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.

Creating and monitoring a feed

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 media source that Elemental Inference will work on.

Topics

- Prepare the source media
- Create the feed in Elemental Inference
- Deliver the source media

Prepare the source media

Stream requirements

The source that you deliver to Elemental Inference must meet the follow requirements of the DASH-IF live media ingest protocol specification, as follows.

- Media fragments: Fragmented CMAF Ingest containerized media fragments.
- Initialization segment: Include an initialization segment with each stream, as follows:

  For video: Streams(default-video.cmfv)/InitializationSegment

  For audio: Streams(default-audio.cmfa)/InitializationSegment
- Futher media segments must following this naming pattern:

  ```
  Streams default-<type>.<ext>/Segment(<sequence-number>)
  ```

  Where:

  <type> is video or audio

  <ext> is cmfv or cmfa

  <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>)`

- Elemental Inference will ingest all media segments (audio and video) for a given sequence number before proceeding to the next sequence number.
- MovieFragmentBox: One per segment.
- Media segment duration: 0-2 seconds.
- Last media segment in the session: A media segment with the `lmsg` brand included in the compatible brands under the SegmentTypeBox.

  If you are using FFMPG, note that currently FFMPG doesn't set the `lmsg` brand to signal end-of-stream, which means that `Elemental Inference` will retain the final buffer it receives. As a workaround, you could send up to 10 seconds of slate, in order to flush the internal buffer.
- Manifest: Not supported.

Media requirements for video

- Codec: H.264 or H.265
- Framerate: 30 frames per second
- Resolution: 1280x720

Media requirements for audio

Codec: AAC

Create the feed in Elemental Inference

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 the name for the feed. The name should help you to identify the media source that you will send to Elemental Inference.
   - Enter an optional description
   - Enable at least one feature in the **AI features** section.

- 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**.
5. Make a note of the feed ARN that Elemental Inference generates. This is the unique *data endpoint* for the feed. You need this endpoint when you deliver the source media, in the next step.
6. Choose the feed by name. The details about the feed appear.
7. In **Feed association**, enter a friendly name for the resource for this feed.

   The resource is the source media that Elemental Inference will work on.

   Each feed has only one resource. All the AI features that you enable will work on this single source.
8. In the **Feed association** section, choose **Save** to confirm the association. The **Feed** information on the page is updated:
   - In **General details**, the status of the feed changes to **Active**.
   - 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.

Deliver the source media

You must format the source media using an encoding application of your choice. You must then use PUTMEDIA on the Elemental Inference feed endpoint to deliver the source media to the data endpoint on the Elemental Inference feed. You can then use the GETMETADATA on the Elemental Inference feed endpoint to obtain the metadata that Elemental Inference generates.

Format the media

The following code shows how to use FFMPG to format the media to follow the requirements in the section called “Prepare the source media”. The commands demux, segment, and containerizes the video and audio.

```
$ 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'
```

Deliver the content

The following code shows how to use CURL to use the PUT command to send the content to the data endpoint of a feed.

You can obtain the data endpoint of a feed by using one of the Elemental Inference APIs or SDKs. For example, use the CreateEndpoint or GetEndpoint operations of the REST API. The endpoint is returned in the response.

Make signed requests to the data endpoint of the Elemental Inference feed. This example assumes that you have exported credentials as environment variables.

```
# 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>)'
```

Query the output media

This following CURL code shows how use to the POST command to query the first second of metadata that is generated by Elemental Inference. After the first frame, the PTS increments by 42 in each metadata returned.

This example shows the metadata returned for the smart crop feature. See below for more information about the metadata for a smart crop.

```
# 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
        }
    ]
}
```

Metadata for a smart crop

For each frame, Elemental Inference creates metadata that identifies a point in that region of interest. This is the point of interest. You can develop a solution that crops and scales the video. The point of interest provides you with a reference point for the cropping and scaling algorithms that you develop.

The point of interest is an x,y coordinate.

- The y coordinate is always the halfway point (the 50% mark) on the y axis. It is not the true y position of the point of interest.
- The x coordinate is the true position (as a percentage) on the x axis.

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 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