tag value length is 256 Unicode characters in UTF-8.
- Allowed characters are letters, numbers, spaces representable in UTF-8, and the following characters: ***. : + = @ _ / -*** (hyphen). Amazon EC2 resources allow any characters.
- Tag keys and values are case sensitive. As a best practice, decide on a strategy for capitalizing tags, and consistently implement that strategy across all resource types. For example, decide

whether to use Costcenter, costcenter, or CostCenter, and use the same convention for all tags. Avoid using similar tags with inconsistent case treatment.

- The aws: prefix is prohibited for tags; it's reserved for AWS use. You can't edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource quota.

## Managing tags

Tags are made up of the Key and Value properties on a resource. You can use the AWS Elemental MediaConnect console, the AWS CLI, or the AWS Elemental MediaConnect API to add, edit, or delete the values for these properties. For information about working with tags, see the following:

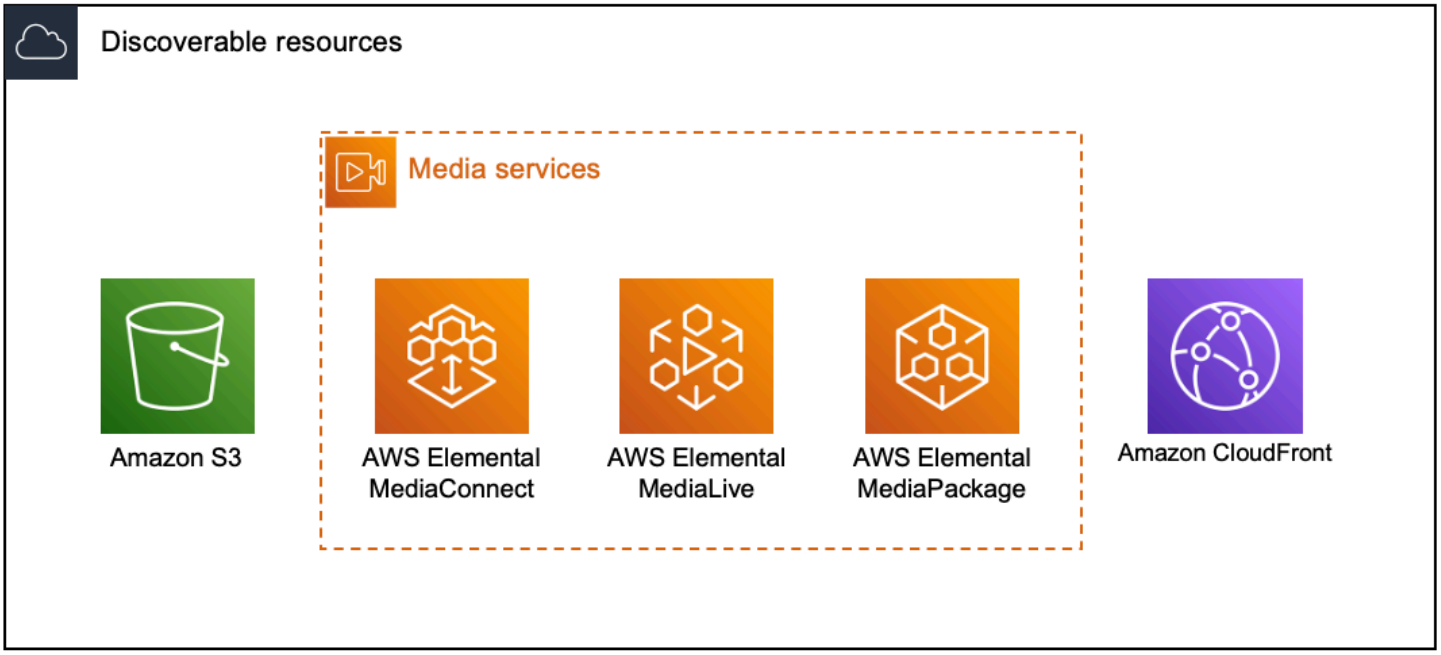
- Resources in the *AWS Elemental MediaConnect API Reference*
- the section called “Managing tags on a flow” in this guide
- the section called “Managing tags on a source” in this guide
- the section called “Managing tags on an output” in this guide
- the section called “Managing tags on an entitlement” in this guide

# Monitoring AWS media services with workflow monitor

Workflow monitor is a tool for the discovery, visualization, and monitoring of AWS media workflows. Workflow monitor is available in the AWS console and API. You can use workflow monitor to discover and create visual mappings of your workflow's resources, called *signal maps*. You can create and manage Amazon CloudWatch alarm and Amazon EventBridge rule templates to monitor the mapped resources. The monitoring templates you create are transformed into deployable AWS CloudFormation templates to allow repeatability. AWS-recommended alarm templates provide predefined best-practice monitoring.

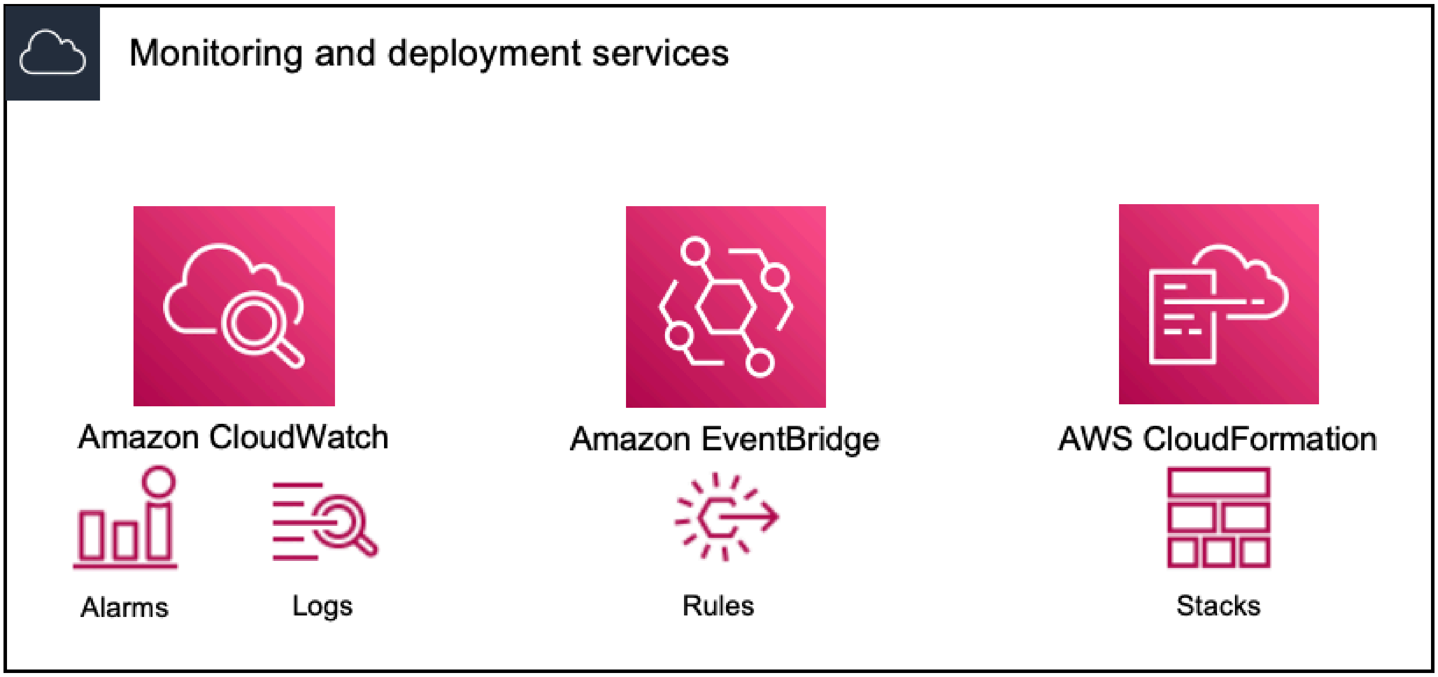
### Discover

Utilize signal maps to automatically discover interconnected AWS resources associated with your media workflow. Discovery can begin at any supported service resource and creates an end-to-end mapping of the workflow. Signal maps can be used as stand-alone visualization tools or enhanced with monitoring templates.



