AWS Elemental MediaConnect: User Guide

Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

What is MediaConnect? .................................................................................................................. 1
   Concepts and terminology ........................................................................................................ 2
   Related services ....................................................................................................................... 4
   Accessing AWS Elemental MediaConnect ............................................................................... 4
   Pricing ..................................................................................................................................... 4
   Regions .................................................................................................................................... 5
Use cases ...................................................................................................................................... 6
   Contribution ............................................................................................................................ 6
   Use case: distribution ............................................................................................................. 8
   Use case: entitlements ............................................................................................................ 8
Setting up ..................................................................................................................................... 10
   Step 1: Sign Up for AWS ........................................................................................................ 10
   Step 2: Create an admin IAM user ......................................................................................... 10
   Step 3: Create non-admin IAM users ..................................................................................... 11
      Step 3a: Create a policy ...................................................................................................... 12
      Step 3b: Create a user group .............................................................................................. 13
      Step 3c: Create users .......................................................................................................... 14
   Step 4. (Optional) Set up encryption ....................................................................................... 14
   Step 5. (Optional) Install the AWS CLI .................................................................................. 15
Getting started ............................................................................................................................. 16
   Prerequisites .......................................................................................................................... 16
   Step 1: Access AWS Elemental MediaConnect ...................................................................... 16
   Step 2: Create a flow ............................................................................................................. 16
   Step 3: Add an output ............................................................................................................ 17
   Step 4: Grant an entitlement .................................................................................................. 17
   Step 5: Share details with your affiliates .............................................................................. 18
   Step 6: Clean up .................................................................................................................... 18
Flows ............................................................................................................................................ 19
   Creating a flow ....................................................................................................................... 19
      Standard source .................................................................................................................. 20
      Entitled source ................................................................................................................... 23
   Viewing a list of flows ............................................................................................................ 24
   Viewing the details of a flow ................................................................................................. 24
   Starting a flow ........................................................................................................................ 26
   Stopping a flow ....................................................................................................................... 26
   Updating a flow ....................................................................................................................... 27
   Managing tags on a flow ......................................................................................................... 27
   Deleting a flow ....................................................................................................................... 28
Sources ........................................................................................................................................ 30
   Source failover ....................................................................................................................... 30
   Adding a source to a flow ....................................................................................................... 31
      Standard source .................................................................................................................. 31
   Updating a source .................................................................................................................. 33
   Removing a source from a flow ............................................................................................. 33
   Confirming the connection of a flow to its source ................................................................. 34
   Source ports .......................................................................................................................... 34
Outputs ........................................................................................................................................ 36
   Adding outputs ....................................................................................................................... 36
   Viewing outputs .................................................................................................................... 39
   Updating outputs .................................................................................................................... 40
   Managing tags on an output ................................................................................................. 41
   Removing outputs .................................................................................................................. 42
   Output destinations ................................................................................................................. 43
Entitlements .................................................................................................................................... 45
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sharing content with other AWS accounts</td>
<td>45</td>
</tr>
<tr>
<td>Granting an entitlement</td>
<td>46</td>
</tr>
<tr>
<td>Updating an entitlement</td>
<td>49</td>
</tr>
<tr>
<td>Managing tags on an entitlement</td>
<td>50</td>
</tr>
<tr>
<td>Revoking an entitlement</td>
<td>51</td>
</tr>
<tr>
<td>Subscribing to content provided by another AWS account</td>
<td>52</td>
</tr>
<tr>
<td>Distributing content</td>
<td>54</td>
</tr>
<tr>
<td>Distributing content across Regions</td>
<td>55</td>
</tr>
<tr>
<td>Distributing content to MediaLive</td>
<td>55</td>
</tr>
<tr>
<td>Distributing content from a MediaLive multiplex</td>
<td>56</td>
</tr>
<tr>
<td>Protocols</td>
<td>57</td>
</tr>
<tr>
<td>Security</td>
<td>58</td>
</tr>
<tr>
<td>Data protection</td>
<td>58</td>
</tr>
<tr>
<td>Static key encryption</td>
<td>59</td>
</tr>
<tr>
<td>SPEKE encryption</td>
<td>62</td>
</tr>
<tr>
<td>Identity and access management</td>
<td>66</td>
</tr>
<tr>
<td>Audience</td>
<td>66</td>
</tr>
<tr>
<td>Authenticating with identities</td>
<td>66</td>
</tr>
<tr>
<td>Managing access using policies</td>
<td>68</td>
</tr>
<tr>
<td>Learn more</td>
<td>69</td>
</tr>
<tr>
<td>How MediaConnect works with IAM</td>
<td>69</td>
</tr>
<tr>
<td>Identity-based policy examples</td>
<td>72</td>
</tr>
<tr>
<td>Resource-based policy examples</td>
<td>74</td>
</tr>
<tr>
<td>Policy examples for secrets in AWS Secrets Manager</td>
<td>77</td>
</tr>
<tr>
<td>Setting up MediaConnect as a trusted service</td>
<td>78</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>80</td>
</tr>
<tr>
<td>Logging and monitoring</td>
<td>81</td>
</tr>
<tr>
<td>Amazon CloudWatch alarms</td>
<td>82</td>
</tr>
<tr>
<td>AWS CloudTrail logs</td>
<td>82</td>
</tr>
<tr>
<td>AWS Trusted Advisor</td>
<td>82</td>
</tr>
<tr>
<td>Compliance validation</td>
<td>82</td>
</tr>
<tr>
<td>Resilience</td>
<td>83</td>
</tr>
<tr>
<td>Infrastructure security</td>
<td>83</td>
</tr>
<tr>
<td>Monitoring</td>
<td>84</td>
</tr>
<tr>
<td>Logging API calls with AWS CloudTrail</td>
<td>84</td>
</tr>
<tr>
<td>AWS Elemental MediaConnect information in CloudTrail</td>
<td>84</td>
</tr>
<tr>
<td>Understanding AWS Elemental MediaConnect log file entries</td>
<td>85</td>
</tr>
<tr>
<td>Monitoring with CloudWatch metrics</td>
<td>86</td>
</tr>
<tr>
<td>Metrics to monitor source health</td>
<td>86</td>
</tr>
<tr>
<td>Metrics to monitor output health</td>
<td>93</td>
</tr>
<tr>
<td>Using metrics to troubleshoot</td>
<td>94</td>
</tr>
<tr>
<td>Monitoring flow health</td>
<td>96</td>
</tr>
<tr>
<td>Tagging resources</td>
<td>96</td>
</tr>
<tr>
<td>Supported resources in AWS Elemental MediaConnect</td>
<td>97</td>
</tr>
<tr>
<td>Tag naming and usage conventions</td>
<td>97</td>
</tr>
<tr>
<td>Managing tags</td>
<td>97</td>
</tr>
<tr>
<td>AWS CLI commands</td>
<td>98</td>
</tr>
<tr>
<td>Best practices</td>
<td>99</td>
</tr>
<tr>
<td>Performance</td>
<td>99</td>
</tr>
<tr>
<td>Availability</td>
<td>99</td>
</tr>
<tr>
<td>Reliability</td>
<td>99</td>
</tr>
<tr>
<td>Security</td>
<td>99</td>
</tr>
<tr>
<td>Quotas</td>
<td>100</td>
</tr>
<tr>
<td>Related information</td>
<td>101</td>
</tr>
<tr>
<td>Document history</td>
<td>102</td>
</tr>
<tr>
<td>AWS Glossary</td>
<td>103</td>
</tr>
</tbody>
</table>
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
• AWS Elemental MediaConnect concepts and terminology (p. 2)
• Related services (p. 4)
• Accessing AWS Elemental MediaConnect (p. 4)
• Pricing for AWS Elemental MediaConnect (p. 4)
• Regions for AWS Elemental MediaConnect (p. 5)

AWS Elemental 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 AWS Elemental 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 AWS Elemental 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.

