Amazon Elastic Transcoder

Developer Guide

API Version 2012-09-25

Amazon Elastic Transcoder: Developer Guide

Table of Contents

What is Amazon Elastic Transcoder? 1
- Accessing Elastic Transcoder 2
  - Regions and Endpoints 2
- Choosing a Region for Your Elastic Transcoder Resources 2
- Limits on Pipelines, Jobs, and Presets 3

Getting Started 4
- 1: Sign Up for AWS 4
- 2: Create a Bucket 5
- 3: Create a Pipeline 6
- 4. Create a Preset 7
- 5. Create a Job 7
- 6: Monitor Progress 8

Working with Pipelines 9
- Creating a Pipeline 9
- Pausing and Reactivating Pipelines 10
- Updating Pipeline Notifications 10
- Listing and Viewing Pipelines 11
- Deleting a Pipeline 11
- Pipeline Settings 12
  - Region 21
  - General Settings 21
  - Configure Amazon S3 Bucket for Transcoded Files and Playlists 12
  - Configure Amazon S3 Bucket for Thumbnails 14
  - Notifications 16

Working with Jobs 17
- Creating a Job 17
- Listing and Viewing Jobs 18
- Canceling a Job 19
- Job Settings 20
  - Region 21
  - General Settings 21
  - Output Details, Part 1 23
  - Output Details, Part 2, Clip and Caption Settings 25
  - Output Details, Part 3 27
  - (Optional) Output Encryption 28
  - (Video/Thumbnails Only) Watermarks 30
  - (AAC/MP3/MP4 Only) Album Art 31
  - (Optional) User Metadata 34
  - (Fragmented MP4/MPEG-TS Outputs Only) Playlist 35
  - (Fragmented MP4/MPEG-TS Outputs Only, Optional) HLS Content Protection 36
  - (Optional) Additional Job Details 38

Working with Presets 39
- Creating a Preset 39
- Listing and Viewing Presets 40
- Changing Presets 40
- Deleting a Preset 40
- Preset Settings 41
  - General Settings 41
  - Video Settings 42
  - Watermark Settings 51
  - Audio Settings 54
  - Thumbnail Settings 56

Amazon S3 Bucket Organization 58

Data Encryption Options 60
- Encryption Options 60

Using KMS ... 62
Getting Ready to Use AWS KMS with Elastic Transcoder ... 62
Connecting AWS KMS ... 63
HLS Content Protection ... 64
Keys for HLS Content Protection ... 64
Streaming HLS Protected Content ... 64
Creating Encrypted Streamed Content ... 65
Notifications ... 66
Captions ... 68
Watermarks ... 69
System Presets ... 70
Controlling Access ... 74
Controlling User Access to Elastic Transcoder ... 75
Example User Policies for Elastic Transcoder ... 75
Restricting Users to Certain Resources ... 77
Pipeline Service Roles ... 79
Default Service Role ... 79
Security Best Practices ... 80
Logging using CloudTrail ... 81
Elastic Transcoder Information in CloudTrail Log Files ... 81
Understanding Elastic Transcoder Log File Entries ... 82
AWS SDK Sample Code ... 86
Background for AWS SDK Sample Code ... 87
HTTP Live Streaming (HLS) ... 87
Notifications ... 87
Setting Up Your Environment ... 87
Java ... 88
Ruby ... 92
PHP ... 95
Python ... 98
Java Samples ... 101
HLS Samples ... 101
Notification Samples ... 101
Ruby Samples ... 104
HLS Samples ... 104
Notification Samples ... 104
PHP Samples ... 107
HLS Samples ... 107
Notification Samples ... 107
Python Samples ... 110
HLS Samples ... 110
Notification Samples ... 110
API Reference ... 113
Using the AWS SDKs ... 113
Making HTTP Requests ... 114
HTTP Header Contents ... 114
HTTP Request Body ... 115
HTTP Responses ... 116
Signing Requests ... 117
Handling Errors ... 118
API Error Codes (Client and Server Errors) ... 118
Errors During Job Processing ... 121
Catching Errors ... 124
Error Retries and Exponential Backoff ... 125
Pipeline Operations ... 127
Create Pipeline ... 128
List Pipelines ... 138
Read Pipeline ... 147

Update Pipeline 156
Update Pipeline Status 167
Update Pipeline Notifications 170
Delete Pipeline 174
Test Role 177
Job Operations 180
Create Job 181
List Jobs by Pipeline 210
List Jobs by Status 230
Read Job 251
Cancel Job 271
Preset Operations 274
Create Preset 275
List Presets 296
Read Preset 315
Delete Preset 334
Document History 337
AWS Glossary 341

What is Amazon Elastic Transcoder?

Topics

- Accessing Elastic Transcoder (p. 2)
- Choosing a Region for Your Elastic Transcoder Resources (p. 2)
- Limits on the Number of Elastic Transcoder Pipelines, Jobs, and Presets (p. 3)

Amazon Elastic Transcoder lets you convert media files that you have stored in Amazon Simple Storage Service (Amazon S3) into media files in the formats required by consumer playback devices. For example, you can convert large, high-quality digital media files into formats that users can play back on mobile devices, tablets, web browsers, and connected televisions.

Elastic Transcoder has four components:

- **Jobs** do the work of transcoding. Each job converts one file into up to 30 formats. For example, if you want to convert a media file into six different formats, you can create files in all six formats by creating a single job.