### Monitor

You can create custom CloudWatch alarm and EventBridge rule templates to monitor the health and status of your media workflows. Best practice alarm templates are available to import into your workflow monitor environment. You can use the best practice alarm templates as they are, or edit them to better fit your workflow. Any templates you create are transformed into AWS CloudFormation templates for repeatable deployment.



**Note**

There is no direct cost for using workflow monitor. However, there are costs associated with the resources created and used to monitor your workflow.

When monitoring is deployed, Amazon CloudWatch and Amazon EventBridge resources are created. When using the AWS Management Console, prior to deploying monitoring to a signal map, you will be notified of how many resources will be created. For more information about pricing, see: CloudWatch pricing and EventBridge pricing.

Workflow monitor uses AWS CloudFormation templates to deploy the CloudWatch and EventBridge resources. These templates are stored in a standard class Amazon Simple Storage Service bucket that is created on your behalf, by workflow monitor, during the deployment process and will incur object storage and recall charges. For more information about pricing, see: Amazon S3 pricing.

Previews generated in the workflow monitor signal map for AWS Elemental MediaPackage channels are delivered from the MediaPackage Origin Endpoint and will incur Data Transfer Out charges. For pricing, see: MediaPackage pricing.

## Components of workflow monitor

Workflow monitor has four major components:

- CloudWatch alarm templates - Define the conditions you would like to monitor using CloudWatch. You can create your own alarm templates, or import predefined templates created by AWS. For more information, see: CloudWatch alarm groups and templates
- EventBridge rule templates - Define how EventBridge sends notifications when an alarm is triggered. For more information, see: EventBridge rule groups and templates
- Signal maps - Use an automated process to create AWS Elemental workflow maps using existing AWS resources. The signal maps can be used to discover resources in your workflow and deploy monitoring to those resources. For more information, see: Workflow monitor signal maps
- Overview - The overview page allows you to directly monitor the status of multiple signal maps from one location. Review metrics, logs, and alarms for your workflows. For more information, see: Workflow monitor overview

## Supported services

Workflow monitor supports automatic discovery and signal mapping of resources associated with the following services:

- AWS Elemental MediaLive
- AWS Elemental MediaPackage
- AWS Elemental MediaConnect
- Amazon S3
- Amazon CloudFront

**Topics**

- Configuring workflow monitor
- Using workflow monitor

## Configuring workflow monitor

To setup workflow monitor for the first time; you create the alarm and event templates, and discover signal maps that are used to monitor your media workflows. The following guide contains the steps necessary to setup both Administrator and Operator level IAM roles, create workflow monitor resources, and deploy monitoring to your workflows.

**Topics**

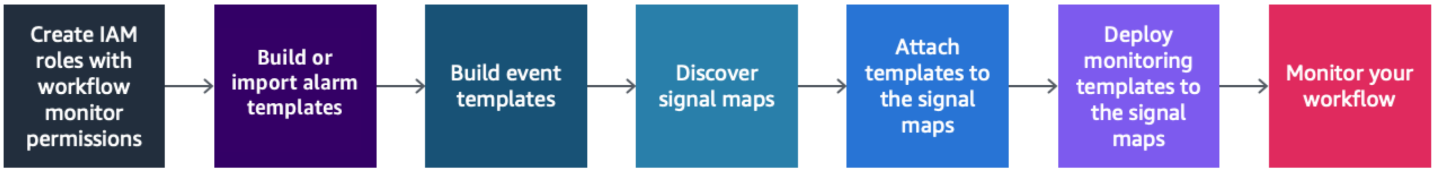
- Getting started with workflow monitor
- Workflow monitor groups and templates
- Workflow monitor signal maps
- Workflow monitor quotas

### Getting started with workflow monitor

The following steps provide a basic overview of using workflow monitor for the first time.

1. Setup workflow monitor IAM permissions for administrator and operator level roles: Workflow monitor IAM policies
2. Build alarm templates or import predefined templates created by AWS: CloudWatch alarms

3. Build notification events that will be delivered by EventBridge: EventBridge rules
4. Discover signal maps using your existing AWS Elemental resources: Signal maps
5. Attach the alarm templates and notification rules to your signal map: Attach templates
6. Deploy the templates to begin monitoring the signal map: Deploy monitoring
7. Monitor and review your workflow monitor resources using the overview section of the AWS console: Overview


Create IAM roles with workflow monitor permissions
Build or import alarm templates
Build event templates
Discover signal maps
Attach templates to the signal maps
Deploy monitoring templates to the signal maps
Monitor your workflow

### Workflow monitor IAM policies

Workflow monitor interacts with multiple AWS services to create signal maps, build CloudWatch and EventBridge resources, and AWS CloudFormation templates. Because workflow monitor interacts with a wide range of services, specific AWS Identity and Access Management (IAM) policies must be assigned for these services. The following examples indicate the necessary IAM policies for both administrator and operator IAM roles.

### Administrator IAM policy

The following example policy is for an administrator-level workflow monitor IAM policy. This role allows for the creation and management of workflow monitor resources and the supported service resources that interact with workflow monitor.

```
                      {
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "cloudwatch:List*",
      "cloudwatch:Describe*",
      "cloudwatch:Get*",
      "cloudwatch:PutAnomalyDetector",
      "cloudwatch:PutMetricData",
      "cloudwatch:PutMetricAlarm",
      "cloudwatch:PutCompositeAlarm",
      "cloudwatch:PutDashboard",
      "cloudwatch:DeleteAlarms",
      "cloudwatch:DeleteAnomalyDetector",
      "cloudwatch:DeleteDashboards",
      "cloudwatch:TagResource",
      "cloudwatch:UntagResource"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "cloudformation:List*",
      "cloudformation:Describe*",
      "cloudformation:CreateStack",
      "cloudformation:UpdateStack",
      "cloudformation:DeleteStack",
      "cloudformation:TagResource",
      "cloudformation:UntagResource"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "cloudfront:List*",
      "cloudfront:Get*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeNetworkInterfaces"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "events:List*",
      "events:Describe*",
      "events:CreateEventBus",
      "events:PutRule",
      "events:PutTargets",
      "events:EnableRule",
      "events:DisableRule",
      "events:DeleteRule",
      "events:RemoveTargets",
      "events:TagResource",
      "events:UntagResource"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "logs:Describe*",
      "logs:Get*",
      "logs:TagLogGroup",
      "logs:TagResource",
      "logs:UntagLogGroup",
      "logs:UntagResource"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "mediaconnect:List*",
      "mediaconnect:Describe*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "medialive:*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "mediapackage:List*",
      "mediapackage:Describe*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "mediapackagev2:List*",
      "mediapackagev2:Get*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "mediapackage-vod:List*",
      "mediapackage-vod:Describe*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "mediatailor:List*",
      "mediatailor:Describe*",
      "mediatailor:Get*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "resource-groups:ListGroups",
      "resource-groups:GetGroup",
      "resource-groups:GetTags",
      "resource-groups:GetGroupQuery",
      "resource-groups:GetGroupConfiguration",
      "resource-groups:CreateGroup",
      "resource-groups:UngroupResources",
      "resource-groups:GroupResources",
      "resource-groups:DeleteGroup",
      "resource-groups:UpdateGroupQuery",
      "resource-groups:UpdateGroup",
      "resource-groups:Tag",
      "resource-groups:Untag"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:*"
    ],
    "Resource": "arn:aws:s3:::workflow-monitor-templates*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "sns:TagResource",
      "sns:UntagResource"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "tag:Get*",
      "tag:Describe*",
      "tag:TagResources",
      "tag:UntagResources"
    ],
    "Resource": "*"
  }
]
}
```

