aws

User Guide

Elemental Inference

Elemental Inference: User Guide

Table of Contents

What Is Elemental Inference?

AWS Elemental Inference is a real-time service that lets you lets you easily apply machine learning models to video, audio, and image content for automated analysis, classification, and insights generation.

Topics

Accessing Elemental Inference

You can access Elemental Inference 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 Inference.
- **AWS SDKs** – If you're using a programming language that AWS provides an SDK for, you can use an SDK to access AWS Elemental Inference. SDKs simplify authentication, integrate easily with your development environment, and provide easy access to Elemental Inference commands. For more information, see Tools for Amazon Web Services.
- **AWS Elemental Inference API** – If you're using a programming language that an SDK isn't available for, see the AWS Elemental Inference 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 PowerShell User Guide.

Pricing for AWS Elemental Inference

As with other AWS products, there are no contracts or minimum commitments for using Elemental Inference. You're charged only for AWS resources that your account uses. Pricing is pay-as-you-go and consists of the following:

- A per-minute charge for each feature (output) that you include in a feed, when the feed is processing source media.
- Savings accrue as you add more features.

For detailed pricing information, see AWS Elemental Inference pricing.

Setting up AWS Elemental Inference

This section provides procedures to set up your organization to use AWS Elemental Inference. It also providers information about determining the IAM permissions that users and other AWS identities require. These permissions let you impose restricted controls on users and other AWS identities, in conformance with the security policies and procedures of your organization.

Topics

Signing up for AWS

Topics

Sign up for an AWS account

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

To sign up for an AWS account

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

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

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

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

Create a user with administrative access

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

Secure your AWS account root user

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

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

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

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

Create a user with administrative access

1. Enable IAM Identity Center.

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

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

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

Sign in as the user with administrative access

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

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

Assign access to additional users

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

   For instructions, see Create a permission set in the *AWS IAM Identity Center User Guide*.

2. Assign users to a group, and then assign single sign-on access to the group.

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

Download tools

The AWS Management Console includes a console for MediaPackage, but if you want to access the services programmatically, see the following:

- The API guides document the operations that the services support and provide links to the related SDK and CLI documentation:
  - AWS Elemental MediaPackage API Reference
- To call an API without having to handle low-level details like assembling raw HTTP requests, you can use an AWS SDK. The AWS SDKs provide functions and data types that encapsulate the functionality of AWS services. To download an AWS SDK and access installation instructions, see the applicable page:
  - Go
  - JavaScript
  - .NET
  - Node.js
  - Python
  - Ruby

  For a complete list of AWS SDKs, see Tools for Amazon Web Services.

- You can use the AWS Command Line Interface (AWS CLI) to control multiple AWS services from the command line. You can also automate your commands using scripts. For more information, see AWS Command Line Interface.
- AWS Tools for Windows PowerShell supports these AWS services. For more information, see AWS Tools for PowerShell Cmdlet Reference.

Quotas for AWS Elemental Inference

Your AWS account has default quotas for each AWS service. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, and other quotas cannot be increased.

To view the quotas for Elemental Inference, open the Service Quotas console. In the navigation pane, choose **AWS services** and select **Elemental Inference**.

To request a quota increase, see Requesting a Quota Increase in the *Service Quotas User Guide*. If the quota is not yet available in Service Quotas, use the limit increase form.

Reference

Media requirements

Topic	Characteristic	Requirements
Container	Live or VOD?	Live only
Video	Codec	H.264 or H.265
	Framerate	30 frames per second
	Aspect ratio	Any
	Resolution	1280x720
Audio	Codec	AAC

Creating an Elemental Inference workflow

You must create a feed and enable at least one AI feature in that feed. After you have created the feed, you must associate one resource, which represents the source media that Elemental Inference will work on.

When you are ready, you must deliver the source media to Elemental Inference. Elemental Inference will produce metadata for each of the features that you set up. You must obtain that metadata and use it to produce the desired media, for example, to produce a video file of an event.

Topics

Step A: Create the feed in Elemental Inference

You must create a feed that contains the features (outputs) that you want to use. After you've created the feed, you must associate a source media (resource) with it.

You can create an Elemental Inference feed using the Elemental Inference console or the AWS CLI.

