AWS Elemental MediaConnect

User Guide

AWS Elemental MediaConnect: User Guide

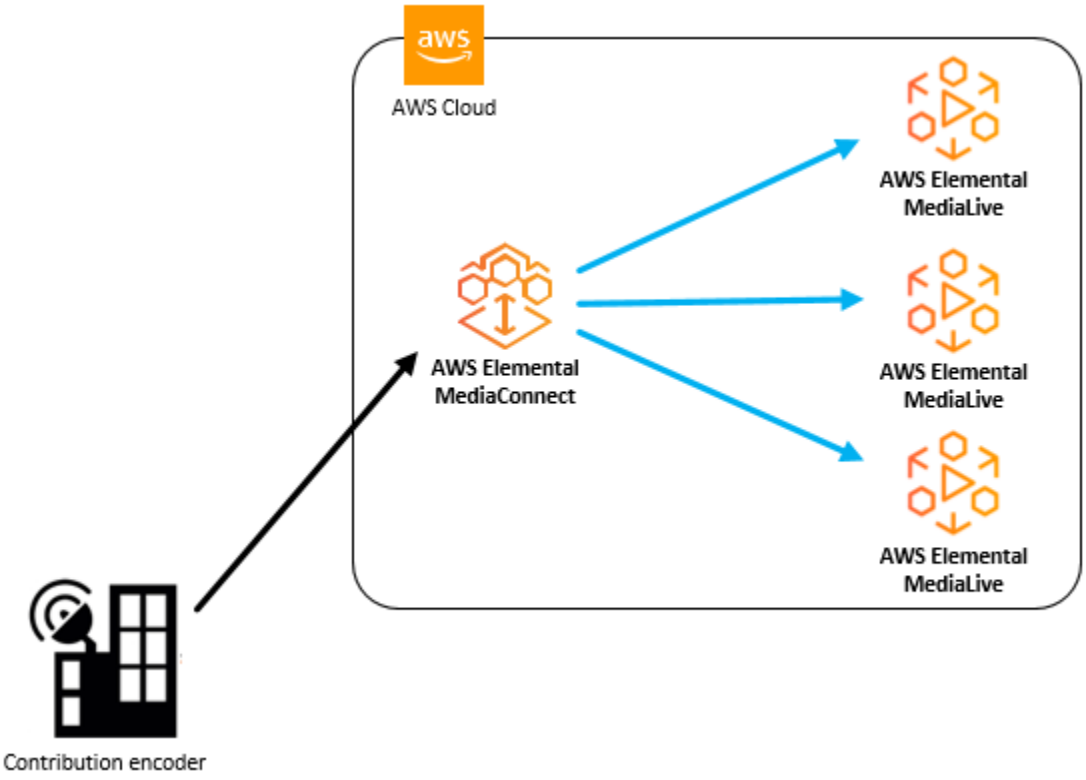
Table of Contents

What Is AWS Elemental MediaConnect?

AWS Elemental MediaConnect is a service that makes it easy for broadcasters and other premium video providers to reliably ingest live video into the AWS Cloud and distribute it to multiple destinations inside or outside the AWS Cloud. AWS Elemental 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, AWS Elemental MediaConnect sends it to outputs that you specify, such as a cloud encoder, another AWS Elemental 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

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 video source 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 20 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 AWS Elemental 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:

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

Sharing

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

Source

External video content that includes configuration information (encryption and source type) and a network address. Each flow has 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 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* (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 AWS Elemental MediaConnect using any of the following methods:

- **AWS Management Console** – The procedures throughout this guide explain how to use the AWS Management Console to perform tasks for 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 AWS Elemental 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 AWS Elemental 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 AWS Elemental MediaConnect API to achieve the results that you want.

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

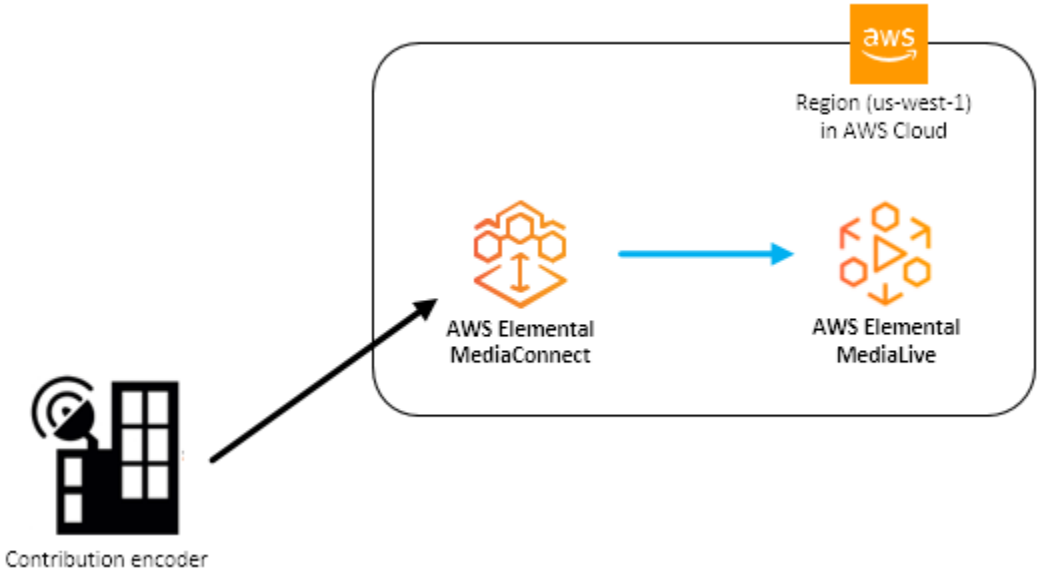
- For **contribution**, use AWS Elemental MediaConnect to ingest content from an on-premises encoder into the AWS Cloud.
- For **distribution**, use AWS Elemental MediaConnect to deliver content to different geographical areas.
- For **entitlements**, use AWS Elemental MediaConnect to share your content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

Topics

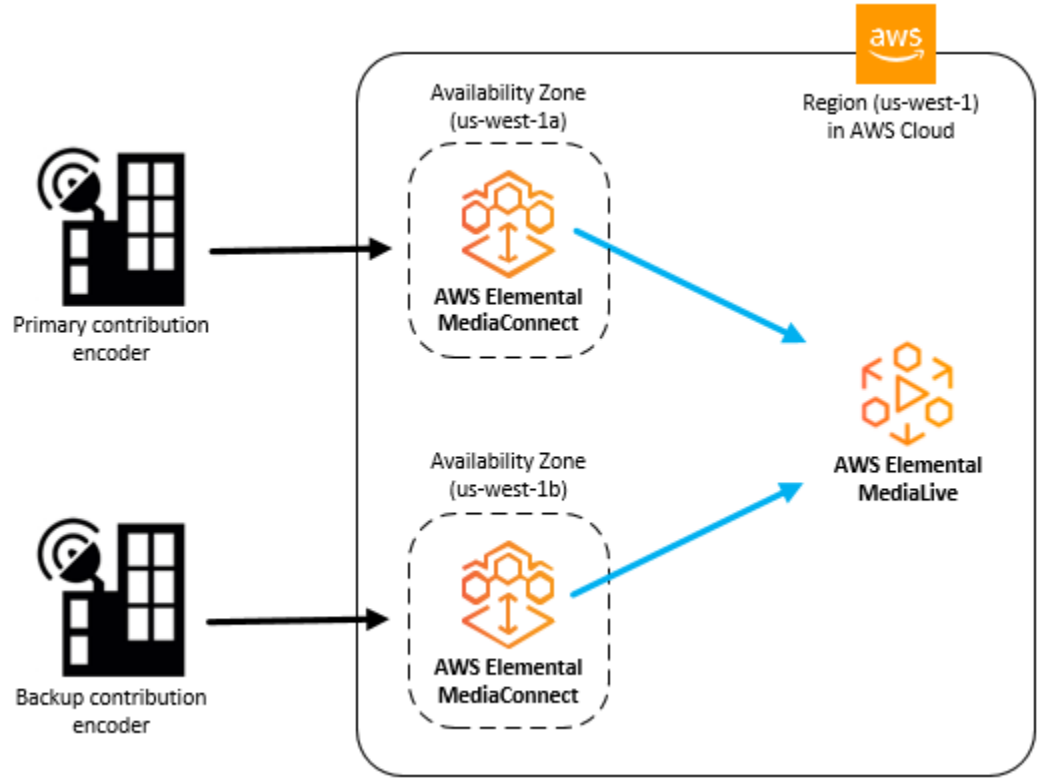
Use Case: Contribution

You can use AWS Elemental MediaConnect to ingest your content from an on-premises contribution encoder into the AWS Cloud. The source for your AWS Elemental 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 AWS Elemental 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.