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 AWS Elemental 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 AWS Elemental 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. Each flow can have up to 50 entitlements.

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 for any outputs or entitlements that you want. AWS Elemental 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. You can also set up entitlements on a flow to allow other AWS accounts to access your content.

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.

Originator account

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

Output

The destination address, protocol, and port that AWS Elemental MediaConnect sends the ingested video to. Each flow can have up to 50 outputs. 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. AWS Elemental 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 AWS Elemental 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.

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 AWS Elemental MediaConnect, these are the resources and their ARN formats:


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 AWS Elemental MediaConnect flow, such as an on-premises encoder. An entitled source comes from an AWS Elemental MediaConnect flow that is owned by another AWS account and has granted an entitlement to your account.
Subscriber account

An AWS account that has 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.

Whitelisting

Allowing a block of Classless Inter-Domain Routing (CIDR) IP addresses to serve as a source to your AWS Elemental 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 MediaConnect (p. 10).

- **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 AWS Elemental MediaConnect

You can access 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 AWS Elemental MediaConnect.

- **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 Elemental MediaConnect API** – If you’re using a programming language that an SDK isn’t available for, see the AWS Elemental MediaConnect API Reference for information about API actions and about how to make API requests.

- **AWS Command Line Interface** – For more information, see the AWS Command Line Interface User Guide.

- **AWS Tools for Windows PowerShell** – For more information, see the AWS Tools for Windows PowerShell User Guide.

Pricing for AWS Elemental MediaConnect

As with other AWS products, there are no contracts or minimum commitments for using AWS Elemental MediaConnect. You are charged a per GB fee when content is processed by the service and a per hour fee for active resources. For more information, see AWS Elemental MediaConnect Pricing.
Regions for AWS Elemental MediaConnect

To reduce data latency in your applications, AWS Elemental MediaConnect offers a regional endpoint to make your requests. To view the list of AWS Regions where MediaConnect is available, see AWS Elemental MediaConnect Regions.
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.

There are three basic use cases for AWS Elemental MediaConnect implementation:

- For **contribution**, use AWS Elemental MediaConnect to ingest content from an on-premises encoder into the AWS Cloud.
- For **distribution**, use AWS Elemental MediaConnect to deliver content to different geographical areas.
- For **entitlements**, use AWS Elemental MediaConnect to share your content with other AWS accounts.

**Topics**

- Use case: contribution (p. 6)
- Use case: distribution (p. 8)
- Use case: entitlements (p. 8)

**Use case: contribution**

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

1. **Primary contribution encoder**
   - Sends content to **AWS Elemental MediaConnect**
   - The output connects to **AWS Elemental MediaLive**

2. **Backup contribution encoder**
   - Sends content to **AWS Elemental MediaConnect**
   - The output connects to **AWS Elemental MediaLive**

Both outputs are located in **Availability Zone (us-west-1a)** and **Availability Zone (us-west-1b)**. The region is **(us-west-1)** in AWS Cloud.
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 AWS Elemental MediaConnect flow.) 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.

This 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 other flows in different AWS Regions. These secondary flows are closer to the receivers, which are located in various cities around the world.

Use case: entitlements

Entitlements allow one AWS account holder to share content 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 AWS Elemental MediaConnect in the same Region as the originator's flow.

This following illustration shows how to share content with another AWS subscriber. The output of the originator’s flow can be used as the source of the subscriber's flow.
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 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” (p. 66).

Topics
- Step 1. Sign Up for AWS (p. 10)
- Step 2. Create an admin IAM user (p. 10)
- Step 3. Create non-admin IAM users (p. 11)
- Step 4. (Optional) Set up encryption (p. 14)
- Step 5. (Optional) Install the AWS CLI (p. 15)

Step 1. Sign Up for AWS

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

To sign up for an AWS account

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.

Step 2. Create an admin IAM user

When you first create an AWS account, you begin with a single 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 do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

In this procedure, you use the AWS account root user to create your first IAM user. You add this IAM user to an Administrators group, to ensure that you have access to all services and their resources in your
account. The next time that you access your AWS account, you should sign in with the credentials for this IAM user.

To create an IAM user with limited permissions, see Step 3. Create non-admin IAM users (p. 11).

**To create an administrator user for yourself and add the user to an administrators group (console)**

1. Use your AWS account email address and password to sign in as the AWS account root user to the IAM console at https://console.aws.amazon.com/iam/.
   
   **Note**
   
   We strongly recommend that you adhere to the best practice of using the Administrator IAM user below and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.

2. In the navigation pane, choose **Users** and then choose **Add user**.
3. For **User name**, enter **Administrator**.
4. Select the check box next to **AWS Management Console access**. Then select **Custom password**, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to **User must create a new password at next sign-in** to allow the new user to reset their password after they sign in.
6. Choose **Next: Permissions**.
7. Under **Set permissions**, choose **Add user to group**.
8. Choose **Create group**.
9. In the **Create group** dialog box, for **Group name** enter **Administrators**.
10. Choose **Filter policies**, and then select **AWS managed -job function** to filter the table contents.
11. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.
   
   **Note**
   
   You must activate IAM user and role access to Billing before you can use the AdministratorAccess permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.

12. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
13. Choose **Next: Tags**.
14. (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.
15. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access Management and Example Policies.

---

**Step 3. Create non-admin IAM users**

Users in the Administrators group for an account have access to all AWS services and resources in that account. This section describes how to create users with permissions that are limited to AWS Elemental MediaConnect.

**Topics**
Step 3a: Create a 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.

To create policies

1. Use your AWS account ID or account alias, and the credentials for your admin IAM 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:

   ```json
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Action": [
           "mediaconnect:*"
         ],
         "Effect": "Allow",
         "Resource": "*"
       },
       {
         "Action": [
           "ec2:DescribeAvailabilityZones"
         ],
         "Effect": "Allow",
         "Resource": "*"
       },
       {
         "Action": [
           "cloudwatch:GetMetricStatistics"
         ],
         "Effect": "Allow",
         "Resource": "*"
       },
       {
         "Action": [
           "iam:PassRole"
         ],
         "Effect": "Allow",
         "Resource": "*"
       }
     ]
   }
   
   This policy allows all actions on all resources in AWS Elemental MediaConnect.
   c. Choose **Review policy**.
   d. On the **Review policy** page, for **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:

   ```json
   {
   "Version": "2012-10-17",
   "Statement": [
   {
   "Action": [
   "mediaconnect:List*",
   "mediaconnect:Describe*"
   ],
   "Effect": "Allow",
   "Resource": "*"
   },
   {
   "Action": [
   "ec2:DescribeAvailabilityZones"
   ],
   "Effect": "Allow",
   "Resource": "*"
   },
   {
   "Action": [
   "cloudwatch:GetMetricStatistics"
   ],
   "Effect": "Allow",
   "Resource": "*"
   },
   {
   "Action": [
   "iam:PassRole"
   ],
   "Effect": "Allow",
   "Resource": "*"
   }
   ]
   }
   ```

   c. Choose Review policy.

   d. On the Review policy page, for Name, enter MediaConnectReadOnlyAccess, and then choose Create policy.

   **Step 3b: Create a user group**

   You can create a user group for each policy and assign users to a group rather than attaching individual policies to each user. Using the following procedure, create two user groups: one for the MediaConnectAllAccess policy and one for the MediaConnectReadOnlyAccess policy.

   **To create user groups**

   1. In the navigation pane of the IAM console, choose Groups.

   2. On the Groups page, create an administrator group using the MediaConnectAllAccess policy:


      b. On the Set Group Name page, enter a name for the group, such as MediaConnectAdmins.

      c. Choose Next Step.
d. On the Attach Policy page, for Filter, choose Customer Managed.

e. In the policy list, choose the MediaConnectAllAccess policy that you created in the procedure in Step 3a: Create a Policy (p. 12).

f. Choose Next Step.

g. On the Review page, verify that the correct policies are added to this group, and then choose Create Group.

3. On the Groups page, create a read-only group using the MediaConnectReaders policy:

   
   b. On the Set Group Name page, enter a name for the group, such as MediaConnectReaders.
   
   c. Choose Next Step.
   
   d. On the Attach Policy page, for Filter, choose Customer Managed.
   
   e. In the policy list, choose the MediaConnectReadOnlyAccess policy that you created in the procedure in Step 3a: Create a Policy (p. 12).
   
   f. Choose Next Step.
   
   g. On the Review page, verify that the correct policies are added to this group, and then choose Create Group.

Step 3c: Create users

Create IAM users for the individuals who require access to AWS Elemental MediaConnect, and add each user to the appropriate user group to ensure that they have the right level of permissions. If you have already created users, skip to step 6 to modify the permissions for the users.

To create users

1. In the navigation pane of the IAM console, choose Users, and then choose Add user.

2. For User name, enter the name that the user will use to sign in to AWS Elemental MediaConnect.

3. Select the check box next to AWS Management Console access, select Custom password, and then enter the new user's password in the box. You can optionally select Require password reset to force the user to create a password the next time the user signs in.

4. Choose Next: Permissions.

5. On the Set permissions for user page, choose Add user to group.

6. In the group list, choose the group with the appropriate attached policy. Remember that permissions levels are as follows:

   - The MediaConnectAdmins group has permissions that allow all actions on all resources in AWS Elemental MediaConnect.
   
   - The MediaConnectReaders group has permissions that allow read-only rights for all resources in AWS Elemental MediaConnect.

7. Choose Next: Review to see the list of group memberships that will be added to the new user.

8. When you are ready to proceed, choose Create user.

Step 4. (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 (p. 59)
• Setting up SPEKE encryption using AWS Elemental MediaConnect (p. 63)

Step 5. (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 (p. 16)
- Step 1: Access AWS Elemental MediaConnect (p. 16)
- Step 2: Create a flow (p. 16)
- Step 3: Add an output (p. 17)
- Step 4: Grant an entitlement (p. 17)
- Step 5: Share details with your affiliates (p. 18)
- Step 6: Clean up (p. 18)

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 (p. 10), and then return to this tutorial.

Step 1: Access AWS Elemental MediaConnect

After you set up your AWS account and create IAM users and 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: AwardsNYCSourc
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 Name, enter AwardsNYCSource.
5. For Protocol, choose Zixi push. AWS Elemental MediaConnect will populate the value of the ingest port.
6. For Stream ID, enter ZixiAwardsNYCFeed.
7. For Whitelist CIDR, enter 10.24.34.0/23.
8. 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 Protocol, choose Zixi push. AWS Elemental MediaConnect populates the value of the port.
6. For Stream ID, enter ZixiAwardsOutput.
7. For Address, enter 198.51.100.0.
8. Choose Create 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 one source, a name, and an Availability Zone. After you create a flow, you can add up to 50 outputs to indicate where you want your content to be sent and how you want it transported.

If you want to share your content with another AWS account, grant an entitlement on the flow. A user of the subscriber account can then create a new AWS Elemental 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 that you create on a flow. Each 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.

Topics
- Creating a flow (p. 19)
- Viewing a list of flows (p. 24)
- Viewing the details of a flow (p. 24)
- Starting a flow (p. 26)
- Stopping a flow (p. 26)
- Updating a flow (p. 27)
- Managing tags on a flow (p. 27)
- Deleting a flow (p. 28)

Creating a flow

A flow consists of one source, a name, and an Availability Zone. The ability to choose an Availability Zone allows you to create multiple flows within an AWS Region for redundancy. After you create a flow, you can add up to 50 outputs and up to 50 entitlements.

The method that you use to create a flow is dependent on the type of source that you want to use for your flow:
- **Standard source (p. 20)** – Uses content from any source that is not an entitled source.
- **Entitled source (p. 23)** – Uses content that is owned by another AWS account that has granted an entitlement to your account.

**Note**
If you want to create a flow that uses redundant sources for failover, create the flow with one of the sources. After the flow is created, add the other source (p. 31). 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.
Creating a flow that uses a standard source

A flow consists of one source, a name, and an Availability Zone. The ability to choose an Availability Zone allows you to create multiple flows within an AWS Region for redundancy. After you create a flow, you can add up to 50 outputs and up to 50 entitlements.

A flow uses a standard source when the content comes from anywhere other than another AWS account (entitled source (p. 23)).

**Important**
If the source of your flow requires encryption, set up encryption (p. 59) before you begin this procedure.

To create a flow that uses a standard source (console)

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.

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 enable 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 **Whitelist 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 **Whitelist 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](https://www.rfc-editor.org/rfc/rfc4632).

   **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.

**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**.

   MediaConnect populates the value of the ingest port.

4. For **Whitelist 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](https://www.rfc-editor.org/rfc/rfc4632).

   **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 **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** (p. 61).
   c. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you **created the secret to store the encryption key** (p. 60).
   d. For **Decryption algorithm**, choose the type of encryption that was used to encrypt the source.

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

   **Note**
   The flow doesn't start automatically. You must **start the flow** (p. 26) manually.

8. **Add outputs** (p. 36) to specify where you want MediaConnect to send the content, or **grant entitlements** (p. 46) to allow users of other AWS accounts to subscribe to your content.

**To create a flow (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:

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

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

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

   The following example shows the return value:

   ```json
   {
       "Flow": {
           "EgressIp": "203.0.113.0",
           "AvailabilityZone": "us-east-1d",
           "Name": "AwardsShow",
           "Status": "STANDBY",
           "Source": {
               "Name": "AwardsShowSource",
               "Protocol": "rtp-fec",
               "WhitelistCidr": "10.24.34.0/23"
           }
       }
   }
   ```
Creating a flow that uses an entitled source

A flow consists of one source, a name, and an Availability Zone. The ability to choose an Availability Zone allows you to create multiple flows within an AWS Region for redundancy. After you create a flow, you can add up to 50 outputs and up to 50 entitlements.

To create a flow that uses an entitled source (console)

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.
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 (p. 26) manually.

8. Add outputs (p. 36) to specify where you want MediaConnect to send the content, or grant entitlements (p. 46) to allow users of other AWS accounts to subscribe to your content.

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:

```json
{
    "Flows": [
        {
            "AvailabilityZone": "us-west-2a",
            "Description": "Temporary listed flow description",
            "Name": "BasketballGame",
            "SourceType": "OWNED",
            "Status": "STOPPING"
        },
        {
            "AvailabilityZone": "us-west-2d",
            "Description": "Temporary listed flow description",
            "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)

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 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:

```
```

The following example shows the return value:

```
{
    "Flow": {
        "EgressIp": "54.201.4.39",
        "AvailabilityZone": "us-east-1b",
        "Status": "ACTIVE",
        "Entitlements": [
            {
                "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"
                },
                "Destination": "192.0.2.0"
            },
            {
                "Port": 3025,
                "Name": "LA",
                "Transport": {
                    "SmoothingLatency": 0,
                    "Protocol": "rtp-fec"
                }
            }
        ]
    }
}
```
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)**

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:

  ```bash
  ```

  The following example shows the return value:

  ```json
  {
    "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)**

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:

  ```bash
  ```

  The following example shows the return value:

  ```json
  {
    "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 (p. 27)
- Updating the source (p. 33)
- Updating outputs (p. 40)
- Updating entitlements (p. 49)

---

**Managing tags on a flow**

You can use tags to help you track the billing and organization for your AWS Elemental MediaConnect flows, entitlements, and outputs. 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 and Cost Management User Guide.