Topics

Creating using the console

This section describes how to use the Elemental Inference console to create an Elemental Inference feed.

Create the feed

1. Open the Elemental Inference console at https://console.aws.amazon.com/elemental-inference/.
2. In the left navigation bar, choose **Feeds**. On the **Feeds** page, choose **Create**.
3. Complete the fields:
   - Enter a friendly name for the feed. You might want to specify a name that helps you to identify the source media that you plan to use with this feed. For example, **`feed-soccer`**.
   - In **AI features** section, enable the features you want to use. Each feature becomes an output in the feed. See the sections after this procedure for information about specific configuration for a feature.
   - Optionally, associate tags with the feed.
4. Choose **Create feed**. The **Feeds** page appears showing a list with one line for each feed. After a few moments, the status of the feed you just created will be **Available**.

   Available means that the feed isn't currently associated with a source media.

Associate the resource

1. In **Feed association**, choose **Add association**. Enter a friendly name for the source media (resource) that you intend for this feed. You might want to specify a name that helps you to identify the feed that this source media belongs to. For example, **`source-soccer`**.
2. In the **Feed association** section, choose **Save** to confirm the association. The **Feed** information on the page is updated:
   - In **Feed association**, the **Integration** field appears, showing the data endpoint for the feed.
   - In **General details**, the status of the feed changes to **Active**, which means that a resource is associated with the feed.
   - In **Outputs**, the status of each output changes to **Enabled**.

     If you want to disable an output or change any other information for the output, select the **Edit** button (a pencil) on the right.

   For information about feed and output status, see the section called “Lifecycle of a feed”.
3. Make a note of the data endpoint (in the **Integration** field). You will need this value in order to deliver the source media to Elemental Inference.

Creating using the CLI

This section describes how to use the AWS CLI to create an Elemental Inference feed.

You must set up a fully-configured feed: resource - feed - output or outputs, where the MediaLive channel is the resource and each output represents one Elemental Inference feature.

1. Use `create-feed` to create a new feed.

   Include one output for each feature you want to implement. Set the status to `ENABLED` in each output.
2. The response includes the following information that you should make a note of:
   - `id`: The feed ID, which you will need for CLI commands on this feed.
   - `arn`: The feed ARN. You can also obtain the ARN using `get-feed`.
   - `dataEndpoints`: The ARN of the data endpoint for this feed. You will use this ARN when you send source media to Elemental Inference for processing, and when you retrieve the metadata that is the result of this processing.
3. After the feed is created, the status of the feed will eventually change to `AVAILABLE`, indicating that it is ready to have a resource (source media) associated with it.
4. Use `associate-feed` to associate the source media with the feed. The source media is the resource for the feed.

   You now have a useable feed: resource - feed - output.

Configuring each feature

Following are details about how to configure each feature (output) that you include in a Elemental Inference feed.

Configuring event clipping

In **Callback config**, you can enter a string that you want Elemental Inference to always include in the event clipping metadata for this output. This information is useful when you later work with Elemental Inference events in Amazon EventBridge. You will be able to filter events using this information, in order to find the events for one feed. The string might identify the sports event in the feed, for example.

Configuring smart crop

There is no specific configuration for smart crop.

Step B: Format the source media

Make sure that the source media meets the requirements.

Format requirements

You must format the source media according to the CMAF Ingest (Interface-1) version 1.2 specification.

The following table identifies specific requirements for Elemental Inference.

Characteristic	Requirement
Media fragments	Fragmented CMAF Ingest containerized media fragments
MovieFragmentBox	One per segment
Initialization segment: naming	Include an initialization segment with each stream, as follows: • For video: `Streams(default-video.cmfv)/InitializationSegment` • For audio: `Streams(default-audio.cmfa)/InitializationSegment`
Media segments: naming	Media segments after the initialization segment must following this naming pattern: `Streams default-< type>.<ext>/Segment (< sequence-number >)` Where: <type> is video or audio <ext> is cmfv or cmfa

Characteristic	Requirement
	<sequence-number> must increase monotonically, although it doesn't have to be contiguous. Each sequence number must match the sequence number in the MovieFragmentHeader box. For example: `Streams default-video.cmfv/Segment( < sequence-number >)`
End of Stream indicator	The last media segment in the session must be: • A media segment with the `lmsg` brand included in the compatible brands under the SegmentTypeBox. If you can't signal the end of stream in this way, there is a workaround. See the section called “Deliver the source media”.