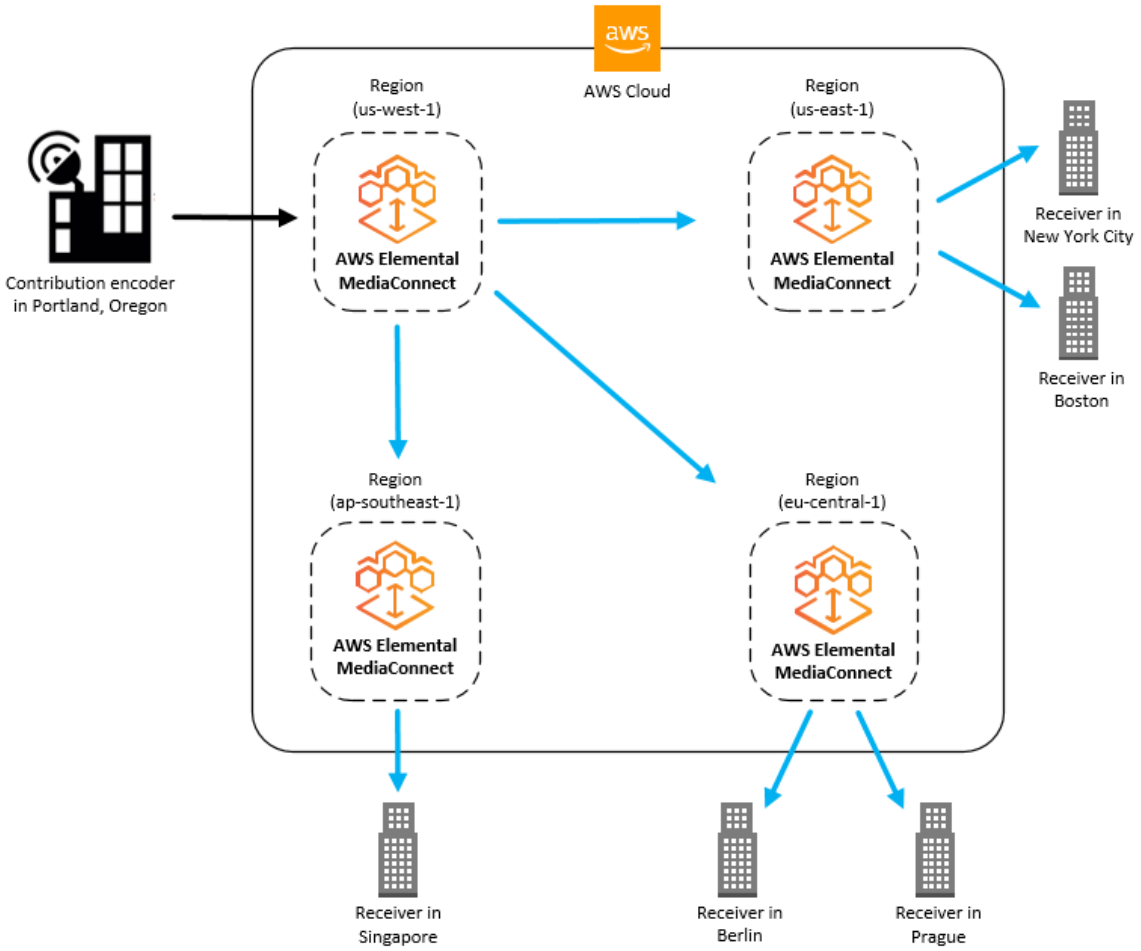
The following illustration shows two on-premises contribution encoders, a primary and a backup, that upload the same content to AWS Elemental MediaConnect in the AWS Cloud. There are two flows, each with one output. Both outputs point to a single AWS Elemental MediaLive channel.



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



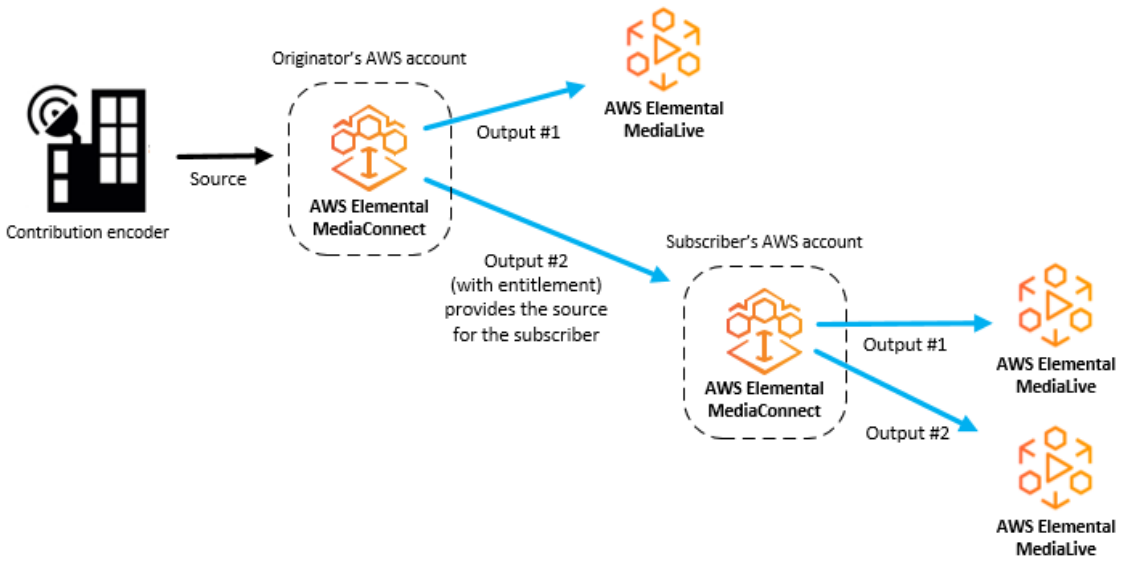
Use Case: Entitlements

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 AWS Elemental 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 AWS Elemental MediaConnect permission to obtain the keys from your Secrets Manager account.

Topics

Step 1. Sign Up for AWS

If you do not have an AWS account, use the following procedure to create one.

To sign up for AWS

1. Open https://aws.amazon.com/, and then choose **Create an AWS Account**.

   Note
   If you previously signed in to the AWS Management Console using AWS account root user credentials, choose **Sign in to a different account**. If you previously signed in to the console using IAM credentials, choose **Sign-in using root account credentials**. Then choose **Create a new AWS account**.

2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code using 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 the section called "Step 3. Create Non-Admin IAM Users" (p. 11).

To create an IAM user for yourself and add the user to an Administrators group

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 of the console, choose **Users**, and then choose **Add user**.
3. For **User name**, type **`Administrator`**.
4. Select the check box next to **AWS Management Console access**, select **Custom password**, and then type the new user's password in the text box. You can optionally select **Require password reset** to force the user to create a new password the next time the user signs in.
5. Choose **Next: Permissions**.
6. On the **Set permissions** page, choose **Add user to group**.
7. Choose **Create group**.
8. In the **Create group** dialog box, for **Group name** type **`Administrators`**.
9. For **Filter policies**, select the check box for **AWS managed - job function**.
10. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.
11. 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.
12. Choose **Next: Tags** to add metadata to the user by attaching tags as key-value pairs.
13. 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 to restrict users' permissions to specific AWS resources, go to 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 3c: Create Users (p. 13)

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**, and then choose **Create policy**.
3. Choose the **JSON** tab and paste the following policy:

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

   This policy allows all actions on all resources in AWS Elemental MediaConnect.
4. Choose **Review policy**.
5. On the **Review policy** page, for **Name**, enter **`MediaConnectAllAccess`**, and then choose **Create policy**.
6. On the **Policies** page, repeat steps 1-5 to create a read-only policy. Use the following policy, and name it **`MediaConnectReadOnlyAccess`**:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "mediaconnect:List*",
                "mediaconnect:Describe*"
            ],
            "Effect": "Allow",
```

```
            "Resource": "*"
        },
        {
            "Action": [
                "ec2:DescribeAvailabilityZones"
                "cloudwatch:GetMetricStatistics"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

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**, and then choose **Create New Group**.
2. On the **Set Group Name** page, enter a name for the group, such as **`MediaConnectAdmins`**.
3. Choose **Next Step**.
4. On the **Attach Policy** page, for **Filter**, choose **Customer Managed**.
5. In the policy list, choose the **MediaConnectAllAccess** policy that you created in the procedure in Step 3a: Create a Policy (p. 12).
6. Choose **Next Step**.
7. On the **Review** page, verify that the correct policies are added to this group, and then choose **Create Group**.
8. On the **Groups** page, repeat steps 1-7 to create a user group with read-only permissions. Use the following guidelines:
   - In step 2, enter a group name such as **`MediaConnectReaders`**.
   - In step 4, choose the **MediaConnectReadOnlyAccess** policy that you created in the procedure in Step 3a: Create a Policy (p. 12).