### Operator IAM policy

The following example policy is for an operator-level workflow monitor IAM policy. This role allows for limited and read-only access to the workflow monitor resources and the supported service resources that interact with workflow monitor.

```
                    {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:List*",
        "cloudwatch:Describe*",
        "cloudwatch:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:List*",
        "cloudformation:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudfront:List*",
        "cloudfront:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeNetworkInterfaces"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:List*",
        "events:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:Describe*",
        "logs:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediaconnect:List*",
        "mediaconnect:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "medialive:List*",
        "medialive:Get*",
        "medialive:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediapackage:List*",
        "mediapackage:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediapackagev2:List*",
        "mediapackagev2:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediapackage-vod:List*",
        "mediapackage-vod:Describe*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "mediatailor:List*",
        "mediatailor:Describe*",
        "mediatailor:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": "arn:aws:s3:::workflow-monitor-templates*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "tag:Get*",
        "tag:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

## Workflow monitor groups and templates

Before you can deploy workflow monitoring to a signal map, you must create the groups and templates for CloudWatch alarms and EventBridge notifications. The CloudWatch templates define what scenarios and thresholds will be used to trigger the alarms. The EventBridge templates will determine how these alarms are reported to you.

If you only want mappings of your connected resources and do not want to use the monitoring template capabilities of workflow monitor, signal maps can be used without CloudWatch and EventBridge templates. For more information about using signal maps, see: Signal maps

**Topics**

- CloudWatch alarm groups and templates
- EventBridge rule groups and templates

### CloudWatch alarm groups and templates

Workflow monitor alarms allow you to use existing CloudWatch metrics as the foundation of alarms for your signal maps. You can create an alarm template group to sort and classify the types of alarming that is important to your workflow. Within each alarm template group, you create alarm templates with specific CloudWatch metrics and parameters that you want to monitor. You can create your own alarm templates or import recommended alarm templates created by AWS. After creating an alarm template group and alarm templates within that group, you can attach one or more of these alarm template groups to a signal map.

You must create an alarm template group first. After you have created an alarm template group, you can create your own templates or use recommended templates created by AWS. If you want to create your own alarm templates, continue on this page. For more information about importing recommended templates, see: Recommended templates

This section covers the creation of CloudWatch alarms using workflow monitor. For more information about how the CloudWatch service handles alarms and details of the alarm components, see: Using CloudWatch alarms in the *Amazon CloudWatch User Guide*

#### Creating alarm template groups

**To create an alarm template group**

1. From the workflow monitor console's navigation pane, select **CloudWatch alarm templates**.
2. Select **Create alarm template group**.
3. Give the alarm template group a unique **Group name** and optional **Description**.
4. Select **Create**, You will be taken to the newly created alarm template group's details page.

#### Creating alarm templates

**To create an alarm template**

1. From the alarm template group's details page, select **Create alarm template**.
2. Give the alarm template a unique **Template name** and optional **Description**.

3. In the **Choose metric** section:

    1. Select a **Target Resource Type**. The target resource type is a resource for the respective service, such as a channel for MediaLive and MediaPackage or a flow for MediaConnect.
    2. Select a **Metric Name**. This is the CloudWatch metric that acts as the foundation for the alarm. The list of metrics will change depending on the selected **Target Resource Type**.

4. In the **Alarm settings** section:

    > **Note**
    >
    > For more information about how the CloudWatch service handles alarms and details of the alarm components, see: Using CloudWatch alarms in the *Amazon CloudWatch User Guide*

    1. Select the **Statistic**. This is a value such as a **Sum** or an **Average** that will be used to monitor the metric.
    2. Select the **Comparison Operator**. This field references the **Threshold** that you set in the next step.
    3. Set a **Threshold**. This is a numeric value that the **Comparison Operator** uses to determine greater than, less than, or equal to status.
    4. Set a **Period**. This is a time value, in seconds. The **Period** is the length of time that the **Statistic**, **Comparison Operator**, and **Threshold** interact to determine if the alarm gets triggered.
    5. Set the **Datapoints**. This value determines how many datapoints are needed to trigger the alarm.
    6. Select how to **Treat Missing Data**. This selection determines how this alarm reacts to missing data.

5. Select **Create** to complete the process.

An example of a completed alarm template could have the following parameters: A MediaConnect flow **Target Resource Type** is monitored for the Disconnections **Metric Name**. The **Statistic** value is set to Sum with a **Comparison Operator** of "greater than or equal to" and a **Threshold** of 10. The **Period** is set to 60 seconds, and only requires 1 out of 1 **Datapoints**. **Treat Missing Data** is set to "ignore."

The result of these settings is: workflow monitor will monitor for disconnections on the flow. If 10 or more disconnections occur within 60 seconds, the alarm will be triggered. 10 or more disconnections in 60 seconds only needs to happen one time for the alarm to be triggered.

### Recommended alarm templates

Workflow monitor's recommended templates are a curated selection of AWS Elemental service metrics with predefined alarm settings appropriate for the metric. If you do not want to create customized alarm templates, recommended templates provide you with best-practice monitoring templates that are created by AWS.

Workflow monitor contains recommended template groups for each supported service. These groups are designed to apply best-practice monitoring to specific types of workflows. Each template group contains a curated selection of alarms configured from service-specific metrics. For example, a recommended template group for a MediaLive multiplex workflow will have a different set of preconfigured metrics than a MediaConnect CDI workflow.

**To use recommended alarm templates**

1. Follow the steps to create an alarm template group, or select an existing one.
2. In the **Alarm templates** section, select **Import**. You will need to import the AWS recommended templates into your template group.
3. Use the **CloudWatch alarm template groups** dropdown to select an AWS recommended group. These groups contain curated alarms for specific services.
4. Select the templates to import using the check boxes. Each template will list its metrics, preconfigured monitoring values, and provide a description of the metric. When you are done selecting templates, select the **Add** button.
5. The selected templates will move to the **Alarm template(s) to import** section. Review your choices and select **Import**.
6. After the import is complete, the selected templates will be added to the template group. If you want to add more templates, repeat the import process.
7. Imported templates can be customized after import. Alarm settings can be modified to fit your alarming needs.

### EventBridge rule groups and templates

CloudWatch uses Amazon EventBridge rules to send notifications. You can send notifications based on event templates you create. You begin by creating an event template group. In that event template group, you create event templates that determine what conditions create a notification and who is notified.

This section covers the creation of EventBridge rules using workflow monitor. For more information about how the EventBridge service uses rules, see: EventBridge rules in the *Amazon EventBridge User Guide*

#### Creating event template groups

#### To create an event template group

1. From the workflow monitor console's navigation pane, select **EventBridge rule templates**.
2. Select **Create event template group**.
3. Give the alarm template group a unique **Group name** and optional **Description**.
4. Select **Create**, You will be taken to the newly created alarm template group's details page.