Example

The following code shows how to use FFMPG to format the media to follow these requirements. The commands demux, segment, and containerize the video and audio. Note the `-init_seg_name` and `-media_seg_name` lines

```
$ mkdir 'Streams(default-video.cmfv)'
$ ffmpeg -i input.mp4 \
-map 0:v:0 -c:v libx264 \
    -profile:v main -pix_fmt yuv420p \
    -g 30 -keyint_min 30 -sc_threshold 0 \
    -force_key_frames 'expr:gte(t,n_forced*1)' \
    -f dash -seg_duration 1 -use_timeline 0 \
    -use_template 1 -remove_at_exit 0  \
    -init_seg_name 'Streams(default-video.cmfv)/InitializationSegment' \
    -media_seg_name 'Streams(default-video.cmfv)/Segment($Number%09d$)' \
    'video.mpd'
```

```
$ mkdir 'Streams(default-audio.cmfa)'$ ffmpeg -i input.mp4 \
    -map 0:a:0 -c:a aac -ar 48000 -ac 2 \
    -f dash -seg_duration 1 -use_timeline 0 \
    -use_template 1 -remove_at_exit 0 \
    -init_seg_name 'Streams(default-audio.cmfa)/InitializationSegment' \
    -media_seg_name 'Streams(default-audio.cmfa)/Segment($Number%09d$)' \
    'audio.mpd'
```

Step C: Deliver the source media

To deliver the source media, you must use an AWS API or SDK. You can't deliver the media using the Elemental Inference console.

1. Obtain the data endpoint for the feed:
   - Using an AWS API or SDK: Use the `GetFeed` operation of Elemental Inference. For information about the operation and parameters, see https://docs.aws.amazon.com/elemental-inference/latest/APIReference/API_GetFeed.
   - Using the Elemental Inference console: Choose Feed in the left navigation bar, then select the feed. In the details page, the data endpoint is in the **Integration** field.
2. Use the Elemental Inference PutMedia operation to deliver the source media to that data endpoint. Make sure to stay within the quotas for the Request rate for PutMedia and the Request rate for PutMedia (in a burst). To view the current quotas, open the Service Quotas console. In the navigation pane, choose **AWS services** and select **AWS Elemental Inference**.

Streaming requirements

You must deliver the source media according to the CMAF Ingest (Interface-1) version 1.2 specification.

The following table identifies specific requirements for Elemental Inference.

Characteristic	Requirement
Media segment duration	0-2 seconds

Characteristic	Requirement
Ingestion order	Elemental Inference will ingest all media segments (audio and video) for a given sequence number before proceeding to the next sequence number
End of Stream indicator (workaround)	Typically, the last media segment includes `lmsg`. However, if you can't signal the end of stream in this way (for example, you are using FFMG), then flush the Elemental Inference internal buffer as follows: • Send up to 10 seconds of slate.
Manifest	CMAF Ingest doesn't support manifests.

Example

The following code sample shows how to use CURL to use the PUT operation to send the content to the data endpoint of a feed. (The example assumes that you have exported your AWS credentials as environment variables, which is standard practice when using the AWS REST API.)

```
# Initialization
$ awscurl --region <region> --service elemental-inference -X PUT \
  'https://<data-endpoint>/v1/feed/<feed-id>/input/0/media/Streams(default-audio.cmfa)/
InitializationSegment' \
  --data-binary -d '@Streams(default-audio.cmfa)/InitializationSegment'

$ awscurl --region <region> --service elemental-inference -X PUT \
  'https://<data-endpoint>/v1/feed/<feed-id>/input/0/media/Streams(default-video.cmfv)/
InitializationSegment' \
  --data-binary -d '@Streams(default-video.cmfv)/InitializationSegment'

# Media
$ awscurl --region <region> --service elemental-inference -X PUT \
 'https://<data-endpoint>/v1/feed/<feed-id>/input/0/media/Streams(default-audio.cmfa)/
Segment(<sequence>)' \
 --data-binary -d '@Streams(default-audio.cmfa)/Segment(<sequence>)'

$ awscurl --region <region> --service elemental-inference -X PUT \
```

```
  'https://<data-endpoint>/v1/feed/<feed-id>/input/0/media/Streams(default-video.cmfv)/
Segment(<sequence>)' \
  --data-binary -d '@Streams(default-video.cmfv)/Segment(<sequence>)'
```

Step D: Query the metadata

Use the Elemental Inference GetEndpoint operation to obtain the metadata that Elemental Inference generates.