  When you create a job, you specify the name of the file that you want to transcode, the names that you want Elastic Transcoder to give to the transcoded files, and several other settings. For each format that you want to transcode into, you also specify a template, known as a *preset* (see below), that contains the audio and video settings that you want to use for the transcoded file or files.
- **Pipelines** are queues that manage your transcoding jobs. When you create a job, you specify which pipeline you want to add the job to. Elastic Transcoder starts processing the jobs in a pipeline in the order in which you added them. If you configure a job to transcode into more than one format, Elastic Transcoder creates the files for each format in the order in which you specify the formats in the job.

  One common configuration is to create two pipelines—one for standard-priority jobs, and one for high-priority jobs. Most jobs go into the standard-priority pipeline; you use the high-priority pipeline only when you need to transcode a file immediately.

  If a pipeline already contains jobs when you create a new job, Elastic Transcoder queues the newest job and begins processing it as soon as resources are available for that pipeline. If the pipeline is already using all of its resources, Elastic Transcoder begins processing the next job in the pipeline when it finishes one of the jobs that it's currently processing.

A pipeline can process more than one job simultaneously, and the time required to complete a job varies significantly based on the size of the file you're converting and the job specifications. Accordingly, jobs don't necessarily complete in the order in which you create them.

You can temporarily stop processing jobs by pausing the pipeline.

- **Presets** are templates that contain most of the settings for transcoding media files from one format to another. Elastic Transcoder includes some default presets for common formats, for example, several iPod and iPhone versions. You can also create your own presets for formats that aren't included among the default presets. You specify which preset you want to use when you create a job.
- **Notifications** let you optionally configure Elastic Transcoder and Amazon Simple Notification Service to keep you apprised of the status of a job: when Elastic Transcoder starts processing the job, when Elastic Transcoder finishes the job, and whether Elastic Transcoder encounters warning or error conditions during processing. Notifications eliminate the need for polling to determine when a job has finished. You configure notifications when you create a pipeline.

Accessing Elastic Transcoder

Elastic Transcoder is a RESTful web service that uses HTTPS as the protocol and JavaScript Object Notation (JSON) as the message format. Your application code can make requests directly to the Elastic Transcoder API. When using the REST API directly, you must write the necessary code to sign and authenticate your requests. For more information about the API and about signing requests, see API Reference (p. 113).

Elastic Transcoder also provides a management console. You can use the console to perform all of the same operations that you can perform using the Elastic Transcoder API. For information about using the console to create and manage pipelines, presets, and jobs, see the applicable topic:

- Working with Jobs (p. 17)
- Working with Pipelines (p. 9)
- Working with Presets (p. 39)

Regions and Endpoints

You create pipelines in a specific AWS region. You always send your Elastic Transcoder requests to a region-specific endpoint. For a list of supported AWS regions, go to the Regions and Endpoints section in the *Amazon Web Services General Reference*.

Choosing a Region for Your Elastic Transcoder Resources

Pipelines and jobs are closely associated with specific regions. When you create pipelines and jobs, they're created in the current region. When you create a new job, you must specify a pipeline in the current region.

You can specify Amazon S3 buckets in a different region than your Elastic Transcoder resources, but we don't recommend it because you'll incur additional charges for transferring files between AWS regions.

For more information on cross-regional fees, see Data Transfer Pricing in Amazon S3 Pricing.

Limits on the Number of Elastic Transcoder Pipelines, Jobs, and Presets

Elastic Transcoder pipelines, jobs, and presets are subject to the following limitations:

- **Pipelines:** For each region, 4 pipelines per AWS account
- **Maximum number of queued jobs:** 100,000 per pipeline
- **Maximum number of outputs:** 30 per job
- **Maximum number of jobs processed simultaneously by each pipeline:**
  - US East (Northern Virginia) Region: 20
  - US West (Northern California) Region: 12
  - US West (Oregon) Region: 20
  - EU (Ireland) Region: 20
  - Asia Pacific (Singapore) Region: 12
  - Asia Pacific (Tokyo) Region: 12
- **Presets:** 50 user-defined presets per AWS account (Elastic Transcoder also includes predefined presets that don't count against the limit.)
- **Maximum rate at which you can submit job requests:**
  - **Create Job:** You can submit two `Create Job` requests per second per AWS account at a sustained rate; brief bursts of 100 requests per second are allowed.
  - **Read Job:** You can submit four `Read Job` requests per second per AWS account at a sustained rate; brief bursts of 100 requests per second are allowed.

You can request higher limits at https://console.aws.amazon.com/support/home#/case/create?issueType=service-limit-increase&limitType=service-code-elastic-transcoders.

Getting Started with Elastic Transcoder

The example in this topic gives you a quick overview of how to use Amazon Elastic Transcoder to transcode media files from one media format into another. You only need to perform a few basic steps to start transcoding your media files using Elastic Transcoder. The first step is signing up for AWS. After that, you create an Amazon S3 bucket and upload a media file that you want to transcode. You then create a pipeline to process your jobs, and create a job to transcode a specific file into a specific format. If you want to transcode to a format for which we don't provide a preset (a template), you can create a custom preset before you create the job.