#### Creating event templates

#### To create an event template

1. From the event template group's details page, select **Create event template**.
2. Give the event template a unique **Template name** and optional **Description**.
3. In the **Rule settings** section:
    1. Select an **Event type**. When selecting an event type, you can choose between several events created by AWS or select **Signal map active alarm** to use an alarm created by an alarm template.
    2. Select a **Target service**. This determines how you would like to be notified of this event. You can select Amazon Simple Notification Service or CloudWatch logs.
    3. After selecting a target service, select a **Target**. This will be a Amazon SNS topic or a CloudWatch log group, depending on your target service selection.
4. Select **Create** to complete the process.

## Workflow monitor signal maps

Signal maps are visual mappings of AWS resources in your media workflow. You can use workflow monitor to start the signal map discovery on any of the supported resource types. During the discovery process, workflow monitor will automatically and recursively map all connected AWS resources. After the signal map has been created, you can use the workflow monitor console to do things like deploy monitoring templates, view metrics, and view details of the mapped resources.

**Topics**

### Creating signal maps

#### To create a signal map

1. From the workflow monitor console's navigation pane, select **Signal maps**.
2. Select **Create signal map**.
3. Give the signal map a **Name** and **Description**.
4. In the **Discover new signal map** section, resources in the current account and selected region are displayed. Select a resource to begin signal map discovery. The selected resource will be the starting point for discovery.
5. Select **Create**. Allow a few moments for the discovery process to complete. After the process is complete, you will be presented with the new signal map.

> **Note**
>
> Previews generated in the workflow monitor signal map for AWS Elemental MediaPackage channels are delivered from the MediaPackage Origin Endpoint and will incur Data Transfer Out charges. For pricing, see: MediaPackage pricing.

### Viewing signal maps

#### Signal map views

After selecting a signal map, you have two views that can be used to monitor or configure the signal map. **Monitor signal map** and **Configure signal map** is a context-sensitive button found in the upper-right of the signal map console section.

If you select the signal map using the **Signal maps** section of the navigation pane, your signal map will be displayed in the configuration view. The configuration view allows you to make changes to the template groups attached to this signal map, deploy the attached templates, and view the basic details and tags of the signal map.

If you select the signal map using the **Overview** section of the navigation pane, your signal map will be displayed in monitoring view. The monitoring view displays the CloudWatch alarms, EventBridge rules, alerts, logs, and metrics for this signal map.

The view can be changed at any time by selecting the **Monitor/Configure signal map** button in the upper-right. The configuration view requires administrator-level IAM permissions. Required IAM permissions can be viewed here: Workflow monitor IAM policies

#### Navigating the signal map

A signal map will contain nodes for every supported AWS resource discovered by workflow monitor. Certain resources, such as MediaLive channels and MediaPackage endpoints can display thumbnail previews of the content, if thumbnail previews are available.

Selecting a resource node, and selecting **View selected resource details** from the **Actions** dropdown menu will take you to the associated service's details page. For example, selecting a MediaLive channel and selecting **View selected resource details** will open the MediaLive console's details page for that channel.

Selecting a resource node will filter the list of active alarms to only that node. If you select the resource's **Target ARN** in the active alarm, you will be taken to the associated service's details page, with the selected resource open.

### Attaching alarm and event templates to your signal map

After you have created alarm and event templates, you need to attach these to a signal map. Any of the alarm and event templates you have created can be attached to any discovered signal maps.

### To attach alarm and event templates to your signal map

1. From the workflow monitor console's navigation pane, select **Signal maps** and select the signal map you want to work with.
2. In the upper-right of the signal map page, in the **CloudWatch alarm template groups** tab, select **Attach CloudWatch alarm template groups**.
   1. In the new section that opens, choose all of the alarm template groups that you want to apply to this signal map, then select **Add**. This will cause the selected alarm template groups to move to the **Attached CloudWatch alarm template groups** section.
   2. Selecting **Save** will save your changes and return you to the signal map page.
3. At the right of the signal map page, select the **EventBridge rule template groups** tab then select **Attach EventBridge rule template groups**.
   1. In the new section that opens, choose all of the event template groups that you want to apply to this signal map, then select **Add**. This will cause the selected rule template groups to move to the **Attached EventBridge rule template groups** section.
   2. Selecting **Save** will save your changes and return you to the signal map page.
4. You have assigned CloudWatch alarm and EventBridge rule templates to the signal map, but the monitoring is not yet deployed. The next section will cover the deployment of the monitoring resources.

## Deploying templates to your signal map

After you have attached the alarm and event templates to your signal map, you must deploy the monitoring. Until the deployment is complete, the monitoring of your signal map will not be active.

Workflow monitor will only deploy alarms that are relevant to the selected signal map. For example, the attached alarm template group might contain alarms for multiple services, such as MediaLive, MediaPackage, and MediaConnect. If the selected signal map only contains MediaLive resources, no MediaPackage or MediaConnect alarms will be deployed.

### To deploy the monitoring templates

1. After attaching alarm and event template groups to your signal map and saving your changes, select **Deploy monitor** in the **Actions** dropdown menu.

2. You will be asked to confirm the deployment and presented with the number of CloudWatch and EventBridge resources that will be created. If you would like to proceed, select **Deploy**.

> **Note**
>
> There is no direct cost for using workflow monitor. However, there are costs associated with the resources created and used to monitor your workflow.
> When monitoring is deployed, Amazon CloudWatch and Amazon EventBridge resources are created. When using the AWS Management Console, prior to deploying monitoring to a signal map, you will be notified of how many resources will be created. For more information about pricing, see: CloudWatch pricing and EventBridge pricing.
> Workflow monitor uses AWS CloudFormation templates to deploy the CloudWatch and EventBridge resources. These templates are stored in a standard class Amazon Simple Storage Service bucket that is created on your behalf, by workflow monitor, during the deployment process and will incur object storage and recall charges. For more information about pricing, see: Amazon S3 pricing.

3. The status of the deployment is displayed next to the name of the signal map. The deployment status is also visible in the **Stacks** section of the AWS CloudFormation console. After a few moments of resource creation and deployment, your signal map monitoring will begin.

### Updating signal maps and underlying resources

If a change is made to your workflow, you might need to rediscover the signal map and redeploy monitoring resources. Workflow monitor is a visualization and monitoring tool that does not have the ability to make any changes to your workflow. Signal maps represent a point-in-time visualization of your workflow. In the event that you add, remove, or significantly modify parts of your media workflow, we recommend that you rediscover the signal map. If you have monitoring resources attached to the signal map, we recommend you redeploy monitoring after the rediscovery process.

### To rediscover a signal map

1. From the workflow monitor console's navigation pane, select **Signal maps** and select the signal map you want to work with.
2. Verify that you are in the **Configure signal map** view. For more information about changing views, see: View signal maps

3. In the upper-right of the signal map page, select the **Actions** dropdown menu. Select **Rediscover**.
4. You will be presented with the rediscovery screen. Select a resource that is a part of the workflow you are rediscovering. Select the **Rediscover** button.
5. The signal map will be rebuilt according to the current workflow. If you need to redeploy monitoring resources, stay on this signal map's page. Any previously attached monitoring templates will remain attached, but will need to be redeployed.