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

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)

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)

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)

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.
Deleting a 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:

```bash
```

The following example shows the return value:

```json
{
  "Status": "DELETING"
}
```
Sources in AWS Elemental MediaConnect

Each AWS Elemental MediaConnect flow has at least one source and as many as two sources. When you create a flow, you specify the source name, protocol, whitelist, and ingest port. A source 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)

Topics
- Source failover (p. 30)
- Adding a source to an existing flow (p. 31)
- Updating the source of a flow (p. 33)
- Removing a source from a flow (p. 33)
- Confirming the connection of a flow to its source (p. 34)
- Source ports (p. 34)

Source failover

Source failover is a setup that involves two redundant sources for a flow. Additionally, MediaConnect must receive content from each source at the same time.

When you enable source failover and specify two sources for a flow, MediaConnect treats both sources as the primary. Neither source is considered a backup of the other. The service uses the two sources for failover activity based on SMPTE 2022-7 compliance on the sources.

**Note**

SMPTE 2022-7 is a standard developed by the Society of Motion Picture and Television Engineers (SMPTE) group. The SMPTE 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.

MediaConnect uses two types of failover method behind the scenes:

- If the two sources are SMPTE 2022-7 compliant, MediaConnect uses content from both sources. The service randomly selects one of the sources to start with. If that source is missing a packet, the service 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.
- If the sources are **not** SMPTE 2022-7 compliant, MediaConnect randomly uses one of the sources to provide content for the flow. If that source fails, the service switches to the other source. The service continues switching back and forth between sources as needed. This setup is sometimes referred to as **hot/hot**.
Adding a 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 (p. 30).

You can only add a second source that is a standard source (p. 31). Failover sources that use an entitlement are not supported.

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 (p. 30).

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

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 100–500 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.
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 Standard 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 Ingest 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 Whitelist 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 **Ingest 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 **Whitelist 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.

Zixi push

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

   **Important**
   AWS Elemental MediaConnect populates the value of the ingest port.

2. For **Whitelist 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.
Updating a source

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 (p. 61).
   c. For Secret ARN, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key (p. 60).
   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)

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:

```
```

The following example shows the return value:

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)

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.
Confirming the connection of a flow to its source

On the AWS Elemental MediaConnect console, you can view the status of the connection between a flow and its source.

To confirm the connection of a flow to its source (console)

2. On the Flows page, choose the name of the flow.
3. Choose the Source tab.
4. View the Source health field. There are two possible values:
   - 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 whitelist CIDR and the protocol configuration.

Source ports

Each source on a flow must use a different port. (The exception is sources that use the Zixi protocol. Those sources all use port 2088.) 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. The following table lists which additional ports, if any, the service reserves.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Ports Needed</th>
<th>Ports Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIST</td>
<td>Port and port+1</td>
<td>The port that you specify, plus one additional port. MediaConnect automatically reserves a port that is +1 from the port that you specified. For example, if you specify port 3000 for this output, the service also reserves port 3001.</td>
</tr>
<tr>
<td>RTP</td>
<td>Port</td>
<td>The port that you specify. This is the only port needed for the output.</td>
</tr>
<tr>
<td>RTP-FEC</td>
<td>Port, port+2, and port+4</td>
<td>The port that you specify, plus two additional ports. MediaConnect automatically reserves ports that are +2 and +4 from the port that you specified. For example, if you specify port 2000 for this output, the service also reserves ports 2002 and 2004 for error correction.</td>
</tr>
<tr>
<td>Protocol</td>
<td>Ports Needed</td>
<td>Ports Required</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Zixi push</td>
<td>Port (2088)</td>
<td>MediaConnect automatically uses port 2088 for these outputs.</td>
</tr>
</tbody>
</table>
Outputs in AWS Elemental MediaConnect

Each flow can have up to 50 outputs. 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.

Another way that you can add outputs to a flow is from an entitlement. You can grant an entitlement (p. 46) 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.

Topics
- Adding outputs to a flow (p. 36)
- Viewing a list of outputs of a flow (p. 39)
- Updating outputs on a flow (p. 40)
- Managing tags on an output (p. 41)
- Removing outputs from a flow (p. 42)
- Output destinations (p. 43)

Adding outputs to a flow

You can add up to 50 outputs for each flow. 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 (p. 46). When the subscriber creates a flow using your content as the source, the service creates an output on your flow.

To add an output to a flow (console)

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. Determine which protocol you want to use for the output.
7. 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 (p. 43).
Adding outputs

**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 (p. 43).

**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.

**Zixi pull**

1. For **Protocol**, choose **Zixi pull**.
2. 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 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 identifier that is assigned to the receiver.
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**.
Adding outputs

c. For **Role ARN**, specify the ARN of the role that you created when you set up encryption (p. 61).

d. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key (p. 60).

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 (p. 43).

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 (p. 61).
   d. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key (p. 60).
   e. For **Encryption algorithm**, choose the type of encryption that you want to use to encrypt the source.

8. 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:

```json
{
   "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-outputs` command:

   ```bash
   ```

   The following example shows the return value:

   ```json
   {
     "Outputs": [
       {
         "Name": "RTPOutput",
         "Port": 5020,
         "Transport": {
           "SmoothingLatency": 100,
           "Protocol": "rtp-fec"
         },
         "Destination": "192.0.2.12",
         "Description": "RTP-FEC 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)**

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)**

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

   ```bash
   ```

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

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:

```bash
```

The following example shows the return value:

```json
{
   "Output": {
      "Address": "192.0.2.12",
      "Encryption": {
         "Algorithm": "aes256",
         "KeyType": "static-key",
         "RoleArn": "arn:aws:iam::111122223333:role/AllowMediaConnect",
      },
      "Name": "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, entitlements, and outputs. 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 and Cost Management User Guide.

To add tags to an output (console)

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)

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)

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 (p. 51).

To remove an output from a flow (console)

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:

```bash
```
The following example shows the return value:

```json
{
    "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.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Destination Definition</th>
<th>Ports Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIST</td>
<td>IP address, port, and port+1</td>
<td>The port that you specify, plus one additional port. The service automatically reserves a port that is +1 from the port that you specified. For example, if you specify port 3000 for this output, the service also reserves port 3001.</td>
</tr>
<tr>
<td>RTP</td>
<td>IP address and port</td>
<td>The port that you specify. This is the only port needed for the output.</td>
</tr>
<tr>
<td>RTP-FEC</td>
<td>IP address, port, port+2, and port+4</td>
<td>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. For example, if you specify port 2000 for this output, the service also reserves ports 2002 and 2004 for error correction.</td>
</tr>
<tr>
<td>Zixi pull</td>
<td>Stream ID and remote ID</td>
<td>The service automatically uses port 2077 for these outputs.</td>
</tr>
</tbody>
</table>
### Output destinations

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Destination Definition</th>
<th>Ports Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zixi push</td>
<td>IP address, stream ID, and port</td>
<td>The port that you specify is the only port needed for the output.</td>
</tr>
</tbody>
</table>
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.