> **Note**
> If you aren't already acquainted with jobs, pipelines, and presets—the basic concepts behind Elastic Transcoder—take a quick look at the short overview topic: What is Amazon Elastic Transcoder? (p. 1)

Topics

- Step 1: Sign Up for AWS (p. 4)
- Step 2: Create an Amazon S3 Bucket or Two, and Upload a Media File (p. 5)
- Step 3: Create a Pipeline (p. 6)
- Step 4: (Optional) Create a Preset (p. 7)
- Step 5: Create a Job (p. 7)
- Step 6: Monitor the Progress of Your Job (p. 8)

Step 1: Sign Up for AWS

To use Elastic Transcoder, you need an AWS account. If you don't already have an account, you'll be prompted to create one when you sign up. You're not charged for any AWS services that you sign up for unless you use them.

To sign up for AWS

1. Open http://aws.amazon.com/, and then click **Sign Up**.
2. Follow the on-screen instructions.

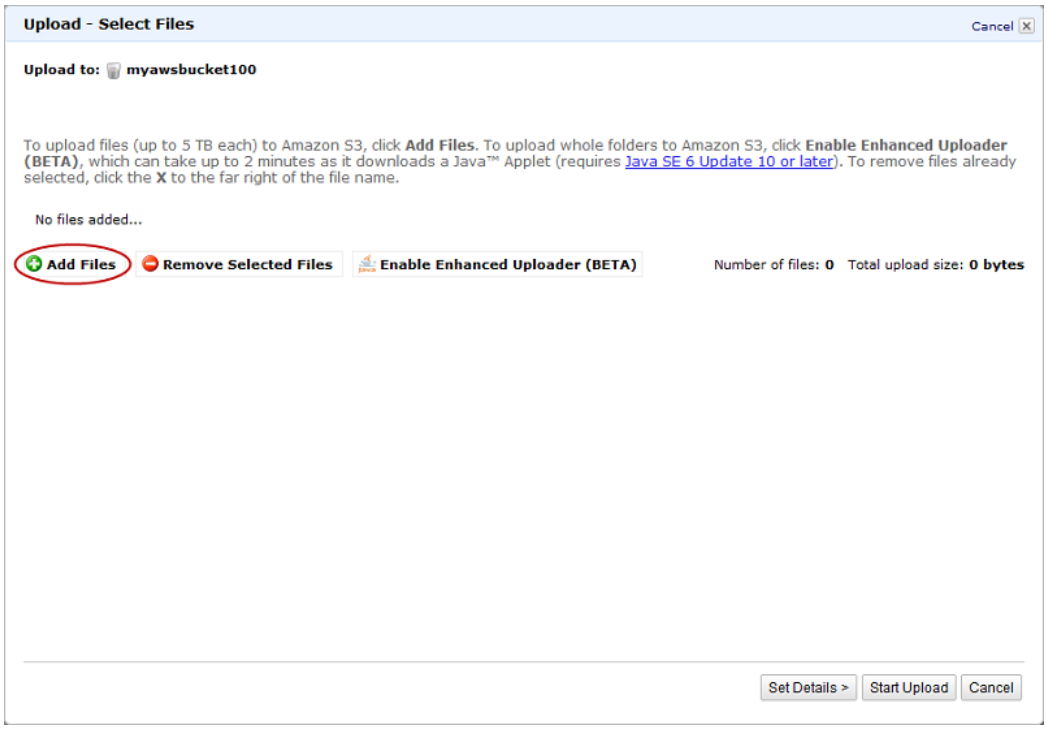
Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Step 2: Create an Amazon S3 Bucket or Two, and Upload a Media File

Create an Amazon S3 bucket for the files that you want to transcode (the input bucket) and another bucket for the transcoded files (the output bucket). You can also use the same bucket for the input bucket and the output bucket.

To create Amazon S3 buckets and upload a media file

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the Amazon S3 console, click **Create Bucket**.
3. In the **Create Bucket** dialog box, enter a bucket name. If you want to create separate input and output buckets, give the bucket an appropriate name.
4. Select a region for your bucket. By default, Amazon S3 creates buckets in the US Standard region. We recommend that you choose a region close to you to optimize latency, minimize costs, or to address regulatory requirements. This is also the region in which you want Elastic Transcoder to do the transcoding.
5. Click **Create**.
6. If you want to create separate buckets for the files that you are transcoding and the files that Elastic Transcoder has finished transcoding, repeat Step 2 through Step 5.
7. In the **Buckets** pane, click the name of your input bucket.
8. Click **Actions** and then click **Upload**.
9. On the **Upload - Select Files** page, click **Add Files**, and upload a media file that you want to transcode.



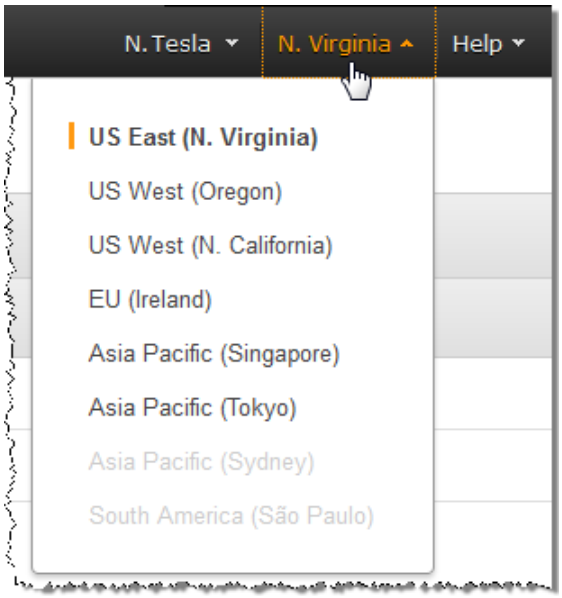
10. Click **Start Upload**.