For example, the following CURL code shows how to use the POST command to query for the metadata for the output named `testOutput`. The query is for the first second of metadata. This one second span is identified by the start PTS of 0 and the end PTS of 1001.

```
# Query the first second of metadata
$ awscurl --service "elemental-inference" --region <region> \
  -X POST 'https://<data-endpoint>/v1/feed/<feed-id>/input/0/metadata' \
  -H "Content-Type: application/json" \
  -d '{"outputName": "testOutput", "timeSpecification": { "ptsBased":
//{ "startPts":0, "endPts": 1001, "timescale": 1000 } }, "parameters":
 {"smartCropping":
//{"frameRate": { "numerator": 24, "denominator": 1}}}}'
```

For information about the metadata returned for each feature, see the following topics.

Topics

- Metadata for smart crop

Metadata for smart crop

The following CURL code shows the query command plus the results when the output `testOutput` is a smart crop output.

```
# Query the first second of metadata
$ awscurl --service "elemental-inference" --region <region> \
  -X POST 'https://<data-endpoint>/v1/feed/<feed-id>/input/0/metadata' \
  -H "Content-Type: application/json" \
  -d '{"outputName": "testOutput", "timeSpecification": { "ptsBased": { "startPts":0,
 "endPts": 1001, "timescale": 1000 } }, "parameters": {"smartCropping": {"frameRate":
 { "numerator": 24, "denominator": 1}}}}'
```

```
{
    "items": [
        {
            "metadata": {
                "smartCropping": {
                    "crop": {
                        "centerPoint": {
                            "scale": 10000,
                            "xPosition": 2176,
                            "yPosition": 6250
                        }
                    }
                }
            },
            "pts": 0,
            "timecode": null
        },
        {
            "metadata": {
                "smartCropping": {
                    "crop": {
                        "centerPoint": {
                            "scale": 10000,
                            "xPosition": 2176,
                            "yPosition": 6250
                        }
                    }
                }
            },
            "pts": 41,
            "timecode": null
        },
        },
        {
            "metadata": {
                "smartCropping": {
                    "crop": {
                        "centerPoint": {
                            "scale": 10000,
                            "xPosition": 2208,
                            "yPosition": 6238
                        }
                    }
                }
```

```
            },
            "pts": 83,
            "timecode": null
        },
.
.
.
        {
            "metadata": {
                "smartCropping": {
                    "crop": {
                        "centerPoint": {
                            "scale": 10000,
                            "xPosition": 2873,
                            "yPosition": 5781
                        }
                    }
                }
            },
            "pts": 1000,
            "timecode": null
        }
    ]
}
```

Using the metadata

For smart crop, Elemental Inference identifies a *region of interest* in each frame. Elemental Inference then generates metadata that identifies the centerpoint in that region. You can develop a solution that uses this metadata to crop and scale the video. The centerpoint provides you with a reference point for the cropping and scaling algorithms that you develop.

The centerpoint is identified using three pieces of data:

- scale is a reference for calculating the positions as a percentage.
- X position is the position of the centerpoint on the X-axis, from the top left corner of the video frame. Always a positive number.
- Y position is the position of the centerpoint on the Y-axis, from the top left corner of the video frame. Always a positive number.