**To redeploy monitoring templates after a signal map rediscovery**

1. After the rediscovery, you will be directed to the updated signal map. To redeploy the monitoring templates, select **Deploy monitor** from the **Actions** dropdown menu.
2. You will be asked to confirm the deployment and presented with the number of any CloudWatch and EventBridge resources that will be created. If you would like to proceed, select **Deploy**.
3. The status of the deployment is displayed next to the name of the signal map. After a few moments of resource creation and deployment, your signal map monitoring will begin.

### Deleting signal maps

If you not longer need a signal map, it can be deleted. If you have monitoring templates deployed on the signal map, the deletion process will ask you to delete any CloudWatch and EventBridge resources that have been deployed to this signal map. Deleting the deployed resources does not affect the templates that created them. This resource deletion is to ensure that you do not have CloudWatch and EventBridge resources that are deployed but not used.

**To delete a signal map**

1. From the workflow monitor console's navigation pane, select **Signal maps** and select the radio button next to the signal map you want to delete.
2. Select the **Delete** button. You will be asked to confirm the deletion of the monitoring resources. Select **Delete** to begin the monitoring resource deletion process.
3. The **Monitor deployment** column will display the current status. When the status has changed to **DELETE_COMPLETE**, select the **Delete** button again.

4. You will be asked to confirm deletion of the signal map. Select **Delete** to proceed and delete the signal map.

## Workflow monitor quotas

The following section contains quota for workflow monitor resources. Each quota is on a "per account" basis. You cannot exceed the following quotas on a single AWS account. These quotas cannot be increased.

**Quotas**

| Resource type | Quota |
|---|---|
| CloudWatch alarm template groups | 20 |
| CloudWatch alarm templates | 200 |
| EventBridge rule template groups | 20 |
| EventBridge rule templates | 200 |
| Signal maps | 30 |
| Signal maps: resource nodes in a single signal map | 50 |
| Signal maps: CloudWatch alarm template groups attached to a single signal map | 5 |
| Signal maps: EventBridge rule template groups attached to a single signal map | 5 |

# Using workflow monitor

Use the **overview** and **signal maps** sections of the workflow monitor console to review the current status of the workflows and any associated alarms, metrics, and logs.

**Topics**

- Workflow monitor overview

- Overview logs and metrics
- Using workflow monitor signal maps

## Workflow monitor overview

The **Overview** section of the workflow monitor console is a dashboard that provides at-a-glance information about your signal maps. In the overview section, you can see the current state of each signal map's monitoring, as well as CloudWatch metrics and any associated CloudWatch logs. You can select any signal map to be taken to that signal maps console page.

### Overview filtering

Using the **Search** bar in the overview section, you can filter the list of signal maps using context sensitive constraints. After selecting the search bar, you will be presented with a list of **Properties** to filter by. Selecting a property will present **Operators** such as Equals, Contains, Does not equal, and Does not contain. Selecting an operator will create a list of resources from the selected property type. Selecting one of these resources will cause the signal map list to only display signal maps that fit the constraint you defined.

## Overview logs and metrics

To view CloudWatch metrics and logs for a signal map, select the radio button next to the name of the signal map. A tabbed interface for both metrics and logs will appear beneath the signal map list.

### CloudWatch Metrics

CloudWatch metrics for the selected signal map will be context-sensitive and only display metrics associated with the services used in that signal maps workflow. You can use the on-screen metrics tools to customize the displayed metric periods and time ranges.

### CloudWatch Logs

If you associated a CloudWatch log group with the signal map, that group will be displayed here.

## Using workflow monitor signal maps

From the **overview** section of the console, you can select a specific signal map to view more information about that signal map and its attached monitoring resources.

After selecting a signal map, you will be presented with the signal map and a number of tabbed section containing more information:

- CloudWatch alarms
- EventBridge rules
- AWS Elemental alerts
- Metrics
- Logs
- Basic details

### Navigating the signal map

A signal map will contain nodes for every supported AWS resource discovered by workflow monitor. Certain resources, such as MediaLive channels and MediaPackage endpoints can display thumbnail previews of the content, if thumbnail previews are available.

Selecting a resource node, and selecting **View selected resource details** from the **Actions** dropdown menu will take you to the associated service's details page. For example, selecting a MediaLive channel and selecting **View selected resource details** will open the MediaLive console's details page for that channel.

Selecting a resource node will filter the list of active alarms to only that node. If you select the resource's **Target ARN** in the active alarm, you will be taken to the associated service's details page, with the selected resource open.

# MediaConnect flow maintenance

AWS Elemental MediaConnect routinely performs maintenance on underlying systems for security, reliability, and operational performance. The maintenance activities include actions such as patching the operating system, updating drivers, or installing software and patches.

> **Note**
>
> As part of the maintenance process, your flow must be restarted.

You can select the day and time that maintenance events occur. This is called a *maintenance window* and is used every time a maintenance event is required. If you need to change the day and time, you can edit the maintenance window.

When maintenance is required for your flow, AWS will assign your flow a **Required by** date. If you do not have a maintenance window configured for the flow, visit Setting maintenance windows. You can view the flows that require maintenance on the MediaConnect console or by using the AWS CLI, visit Viewing flows that require maintenance. When a **Required by** date has been assigned to your flow, you can select a specific date for that maintenance to occur. The selected **Maintenance date** will only apply to the next maintenance event.

If you do not configure a maintenance window, AWS selects a maintenance window for you —automatically. We recommend that you set a maintenance window for each flow and allow MediaConnect to perform the restart automatically during that window. Allowing MediaConnect to perform the restart results in less downtime for your flow. If a flow requires maintenance and you choose to manually restart the flow, the status of that flow's maintenance will change to **Canceled**. The manually restarted flow will still apply the required updates, but you will not receive the **Completed successfully** status. Since you performed the restart manually, the maintenance is considered **Canceled** because MediaConnect no longer requires updates for that flow.

The duration of the maintenance window is two hours.

> **Important**
>
> The two hour window duration does not mean the flow will be affected for two hours. The flow will perform a normal stop and start at some point within the two hour window.

Example: If you configure a flow's maintenance window **Start hour** to be 02:00, the flow will restart at some point between 02:00 and 04:00.

In the event that maintenance does not occur at the scheduled date and time, MediaConnect will reschedule it to occur in the following week’s maintenance window, or automatically set a new window if you don't have one configured.

**Topics**

- Viewing flows that require maintenance
- Setting maintenance windows

# Viewing flows that require maintenance

You can view flows that require maintenance in the MediaConnect console or by using the AWS CLI.

> **Note**
>
> If your flow does not have a **Required by date** (console) or a **MaintenanceDeadline** (AWS CLI), maintenance is not currently required for that flow.

### To view the flows that require maintenance (console)

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. In the navigation pane, choose **Flows**.
3. In the **Maintenance window** column, you can view the **Required by date**. Alternatively, you can view the **Required by date** on an individual flows **Details** page.
4. All listed flows must be restarted by the date shown.

### To view the flows that require maintenance (AWS CLI)

- In the AWS CLI, you can use the `list-flows` command to view all flows and their maintenance statuses. Additionally, you can view a specific flows maintenance status by using the `describe-flow` command:

  ```
  aws mediaconnect list-flows
  ```

or

```
aws mediaconnect describe-flow --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame
```

The following example shows the return value of `list-flows`. The return value for `describe-flow` uses a similar structure.

In this example, the Flow named *BasketballGame* has a **MaintenanceDay** and **MaintenanceStartHour** set for recurring maintenance. The Flow named *AwardsShow* has the **MaintenanceDay** and **MaintenanceStartHour** set, but also a **MaintenanceDeadline**. The **MaintenanceDeadline** is the required due date for maintenance restarts on this flow. The *AwardsShow* flow has also scheduled a specific date for the maintenance restarts to occur, seen in the **MaintenanceScheduledDate** value. The **MaintenanceScheduledDate** must occur before the **MaintenanceDeadline**:

```
{
    "Flows": [
        {
            "AvailabilityZone": "us-west-2d",
            "Description": "Example flow description",
            "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame",
            "Name": "BasketballGame",
            "SourceType": "OWNED",
            "Status": "STANDBY",
            "Maintenance": {
                "MaintenanceDay": "Monday",
                "MaintenanceStartHour": "08:00"}
        },
        {
            "AvailabilityZone": "us-west-2b",
            "Description": "Example flow description",
            "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:2-3aBC45dEF67hiJ8k-2AbC34DE5fGa6:AwardsShow",
            "Name": "AwardsShow",
            "SourceType": "OWNED",
            "Status": "ACTIVE",
            "Maintenance": {
                "MaintenanceDay": "Saturday",
                "MaintenanceDeadline": "2021-10-25T22:15:56Z",
                "MaintenanceScheduledDate": "2021-10-23",
                "MaintenanceStartHour": "23:00"}
         }
    ]
}
```

# Setting maintenance windows

You can select the day and time that maintenance events occur. This is called a *maintenance window*. These windows help minimize maintenance impact on your production.

A maintenance window is used every time a maintenance event is required. You can set a maintenance window while creating a flow, or add the window to an existing flow. To change the day and time of a maintenance window, you can use the MediaConnect console or the AWS CLI. Also, if maintenance is required, you can set a specific date for the maintenance to occur. The date you select must be before the required maintenance date.

If you don't set a maintenance window, MediaConnect restarts the flows for you. We recommend that you set a maintenance window for each flow that requires maintenance.

**To set a maintenance window (console)**

1. Open the MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. In the navigation pane, choose **Flows**. When a flow requires maintenance, it will display a **Required by** date under the **Maintenance window** column.
3. Select the flow or flows. You can set a unique maintenance window for each flow. Alternatively, you can set maintenance windows in bulk by selecting multiple flows.
4. Under the **Flow actions** drop-down menu, select **Edit flow maintenance window**.
5. - Select the day of the week maintenance will occur in the **Start day** field.
   - Select the time maintenance will occur in the **Start hour** field. Time is presented in UTC.
   - If maintenance is required, you have the option to select a specific date in the **Maintenance window date** field. The selected date must occur before the required maintenance date and time.
   - Select **Update**.
6. You can verify the window by viewing the **Maintenance window** column on the **Flows** dashboard.

### To set a maintenance window (AWS CLI)

1. In the AWS CLI, use the `update-flow` command with the `--maintenance` option. You will also need to use the `--flow-arn` option to specify which flow you are working with.

   The `--maintenance` option accepts the following arguments:

   - `MaintenanceDay`
   - `MaintenanceStartHour`
   - `MaintenanceScheduleDate` - This argument is only accepted when there is a required maintenance date assigned by AWS.

2. Use the following command to update the reoccurring maintenance day and time. The maintenance day and time can be configured at any time, regardless of required maintenance status.

   ```
   aws mediaconnect update-flow --flow-arn arn:aws:mediaconnect:us-
   east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame --
   maintenance MaintenanceDay='Tuesday',MaintenanceStartHour='10:00'
   ```

   The following example shows the return value when only setting the **MaintenanceDay** and **MaintenanceStartHour**:

   ```
   {
       "Flows": [
           {
               "AvailabilityZone": "us-west-2d",
               "Description": "Example flow description",
               "FlowArn": "arn:aws:mediaconnect:us-
   east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame",
               "Name": "BasketballGame",
               "SourceType": "OWNED",
               "Status": "STANDBY",
               "Maintenance": {
                   "MaintenanceDay": "Tuesday",
                   "MaintenanceStartHour": "10:00"}
           }
       ]
   }
   ```

3. Use the following command to set a specific maintenance date, in addition to setting the reoccurring maintenance day and time. The maintenance scheduled date can only be set when AWS requires maintenance on the flow.

   ```
   aws mediaconnect update-flow --flow-arn arn:aws:mediaconnect:us-
   east-1:111122223333:flow:2-3aBC45dEF67hiJ8k-2AbC34DE5fGa6:AwardsShow --maintenance
    MaintenanceDay='Saturday',MaintenanceStartHour='23:00',MaintenanceScheduledDate='2021-10-2
   ```

   The following example shows the return value when setting the **MaintenanceDay**, **MaintenanceStartHour**, and **MaintenanceScheduledDate**:

   ```
   {
       "Flows": [
           {
               "AvailabilityZone": "us-west-2b",
               "Description": "Example flow description",
               "FlowArn": "arn:aws:mediaconnect:us-
   east-1:111122223333:flow:2-3aBC45dEF67hiJ8k-2AbC34DE5fGa6:AwardsShow",
               "Name": "AwardsShow",
               "SourceType": "OWNED",
               "Status": "ACTIVE",
               "Maintenance": {
                   "MaintenanceDay": "Saturday",
                   "MaintenanceDeadline": "2021-10-25T22:15:56Z",
                   "MaintenanceScheduledDate": "2021-10-23",
                   "MaintenanceStartHour": "23:00"}
            }
       ]
   }
   ```

The selected day and time are used for all future recurring maintenance events on that flow. Repeat these steps to add or edit additional maintenance windows. After maintenance is complete, the **Maintenance status** column on the **Flows** dashboard will display **No maintenance required**.

# Best practices for MediaConnect

For the best performance and availability, follow best practices when you configure your AWS Elemental MediaConnect flows.

## Performance

The following best practices describe how to optimize the performance of transport stream flows:

- Ensure you have set up your transport stream flows with an aggregate output bandwidth of up to 400 Mb/s. MediaConnect is designed to work with an aggregate output bandwidth of 400 Mb/s.

  *aggregate output bandwidth* = (bitrate of the source) x (number of outputs)

  For example, if your flow has a source with a bitrate of 80 Mb/s and 5 outputs, the aggregate output bandwidth is 400 Mb/s. Likewise, a flow that has a source with a bitrate of 20 Mb/s and sends content to 20 outputs also has an aggregate output bandwidth of 400 Mb/s.

  > **Note**
  >
  > Because you can specify two destinations for a single ST 2110 JPEG XS output, those outputs should be counted twice in this calculation.

- You can set up transport stream flows with bitrates up to 120 megabits per second (Mb/s) with mezzanine-quality live video.
- You can use up to 20 Fujitsu outputs. In addition to the 20 Fujitsu outputs, you can use up to 30 of any other non-Fujitsu outputs. Aggregate output bandwidth must not exceed 400 Mb/s.

The following best practices describe how to optimize the performance of CDI flows:

- You can use up to 10 outputs for CDI flows. In addition, 4Kp60 CDI flows support 10 ST 2110 JPEG XS outputs, but only 4 CDI outputs.

The following best practices describe how to optimize the performance of Gateways:

- The API can be used to start multiple bridges at one time. If you are starting multiple bridges using the API, we recommend starting no more than 10 at one time. If you need to start more than 10 bridges, use multiple requests.

## Availability

- To minimize packet loss, use Forward Error Correction (FEC) or automatic repeat request (ARQ) based protocols such as the Zixi or RTP-FEC protocol. These protocols are designed to minimize packet loss between the source and destination devices.