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. You can also specify the percentage of the entitlement data transfer fee that you want the subscriber to be responsible for.

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 an AWS Elemental 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.
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 (p. 59) or Secure Packager and Encoder Key Exchange (SPEKE) (p. 63).

To grant an entitlement on a flow (console)

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 he or she creates and starts a flow that is based on this entitlement.

9. 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 (p. 61).
   4. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key (p. 60).
   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 (p. 64).

   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:
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.

10. At the bottom of the page, choose **Grant entitlement**.

11. On the **Entitlements** tab, locate the new entitlement in the list.

12. Make a note of the entitlement ARN.

13. 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

**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:

   ```json
   [
   {
   "Description": "For AnyCompany",
   "Encryption": [
   {
   "Algorithm": "aes128",
   "KeyType": "static-key",
   "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
   }
   ],
   "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:

   ```bash
   ```

   The following example shows the return value:

   ```json
   {
   "Entitlements": [
   ```
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)

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:

```
```
The following example shows the return value:

```json
{
  "Entitlement": {
    "Name": "AnyCompany_Entitlement",
    "Description": "For AnyCompany Affiliate",
    "Encryption": {
      "KeyType": "static-key",
      "Algorithm": "aes128",
      "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
    },
    "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, entitlements, and outputs. 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 and Cost Management User Guide*.

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

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)

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)**

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 and the associated output is removed from your flow.

**To revoke an entitlement (console)**

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:

  ```
  ```

The following example shows the return value:

```json
{
}
```
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.

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 (p. 59), store the encryption key (p. 60) 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)

2. Verify that you are logged in to the same AWS Region that the originator’s flow is in.
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 (p. 61).

c. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key (p. 60).

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 (p. 26) manually.

10. **Add outputs** (p. 36) to specify where you want AWS Elemental MediaConnect to send the content, or **grant entitlements** (p. 46) to allow users of other AWS accounts to subscribe to your content.
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 other 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 (p. 55)
- Distributing content to AWS Elemental MediaLive (p. 55)
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 (p. 19) and add outputs to a flow (p. 36).

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.
   - Ingest port: Enter 2088.
   - Whitelist 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, but in different Availability Zones (such as us-east-1a and us-east-1b). 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 (p. 52).

3. Add outputs (p. 36) to distribute the content.
Protocols in AWS Elemental MediaConnect

AWS Elemental MediaConnect supports the following protocols for incoming (source) and outgoing (output) live video streams:

- **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 AWS Elemental 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.

- **Reliable Internet Stream Transport (RIST)** is a highly available, low-latency protocol that is suitable for long-distance applications. AWS Elemental MediaConnect doesn't support encryption for sources or outputs that use the RIST 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.

- **Real-Time Transport Protocol (RTP)** has wide applicability and takes less bandwidth than RTP-FEC. AWS Elemental MediaConnect doesn't support encryption for sources or outputs that use the RTP protocol.
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 (p. 58)
- Identity and access management for AWS Elemental MediaConnect (p. 66)
- Logging and monitoring (p. 81)
- Compliance validation for AWS Elemental MediaConnect (p. 82)
- Resilience in AWS Elemental MediaConnect (p. 83)
- Infrastructure security in AWS Elemental MediaConnect (p. 83)

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 two 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).

**Note**

Encryption is supported only for entitlements and for sources and outputs that use the Zixi protocol.

**Topics**

- Static key encryption in AWS Elemental MediaConnect (p. 59)
- SPEKE encryption in AWS Elemental MediaConnect (p. 62)

**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 (p. 59)
- Setting up static key encryption using AWS Elemental MediaConnect (p. 59)

**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 (p. 59).

**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 (p. 60) – Store your encryption key as a secret in AWS Secrets Manager.

Step 2 (p. 60) – Create an IAM policy that allows AWS Elemental MediaConnect to read the secret that you stored in AWS Secrets Manager.

Step 3 (p. 61) – 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
Static key encryption is supported only for entitlements, and for sources and outputs that use the Zixi protocol.

Step 1: Store your encryption key in AWS Secrets Manager

To use static key encryption to encrypt your AWS Elemental MediaConnect content, you must create a secret in AWS Secrets Manager to store your encryption key. You also must use the same AWS account to create the secret that you use to create the AWS Elemental MediaConnect resource (source, output, or entitlement) that uses the secret. AWS Elemental MediaConnect does not support cross-account sharing of secrets.

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.
3. On the Store a new secret page, for Select secret type, choose Other type of secrets.
4. For Specify the key/value pairs to be stored in this secret, choose Plaintext.
5. Clear any text in the box and replace it with the encryption key.
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 (p. 60), 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:
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 (p. 77).

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 (p. 60), 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 (p. 60), 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 Relationship**.


   The policy document should now look like this:

   ```json
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "Service": "mediaconnect.amazonaws.com",
         },
         "Action": "sts:AssumeRole",
       },
     ],
   }
   ```

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 (p. 45). 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 (p. 62)
- Setting up SPEKE encryption using AWS Elemental MediaConnect (p. 63)

---

**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.
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 (p. 63).

### 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. (p. 63)** – 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 (p. 64)** – 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. (p. 64)** – 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 (p. 46) in MediaConnect.

**Step 2: Create an IAM policy to allow API Gateway to act as your proxy**

In step 1 (p. 63), 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:

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "execute-api:Invoke"
         ],
         "Resource": [
            "arn:aws:execute-api:us-west-2:111122223333:1abcdefg1/*/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.

5. For Name, enter APIGateway-Proxy-Access.
6. Choose Create policy.

**Step 3: Create an IAM role with a trusted relationship**

In step 2 (p. 64), 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:

   ```json
   {
   "Version": "2012-10-17",
   "Statement": [
   {
   "Effect": "Allow",
   "Principal": {
   "Service": "mediaconnect.amazonaws.com"
   },
   "Action": "sts:AssumeRole"
   }
   ]
   }
   ```

15. Choose **Update Trust Policy**.

---

65
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.

This section provides background and additional information about the setup procedures that you follow to use MediaConnect. See Setting up (p. 10).

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work 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 (p. 80).

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 employees 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 (p. 69).

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 (p. 72).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see The IAM Console and Sign-in Page in the IAM User Guide.

You must be authenticated (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication, or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the AWS Management Console, use your password with your root user email or your IAM user name. You can access AWS programmatically using your root user or IAM user access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don't use AWS tools, you must sign the request yourself. Do this using Signature Version 4, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the AWS General Reference.

Regardless of the authentication method that you use, you might also 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 Using Multi-Factor Authentication (MFA) in AWS in the IAM User Guide.
AWS Account Root User

When you first create an AWS account, you begin with a single 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 do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM users and groups

An IAM user is an identity within your AWS account that has specific permissions for a single person or application. An IAM user can have long-term credentials such as a user name and password or a set of access keys. To learn how to generate access keys, see Managing Access Keys for IAM Users in the IAM User Guide. When you generate access keys for an IAM user, make sure you view and securely save the key pair. You cannot recover the secret access key in the future. Instead, you must generate a new access key pair.