You can use this data to calculate the centerpoint pixel position in output video of any resolution. The formulas for finding the centerpoint are:

(*X position*) x width of output video / *scale*

(*Y position*) x height of output video / *scale*

Example 1

For example, if the output video is 1920 x 1080, then the following applies to the first piece of data in the metadata example:

- The X pixel position is 2176 x 1920 / 10000 = pixel 417.792 or 418 rounded up
- The Y pixel position is 6250 x 1080 / 10000 = pixel 675

Example 2

Or if the output video is 1280 x 720, then the following applies:

- The X pixel position is 2176 x 1280 / 10000 = pixel 278.528 or 279 rounded up
- The Y pixel position is 6250 x 720 / 10000= pixel 450

Work with existing Elemental Inference feeds

You can make the following changes to an existing feed:

- Revise the properties of any features (outputs) that have configuration properties. For example, you can change the callback metadata value in an event clipping output.
- Add outputs up to the maximum allowed in one feed.
- Remove outputs.
- Enable or disable outputs (change the status). For information about the status of outputs, see the section called “Status of an output”.

You can make these changes using the Elemental Inference console or the AWS CLI.

Topics

Revising using the console

1. If you want to add a feature (an output), make sure that you have room in the enabled outputs quota for Elemental Inference. The list of quotas is sorted alphabetically. Look for quotas that don't start with "Request rate for".
2. Open the Elemental Inference console at https://console.aws.amazon.com/elemental-inference/.
3. In the left navigation bar, choose **Feeds**. On the **Feeds** page, select the feed. The feed details page appears.
4. Take the appropriate action, as follows.

Action	Description
To change output properties	In the section for the output, choose the edit icon and make any changes. Then choose **Save** on the dialog.

Action	Description
To enable or disable an output	In the section for the output, choose the edit icon and change the **Status** field. Then choose **Save** on the dialog.
To add an output	In the **Feed outputs** tab, choose Add output. In the dialog that appears, enter a name and optional description, then choose the feature type. Then choose **Add** on the dialog.
To remove an output	In the section for that output, choose the delete icon on the right side of the section.

Revising using the CLI

To add or delete features in a feed, use the `AssociateFeed` or `UpdateFeed` action. `AssociateFeed` is a PATCH operation. `UpdateFeed` is a PUT operation.

For information about the differences between these two operations and about which operation to use for specific changes, see the information about these operations in the https://docs.aws.amazon.com/elemental-inference/latest/APIReference/WelcomeAWS Elemental Inference API Reference.

Monitoring Elemental Inference activity

You can monitor activity in Elemental Inference on the AWS Elemental Inference console or using the AWS CLI or an AWS API or SDK.

In addition, implements the monitoring capabilities of several AWS services, identified in the following topics.

Topics

Lifecycle of an AWS Elemental Inference workflow

When you use Elemental Inference, you create a feed, include one or more outputs, and associate a resource. Each output represents one Elemental Inference feature, such as smart crop. Each resource represents the source media that works with the feed.

Lifecycle and status of a feed

The typical lifecycle of a feed is as follows:

CREATING -> AVAILABLE -> ACTIVE -> ARCHIVED -> DELETING -> DELETED

All of these statuses are one-way. Most importantly, you can't unarchive a feed. Typically the status changes as follows:

- CREATING to AVAILABLE. This is always the initial transtion.

  AVAILABLE means that the feed doesn't have a resource associated with it.
- AVAILABLE to ACTIVE. After you associate the resource, the feed becomes ACTIVE.

ACTIVE means that you can stream media to the feed, using `PutMedia`.

ACTIVE doesn't mean that you've called PutMedia and the feed is processing the media. Elemental Inference doesn't monitor streaming because each PutMedia call occurs in milliseconds, which means it is impossible to track in a meaningful way.

- ACTIVE to ARCHIVED. When you no longer need the feed, you can archive it in one of these ways:
  - On the console, display the feed details and choose **Archive**.
  - On the CLI, use the `DisassociateFeed` operation, which indirectly sets the status to ARCHIVED.

  When the feed is ARCHIVED, you can't use it any more. Your quota for feeds decreases by 1.
- When you are ready, delete the feed. The feed will disappear from the array of lists, so you won't actually see it with a DELETED status.

Status of a resource

A resource doesn't have a status. Keep in mind that the resource is really just a way of identifying that the feed is intended for a specific source media.

Status of an output

Related to the feed status, the output also has a status:

- ENABLED means that Elemental Inference will generate metadata for the output when it is processing the source media.
- DISABLED means that it won't generate new metadata.

If you create an output using the console, its initial status is ENABLED. If you create it using the CLI, you choose the initial status.