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. Set Up a Policy for AWS Elemental MediaConnect

If any of your sources or outputs are encrypted, you must store your encryption keys in secrets that you save in AWS Secrets Manager. To allow other services, like AWS Elemental MediaConnect, to read secrets that you save in Secrets Manager, set up a policy that allows read access to those secrets.

To set up a policy for AWS Elemental MediaConnect

1. In the navigation pane of the IAM console, choose **Policies**.
2. Choose **Create policy**, and then choose the **JSON** tab.
3. Enter a policy that uses the following format:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetResourcePolicy",
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:ListSecretVersionIds"
      ],
      "Resource": [
        "arn:aws:secretsmanager:us-west-2:111122223333:secret:aes128-1a2b3c",
        "arn:aws:secretsmanager:us-west-2:111122223333:secret:aes192-4D5e6F",
        "arn:aws:secretsmanager:us-west-2:111122223333:secret:aes256-7g8H9i"
      ]
    }
  ]
}
```

   In the `Resource` section, each line represents the ARN of a different secret that you have created. For more examples, see the section called “IAM Policy Examples for Secrets in AWS Secrets Manager” (p. 42).
4. Choose **Review policy**.
5. For **Name**, enter a name for your policy such as **`SecretsManagerForMediaConnect`**.

6. Choose **Create policy**.

Step 5. Set Up AWS Elemental MediaConnect as a Trusted Entity

After you set up a policy that allows read access to secrets that you save in Secrets Manager, you create a role and assign the policy that you created to that role. You also need to identify AWS Elemental MediaConnect as a trusted entity.

To set up permissions for AWS Elemental MediaConnect

1. In the navigation pane of the IAM console, choose **Roles**.
2. On the **Role** page, choose **Create role**. The **Create role** page appears.
3. 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 AWS Elemental MediaConnect instead of EC2.
5. Choose **Next: Permissions**.
6. For **Attach permissions policies**, enter the name of the policy that you created in the previous step, such as **`SecretsManagerForMediaConnect`**.
7. Select the check box next to **SecretsManagerReadWrite**, 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**.
13. For **Policy Document**, change `ec2.amazonaws.com` to `mediaconnect.amazonaws.com`.

    The policy document should now look like this:

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

14. Choose **Update 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`.

Step 6. Set Up Encryption (optional)

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. You store your encryption keys in AWS Secrets Manager, and then give AWS Elemental MediaConnect permission to obtain the encryption keys from your Secrets Manager account. For more information, see the section called “Encryption in Transit” (p. 40).

Step 7. Install the AWS CLI (optional)

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

Getting Started with AWS Elemental MediaConnect

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

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

Topics

Prerequisites

Before you can use AWS Elemental MediaConnect, you need an AWS account and the appropriate permissions to access, view, and edit AWS Elemental MediaConnect components. Complete the steps in *Setting Up* (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 AWS Elemental MediaConnect console at https://console.aws.amazon.com/mediaconnect/.

Step 2: Create a Flow

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

- Flow name: AwardsNYCShow

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

To create a flow

1. On the **Flows** page, choose **Create flow**.
2. In the **Details** section, for **Name**, enter **`AwardsNYCShow`**.
3. For **Availability Zone**, choose **Any**.
4. In the **Source** section, for **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 20 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 20 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 5 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 15 subscribers create flows, your **`BasketballGame`** flow reaches its maximum number of outputs (5 for the original outputs that you created and another 15 that the service created for the subscribing flows). When the 16th subscriber tries to create a flow using **`BasketballGame`** as a source, the service returns an error.

Topics

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 20 outputs and up to 50 entitlements.

> **Important**
> If the source or any of the outputs of your flow require encryption, store the encryption key (p. 41) in AWS Secrets Manager before you begin this procedure.

To create a flow (console)

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

2. On the **Flows** page, choose **Create flow**.
3. In the **Details** section, for **Name**, specify a name for your flow. This name will become part of the ARN for this flow.

   Note
   AWS Elemental 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** and the service will randomly assign an Availability Zone within the current AWS Region.
5. Determine which type of source you are using:

   - A standard source with RTP or RTP-FEC protocol
   - A standard source with Zixi push protocol
   - An entitled source (a flow that is owned by another AWS account and has granted an entitlement to your account)
6. For specific instructions based on your source type and protocol, choose one of the following tabs:

   Standard source with 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 AWS Elemental 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.
   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.
   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.

   Standard source with 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 AWS Elemental MediaConnect console. It is not visible to anyone outside of the current AWS account.
   3. For **Protocol**, choose **Zixi push**.

      AWS Elemental 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.
   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 during setup (when you set up AWS Elemental MediaConnect as a trusted entity (p. 15)).
   c. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key (p. 41).
   d. For **Decryption algorithm**, choose the type of encryption that was used to encrypt the source.

Entitled source

1. In the **Source** section, for **Source type** choose **Entitled source**.
2. 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. AWS Elemental MediaConnect will filter the list to include only entitlements with a name that matches what you enter.

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

   Note
   The flow doesn't start automatically. You must start the flow (p. 25) manually.

8. Add outputs (p. 30) to specify where you want AWS Elemental MediaConnect to send the content, or grant entitlements (p. 35) 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:

```
{
  "Description": "Awards show in NYC on 2018-11-27",
  "Name": "AwardsShow",
  "Outputs": [
    {
      "Address": "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:

```
aws mediaconnect create-flow --cli-input-json file://rtp.json --region us-east-1 --
profile PMprofile
```

The following example shows the return value:

```
{
  "Flow": {
    "AvailabilityZone": "us-east-1d",
    "Entitlements": [
    ],
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
    "Name": "AwardsShow",
    "Outputs": [
      {
        "Address": "198.51.100.12,
        "Description": "RTP-FEC Output",
        "Name": "AwardsShowOutput",
        "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:1:2-3aBC45dEF67hiJ89-c34de5fG678h:AwardsShowOutput",
        "Port": 5040,
        "Protocol": "rtp-fec",
      },
    ],
    "Source": {
      "IngestIp": "198.51.100.15",
      "IngestPort": 5010,
      "Name": "AwardsShowSource",
      "Protocol": "rtp-fec",
      "SourceArn": "arn:aws:mediaconnect:us-
east-1:111122223333:source:1:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:AwardsShowSource",
      "WhitelistCidr": "10.24.34.0/23",
    },
    "Status": "STANDBY",
  },
}
```

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 AWS Elemental 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 --region us-east-1 --profile PMprofile
```

The following example shows the return value:

```
{
  "Flows": [
    {
      "AvailabilityZone": "us-west-2a",
```

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

Viewing the Details of a Flow

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

To view the details of a flow (console)

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

   The details page for that flow appears. This page is divided into three 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 **Entitlements** tab shows any entitlements that you have granted on this flow.
   - The **Outputs** tab shows details for each output that you created for this flow.

To view the details of a flow (AWS CLI)

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

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

The following example shows the return value:

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