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:

- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.

- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as federated users. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated Users and Roles in the IAM User Guide.

- **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.

- **AWS service access** – A service role is an IAM role that a service assumes to perform actions in your account on your behalf. When you set up some AWS service environments, you must define a role for the service to assume. This service role must include all the permissions that are required for the service to access the AWS resources that it needs. Service roles vary from service to service, but many allow you to choose your permissions as long as you meet the documented requirements for that
service. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. You can create, modify, and delete a service role from within IAM. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see Creating a Role to Delegate Permissions to an AWS Service in the IAM User Guide.

- **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, 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 IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when an entity (root user, IAM user, or IAM role) 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.

An IAM administrator can use policies to specify who has access to AWS resources, and what actions they can perform on those resources. Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

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, role, or group. These policies control what actions that identity 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 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 (p. 69)
- Identity-based policy examples (p. 72)
- Resource-based policy examples (p. 74)
- Policy examples for secrets in AWS Secrets Manager (p. 77)
- Troubleshooting (p. 80)

**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**

- MediaConnect identity-based policies (p. 69)
- MediaConnect resource-based policies (p. 71)
- Authorization based on MediaConnect tags (p. 71)
- MediaConnect IAM roles (p. 71)

**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**

The Action element of an IAM identity-based policy describes the specific action or actions that will be allowed or denied by the policy. Policy actions usually have the same name as the associated AWS API operation. The action is used 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**

The Resource element specifies the object or objects to which the action applies. Statements must include either a Resource or a NotResource element. You specify a resource using an ARN or using the wildcard (*) to indicate that the statement applies to all resources.

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:

```
```

To specify all flows that belong to a specific account, use the wildcard (*):

```
```
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**

MediaConnect doesn't provide any service-specific condition keys, but it does support using some global 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 (p. 72).

**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 are very powerful. They 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 Using AWS Managed Policies** – To start using MediaConnect quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see Get Started Using Permissions With AWS Managed Policies in the IAM User Guide.

- **Grant Least Privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see Grant Least Privilege in the IAM User Guide.

- **Enable MFA for Sensitive Operations** – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see Using Multi-Factor Authentication (MFA) in AWS in the IAM User Guide.

- **Use Policy Conditions for Extra Security** – To the extent that it's practical, define the conditions under which your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see IAM JSON Policy Elements: Condition 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.
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.
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:GetMetricStatistics 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.

```json
{
"Version": "2012-10-17",
"Statement": [
{
"Action": [
"mediaconnect:List*",
"mediaconnect:Describe*"
],
"Effect": "Allow",
"Resource": "*
}
},
{
"Action": [
"ec2:DescribeAvailabilityZones"
],
"Effect": "Allow",
"Resource": "*
}
}
Allow all AWS Elemental MediaConnect actions on a specific resource

Every user of AWS Elemental MediaConnect must have a policy that defines permissions on MediaConnect resources. The IAM policy below provides the following permissions:

- The section for the `mediaconnect:*` action allows all AWS Elemental MediaConnect actions on a specific resource.
- 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 and should apply to all resources.
- The section for the `cloudwatch:GetMetricStatistics` action allows the service to obtain metrics from Amazon CloudWatch. This portion of the policy is required and should apply to all resources.
- The section for the `iam:PassRole` action allows IAM to pass a role to AWS Elemental MediaConnect, allowing 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. It can be applied to all resources, as shown in the example below, or to a specific resource.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "mediaconnect:*"
            ],
            "Effect": "Allow",
            "Resource": [
            ]
        },
        {
            "Action": [
                "ec2:DescribeAvailabilityZones"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "cloudwatch:GetMetricStatistics"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```
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:GetMetricStatistics` 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.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": ["mediaconnect:*"],
            "Effect": "Allow",
            "Resource": "*
        },
        {
            "Action": ["ec2:DescribeAvailabilityZones"],
            "Effect": "Allow",
            "Resource": "*
        },
        {
            "Action": ["cloudwatch:GetMetricStatistics"],
            "Effect": "Allow",
            "Resource": "*
        },
        {
            "Action": ["iam:PassRole"],
            "Effect": "Allow",
            "Resource": "*
        }
    ]
}
```
IAM policy examples for secrets in AWS Secrets Manager

During setup, you create an IAM policy (p. 60) 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 (p. 77)
- Allow read access to all secrets created in a specific Region in AWS Secrets Manager (p. 77)
- Allow read access to all resources in AWS Secrets Manager (p. 78)

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:aes256-7g8H9i"
            ]
        }
    ]
}
```

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:aes256-7g8H9i"
            ]
        }
    ]
}
```
AWS Elemental MediaConnect User Guide
Setting up MediaConnect as a trusted service

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"
            ],
            "Resource": ["*"
        }
    ]
}
```

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. (p. 78)** – Create an IAM policy that governs which actions you want to allow.

**Step 2 (p. 79)** – 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 examples for secrets in AWS Secrets Manager (p. 77)
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 (p. 78), 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 (p. 78).
7. Select the check box next to the name of the policy, and then choose **Next: Review**.
8. 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.
9. For **Role description**, replace the default text with a description that will help you remember the purpose of this role.
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 Relationship**.

The policy document should now look like this:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "mediaconnect.amazonaws.com",
      },
      "Action": "sts:AssumeRole",
    ],
  }
}
```

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.
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 (p. 80)
• I am not authorized to perform iam:PassRole (p. 80)
• I want to view my access keys (p. 80)
• I'm an administrator and want to allow others to access MediaConnect (p. 81)
• I want to allow people outside of my AWS account to access my MediaConnect resources (p. 81)

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 IAM 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 am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the iam:PassRole action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password. Ask that person to update your policies to allow you to pass a role to MediaConnect.

Some AWS services allow you to pass an existing role to that service, instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named marymajor tries to use the console to perform an action in MediaConnect. However, the action requires the service to have permissions granted by a service role. Mary does not have permissions to pass the role to the service.

User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole

In this case, Mary asks her administrator to update her policies to allow her to perform the iam:PassRole action.

I want to view my access keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can’t view your secret access key again. If you lose your secret key, you must create a new access key pair.
Access keys consist of two parts: an access key ID (for example, AKIAIOSFODNN7EXAMPLE) and a secret access key (for example, wJalrXtNfeQXrQy7URtQ8ipQXrQy7URtQ8ip). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

**Important**
Do not provide your access keys to a third party, even to help find your canonical user ID. By doing this, you might give someone permanent access to your account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see Managing Access Keys in the IAM User Guide.

**I'm an administrator and want to allow others to access MediaConnect**

To allow others to access MediaConnect, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in MediaConnect.

To get started right away, see Creating Your First IAM Delegated User and Group in the IAM User Guide.

**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 (p. 69).
- 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 (p. 84).

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 (p. 86).

**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 (p. 84).

**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**

Third-party auditors assess the security and compliance of AWS Elemental MediaConnect as part of multiple AWS compliance programs. These include SOC, PCI, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. 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 MediaConnect is determined by the sensitivity of your data, your company’s compliance objectives, and applicable laws and regulations. AWS provides resources to help:

- **Security and Compliance Quick Start Guides** – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- **Architecting for HIPAA Security and Compliance Whitepaper** – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- **AWS Compliance Resources** – This collection of workbooks and guides might apply to your industry and location.
• **AWS Config** – This AWS 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 that helps you check your compliance with security industry standards and best practices.