You can change the output status when a feed is any of the statuses before ARCHIVED. You can change the status at any time, including when the source media is being streamed to the feed (in other words, you've called PutMedia).

You can't change the status of the output after the feed is ARCHIVED. There is never any need to do this.

Lifecycle of metadata

Elemental Inference produces metadata for a feature type (for example, smart crop) when media is being streamed to the feed and the output for that feature is ENABLED.

Elemental Inference retains metadata for 24 hours, for both ENABLED and DISABLED outputs.

It continually discards data that is older than 24 hours, for both ENABLED and DISABLED outputs.

Monitoring AWS Elemental Inference on the console

You can monitor a feed using the Elemental Inference console.

1. On the Elemental Inference console, in the navigation pane, choose **Feeds**.
2. The **Feeds** page shows a list of your feeds. Each line in the list provides basic information about the feed, including its status. For information about statuses, see the section called “Lifecycle of a feed”.
3. To view more details about a feed, choose the name of that feed. The **Feed details** page appears. Information appears, as described in the following sections.

General details panel and feed association panel

Feed ID

The ID of the Elemental Inference feed. The ID is identical to the last portion of the ARN of the feed.

ARN

The ARN of the feed.

Status of the feed

These statuses are listed in lifetime order, from **CREATING** to **ARCHIVED**. Note the following:

- A newly created feed typically transitions immediately from **CREATING** to **AVAILABLE** to **ACTIVE**. **ACTIVE** means that the feed is associated with its resource.
- When MediaLive deletes a feed, its status changes to **DELETED**, then after a short period, it changes to **ARCHIVED**. There is no way to change the status of a feed that is **DELETED** or **ARCHIVED**.

Feed outputs tab

In this tab, one panel appears for each feature that you have enabled in the channel. Each panel includes the following information:

- The output status. For more information about status, see the section called "Lifecycle of a feed".
- From association: A value of true means that the output was created using the `AssociateFeed` operation.

  If your organization uses AWS Elemental MediaLive to set up Elemental Inference features in a channel, a value of true indicates that you used MediaLive to create the feed and the output.

Preview metadata for smart crop

This output tab also includes a viewer for historical smart cropping metadata. To retrieve metadata:

Preview metadata for event clipping

This output tab also includes a viewer for historical event clipping metadata.

Monitoring AWS Elemental Inference using the AWS CLI

To monitor feeds and outputs using an AWS API, use the `GetFeed` operation of Elemental Inference. For more information, see GetFeed in the *AWS Elemental Inference API Reference*.

For information about how the feed and outputs change status through the lifetime of the feed, see the section called "Lifecycle of a feed".

Monitoring AWS Elemental Inference with Amazon CloudWatch

You can monitor AWS Elemental Inference using CloudWatch. CloudWatch 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 about using CloudWatch with data, see the Amazon CloudWatch User Guide.

For information about the metrics that CloudWatch produces from CloudWatch data, see the sections that follow.

Topics

Components of a metric

AWS Elemental Inference collects data that is the basis for metrics. It collects these *datapoints* every second and sends them immediately to Amazon CloudWatch. You can use CloudWatch to generate *metrics* for these datapoints.

A metric is a collection of datapoints that has had an aggregation (a *statistic*) applied and that has a *period* and a *time range*. For example, you can request the Dropped frames metric as a sum (the statistic) for a 1-minute period over 10 minutes (the time range). This result of this request is 10 metrics (because the range divided by the period is 10).

Elemental Inference supports all the statistics offered by CloudWatch. However, some statistics aren't useful for Elemental Inference metrics. In the description of metrics later in this chapter, we include the recommended statistics for each metric.

Each Elemental Inference metric includes one or two specific sets of dimensions.

Pricing to view Elemental Inference metrics

For information about charges to view metrics on the CloudWatch console or to retrieve metrics using a CloudWatch API, see the Amazon CloudWatch User Guide.

Service availability and API metrics

API Latency

The duration of API calls from request initiation to response completion.

Details

- Name: ApiLatency
- Supported dimension sets: Feed
- Recommended statistic: p50, p95, p99, Average
- Units: Milliseconds
- Meaning of zero: Elemental Inference throttled requests because of request volume was too high. Typically, API requests fail with error code 429.
- Meaning of no datapoints: There were no API calls on the specified feed.