```
        "Name": "RTPOutput",
        "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:1:2-3aBC45dEF67hiJ89-c34de5fG678h:RTPOutput",
        "Port": 5020,
        "Protocol": "rtp-fec"
      }
    ],
    "Source": {
      "IngestIp": "195.51.100.21",
      "IngestPort": 5010,
      "Name": "BasketballGameSource",
      "Protocol": "rtp-fec",
      "SourceArn": "arn:aws:mediaconnect:us-
east-1:111122223333:source:1:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:BasketballGameSource",
      "WhitelistCidr": "10.24.34.0/23"
    },
    "Status": "STANDBY"
  }
}
```

Starting a Flow

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

To start a flow (console)

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

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

To start a flow (AWS CLI)

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

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

The following example shows the return value:

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

Stopping a Flow

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

To stop a flow (console)

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

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

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

To stop a flow (AWS CLI)

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

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

  The following example shows the return value:

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

Updating a Flow

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

- Update source (p. 28)
- Update outputs (p. 32)
- Update entitlements (p. 36)

Deleting a Flow

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

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

To delete a flow (console)

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

The details page for that flow appears.

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

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

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

To delete a flow (AWS CLI)

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

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

The following example shows the return value:

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

Sources in AWS Elemental MediaConnect

Each AWS Elemental MediaConnect flow can have one source. 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

Updating the Source of a Flow

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

To update the source of an existing flow (console)

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

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

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

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

The following example shows the return value:

```
{
  "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
  "Source": {
        "IngestIp": "203.0.113.20",
        "Description": "NYC awards show",
```

```
        "Transport": {
            "Protocol": "rtp",
            "MaxBitrate": 80000000
        },
        "WhitelistCidr": "10.24.34.0/24",
        "SourceArn": "arn:aws:mediaconnect:us-
east-1:111122223333:source:1:2-3aBC45dEF67hiJ89-c34de5fG678h:AwardsShowSource",
        "IngestPort": 5000,
        "Name": "AwardsShowSource"
    }
}
```

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)

1. Open the AWS Elemental MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
2. On the **Flows** page, choose the name of the flow.
3. Choose the **Source** tab.
4. View the **Source health** field. A value of **Connected** indicates that the flow has connected successfully to its source.

Outputs in AWS Elemental MediaConnect

Each flow can have up to 20 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 outputs can be added to a flow is from an entitlement. You can grant an entitlement (p. 35) 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

You can add up to 20 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, do not create the output. Instead, grant an entitlement (p. 35). 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)

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

   The details page for that flow appears.
3. Choose the **Outputs** tab.
4. Choose **Add output**.
5. For **Name**, specify a name for your output. This value is an identifier that is visible only on the AWS Elemental MediaConnect console and is not visible to the end user.
6. Determine which protocol you want to use for the output.
7. For specific instructions based on the protocol you want to use, choose one of the following tabs:

   RTP or RTP-FEC

   1. For **Protocol**, choose RTP or RTP-FEC.
   2. For **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.
4. For **Smoothing latency**, specify the transmission rate for the output. We recommend that you specify a value between 100 and 1,000 ms. If you leave this field blank, the service will use the default value of 100 ms.

Zixi push

1. For **Protocol**, choose **Zixi push**.
2. For **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.
4. For **Stream ID**, enter the stream ID set in the Zixi receiver.

   Important
   If you leave 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 need to 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 leave this field blank, the service will use 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 during setup (when you set up AWS Elemental MediaConnect as a trusted entity (p. 15)).
   d. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key (p. 41).
   e. For **Encryption algorithm**, choose the type of encryption that you want to use to encrypt the source.
8. Choose **Add output**.

Viewing a List of Outputs of a Flow

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

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

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

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

   A list of outputs for that flow appears.

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

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

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

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