Step 3: Create a Pipeline

A pipeline manages the jobs that transcode your files. Typically, you'll create two or more pipelines, for example, to manage standard-priority jobs and high-priority jobs. In this example, you'll create a single pipeline. For more information about pipelines, see Working with Pipelines (p. 9).

To create a pipeline using the Elastic Transcoder console

1. Open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. In the navigation bar of the Elastic Transcoder console, select the region in which you want to create the pipeline.



3. In the left pane of the console, click **Pipelines**.
4. On the **Pipelines** page, click **Create New Pipeline**.
5. Enter the applicable values. For more information about each field, see Settings that You Specify When You Create an Elastic Transcoder Pipeline (p. 12).

   Note
   If you want to monitor the progress of your job, specify the Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder begins processing a job and when Elastic Transcoder has completed the job.

6. Click **Create Pipeline**.

Step 4: (Optional) Create a Preset

A preset is a template that contains the settings that you want Elastic Transcoder to apply during the transcoding process, for example, the codec and the resolution that you want in the transcoded file. When you create a job, you specify which preset you want to use. We provide system presets that create media files that play on many common devices and system presets that target specific devices. For maximum compatibility, choose a preset that creates output that plays on a wide range of devices. For optimum quality and file size, choose a preset that creates output for a specific device or class of devices. For the current list of system presets, see the list of presets on the Elastic Transcoder detail page.

If none of the system presets contain the settings that you need, you can create your own presets. For more information, see Creating a Preset in Elastic Transcoder (p. 39).

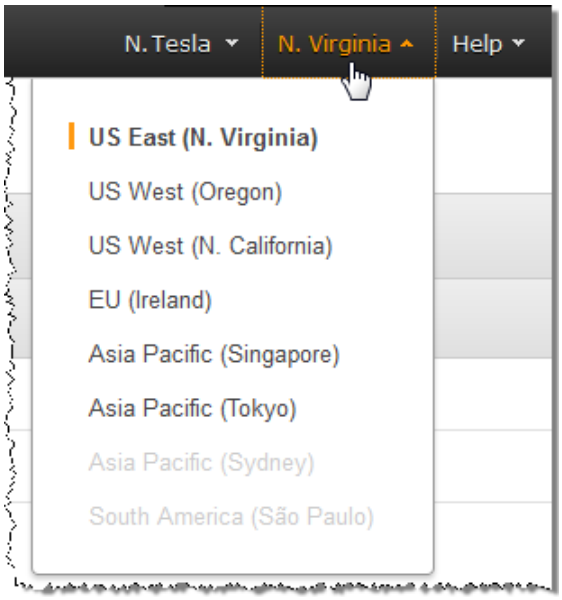
Step 5: Create a Job

A job does the work of transcoding. You specify the name of the file that you want to transcode (the input file), the name that you want Elastic Transcoder to give the transcoded file, the preset that you want Elastic Transcoder to use, and a few other settings. Elastic Transcoder gets the input file from the Amazon S3 input bucket that you specified in your pipeline, transcodes the file, and saves the transcoded file or files in the Amazon S3 output bucket that you specified in the pipeline.

For more information about jobs, see Working with Jobs (p. 17).

To create a job using the Elastic Transcoder console

1. Open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. In the navigation bar of the Elastic Transcoder console, select the region in which you want to create the job.



3. In the left pane of the console, click **Pipelines**. (You create the job in the pipeline—the queue—that you want to use to transcode the file.)
4. On the **Pipelines** page, click **Create New Job**.
5. Enter the applicable values. For more information about each field, see Settings that You Specify When You Create an Elastic Transcoder Job (p. 20).
6. Click **Create Job**.

Step 6: Monitor the Progress of Your Job

When you created a pipeline in Step 3: Create a Pipeline (p. 6), you had the option to configure notifications, so Elastic Transcoder sends a message to an Amazon Simple Notification Service (Amazon SNS) topic when Elastic Transcoder begins processing a job and finishes processing a job. If you configured notifications and if you subscribed to the applicable Amazon SNS topic, you can monitor the progress of your job.

Working with Pipelines

Topics

- Creating a Pipeline in Elastic Transcoder (p. 9)
- Pausing and Reactivating Pipelines in Elastic Transcoder (p. 10)
- Updating Pipeline Notifications in Elastic Transcoder (p. 10)
- Listing and Viewing Pipelines in Elastic Transcoder (p. 11)
- Deleting an Elastic Transcoder Pipeline (p. 11)
- Settings that You Specify When You Create an Elastic Transcoder Pipeline (p. 12)

Pipelines are queues that manage your transcoding jobs. When you create a job, you specify the pipeline to which you want to add the job. Elastic Transcoder starts processing the jobs in a pipeline in the order in which you added them.

One common configuration is to create two pipelines—one for standard-priority jobs, and one for high-priority jobs. Most jobs go into the standard-priority pipeline; you use the high-priority pipeline only when you need to transcode a file immediately.

If there are other jobs in a pipeline when you create a job, Elastic Transcoder starts processing the new job when resources are available. A pipeline can process more than one job simultaneously, and the time required to complete a job varies significantly based on the size of the file you're converting and the job specifications. As a result, jobs don't necessarily complete in the order in which you create them.