- Because packet loss is present on any network, even in fully managed networks such as the AWS Cloud, you should create and manage redundant connections throughout your workflows. In MediaConnect, there are multiple ways to add redundancy to your workflow:
  - Create flows in at least two different Availability Zones.
  - Add a second source to each flow. If there are errors in the stream, MediaConnect can use packets from a redundant source or switch to the redundant source completely.
- We recommend that your organization create a VPC specifically for all AWS Media Services. A single VPC will help to ensure the availability of IP addresses, help in setting up appropriate rules in the security groups, and help to ensure that a network administrator doesn't accidentally delete elastic network interfaces.

## Reliability

- Set up Amazon CloudWatch metrics and alarms to track the health of your source. For information about which metrics to monitor, see Monitoring and tagging.

## Security

- The CIDR block on the flow source should be as precise as possible. Include only the IP addresses that you want to contribute content to your flow. If the CIDR block is too wide, it allows for the possibility of outside parties sending content to your flow.
- When you create a new SRT password to encrypt an SRT output, you must create that password in AWS Secrets Manager. AWS Secrets Manager does not enforce a specific password policy. However, we recommend the following password policy:
  - Minimum password length of 10 characters and a maximum length of 80 characters

- Minimum of three of the following mix of character types: uppercase, lowercase, numbers, and **`! @ # $ % ^ & * ( ) _ + - = [ ] { } | '`** symbols
- Not be identical to your AWS account name or email address

# Quotas in AWS Elemental MediaConnect

The following table describes quotas, formerly referred to as *limits*, in AWS Elemental MediaConnect. For information about quotas that can be changed, see AWS Service Quotas.

| Resource | Default Quota | Comments |
|---|---|---|
| Entitlements | 50 per flow | The maximum number of entitlements that you can grant on a flow.<br>You cannot increase this quota. |
| Flows | 20 per AWS Region | The maximum number of flows that you can create in each AWS Region.<br>You can request a quota increase. |
| Outputs | 50 per transport stream flow<br>10 per CDI flow | The maximum number of outputs that a flow can have.<br>You cannot increase this quota. |
| Sources | 2 per transport stream flow<br>1 per CDI flow | The maximum number of sources that a flow can have.<br>You cannot increase this quota. |
| VPC interfaces | 2 ENA interfaces and 1 EFA interface per flow | The maximum number of VPC interfaces that a flow can have.<br>You cannot increase this quota. |

**Note**

To optimize performance, we recommend that you set up your workflow for an aggregate output bandwidth of 400 Mb/s or less. For more information, see Best practices.

## Limits for API requests

The following table describes the limits for API request frequency in MediaConnect. These limits are not quotas that you can increase. If you exceed these limits, MediaConnect returns an HTTP 429 (`too many requests`) error.

| API method | Limit |
|---|---|
| Frequency of API requests - steady state | 5 requests per second for each account in a Region.<br>This limit is not a quota that you can increase. |
| Frequency of API requests - burst mode<br>Burst mode allows a temporary overrun of the steady state limit.<br>If API requests exceed the burst mode limit, MediaConnect will throttle the limit and return a 429 error.<br>The limit will refill at a rate of 5 requests per second. | 30 requests per second for each account in a Region.<br>This limit is not a quota that you can increase. |

**Note**

If your application exceeds these limits, we recommend that you implement exponential backoff for retries. For more information, see Error Retries and Exponential Backoff in AWS in the *Amazon Web Services General Reference*.

# Reference: Supported media standards

> **Important**
>
> MediaConnect complies with and implements many media industry standards from different organizations. This reference is not intended to be a comprehensive list, but contains highlighted standards from specific organizations.

## Video Services Forum: technical recommendations

AWS Elemental MediaConnect supports *technical recommendations (TR)* from the *Video Services Forum (VSF)* for some features. This reference guide can be used to identify which TRs are supported by MediaConnect. For more information about technical recommendations, visit the VSF website: VSF technical recommendations

**Supported VSF technical recommendations**

| Technical recommendation | Description |
|---|---|
| **TR-06-01**: Reliable Internet Stream Transport (RIST) [Simple Profile] | This technical recommendation is for RIST *Simple Profile* support only. MediaConnect does not support Main, Enhanced, or Scalable Profiles when using RIST. |
| **TR-07**: Transport of JPEG XS Video in MPEG-2 Transport Stream (TS) over IP<br>**Important**<br>TR-07 is automatically invoked when you use a supported protocol and the maximum bitrate is greater than 200 Mbps. | MediaConnect supports JPEG XS transport in MPEG-2 TS over IP with the following requirements and limitations:<br>• As a source:<br>  ◦ Only a redundant RTP or RTP-FEC protocol is supported.<br>  ◦ The maximum source bitrate is 500 Mbps.<br>• As an output:<br>  ◦ The output protocol can be RTP or RTP-FEC. |

| Technical recommendation | Description |
|---|---|
| | • Up to four total outputs can be used, but aggregate bandwidth must not exceed 1250 Mbps. |

| Technical recommendation | Description |
|---|---|
| **TR-08**: Transport of JPEG XS Video in ST 2110-22<br>Note<br>For JPEG XS passthrough flows where the video frames are not encoded by MediaConnect, the video frames are not decoded. As a result, no validation of TR-08 compliance is performed. | MediaConnect supports JPEG XS transport over SMPTE ST 2110-22 with the following requirements and limitations:<br>• A High profile is required. Using the Main profile will not cause errors, but will be ignored by MediaConnect.<br>• An interlace mode of 01 (top-field first) is required for interlaced signals.<br>• A sublevel of either 3 bits-per-pixel or 4 bits-per-pixel is required. The sublevel depends on the level of compression and pixel bit depth you are using.<br>• Video Description Boxes placed in the encoded video frames will reflect compliant values for profile, interlace mode, and sublevel.<br>• Networked Media Open Specification (NMOS) registration is not supported.<br>• Real-time Transport Protocol (RTP) sequential packet transmission mode only.<br>• Codestream packetization mode only. Slice mode is not supported.<br>Supported color space, bit depth, and chroma sampling configurations:<br>• YCbCr 10-bit 4:2:2<br>• RGB 10-bit 4:4:4<br>• RGB 12-bit 4:4:4 |

# SMPTE-2022

MediaConnect supports many SMPTE (Society of Motion Picture and Television Engineers) standards. The following table is specific to SMPTE-2022 and includes a selection of standards. It is not a comprehensive list of all supported SMPTE standards.

**Supported SMPTE-2022 standards**

| Standard | Description |
|---|---|
| **SMPTE-2022-7**: Seamless Protection Switching of RTP | • Sources: MediaConnect supports RTP sources that comply with this standard. For more information about source failover, see Source failover<br>• Outputs: RTP and RTP-FEC outputs are compliant with the SMPTE 2022-7 standard. If your downstream receiver supports 2022-7 source merging, RTP and RTP-FEC outputs will be compatible. |

# Additional resources

Learn more about AWS Elemental MediaConnect and other AWS resources.

**Topics**

## AWS Elemental MediaConnect open-source attributions

To view the open-source components used by MediaConnect, download the following file:

- MediaConnectOpenSourceAttributions.zip

## AWS Elemental MediaConnect related information

The following list contains related resources that you'll find useful as you work with AWS Elemental MediaConnect.

- Classes & Workshops – Links to role-based and specialty courses, in addition to self-paced labs to help sharpen your AWS skills and gain practical experience.
- AWS Developer Center – Explore tutorials, download tools, and learn about AWS developer events.
- AWS Developer Tools – Links to developer tools, SDKs, IDE toolkits, and command line tools for developing and managing AWS applications.
- Getting Started Resource Center – Learn how to set up your AWS account, join the AWS community, and launch your first application.
- Hands-On Tutorials – Follow step-by-step tutorials to launch your first application on AWS.
- AWS Whitepapers – Links to a comprehensive list of technical AWS whitepapers, covering topics such as architecture, security, and economics and authored by AWS Solutions Architects or other technical experts.
- AWS Support Center – The hub for creating and managing your AWS Support cases. Also includes links to other helpful resources, such as forums, technical FAQs, service health status, and AWS Trusted Advisor.