```
{
  "Flow": {
    "AvailabilityZone": "us-east-1d",
    "Entitlements": [],
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame",
    "Name": "BasketballGame",
    "Outputs": [
      {
        "Address": "192.0.2.12",
        "Description": "RTP-FEC Output",
        "Name": "NYCOutput",
        "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:1:2-3aBC45dEF67hiJ89-c34de5fG678h:NYCOutput",
        "Port": 5020,
        "Protocol": "rtp-fec"
      },
      {
        "Address": "198.51.100.8",
        "Description": "RTP Output",
        "Name": "DCOutput",
        "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:1:2-987655dEF67hiJ89-c34de5fG678h:DCOutput",
        "Port": 5110,
        "Protocol": "rtp"
      }
    ],
    "Source": {
      "IngestIp": "195.51.100.21",
      "IngestPort": 5010,
      "Name": "BasketballGameSource",
      "Protocol": "rtp-fec",
      "SourceArn": "arn:aws:mediaconnect:us-
east-1:111122223333:source:1:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:BasketballGameSource",
      "WhitelistCidr": "10.24.34.0/23"
    },
    "Status": "STANDBY"
  }
}
```

Updating Outputs on a Flow

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

To update an output on a flow (console)

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

A list of outputs for that flow appears.

4. Choose the option for the output that you want to update.
5. Choose **Update**.
6. Make the appropriate changes, and then choose **Save**.

To update a flow output (AWS CLI)

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

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

The following example shows the return value:

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

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. 36).

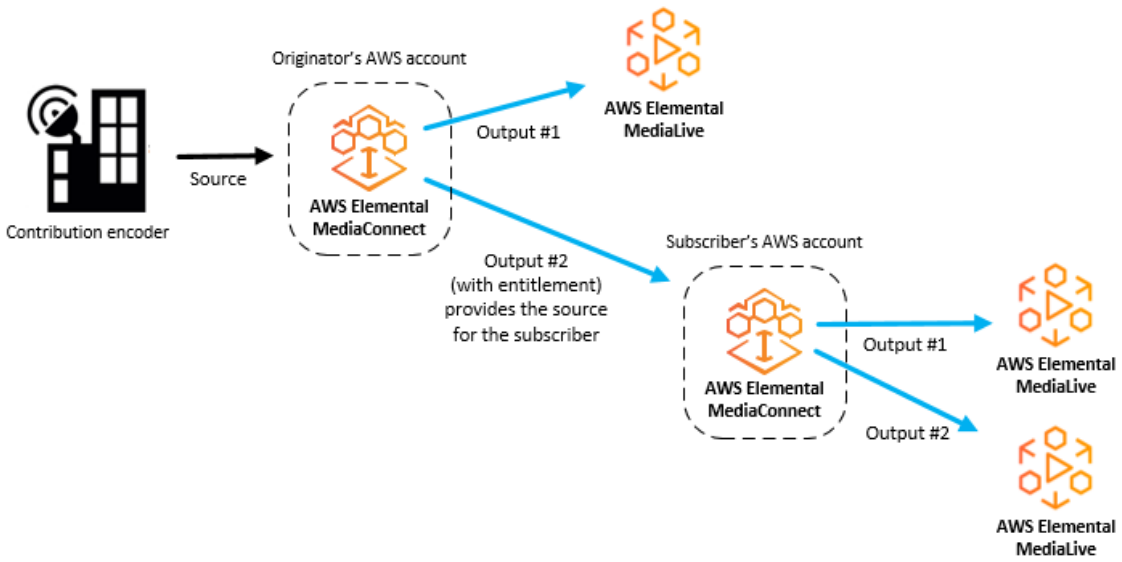
To remove an output from a flow (console)

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

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

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.



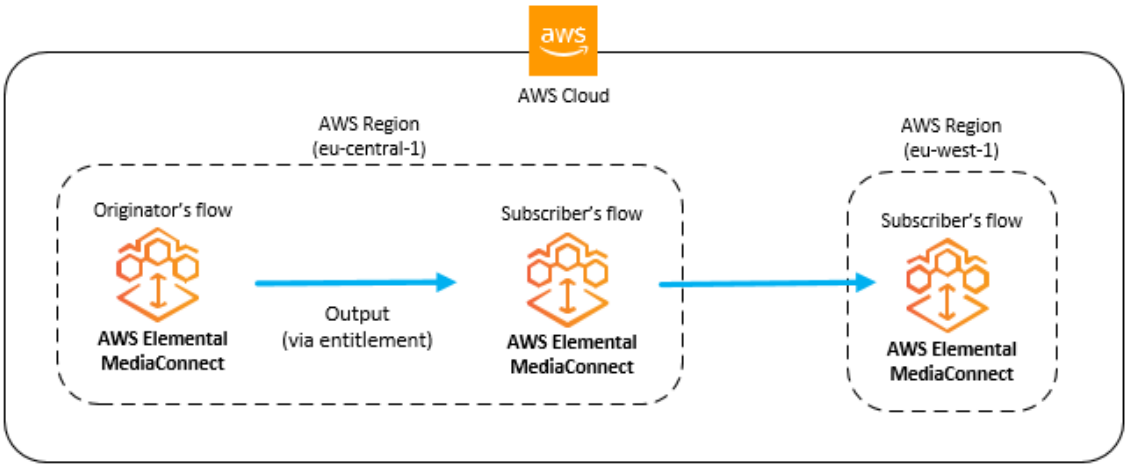
Topics

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 20 maximum outputs that you can have on your flow. Although you can grant up to 50 entitlements on a flow, the service will not allow subscribers to create new flows based on an entitlement once your flow has reached the maximum 20 outputs.

You can grant, update, and revoke entitlements at any time, even on an active flow.

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.



Topics

Granting an Entitlement on a Flow

You can add an entitlement to an existing flow to share your content with another AWS account (the subscriber account). The subscriber can create an AWS Elemental MediaConnect flow, 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.

> **Important**
> If you want to encrypt the video as it is sent from your flow to the subscriber's flow, store the encryption key (p. 41) in AWS Secrets Manager before you begin this procedure.

The subscriber can use an entitlement only once.

To add an entitlement on a flow (console)

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

   The details page for that flow appears.
3. Choose the **Entitlements** tab.
4. Choose **Grant entitlement**. The **Grant entitlement** page appears.
5. For **Name**, specify a name for the entitlement that will help you and the subscriber differentiate this flow from other flows. The name also becomes part of the entitlement ARN, which is visible to the subscriber.
6. For **Subscriber account ID**, specify the subscriber's 12-digit AWS account ID. Don't include hyphens in the ID.
7. For **Description**, specify a description that will help you identify this entitlement later. The description is visible only on the AWS Elemental MediaConnect console for your account.
8. If you want to encrypt the video as it is sent from your flow to the subscriber's flow, 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 during setup (when you set up AWS Elemental MediaConnect as a trusted entity (p. 15)).
d. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key (p. 41).
e. For **Encryption algorithm**, choose the type of encryption that you want to use to encrypt the source.

9. At the bottom of the page, choose **Grant entitlement**.
10. On the **Entitlements** tab, locate the new entitlement in the list.
11. Make a note of the entitlement ARN.
12. 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

Updating an Entitlement

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

To update an entitlement (console)

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

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

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)

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

   The details page for that flow appears.

3. Choose the **Entitlements** tab.
4. Choose the option for the entitlement that you want to revoke.
5. Choose **Revoke**.

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.

Before you can create your flow, you need 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

> **Important**
> If the entitlement is encrypted, store the encryption key (p. 41) in AWS Secrets Manager before you begin this procedure.

You can use an entitlement only once.

To create a flow based on an entitlement (console)

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

   > **Tip**
   > You can click in this field and start typing the entitlement name. AWS Elemental MediaConnect will filter the list to include only entitlements with a name that matches what you type.
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 during setup (when you set up AWS Elemental MediaConnect as a trusted entity (p. 15)).
   c. For **Secret ARN**, specify the ARN that AWS Secrets Manager assigned when you created the secret to store the encryption key (p. 41).
   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. 25) manually.

10. Add outputs (p. 30) to specify where you want AWS Elemental MediaConnect to send the content, or grant entitlements (p. 35) to allow users of other AWS accounts to subscribe to your content.

Protocols in AWS Elemental MediaConnect

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

- **Zixi** is the most reliable protocol offered. We strongly recommend that you use this protocol for your AWS Elemental MediaConnect sources and outputs whenever possible, due to its reliability and ability to stream over longer distances. If your encoder is not capable of using Zixi, you can use Zixi feeder software. The Zixi receiver software (a version specifically for AWS Elemental MediaConnect is available here) can also be used to receive Zixi protocol in environments and on devices that don't support Zixi natively. If you are setting up multiple flows for distribution, we recommend that you use Zixi as the protocol to send content between flows.
- **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.
- **RTP** has wide applicability and takes less bandwidth than RTP-FEC.

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 are built to meet the requirements of the most security-sensitive organizations.

Use the following topics to learn how to secure your AWS Elemental MediaConnect resources.

Topics

Data Protection for AWS Elemental MediaConnect

You can protect your data using tools that are provided by AWS. AWS Elemental MediaConnect can decrypt your incoming video (source) and encrypt your outgoing video (outputs). You can store encryption information securely in AWS Secrets Manager, and then set up an IAM policy to allow AWS Elemental MediaConnect to communicate with Secrets Manager to obtain the encryption credentials as needed.

> **Note**
> Encryption is supported only for sources and outputs that use Zixi protocol.

Topics

Encryption in Transit

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. You store your encryption keys in AWS Secrets Manager, and then give AWS Elemental MediaConnect permission to obtain the encryption keys from your Secrets Manager account.

Topics

Setting Up Encrypted Sources in AWS Elemental MediaConnect

If your source is encrypted, you must save the encryption key in AWS Secrets Manager. You must also make sure that the IAM policy that you created during setup includes this new secret.

> **Note**
> Encryption is supported only for sources that use the Zixi protocol.

To set up an encrypted source (console)

1. Obtain the encryption key from the entity that manages the source.
2. Store the encryption key (p. 41) in Secrets Manager.
3. Make a note of the secret ARN from Secrets Manager. You will need this information later in this procedure.
4. Open the IAM console at https://console.aws.amazon.com/iam/.
5. Make sure that the IAM policy that you created during setup (p. 14) includes the new secret that you just created.
6. Open the AWS Elemental MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
7. Create your flow (p. 20). When you specify the source details, choose to decrypt the source. You will need the ARN of the secret that you created earlier in this procedure.

Setting Up Encrypted Outputs in AWS Elemental MediaConnect

If you want to encrypt your flow output, you must save the encryption key in AWS Secrets Manager. You must also make sure that the IAM policy that you created during setup includes this new secret.

Note
Encryption is supported only for outputs that use the Zixi protocol.

To set up an encrypted output (console)

1. Determine the encryption key that you want to use to encrypt the output.
2. Store the encryption key (p. 41) in Secrets Manager.
3. Make a note of the secret ARN from Secrets Manager. You will need this information later in this procedure.
4. Open the IAM console at https://console.aws.amazon.com/iam/.
5. Make sure that the IAM policy that you created during setup (p. 14) includes the new secret that you just created.
6. Open the AWS Elemental MediaConnect console at https://console.aws.amazon.com/mediaconnect/.
7. Create an output (p. 20) on your flow. When you specify the source details, choose to encrypt the output. You will need the ARN of the secret that you created earlier in this procedure.

Key Management in AWS Elemental MediaConnect

You can protect your content from unauthorized use through encryption. Store your encryption keys in AWS Secrets Manager, and then give AWS Elemental MediaConnect permission to obtain the encryption keys from your Secrets Manager account.

Storing Encryption Keys in AWS Secrets Manager

The Secrets Manager secret that stores your encryption keys must be created using the same AWS account that creates the flow. AWS Elemental MediaConnect does not support cross-account sharing of secrets.

To store encryption keys in Secrets Manager (console)

1. Sign in to the AWS Secrets Manager console at https://console.aws.amazon.com/secrets-manager/.
2. Choose **Store a new secret**.

3. In the **Select secret type** section, choose **Other type of secrets**.
4. In the **Specify the key/value pairs to be stored for this secret** section, choose **Plain text**.
5. Clear any text in the box and replace it with the password value.
6. Keep the **Select the encryption key** set to **DefaultEncryptionKey**.
7. Choose **Next**.
8. For **Secret name**, specify a name for your password. For example, **`2018-12-01_baseball-game-source`**.
9. Choose **Next**.
10. In the **Configure automatic rotation** section, choose **Disable automatic rotation**.
11. Choose **Next**, and then choose **Store**.
12. The details page for your new secret appears, showing information such as the secret ARN. You will need this value when you create a flow that uses the encryption key that you just stored.

IAM Policy Examples for Secrets in AWS Secrets Manager

During setup, you create an IAM policy (p. 14) that allows read access to 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 AWS Elemental MediaConnect needs only read access to secrets, all the examples in this section show only the actions necessary to read the values that you store.

Topics

Allow Read Access to Specific Secrets in AWS Secrets Manager

The following IAM policy allows read access to specific resources (secrets) that you create in AWS Secrets Manager:

```
{
                "Version": "2012-10-17",
                "Statement": [
                  {
                    "Effect": "Allow",
                    "Action": [
                      "secretsmanager:GetResourcePolicy",
                      "secretsmanager:GetSecretValue",
                      "secretsmanager:DescribeSecret",
                      "secretsmanager:ListSecretVersionIds"
                    ],
                    "Resource": [
                      "arn:aws:secretsmanager:us-west-2:111122223333:secret:aes128-1a2b3c",
                      "arn:aws:secretsmanager:us-west-2:111122223333:secret:aes192-4D5e6F",
                      "arn:aws:secretsmanager:us-west-2:111122223333:secret:aes256-7g8H9i"
                    ]
                  }
                ]
```

```
        }
```

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:*",
                ]
              }
            ]
          }
```

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": ["*"]
              }
            ]
          }
```

Authentication and Access Control for AWS Elemental MediaConnect

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS Elemental MediaConnect resources. Administrators use IAM to control who is *authenticated* (signed in) and *authorized* (has permissions) to use AWS Elemental MediaConnect resources. IAM is a feature of your AWS account offered at no additional charge.

Topics

Introduction to Authorization and Access Control

Authentication – To sign in to AWS, you must use root user credentials (not recommended), IAM user credentials, or temporary credentials using IAM roles. To learn more about these entities, see What is Authentication? (p. 48).

Access Control – AWS administrators use policies to control access to AWS resources, such as the AWS Elemental MediaConnect output. To learn more, see What is Access Control? (p. 49) and What are Policies? (p. 51).

> **Important**
> All resources in an account are owned by the account, regardless of who created those resources. You must be granted access to create a resource. However, just because you created a resource does not mean that you automatically have full access to that resource. An administrator must explicitly grant permissions for each action that you want to perform. That administrator can also revoke your permissions at any time.

To help you understand the basics of how IAM works, review the following terms:

- **Resources** – AWS services, such as AWS Elemental MediaConnect and IAM, are made up of objects called resources. You can create, manage, and delete these resources from the service. IAM resources include users, groups, roles, and policies.
  - **Users** – An IAM user represents the person or application who uses its credentials to interact with AWS. A user consists of a name, a password to sign into the AWS Management Console, and up to two access keys that can be used with the AWS CLI or AWS API.
  - **Groups** – An IAM group is a collection of IAM users. You can use groups to specify permissions for its member users. This makes it easier for you to manage permissions for multiple users.
  - **Roles** – An IAM role does not have any long-term credentials (password or access keys) associated with it. A role can be assumed by anyone who needs it and has permissions. An IAM user can assume a role to temporarily take on different permissions for a specific task. Federated users can assume a role by using an external identity provider that is mapped to the role. Some AWS services can assume a *service role* to access AWS resources on your behalf.
  - **Policies** – Policies are JSON policy documents that define the permissions for the object to which they are attached. AWS supports *identity-based policies* that you attach to identities (users, groups, or roles). Some AWS services allow you to attach *resource-based policies* to resources to control what a principal (person or application) can do to that resource. AWS Elemental MediaConnect does not support resource-based policies.
- **Identities** – Identities are IAM resources for which you can define permissions. These include users, groups, and roles.
- **Entities** – Entities are IAM resources that you use for authentication. These include users and roles.
- **Principals** – In AWS, a principal is a person or application that uses an entity to sign in and make requests to AWS. As a principal, you can use the AWS Management Console, the AWS CLI, or the AWS API to perform an operation (such as deleting an output). This creates a *request* for that operation. Your request specifies the *action*, *resource*, *principal*, *principal account*, and any additional information about your request. All of this information provides AWS with *context* for your request. AWS checks all the policies that apply to the context of your request. AWS authorizes the request only if each part of your request is allowed by the policies.

To view a diagram of the authentication and access control process, see Understanding How IAM Works in the *IAM User Guide*. For details about how AWS determines whether a request is allowed, see Policy Evaluation Logic in the *IAM User Guide*.

Permissions Required

To use AWS Elemental MediaConnect or to manage authorization and access control for yourself or others, you must have the correct permissions.

Permissions Required to Use the AWS Elemental MediaConnect Console

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 AWS Elemental MediaConnect resources in your AWS account. If you create an identity-based permissions policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities with that policy.

To ensure that those entities can still use the AWS Elemental MediaConnect console, also attach the following AWS managed policy to the user, as described in Creating Policies on the JSON Tab:

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

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, you need only the permissions that match the API operation you're trying to perform.

Permissions Required for Authentication Management

To manage your own credentials, such as your password, access keys, and multi-factor authentication (MFA) devices, your administrator must grant you the required permissions.

As an AWS administrator, you need full access to IAM so that you can create and manage users, groups, roles, and policies in IAM. You should use the AdministratorAccess AWS managed policy that includes

full access to all of AWS. This policy does not provide access to the AWS Billing and Cost Management console or allow tasks that require root user credentials. For more information, see AWS Tasks That Require AWS Account Root User Credentials in the *AWS General Reference*.

Warning
Only an administrator user should have full access to AWS. Anyone with this policy has permission to fully manage authentication and access control, in addition to modifying every resource in AWS. To learn how to create this user, see Create your IAM Admin User (p. 10).

Permissions Required for Access Control

If your administrator provided you with IAM user credentials, they attached policies to your IAM user to control what resources you can access. To view the policies attached to your user in the AWS Management Console, you must have the following permissions:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": [
                "arn:aws:iam::*:user/${aws:username}"
            ]
        },
        {
            "Sid": "ListUsersViewGroupsAndPolicies",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

If you need additional permissions, ask your administrator to update your policies to allow you to access the actions that you require.

Understanding How AWS Elemental MediaConnect Works with IAM

Services can work with IAM in several ways:

- **Actions** – AWS Elemental MediaConnect supports using actions in a policy. This allows an administrator to control whether an entity can complete an operation in AWS Elemental

MediaConnect. For example, to allow an entity to view update an output on a flow by performing the `UpdateFlowOutput` AWS API operation, an administrator must attach a policy that allows the `iam:UpdateFlowOutput` action.

- **Resource-level permissions** – AWS Elemental MediaConnect supports resource-level permissions. Resource-level permissions allow you to use ARNs to specify individual resources in the policy. Although AWS Elemental MediaConnect supports this feature, some individual actions, such as `List*` actions, do not support specifying an ARN because they are designed to return multiple resources.
- **Resource-based policies** – AWS Elemental MediaConnect does not support resource-based policies. Resource-based policies allow you to attach a policy to a resource within the service. Resource-based policies include a `Principal` element to specify which IAM identities can access that resource.
- **Authorization based on tags** – AWS Elemental MediaConnect does not support authorization-based tags. This feature allows you to use resource tags in the condition of a policy. For example, you might create a policy that allows tag owners full access to Amazon RDS resources that they have tagged. You do this by using a condition key such as `rds:db-tag/Owner`.
- **Temporary credentials** – AWS Elemental MediaConnect supports temporary credentials. This feature allows you 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.
- **Service-linked roles** – AWS Elemental MediaConnect does not support service roles. This feature allows a service to assume a service-linked role on your behalf. This role allows the service 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.
- **Service roles** – AWS Elemental MediaConnect does not support 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, this might break the functionality of the service.

Troubleshooting Authentication and Access Control

Use the following information to help you diagnose and fix common issues that you might encounter when working with IAM.

Topics

I am not authorized to perform an action in AWS Elemental MediaConnect

If you receive an error in the AWS Management Console that tells you that you're not authorized to perform an action, then you must contact the administrator that provided you with your user name and password.

The following example error occurs when an IAM user named my-user-name tries to use the console to perform the AddFlowOutput action, but does not have permissions.