You can temporarily pause a pipeline so it stops processing jobs. This is useful if you want to cancel one or more jobs, which you can do only until Elastic Transcoder starts processing the jobs.

Creating a Pipeline in Elastic Transcoder

You can create a pipeline using the AWS Management Console or using the Elastic Transcoder Create Pipeline API action. The following procedure explains how to create a pipeline using the console. For information about how to create a pipeline using the API, see Create Pipeline (p. 128).

To create a pipeline using the Elastic Transcoder console

1. Sign in to the AWS Management Console and open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.

2. In the navigation bar of the Elastic Transcoder console, select the region in which you want to create the pipeline. This region is where you'll create your jobs.
3. In the navigation (left) pane of the console, click **Pipelines**.
4. On the **Pipelines** page, click **Create New Pipeline**.
5. Enter the applicable values. For more information about each field, see Settings that You Specify When You Create an Elastic Transcoder Pipeline (p. 12).
6. Click **Create Pipeline**.

Pausing and Reactivating Pipelines in Elastic Transcoder

If you want to cancel a job, we recommend that you first pause the corresponding pipeline so Elastic Transcoder doesn't start processing the job. After the status of a job changes from **Submitted** to **Progressing**, you can't cancel it.

The following procedure explains how to pause and reactivate a pipeline by using the console. For information about how to pause and reactivate a pipeline by using the API, see Update Pipeline Status (p. 167).

To pause or reactivate a pipeline using the Elastic Transcoder console

1. Sign in to the AWS Management Console and open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. In the navigation bar of the Elastic Transcoder console, select the region in which you want to pause or reactivate a pipeline.
3. In the navigation (left) pane, click **Pipelines**.
4. Select the check box next to the pipeline that you want to pause or reactivate.
5. Click **Pause** or **Activate** as applicable.

Updating Pipeline Notifications in Elastic Transcoder

When you create a pipeline, you can optionally configure Elastic Transcoder to send a message to an Amazon Simple Notification Service (Amazon SNS) topic when the status of a job changes, including when Elastic Transcoder starts or finishes processing a job, and when Elastic Transcoder encounters a warning or error condition while processing a job. You can change whether you want Elastic Transcoder to send a message, and, if so, you can change which SNS topic to send the message to.

Amazon SNS offers a variety of notification options, including the ability to send Amazon SNS messages to Amazon Simple Queue Service (Amazon SQS) queues. For more information, see the Amazon Simple Notification Service Developer Guide.

The following procedure explains how to update notifications using the console. For information about how to update notifications using the API, see Update Pipeline Notifications (p. 170).

To update pipeline notifications using the Elastic Transcoder console

1. Sign in to the AWS Management Console and open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.

2. In the navigation bar of the Elastic Transcoder console, select the region in which you want to pause or reactivate a pipeline.
3. In the navigation (left) pane, click **Pipelines**.
4. Select the check box next to the pipeline for which you want to change notifications.
5. Click **Edit**.
6. Change values as applicable. For more information, see Settings that You Specify When You Create an Elastic Transcoder Pipeline (p. 12).
7. Click **Save** to save your changes.

Listing and Viewing Pipelines in Elastic Transcoder

You can list the pipelines associated with the current AWS account, and you can also view the settings for a specified pipeline. The following procedure explains how to list pipelines and how to view settings for a pipeline by using the console.

For information about how to use the API to:

- List pipelines for the current AWS account, see List Pipelines (p. 138).
- Get settings for a specified pipeline, see Read Pipeline (p. 147).

To list pipelines and view pipeline settings using the Elastic Transcoder console

1. Sign in to the AWS Management Console and open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. In the navigation bar of the Elastic Transcoder console, select the region in which you want to list jobs.
3. In the navigation (left) pane of the console, click **Pipelines**. The right pane lists the pipelines that are associated with the current account.
4. To display detailed information about a pipeline, click the icon next to the pipeline.

Deleting an Elastic Transcoder Pipeline

You can delete a pipeline by using the AWS Management Console or by using the Elastic Transcoder Delete Pipeline API. The following procedure explains how to delete pipelines using the console. For information about how to delete pipelines using the API, see Delete Pipeline (p. 174).

Note
You can't delete a pipeline that contains unprocessed jobs.

To delete a pipeline using the Elastic Transcoder console

1. Sign in to the AWS Management Console and open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. In the navigation bar of the Elastic Transcoder console, select the region that contains the pipeline that you want to delete.
3. In the navigation (left) pane of the console, click **Pipelines**.
4. Select the check box for the pipeline that you want to delete.
5. Click **Remove**.

Settings that You Specify When You Create an Elastic Transcoder Pipeline

When you create a pipeline using the Elastic Transcoder console, you can specify the following values.

Topics

- Region (p. 21)
- General Settings (p. 21)
- Configure Amazon S3 Bucket for Transcoded Files and Playlists (p. 12)
- Configure Amazon S3 Bucket for Thumbnails (p. 14)
- Notifications (p. 16)

Region

Elastic Transcoder creates your pipeline in the region you are in. Choose the region you plan to run jobs in.

General Settings

Pipeline Name
: The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced. The maximum length of a pipeline name is 40 characters.

Input Bucket
: The Amazon S3 bucket in which you saved the media files that you want to transcode and the graphics files, if any, that you want to use for watermarks.

IAM Role
: The IAM Amazon Resource name (ARN) for the role that you want Elastic Transcoder to use to transcode jobs for this pipeline.