API Request Count

The number of API calls made.

Details

- Name: ApiRequestCount
- Supported dimension sets:
  - Feed
  - Feed, StatusCode: to monitor API requests that result in a specific category HTTP response codes.
- Recommended statistic: Sum
- Meaning of zero: No API requests were made to the specified feed during the time period.
- Meaning of no datapoints: There were no API calls on the specified feed.

Elemental Inference performance metrics

GetMetadata request count

The number of GetMetadata API requests made to retrieve metadata.

Details

- Name: GetMetadataRequestCount
- Supported dimension sets:
  - Feed

  - Feed, StatusCode: to monitor API requests that result in a specific category HTTP response codes.
- Recommended statistic: Sum
- Units: Count
- Meaning of zero: This metric will never emit zero.
- Meaning of no datapoints: There were no GetMetadata API calls on the specified feed.

Processing fault count

The total number of processing failures.

Details

- Name: ProcessingFaultCount
- Supported dimension sets: Feed
- Recommended statistic: Sum
- Units: Count
- Meaning of zero: All video segments submitted to the feed were processed successfully without any failures.
- Meaning of no datapoints: No video segments were submitted. (PutMedia hasn't been called.)

Processing latency

The end-to-end time from when a video segment is submitted using PutMedia until the metadata is available to be retrieved using GetMetadata.

Details

Name: ProcessingLatency

Supported dimension sets:

- Feed
- Feed, Feature

Recommended statistics: p50, p95, p99, Average

Units: Milliseconds

Meaning of zero: There is probably an issue with the source media. For example, if the timestamp is wrong, it's possible that the latency will seem to be 0.

Meaning of no datapoints: No video segments were submitted (PutMedia hasn't been called.)

PutMedia request count

The number of PutMedia API requests made to submit video segments for processing.

Details

- Name: PutMediaRequestCount
- Supported dimension sets:
  - Feed,
  - Feed, StatusCode: to monitor API requests that result in a specific category HTTP response codes.

  Recommended statistic: Sum
- Units: Count
- Meaning of zero: This metric will never emit zero.
- Meaning of no datapoints: There were no PutMedia API calls on the specified feed.

Monitoring AWS Elemental Inference events in Amazon EventBridge

Elemental Inference automatically turns information about event clipping metadata into events in EventBridge. You can use Amazon EventBridge to manage these events. For example, you can create event rules and deliver the events in emails or SMS messages. You can deliver events to a number of destinations.

For information about the specific events in Elemental Inference that are sent to EventBridge, see the sections that follow.

For information about working with AWS Elemental Inference events in EventBridge, see Amazon EventBridge user guide.

Topics

Clip Metadata Generated event

This event is published when Elemental Inference detects an interesting event in a stream that you are pushing to Elemental Inference. The event includes the following information:

- `resource`: The ARN of the Elemental Inference feed that this clipping event belongs to.
- `timescale`
- `startPts` and `endPts`: The start and end presentation time stamp in the source video. This range identifies the start and end of the interesting event.
- `description`: The field is optional.
- `tags`

The following message is an example of this event.

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "Clip Metadata Generated",
    "source": "aws.elemental-inference",
    "account": "111122223333",
    "time": "2026-03-09T08:41:11Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:elemental-inference:us-west-2:feed/vbphju6m7nohlpcs3sd"
    ],
    "detail": {
        "timescale": 90000,
        "startPts": 159574109479500,
        "endPts": 159574109560500,
        "description": "",
        "tags": [
            "goal"
        ]
    }
}
```

Logging Elemental Inference API calls with AWS CloudTrail

AWS Elemental Inference is integrated with AWS CloudTrail. *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.

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

Elemental Inference information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Elemental Inference, 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 Elemental Inference, 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

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 Elemental Inference 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 action, the date and time of the action, 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. The example shows the entry for one API call. The call is made by the identity specified in `userIdentity`, in this case a user who has assumed the Administrator role with the session name `santosp`. The call was a `CreateFeed` operation coming from the AWS CLI (as specified in `userAgent`) running on a computer with the IP address 203.0.113.33