```
User: arn:aws:iam::123456789012:user/my-user-name is not authorized to
 perform: MediaConnect:AddFlowOutput on resource: my-example-output
```

For this example, ask your administrator to update your policies to allow you to access the `my-example-output` resource using the `MediaConnect:AddFlowOutput` action.

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

To allow others to access AWS Elemental 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 AWS Elemental MediaConnect.

I want to understand IAM without becoming an expert

To learn more about IAM terms, concepts, and procedures, see the following pages:

- What is Authentication? (p. 48)
- What is Access Control? (p. 49)
- What are Policies? (p. 51)

What is Authentication?

Authentication is how you sign in to AWS using your credentials.

As a principal, you must be *authenticated* (signed in to AWS) using an entity (root user, IAM user, or IAM role) to send a request to AWS. An IAM user can have long-term credentials such as a user name and password or a set of access keys. When you assume an IAM role, you are given temporary security credentials.

To authenticate from the AWS Management Console as a user, you must sign in with your user name and password. To authenticate from the AWS CLI or AWS API, you must provide your access key and secret key or temporary credentials. AWS provides SDK and CLI tools to cryptographically sign your request using your credentials. If you don't use AWS tools, you must sign the request yourself. 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.

As a principal, you can sign in to AWS using the following entities (users or roles):

- **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 user** – An IAM user is an entity within your AWS account that has specific permissions. AWS Elemental MediaConnect supports *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*.
- **IAM role** – An IAM role is an IAM identity that you can create in your account that has specific permissions. It is similar to an *IAM user*, but it is not associated with a specific person. An IAM role enables you to obtain temporary access keys that can be used to access AWS services and resources. IAM roles with temporary credentials are useful in the following situations:
  - **Federated user access** – Instead of creating an IAM user, you can use existing user 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*.

- **Temporary user permissions** – An IAM user can assume a role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow 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). AWS Elemental MediaConnect does not support these resource-based policies. For more information about choosing whether to use a role or a resource-based policy to allow cross-account access, see Controlling Access to Principals in a Different Account (p. 51).
- **AWS service access** – You can use an IAM role in your account to grant an AWS service permissions to access your account's resources. 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 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*.

What is Access Control?

After you sign in (are authenticated) to AWS, your access to AWS resources and operations is controlled using policies. Access control is also known as authorization.

During authorization, AWS uses values from the request context to check for policies that apply. It then uses the policies to determine whether to allow or deny the request. Most policies are stored in AWS as JSON documents and specify the permissions that are allowed or denied for principals. For more information about the structure and contents of JSON policy documents, see What are Policies? (p. 51).

Policies let an administrator 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 view their own access keys. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or they can add the user to a group that has the intended permissions. When an administrator then give permissions to a group, all users in that group get those permissions.

You might have valid credentials to authenticate your requests, but unless an administrator grants you permissions you cannot create or access AWS Elemental MediaConnect resources. For example, you must have explicit permissions to create an AWS Elemental MediaConnect output.

As an administrator, you can write a policy to control access to the following:

- **AWS for Principals (p. 50)** – Control what the person making the request (the *principal*) is allowed to do.
- **IAM Identities (p. 50)** – Control which IAM identities (groups, users, and roles) can be accessed and how.
- **IAM Policies (p. 50)** – Control who can create, edit, and delete customer managed policies, and who can attach and detach all managed policies.

- **AWS Resources (p. 50)** – Control who has access to resources using an identity-based policy or a resource-based policy.
- **AWS Accounts (p. 51)** – Control whether a request is allowed only for members of a specific account.

Controlling Access for Principals

Permissions policies control what you, as a principal, are allowed to do. An administrator must attach an identity-based permissions policy to the identity (user, group, or role) that provides your permissions. Permissions policies allow or deny access to AWS. Administrators can also set a permissions boundary for an IAM entity (user or role) to define the maximum permissions that the entity can have. Permissions boundaries are an advanced IAM feature. For more information about permissions boundaries, see Permissions Boundaries for IAM Identities in the *IAM User Guide*.

For more information and an example of how to control AWS access for principals, see Controlling Access for Principals in the *IAM User Guide*.

Controlling Access to Identities

Administrators can control what you can do to an IAM identity (user, group, or role) by creating a policy that limits what can be done to an identity, or who can access it. Then attach that policy to the identity that provides your permissions.

For example, an administrator might allow you to reset the password for three specific users. To do this, they attach a policy to your IAM user that allows you to reset the password for only yourself and users with the ARN of the three specified users. This allows you to reset the password of your team members but not other IAM users.

For more information and an example of using a policy to control AWS access to identities, see Controlling Access to Identities in the *IAM User Guide*.

Controlling Access to Policies

Administrators can control who can create, edit, and delete customer managed policies, and who can attach and detach all managed policies. When you review a policy, you can view the policy summary that includes a summary of the access level for each service within that policy. AWS categorizes each service action into one of four *access levels* based on what each action does: `List`, `Read`, `Write`, or `Permissions management`. You can use these access levels to determine which actions to include in your policies. For more information, see Understanding Access Level Summaries Within Policy Summaries in the *IAM User Guide*.

> **Warning**
> You should limit `Permissions Management` access level permissions in your account. Otherwise your account members can create policies for themselves with more permissions than they should have. Or they can create separate users with full access to AWS.

For more information and an example for how to control AWS access to policies, see Controlling Access to Policies in the *IAM User Guide*.

Controlling Access to Resources

Administrators can control access to resources using an identity-based policy or a resource-based policy. In an identity-based policy, you attach the policy to an identity and specify what resources that identity can access. In a resource-based policy, you attach a policy to the resource that you want to control. In the policy, you specify which principals can access that resource.

For more information, see Controlling Access to Resources in the *IAM User Guide*.

Resource Creators Do Not Automatically Have Permissions

All resources in an account are owned by the account, regardless of who created those resources. The AWS account root user is the account owner, and therefore has permission to perform any action on any resource in the account.

> **Important**
> 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. To view the tasks that require you to sign in as the root user, see AWS Tasks That Require Root User.

Entities (users or roles) in your account must be granted access to create a resource. But just because they create a resource does not mean they automatically have full access to that resource. You must explicitly grant permissions for each action. Additionally, you can revoke those permissions at any time, as long as you have access to manage user and role permissions.

Controlling Access to Principals in a Different Account

Administrators can use AWS resource-based policies, IAM cross-account roles, or the AWS Organizations service to allow principals in another account to access resources in your account.

For some AWS services, you can grant cross-account access to your resources. To do this, you attach a policy directly to the resource that you want to share, instead of using a role as a proxy. If the service supports this policy type, then the resource that you want to share must also support resource-based policies. Unlike a user-based policy, a resource-based policy specifies who (in the form of a list of AWS account ID numbers) can access that resource. AWS Elemental MediaConnect does not support resource-based policies.

Cross-account access with a resource-based policy has some advantages over a role. With a resource that is accessed through a resource-based policy, the principal (person or application) still works in the trusted account and does not have to give up his or her user permissions in place of the role permissions. In other words, the principal has access to resources in the trusted account *and* in the trusting account at the same time. This is useful for tasks such as copying information from one account to another. For more information about using cross-account roles, see Providing Access to an IAM User in Another AWS Account That You Own in the *IAM User Guide*.

AWS Organizations offers policy-based management for multiple AWS accounts that you own. With Organizations, you can create groups of accounts, automate account creation, and apply and manage policies for those groups. Organizations enables you to centrally manage policies across multiple accounts, without requiring custom scripts and manual processes. Using AWS Organizations, you can create Service Control Policies (SCPs) that centrally control AWS service use across AWS accounts. For more information, see What Is AWS Organizations? in the *AWS Organizations User Guide*.

What are 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 entity or resource, defines their permissions. AWS evaluates these policies when a principal, such as a user, makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, if a policy allows the GetUser action, then a user with that policy can get user information from the AWS Management Console, the AWS CLI, or the AWS API. When you create an IAM

user, you can set up the user to allow console or programmatic access. The IAM user can sign in to the console using a user name and password. Or they can use access keys to work with the CLI or API.

The following policy types, listed in order of frequency, can affect whether a request is authorized. For more details, see Policy Types in the *IAM User Guide*.

- **Identity-based policies** – You can attach managed and inline policies to IAM identities (users, groups to which users belong, and roles).
- **Resource-based policies** – You can attach inline policies to resources in some AWS services. The most common examples of resource-based policies are Amazon S3 bucket policies and IAM role trust policies. AWS Elemental MediaConnect does not support resource-based policies.
- **Organizations SCPs** – You can use an AWS Organizations service control policy (SCP) to apply a permissions boundary to an AWS Organizations organization or organizational unit (OU). Those permissions are applied to all entities within the member accounts.
- **Access control lists (ACLs)** – You can use ACLs to control what principals can access a resource. ACLs are similar to resource-based policies, although they are the only policy type that does not use the JSON policy document structure. AWS Elemental MediaConnect does not support ACLs.

These policies types can be categorized as *permissions policies* or *permissions boundaries*.

- **Permissions policies** – You can attach permissions policies to a resource in AWS to define the permissions for that object. Within a single account, AWS evaluates all permissions policies together. Permissions policies are the most common policies. You can use the following policy types as permissions policies:
  - **Identity-based policies** – When you attach a managed or inline policy to an IAM user, group, or role, the policy defines the permissions for that entity.
  - **Resource-based policies** – When you attach a JSON policy document to a resource, you define the permissions for that resource. The service must support resource-based policies.
  - **Access control lists (ACLs)** – When you attach an ACL to a resource, you define a list of principals with permission to access that resource. The resource must support ACLs.
- **Permissions boundaries** – You can use policies to define the permissions boundary for an entity (user or role). A permissions boundary controls the maximum permissions that an entity can have. Permissions boundaries are an advanced AWS feature. When more than one permissions boundary applies to a request, AWS evaluates each permissions boundary separately. You can apply a permissions boundary in the following situations:
  - **Organizations** – You can use an AWS Organizations service control policy (SCP) to apply a permissions boundary to an AWS Organizations organization or organizational unit (OU).
  - **IAM users or roles** – You can use a managed policy for a user or role's permissions boundary. For more information, see Permissions Boundaries for IAM Entities in the *IAM User Guide*.

Topics

Identity-based Policies

You can attach policies to IAM identities. For example, you can do the following:

- **Attach a permissions policy to a user or a group in your account** – To grant a user permissions to create an AWS Elemental MediaConnect resource, such as an output, you can attach a permissions policy to a user or a group to which the user belongs.

- **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, the administrator in account A can create a role to grant cross-account permissions to another AWS account (for example, account B) or an AWS service as follows:
  1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in account A.
  2. Account A administrator attaches a trust policy to the role identifying account B as the principal who can assume the role.
  3. Account B administrator can then delegate permissions to assume the role to any users in account B. Doing this allows users in account B to create or access resources in account A. The principal in the trust policy can also be an AWS service principal if you want to grant an AWS service permissions to assume the role.

  For more information about using IAM to delegate permissions, see Access Management in the *IAM User Guide*.

For more information about users, groups, roles, and permissions, see Identities (Users, Groups, and Roles) in the *IAM User Guide*.

Resource-based Policies

Resource-based policies are JSON policy documents that you attach to a resource. These policies allow you to specify what actions a specified principal can perform on that resource and under what conditions. The most commonly known resource-based policy is an Amazon S3 bucket. Resource-based policies are inline policies that exist only on the resource. There are no managed resource-based policies.

Granting permissions to members of other AWS accounts using a resource-based policy has some advantages over an IAM role. For more information, see How IAM Roles Differ from Resource-based Policies in the *IAM User Guide*.

AWS Elemental MediaConnect does not support resource-based policies.

Policy Access Level Classifications

In the IAM console, actions are grouped using the following access level classifications:

- **List** – Provide permission to list resources within the service to determine whether an object exists. Actions with this level of access can list objects but cannot see the contents of a resource. Most actions with the **List** access level cannot be performed on a specific resource. When you create a policy statement with these actions, you must specify **All resources** (`"*"`).
- **Read** – Provide permission to read but not edit the contents and attributes of resources in the service. For example, the Amazon S3 actions `GetObject` and `GetBucketLocation` have the **Read** access level.
- **Write** – Provide permission to create, delete, or modify resources in the service. For example, the Amazon S3 actions `CreateBucket`, `DeleteBucket` and `PutObject` have the **Write** access level.
- **Permissions management** – Provide permission to grant or modify resource permissions in the service. For example, most IAM and AWS Organizations policy actions have the **Permissions management** access level.

  Tip
  To improve the security of your AWS account, restrict or regularly monitor policies that include the **Permissions management** access level classification.
- **Tagging** – Provide permission to create, delete, or modify tags that are attached to a resource in the service. For example, the Amazon EC2 `CreateTags` and `DeleteTags` actions have the **Tagging** access level.

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 AWS Elemental MediaConnect, report when something is wrong, and take automatic actions when appropriate:

- *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.
- *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.

Monitoring AWS Elemental MediaConnect with Amazon CloudWatch

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.

AWS Elemental MediaConnect sends the following metrics to CloudWatch:

- ARQRecovered
- ARQRequests
- CATError
- ConnectedOutputs
- ContinuityCounter
- CRCError
- FECPackets
- FECRecovered
- NotRecoveredPackets
- OverflowPackets
- PacketLossPercent

- PATError
- PCRAccuracyError
- PCRError
- PIDError
- PMTError
- PTSError
- RecoveredPackets
- RoundTripTime
- SourceBitRate
- TransportError
- TSByteError
- TSSyncLoss

Logging AWS Elemental MediaConnect API Calls with AWS CloudTrail

AWS Elemental MediaConnect is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Elemental MediaConnect. CloudTrail captures all API calls for AWS Elemental MediaConnect as events. The calls captured include calls from the AWS Elemental MediaConnect console and code calls to the AWS Elemental MediaConnect API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS Elemental MediaConnect. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to AWS Elemental MediaConnect, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

AWS Elemental MediaConnect Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS Elemental MediaConnect, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for AWS Elemental MediaConnect, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All AWS Elemental MediaConnect actions are logged by CloudTrail and are documented in the AWS Elemental MediaConnect API Reference. For example, calls to the `CreateFlow`, `StartFlow` and `UpdateFlowOutput` operations generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

Understanding AWS Elemental MediaConnect Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested operation, the date and time of the operation, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `DescribeFlow` operation:


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "ABCDEFGHIJKL123456789",
    "arn": "arn:aws:sts::111122223333:user/testUser",
    "accountId": "111122223333",
    "accessKeyId": "ABCDE12345EFGHIJKLMN",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-11-16T20:34:51Z",
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "ABCDEFGHIJKL123456789",
        "arn": "arn:aws:iam::111122223333:role/Administrator",
        "accountId": "111122223333",
        "userName": "Administrator",
      },
    },
  },
  "eventTime": "2018-11-16T20:34:52Z",
  "eventSource": "mediaconnect.amazonaws.com",
  "eventName": "DescribeFlow",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "203.0.113.17",
  "userAgent": "aws-cli/1.15.40 Python/3.6.5 Darwin/16.7.0 botocore/1.10.40",
  "requestParameters": {
    "flowArn": "arn%3Aaws%3Amediaconnect%3Aus-west-2%111122223333%3Aflow
%3A1-23aBC45dEF67hiJ8-12AbC34DE5fG%3AAwardsShow",
  },
  "responseElements": {
  },
```

```
  "requestID": "1a2b3c4d-1234-5678-1234-1a2b3c4d5e6f",
  "eventID": "987abc65-1a2b-3c4d-5d6e-987abc654def",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333",
}
```