AWS KMS Key ARN
: The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

  If you use either **S3** or **S3-AWS-KMS** as your **Encryption:Mode**, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an **Encryption:Mode** of **AES-PKCS7**, **AES-CTR**, or **AES-GCM**.

Configure Amazon S3 Bucket for Transcoded Files and Playlists

Use the settings in this section to specify the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists for jobs that are submitted to this pipeline, the Amazon S3 storage class that you want to assign to the files, which users you want to have access to the files, and the type of access you want users to have.

Bucket
: The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists (if applicable) for jobs that you submit to this pipeline.

Storage Class

The Amazon S3 storage class, **Standard** or **Reduced Redundancy**, that you want Elastic Transcoder to assign to the transcoded files and playlists that it stores in your Amazon S3 bucket. For more information, see Reduced Redundancy Storage in the *Amazon Simple Storage Service Developer Guide*.

Grantee Type

Specify how you want to identify the users or groups that you want to have access to transcoded files and playlists. When you select a grantee type, the Grantee Type field appears:

- **Canonical:** The value of **Grantee** is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see Access Control List (ACL) Overview in the *Amazon Simple Storage Service Developer Guide*. For more information about using CloudFront origin access identities to require that users use CloudFront URLs instead of Amazon S3 URLs, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content in the *Amazon CloudFront Developer Guide*.

  Important
  A canonical user ID is not the same as an AWS account number.

- **Email:** The value of **Grantee** is the registered email address of an AWS account.
- **Group:** The value of **Grantee** is one of the following predefined Amazon S3 groups: **AllUsers**, **AuthenticatedUsers**, or **LogDelivery**.

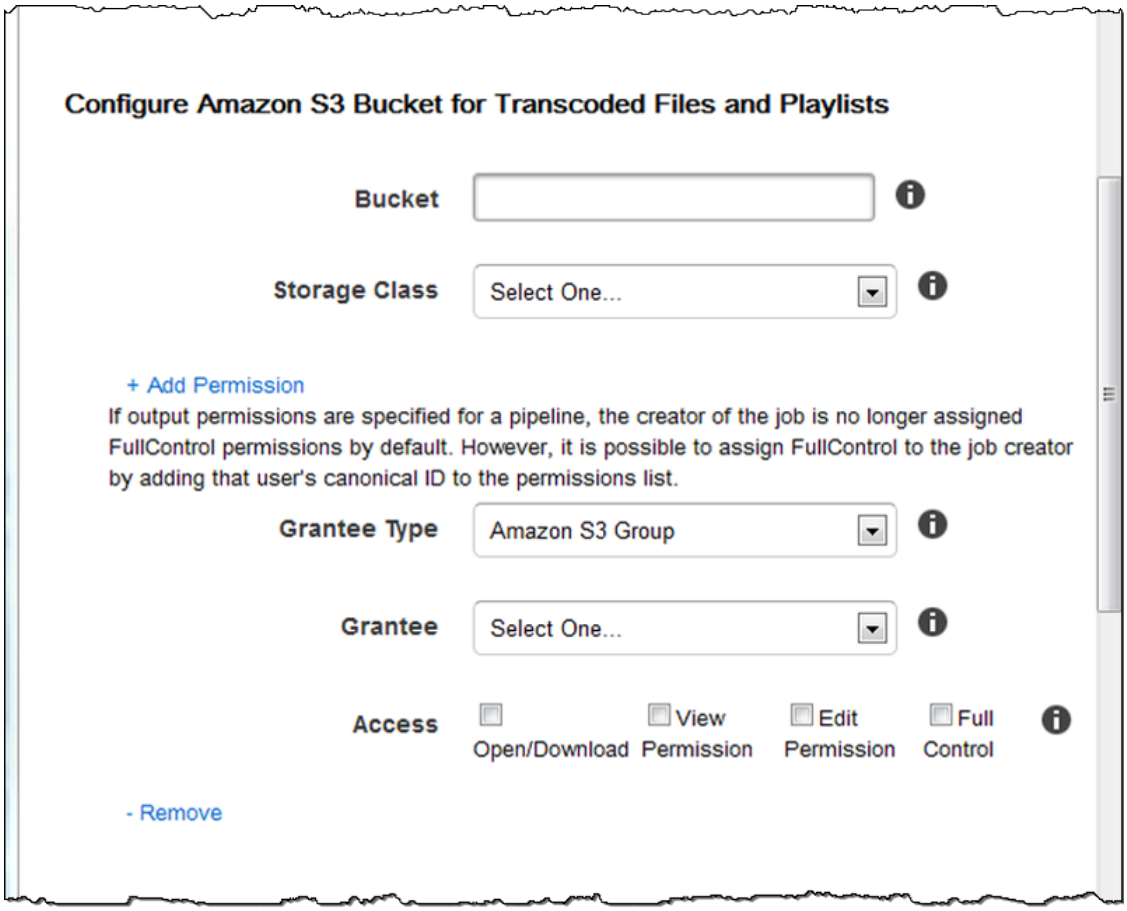
Grantee

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group. For more information, see Grantee Type.