```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAEXAMPLEID:santosp",
    "arn": "arn:aws:sts::111122223333:assumed-role/Administrator/santosp",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAEXAMPLEID",
        "arn": "arn:aws:iam::111122223333:role/Administrator",
        "accountId": "111122223333",
        "userName": "Administrator"
      },
      "attributes": {
        "creationDate": "2024-11-14T19:00:28Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2024-11-14T19:00:29Z",
  "eventSource": "elemental-inference.amazonaws.com",
  "eventName": "CreateFeed",
  "awsRegion": "us-west-2",
```

```
  "sourceIPAddress": "203.0.113.33",
  "userAgent": "aws-cli/2.32.22 md/awscrt#0.29.1 ua/2.1 os/
linux#6.1.159-182.297.amzn2023.x86_64 md/arch#x86_64 lang/python#3.9.25 md/
pyimpl#CPython cfg/retry-mode#standard md/installer#source md/distrib#amzn.2023 md/
prompt#off md/command#elemental-inference.create-feed",
  "requestParameters": {
    "outputs": [
      {
        "name": "out1",
        "outputConfig": {
          "cropping": {}
        },
        "status": "ENABLED"
      }
    ],
    "name": "live-studio-feed",
    "x-amzn-client-token": "1111aaaa-e840-4416-8ee8-a95d5ed7c031"
  },
  "responseElements": {
    "outputs": [
      {
        "name": "out1",
        "outputConfig": {
          "cropping": {}
        },
        "status": "ENABLED"
      }
    ],
    "dataEndpoints": [
      "https://abc123example.elemental-inference-data.us-west-2.amazonaws.com/"
    ],
    "name": "live-studio-feed",
    "id": "abc123example",
    "arn": "arn:aws:elemental-inference:us-west-2:111122223333:feed/abc123example",
    "status": "CREATING",
    "tags": {}
  },
  "requestID": "d2f882ac-1a9d-11e9-a0e5-afe6a8c88993",
  "eventID": "ebbe0290-7a1b-4053-a219-367404e0fe96",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::Elemental-Inference::Feed",
```

```
      "ARN": "arn:aws:elemental-inference:us-west-2:111122223333:feed/abc123example"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}
```

Security in AWS Elemental Inference

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS Elemental Inference, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Elemental Inference. The following topics show you how to configure Elemental Inference to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Elemental Inference resources.

Topics

Data protection in AWS Elemental Inference

The AWS shared responsibility model applies to data protection in AWS Elemental Inference. As described in this model, AWS is responsible for protecting the global infrastructure that runs all

of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Elemental Inference or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Identity and Access Management for AWS Elemental Inference

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Elemental Inference resources. IAM is an AWS service that you can use with no additional charge.

Topics

Audience

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user** - request permissions from your administrator if you cannot access feature
- **Service administrator** - determine user access and submit permission requests
- **IAM administrator** - write policies to manage access

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users to use federation with an identity provider to access AWS services using temporary credentials.

A *federated identity* is a user from your enterprise directory, web identity provider, or Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by switching from a user to an IAM role (console) or by calling an AWS CLI or AWS API operation. For more information, see Methods to assume a role in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. You must specify a principal in a resource-based policy.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

Compliance validation for AWS Elemental Inference

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more information about your compliance responsibility when using AWS services, see AWS Security Documentation.

Resilience in AWS Elemental Inference

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure security in AWS Elemental Inference

As a managed service, AWS Elemental Inference is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud

Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access Elemental Inference through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Document history for user guide

The following table describes the documentation for this release of AWS Elemental Inference

- **API version: latest**

Change	Description	Date
New information	New information has been added to all sections of this guide.	April 8, 2026
New service and guide	This is the initial release of the AWS Elemental Inference service and the AWS Elemental Inference User Guide.	February 24, 2026
Infrastructure security	The information in this section has been revised. Specifically, we now require TLS 1.2 and we recommend TLS 1.3.	June 24, 2023
Data protection	The information in this section has been revised. Specifically, we now require TLS 1.2 and we recommend TLS 1.3.	June 24, 2023
AWS Identity and Access Management	Updated guide to align with the IAM best practices . For more information, see Security best practices in IAM.	February 14, 2023