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.

Command	Applies to	Description
Entitlements	`grant-flow-entitlements`	Grants entitlements on a flow.
Entitlements	`list-entitlements`	Displays a list of all entitlements that have been granted to this account.
Entitlements	`revoke-flow-entitlement`	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.
Entitlements	`update-flow-entitlement`	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.
Flows	`create-flow`	Creates a flow.
Flows	`delete-flow`	Deletes a flow. You must stop a flow before you can delete it.
Flows	`describe-flow`	Retrieves information about a flow in your account.
Flows	`list-flows`	Lists all the flows that are associated with your account.
Flows	`start-flow`	Starts a flow.
Flows	`stop-flow`	Stops a flow.
Outputs	`add-flow-outputs`	Adds outputs to a flow.
Outputs	`remove-flow-output`	Removes the specified outputs from a flow.
Outputs	`update-flow-output`	Updates the specified output of a flow.
Source	`update-flow-source`	Updates the source of a flow.

Limits in AWS Elemental MediaConnect

The following table describes limits in AWS Elemental MediaConnect. For information about limits that can be changed, see AWS Service Limits.

Resource	Default Limit	Comments
Entitlements	50 per flow	The maximum number of entitlements that you can grant on a flow.
Flows	20 per AWS Region	The maximum number of flows that you can create in each AWS Region.
Outputs	20 per flow	The maximum number of outputs that a flow can have.
Sources	1 per flow	The maximum number of sources that you can assign to a flow.

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.

update-history-change	update-history-description	update-history-date
New service and guide (p. 1)	This is the initial release of the media ingest and transport service, AWS Elemental MediaConnect, and the *AWS Elemental MediaConnect User Guide*.	November 27, 2018

Note

- The AWS Media Services are not designed or intended for use with applications or in situations requiring fail-safe performance, such as life safety operations, navigation or communication systems, air traffic control, or life support machines in which the unavailability, interruption or failure of the services could lead to death, personal injury, property damage or environmental damage.

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.