Access

The permission that you want to give to the AWS user that you specified in Grantee. Permissions are granted on the transcoded files and playlists that Elastic Transcoder adds to the bucket. Valid values include:

- **Open/Download:** The grantee can read the objects and metadata for transcoded files and playlists that Elastic Transcoder adds to the Amazon S3 bucket.
- **View Permissions:** The grantee can read the object ACL for transcoded files and playlists that Elastic Transcoder adds to the Amazon S3 bucket.
- **Edit Permissions:** The grantee can write the ACL for transcoded files and playlists that Elastic Transcoder adds to the Amazon S3 bucket.
- **Full Control:** The grantee has permissions to read objects, and view and edit the ACL for transcoded files and playlists that Elastic Transcoder adds to the Amazon S3 bucket.



Configure Amazon S3 Bucket for Thumbnails

Use the settings in this section to specify the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnails for jobs that are submitted to this pipeline, the Amazon S3 storage class that you want to assign to the thumbnails, which users you want to have access to the thumbnails, and the type of access you want users to have.

Bucket
The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnails for jobs that you submit to this pipeline.

Storage Class
The Amazon S3 storage class, **Standard** or **Reduced Redundancy**, that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket. For more information, see Reduced Redundancy Storage in the *Amazon Simple Storage Service Developer Guide*.

Grantee Type
Specify how you want to identify the users or groups that you want to have access to thumbnails. When you select a grantee type, the Grantee Type field appears:

- **Canonical:** The value of **Grantee** is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see Access Control List (ACL) Overview in the *Amazon Simple Storage Service Developer Guide*. For more information about using CloudFront origin access identities to require that users use CloudFront URLs instead of Amazon S3 URLs, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content in the *Amazon CloudFront Developer Guide*.

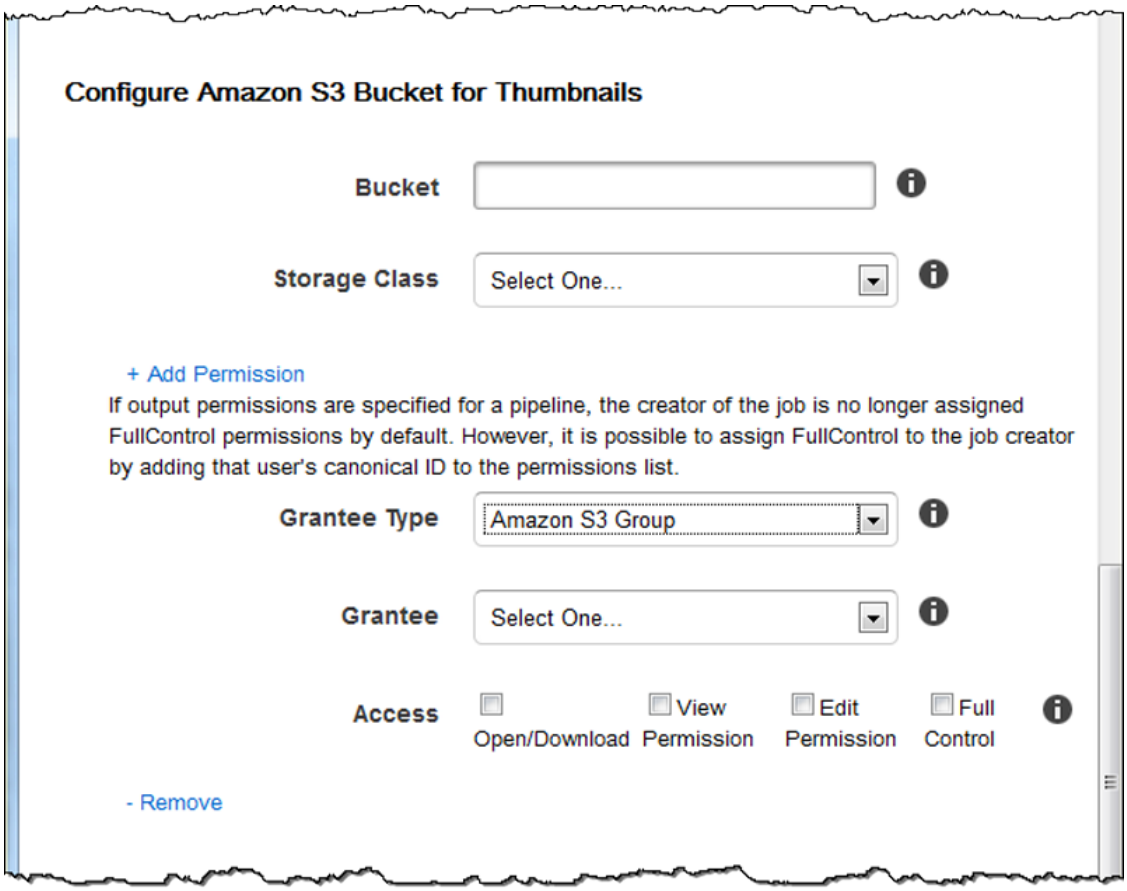
Important
A canonical user ID is not the same as an AWS account number.

- **Email:** The value of **Grantee** is the registered email address of an AWS account.
- **Group:** The value of **Grantee** is one of the following predefined Amazon S3 groups: **AllUsers**, **AuthenticatedUsers**, or **LogDelivery**.

Grantee
The AWS user or group that you want to have access to thumbnails. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group. For more information, see Grantee Type.

Access
The permission that you want to give to the AWS user that you specified in Grantee. Permissions are granted on the thumbnails that Elastic Transcoder adds to the bucket. Valid values include:

- **Open/Download:** The grantee can read the objects and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **View Permissions:** The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **Edit Permissions:** The grantee can write the ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **Full Control:** The grantee has permissions to read objects, and view and edit the ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.