---

**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](https://docs.aws.amazon.com/AmazonS3/latest/userguide/AmazonS3uckleatures.html).

---

**Infrastructure security in AWS Elemental MediaConnect**

As a managed service, AWS Elemental MediaConnect is protected by the AWS global network security procedures that are described in the [Amazon Web Services: Overview of Security Processes](https://aws.amazon.com/security/) whitepaper.

You use AWS published API calls to access MediaConnect through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). 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](https://docs.aws.amazon.com/IAM/latest/userguide/using-ses.html) (AWS STS) to generate temporary security credentials to sign requests.
Monitoring 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:

- **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** 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](#).

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**
Understanding AWS Elemental MediaConnect log file entries

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:

```json
{
  "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",
}
```
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 statistics 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. 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.

For AWS Elemental MediaConnect, you might want to watch PacketLossPercent and send an email to yourself when that metric reaches a certain threshold.

To view metrics using the CloudWatch console

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

1. Sign in to the AWS Management Console and open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose Metrics.
3. Under All metrics, choose the AWS/MediaConnect namespace.
4. Choose the metric dimension to view the metrics (for example, choose flow to view metrics per flow).

To view metrics using the AWS CLI

• At a command prompt, use the following command:

```bash
aws cloudwatch list-metrics --namespace "AWS/MediaConnect"
```

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.

Note

Metrics tracked by MediaConnect adhere to the standard as defined by the TR 101 290 spec.
Network metrics

The following table lists network metrics that AWS Elemental MediaConnect sends to CloudWatch.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARQRecovered</td>
<td>The number of dropped packets that were recovered by automatic repeat request (ARQ). This metric applies only to sources that use the RIST protocol or the Zixi protocol.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
<tr>
<td>ARQRequests</td>
<td>The number of retransmitted packets that were requested through automatic repeat request (ARQ) and received. This metric applies only to sources that use the RIST protocol or the Zixi protocol.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
<tr>
<td>Connected</td>
<td>The status of the source. A value of one indicates that the source is connected and a value of zero indicates that the source is disconnected. This metric applies only to sources that use the Zixi protocol.</td>
</tr>
<tr>
<td></td>
<td>Units: None</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
<tr>
<td>Disconnections</td>
<td>The number of times that the source status changed from connected to disconnected.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• Availability Zone • All flows</td>
<td>DroppedPackets: The number of packets that were lost during transit. This value is measured before any error correction takes place. Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions: • Flow ARN • Availability Zone • All flows</td>
</tr>
<tr>
<td></td>
<td>FECPackets: 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 protocol or the Zixi protocol. Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions: • Flow ARN • Availability Zone • All flows</td>
</tr>
<tr>
<td></td>
<td>FECRecovered: 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 protocol or the Zixi protocol. Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions: • Flow ARN • Availability Zone • All flows</td>
</tr>
<tr>
<td></td>
<td>NotRecoveredPackets: The number of packets that were lost during transit and were not recovered by error correction. Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions: • Flow ARN • Availability Zone • All flows</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OverflowPackets</td>
<td>The number of packets that were lost in transit because the video required more buffer than was available.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
<tr>
<td>PacketLossPercent</td>
<td>The percentage of packets that were lost during transit, even if they were recovered.</td>
</tr>
<tr>
<td></td>
<td>Units: Percent</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
<tr>
<td>RecoveredPackets</td>
<td>The number of packets that were lost during transit, but recovered.</td>
</tr>
<tr>
<td></td>
<td>Units: Counter</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
<tr>
<td>RoundTripTime</td>
<td>The amount of time it takes for the source to send a signal and receive an acknowledgment from AWS Elemental MediaConnect. This metric applies only to sources that use the Zixi protocol.</td>
</tr>
<tr>
<td></td>
<td>Units: Milliseconds</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
<tr>
<td>TotalPackets</td>
<td>The total number of packets that were received.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
</tbody>
</table>
## TR 101 290 Priority 1 metrics

The following table lists TR 101 290 Priority 1 metrics that AWS Elemental MediaConnect sends to CloudWatch.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
</table>
| ContinuityCounter | The number of times that a continuity error occurred. This error indicates an incorrect packet order or lost packets. Units: Count. Valid dimensions:  
|               | • Flow ARN  
|               | • Availability Zone  
|               | • 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 and points to the program map tables (PMTs). The decoder needs the PAT to do its job. Units: Count. Valid dimensions:  
|               | • Flow ARN  
|               | • Availability Zone  
|               | • 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 (TS) has been multiplexed and then remultiplexed. Units: Count. Valid dimensions:  
|               | • Flow ARN  
|               | • Availability Zone  
|               | • 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. Units: Count. Valid dimensions:  
|               | • Flow ARN  
|               | • Availability Zone  
|               | • All flows                                                                                                         |
### Metrics to monitor source health

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TSByteError</strong></td>
<td>The number of times that a transport stream (TS) byte error occurred. This error indicates that the sync byte did not appear after the prescribed number of bytes.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
<tr>
<td><strong>TSSyncLoss</strong></td>
<td>The number of times that a TS sync loss error occurred. This error happens after two or more consecutive TS byte errors.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
</tbody>
</table>

### TR 101 290 Priority 2 metrics

The following table lists TR 101 290 Priority 2 metrics that AWS Elemental MediaConnect sends to CloudWatch.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CATError</strong></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
<tr>
<td><strong>CRCError</strong></td>
<td>The number of times that a cyclic redundancy check (CRC) error occurred. This error happens when a CRC determines that data is corrupted.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
</tbody>
</table>
### Metrics to monitor source health

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
</table>
| PCR Accuracy Error| 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. Units: Count<br>Valid dimensions:  
  - Flow ARN  
  - Availability Zone  
  - All flows |
| PCR Error         | 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. Units: Count<br>Valid dimensions:  
  - Flow ARN  
  - Availability Zone  
  - All flows |
| PTS Error         | 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 is scrambled. Units: Count<br>Valid dimensions:  
  - Flow ARN  
  - Availability Zone  
  - All flows |
| Transport Error   | 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. Units: Count<br>Valid dimensions:  
  - Flow ARN  
  - Availability Zone  
  - All flows |
Source metrics

The following table lists source metrics that AWS Elemental MediaConnect sends to CloudWatch.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceBitRate</td>
<td>The bitrate of the incoming (source) video.</td>
</tr>
<tr>
<td></td>
<td>Units: bits per second (b/s)</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectedOutputs</td>
<td>The number of outputs that are currently connected. This metric applies only to outputs that use the Zixi protocol.</td>
</tr>
<tr>
<td></td>
<td>Units: Count</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OutputConnected</td>
<td>The status of the output. A value of one indicates that the output is connected, and a value of zero indicates that the output is disconnected. This metric applies only to outputs that use the Zixi protocol.</td>
</tr>
<tr>
<td></td>
<td>Units: None</td>
</tr>
<tr>
<td></td>
<td>Valid dimensions:</td>
</tr>
<tr>
<td></td>
<td>• Output ARN</td>
</tr>
<tr>
<td></td>
<td>• Flow ARN</td>
</tr>
<tr>
<td></td>
<td>• Availability Zone</td>
</tr>
<tr>
<td></td>
<td>• All flows</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OutputDisconnections</td>
<td>The number of times that the output status changed from connected to disconnected. This metric applies only to outputs that use the Zixi protocol.</td>
</tr>
</tbody>
</table>
### 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 RTP protocol (p. 94)
- Metrics to watch if your source uses the RTP-FEC protocol (p. 94)
- Metrics to watch if your source uses the RIST protocol (p. 95)
- Metrics to watch if your source uses the Zixi push protocol (p. 95)
- Metrics to watch if your source comes from an entitlement (p. 95)