- AWS Support – The primary webpage for information about AWS Support, a one-on-one, fast-response support channel to help you build and run applications in the cloud.
- Contact Us – A central contact point for inquiries concerning AWS billing, account, events, abuse, and other issues.
- AWS Site Terms – Detailed information about our copyright and trademark; your account, license, and site access; and other topics.

# Document history for user guide

The following table describes the documentation for this release of AWS Elemental MediaConnect. For notification about updates to this documentation, you can subscribe to an RSS feed.

| Change | Description | Date |
|---|---|---|
| Workflow monitor | Analyze AWS media services and create signal maps, visualizations of the media workflow, between those services. Use the signal maps to generate monitoring alarms and notifications using CloudWatch, EventBridge, and AWS CloudFormation. | April 11, 2024 |
| Updated MediaConnect Gateway operating system recommendation | The recommended OS for MediaConnect Gateway has been updated from RHEL 8 to Ubuntu 20.04. | March 11, 2024 |
| Source stream monitoring: Console | Detailed information about MediaConnect flow source streams can be viewed using source metadata monitoring in the MediaConnect console. Source metadata monitorin g displays media information about the transport stream and its programs. | March 8, 2024 |
| Source stream monitoring: API | Detailed information about MediaConnect flow source streams can be viewed using the source metadata monitoring API. Source | December 22, 2023 |

| | | |
|---|---|---|
| | metadata monitoring displays media information about the transport stream and its programs. | |
| VSF TR-07 support | The supported media standards reference section has been updated to reflect MediaConnect's implement ation of the Video Services Forum's TR-07 (Transport of JPEG XS Video in MPEG-2 Transport Stream over IP). | December 8, 2023 |
| Limits for API requests | The guide has been updated to include limits for API requests per second. | November 2, 2023 |
| AWS Elemental Link UHD devices with MediaConnect | You can now use AWS Elemental Link UHD devices and the Zixi push protocol as a source for MediaConnect flows. | September 11, 2023 |
| MediaConnect media metrics | The user guide has been updated to include new CloudWatch metrics for monitoring the health of the media transmitted using MediaConnect. | September 7, 2023 |
| MediaConnect high resolution metrics | MediaConnect metrics can now be viewed in intervals as short as one second. | June 22, 2023 |

| | | |
|---|---|---|
| Supported media standards reference | This guide has been updated to include a reference list of media industry standards that are supported by MediaConnect. | June 9, 2023 |
| SRT failover | You can now enable source failover and add a second source to flows with SRT (listener or caller) sources. | May 1, 2023 |
| Failover support table | A new table has been added that defines which source protocols can support failover. | May 1, 2023 |
| MediaConnect Gateway metrics | The user guide has been updated to include new CloudWatch metrics for the MediaConnect Gateway feature. | April 13, 2023 |
| AWS managed policy - New policy | The MediaConnectGatewayInstanceRolePolicy has been created. | April 13, 2023 |
| AWS managed policy - New policy | The AWSMediaConnectServicePolicy has been created. | April 13, 2023 |
| AWS Elemental MediaConnect Gateway | A new feature has been released called MediaConnect Gateway. MediaConnect Gateway in an on-premises implementation of MediaConnect. | April 13, 2023 |

| | | |
|---|---|---|
| AWS service-linked role - New role | The AWSServiceRoleForM ediaConnect role has been created. | April 13, 2023 |
| Updated the IAM guidance for MediaConnect | Updated guide to align with the IAM best practices . For more information, see Security best practices in IAM. | February 14, 2023 |
| Health CloudWatch events | New flow, source, and output health monitoring CloudWatc h events have been added to MediaConnect. | February 8, 2023 |
| Color support for CDI protocols | A new table has been added that defines color space, bit depth, and chroma sampling support for CDI protocols. | November 4, 2022 |
| MediaConnect Alerts: stream errors | The user guide has been updated to include informati on about stream error Alerts. | October 27, 2022 |
| SRT caller sources and outputs | You can now use the SRT caller protocol for sources and outputs. | September 19, 2022 |
| Source and Output protocol table | A new table has been added that defines which protocols can be used for sources, outputs, or both. | August 5, 2022 |
| Maintenance CloudWatch metrics | The user guide has been updated to include new CloudWatch metrics for MediaConnect maintenance. | August 1, 2022 |

| | | |
|---|---|---|
| Maintenance CloudWatch event | The user guide has been updated to include a new CloudWatch event for MediaConnect maintenance. | August 1, 2022 |
| SRT password encryption | Documentation for SRT password encryption has been added to the guide. | May 31, 2022 |
| Maintenance windows | You can now schedule maintenance windows for MediaConnect to perform maintenance on your flows. You can schedule the maintenance using the new scheduling tools in the console or API. | March 22, 2022 |
| Fujitsu-QoS sources and outputs | You can now use the Fujitsu-QoS protocol for sources and outputs to transport content to and from Fujitsu devices. | December 20, 2021 |
| Maintenance windows | You can now schedule maintenance windows for MediaConnect to perform maintenance on your flows by creating a support case. | August 31, 2021 |
| Source failover | When you enable source failover, you can now specify one of two sources as the primary source. You can choose between two failover modes to prevent any disruption to the video stream. | June 11, 2021 |

| | | |
|---|---|---|
| CDI workflows | MediaConnect now supports JPEG XS for AWS Cloud Digital Interface (AWS CDI) uncompressed workflows. | May 17, 2021 |
| Listener address | For flows that use listener protocols, you can now easily locate an output's outbound IP address for a private internet. | April 14, 2021 |
| SRT-listener sources and outputs | You can now use the SRT-listener protocol for sources and outputs. | March 16, 2021 |
| Reservations | You can now purchase reservations, which provide a disconted hourly rate in exchange for a commitment to use a specific amount of outbound bandwidth each month over the course of a specified duration. | September 30, 2020 |
| Disabling entitlements | You can now disable an entitlement to temporarily stop streaming content to the subscriber’s flow. When you're ready to reinstate access, you can enable the entitlement. | July 24, 2020 |
| Source health metrics | In the MediaConnect console, you can view Amazon CloudWatch metrics that show the health of the source over a period of time. | May 11, 2020 |

| | | |
|---|---|---|
| VPC outputs | You can now add an output to send content from your AWS Elemental MediaConnect flow to your VPC without going over the public internet. | April 7, 2020 |
| VPC sources | You can now connect your VPC to your AWS Elemental MediaConnect flow and send content to your flow without going over the public internet. | March 31, 2020 |
| Source failover | You can now enable source failover and add a second (redundant) source to your flow. | March 13, 2020 |
| Service quotas (outputs) | You can now add up to 50 outputs to each transport stream flow. | February 7, 2020 |
| Sharing the entitlement data transfer fee with the subscriber | When you grant an entitleme nt, you can now specify the percentage of the entitleme nt data transfer fee that you want the subscriber to be responsible for. | September 16, 2019 |
| RIST sources and outputs | You can now use the RIST protocol for sources and outputs. | September 11, 2019 |
| Zixi pull outputs | You can now add outputs that use the Zixi pull protocol. | July 26, 2019 |
| SPEKE support | You can now encrypt the contents of your entitlements using (SPEKE). | June 25, 2019 |

| | | |
|---|---|---|
| Service quotas (flows) | You can now request an increase to the quota of 20 flows per AWS Region. | March 14, 2019 |
| New service and guide | This is the initial release of the media ingest and transport service, AWS Elemental MediaConnect, and the *AWS Elemental MediaConnect User Guide*. | November 27, 2018 |

**Note**

- The AWS Media Services are not designed or intended for use with applications or in situations requiring fail-safe performance, such as life safety operations, navigation or communication systems, air traffic control, or life support machines in which the unavailability, interruption or failure of the services could lead to death, personal injury, property damage or environmental damage.

# AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.