#### 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
- PacketLossPercent
- 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
- PacketLossPercent
- RecoveredPackets
- RoundTripTime
- TotalPackets

---

**Metric | Description**

<table>
<thead>
<tr>
<th>Units: Count</th>
<th>Valid dimensions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Output ARN</td>
<td>• Flow ARN</td>
</tr>
<tr>
<td>• Availability Zone</td>
<td>• All flows</td>
</tr>
</tbody>
</table>
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 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
- PacketLossPercent
- 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
- PacketLossPercent
- RecoveredPackets
- RoundTripTime
- TotalPackets
Monitoring the health of a flow

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:

- 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.

To monitor the health of a flow (console)

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.

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 and Cost Management User Guide.

The following sections provide more information about tags for AWS Elemental MediaConnect.
Supported resources in AWS Elemental MediaConnect

The following resources in AWS Elemental MediaConnect support tagging:

- Flows
- Entitlements
- Outputs

For information about adding and managing tags, see Managing tags (p. 97).

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” (p. 27) in this guide
- the section called “Managing tags on an entitlement” (p. 50) in this guide
- the section called “Managing tags on an output” (p. 41) in this guide
# AWS CLI commands for AWS Elemental MediaConnect

The following table shows the AWS CLI commands that you can use to create or modify flows in AWS Elemental MediaConnect.

<table>
<thead>
<tr>
<th>Command</th>
<th>Applies to</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entitlements</td>
<td>grant-flow-entitlements</td>
<td>Grants entitlements on a flow.</td>
</tr>
<tr>
<td>Entitlements</td>
<td>list-entitlements</td>
<td>Displays a list of all entitlements that have been granted to this account.</td>
</tr>
<tr>
<td>Entitlements</td>
<td>revoke-flow-entitlement</td>
<td>Revokes the specified entitlement from a flow. After an entitlement is revoked, the content becomes unavailable to the subscriber and the associated output is removed.</td>
</tr>
<tr>
<td>Entitlements</td>
<td>update-flow-entitlement</td>
<td>Updates the specified entitlement on the specified flow. You can change an entitlement's description, subscriber account ID, and encryption. If you change the subscriber account ID, the service removes the output that was generated when the original subscriber set up their flow.</td>
</tr>
<tr>
<td>Flows</td>
<td>create-flow</td>
<td>Creates a flow.</td>
</tr>
<tr>
<td>Flows</td>
<td>delete-flow</td>
<td>Deletes a flow. You must stop a flow before you can delete it.</td>
</tr>
<tr>
<td>Flows</td>
<td>describe-flow</td>
<td>Retrieves information about a flow in your account.</td>
</tr>
<tr>
<td>Flows</td>
<td>list-flows</td>
<td>Lists all the flows that are associated with your account.</td>
</tr>
<tr>
<td>Flows</td>
<td>start-flow</td>
<td>Starts a flow.</td>
</tr>
<tr>
<td>Flows</td>
<td>stop-flow</td>
<td>Stops a flow.</td>
</tr>
<tr>
<td>Outputs</td>
<td>add-flow-outputs</td>
<td>Adds outputs to a flow.</td>
</tr>
<tr>
<td>Outputs</td>
<td>remove-flow-output</td>
<td>Removes the specified outputs from a flow.</td>
</tr>
<tr>
<td>Outputs</td>
<td>update-flow-output</td>
<td>Updates the specified output of a flow.</td>
</tr>
<tr>
<td>Source</td>
<td>update-flow-source</td>
<td>Updates the source of a flow.</td>
</tr>
</tbody>
</table>
Best practices for AWS Elemental MediaConnect

For the best performance and availability, follow best practices when you configure your AWS Elemental MediaConnect flows.

Performance

- AWS Elemental MediaConnect is designed to work with mezzanine quality live video and has been tested with bitrates up to 120 megabits per second (Mb/s).
- AWS Elemental MediaConnect is designed to work with an aggregate output bandwidth of 400 Mb/s. The aggregate output bandwidth is calculated by multiplying the bitrate of the source by the 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.

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 (p. 57) 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 (p. 31) 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.

Reliability

- Set up Amazon CloudWatch metrics and alarms to track the health of your source. For information about which metrics to monitor, see Monitoring (p. 84).

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.
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.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default Quota</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entitlements</td>
<td>50 per flow</td>
<td>The maximum number of entitlements that you can grant on a flow.</td>
</tr>
<tr>
<td>Flows</td>
<td>20 per AWS Region</td>
<td>The maximum number of flows that you can create in each AWS Region. You can request a quota increase.</td>
</tr>
<tr>
<td>Outputs</td>
<td>50 per flow</td>
<td>The maximum number of outputs that a flow can have.</td>
</tr>
<tr>
<td>Sources</td>
<td>2 per flow</td>
<td>The maximum number of sources that a flow can have.</td>
</tr>
</tbody>
</table>

**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 (p. 99).
AWS Elemental MediaConnect
related information

The following table lists related resources that you'll find useful as you work with AWS Elemental MediaConnect.

- **Classes & Workshops** – Links to role-based and specialty courses as well as self-paced labs to help sharpen your AWS skills and gain practical experience.
- **AWS Developer Tools** – Links to developer tools, SDKs, IDE toolkits, and command line tools for developing and managing AWS applications.
- **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 web page 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.

<table>
<thead>
<tr>
<th>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source failover (p. 30)</td>
<td>You can now enable source failover and add a second (redundant) source to your flow.</td>
<td>March 13, 2020</td>
</tr>
<tr>
<td>Service quotas (outputs) (p. 100)</td>
<td>You can now add up to 50 outputs to each flow.</td>
<td>February 7, 2020</td>
</tr>
<tr>
<td>Sharing the entitlement data transfer fee with the subscriber (p. 46)</td>
<td>When you grant an entitlement, you can now specify the percentage of the entitlement data transfer fee that you want the subscriber to be responsible for.</td>
<td>September 16, 2019</td>
</tr>
<tr>
<td>RIST sources and outputs (p. 57)</td>
<td>You can now use the RIST protocol for sources and outputs.</td>
<td>September 11, 2019</td>
</tr>
<tr>
<td>Zixi pull outputs (p. 36)</td>
<td>You can now add outputs that use the Zixi pull protocol.</td>
<td>July 26, 2019</td>
</tr>
<tr>
<td>SPEKE support (p. 63)</td>
<td>You can now encrypt the contents of your entitlements using (SPEKE).</td>
<td>June 25, 2019</td>
</tr>
<tr>
<td>Service quotas (flows) (p. 100)</td>
<td>You can now request an increase to the quota of 20 flows per AWS Region.</td>
<td>March 14, 2019</td>
</tr>
<tr>
<td>New service and guide (p. 1)</td>
<td>This is the initial release of the media ingest and transport service, AWS Elemental MediaConnect, and the AWS Elemental MediaConnect User Guide.</td>
<td>November 27, 2018</td>
</tr>
</tbody>
</table>

**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 General Reference.