Notifications

Use the settings in this section to configure Elastic Transcoder to notify you when the status of a job changes.

On Progressing Event
The Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process the job.

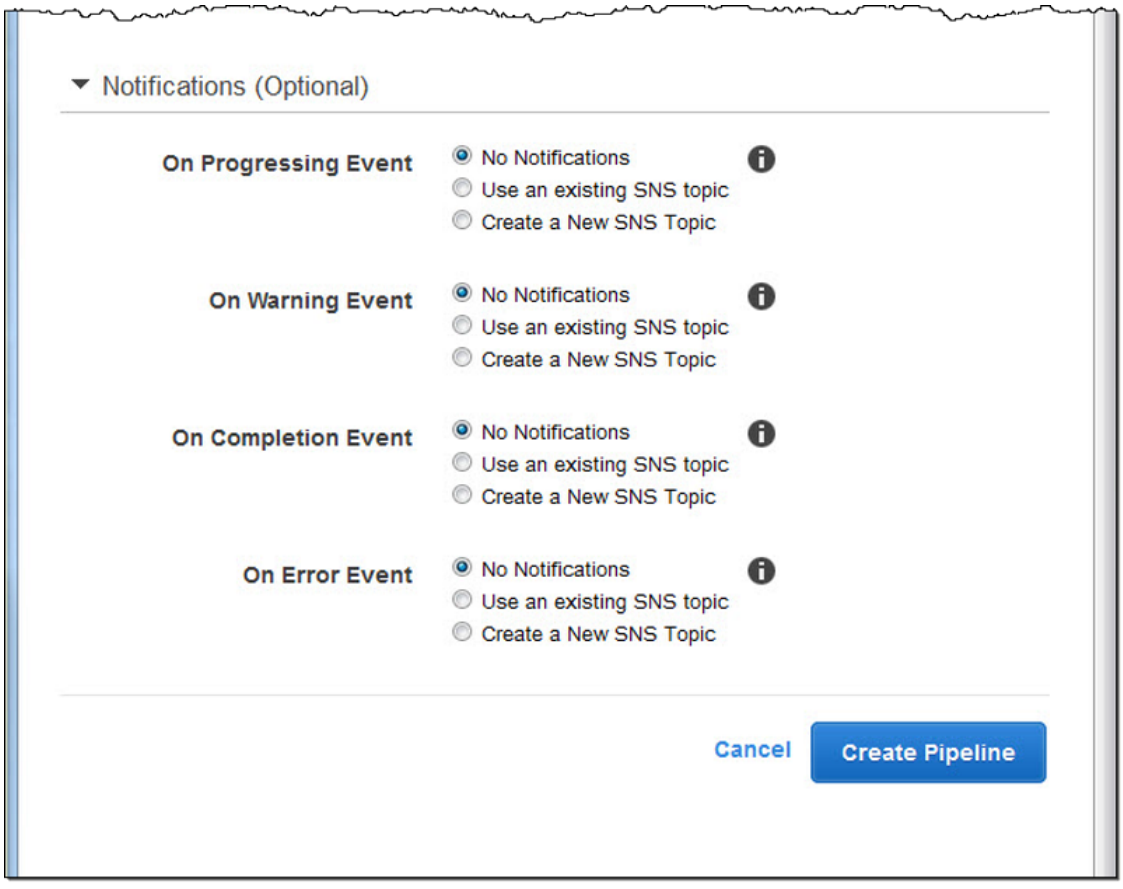
> **Important**
> To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

Amazon SNS offers a variety of notification options, including the ability to send Amazon SNS messages to Amazon Simple Queue Service (Amazon SQS) queues. For more information, see the Amazon Simple Notification Service Developer Guide.

On Completed Event
The Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing the job.

On Warning Event
The Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition.

On Error Event
The Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition.



Working with Jobs

Topics

- Creating a Job in Elastic Transcoder (p. 17)
- Listing Jobs and Viewing Job Settings in Elastic Transcoder (p. 18)
- Canceling an Elastic Transcoder Job (p. 19)
- Settings that You Specify When You Create an Elastic Transcoder Job (p. 20)

A job does the work of transcoding a media file from one format into another format. When you create a job, you specify the information that Elastic Transcoder needs to perform the transcoding: which file to transcode, what to name the transcoded file, which preset to use (a preset is a template that contains the settings that you want Elastic Transcoder to apply), and so on.

Creating a Job in Elastic Transcoder

You can create a job using the AWS Management Console or using the Elastic Transcoder Create Job API action. The following procedure explains how to create a job by using the console. For information about how to create a job using the API, see Create Job (p. 181).

You can configure Elastic Transcoder to notify you when the status of a job changes, including when Elastic Transcoder starts and finishes processing a job, and when it encounters a warning or error condition. For more information, see Creating a Pipeline in Elastic Transcoder (p. 9).

You cannot update a job after you have created it. If you need to change settings in a job, cancel it, create a new job based on the one that you canceled, update the applicable values, and create the new job.

> **Note**
> Before you can create a job, you must create the pipeline (the queue) that will manage the job. For more information about creating a pipeline, see Creating a Pipeline in Elastic Transcoder (p. 9). In addition, if you want to transcode a file using settings other than those provided in the Elastic Transcoder default presets, you must create a new preset. For more information about creating a preset, see Creating a Preset in Elastic Transcoder (p. 39).

To create a job using the Elastic Transcoder console

1. Sign in to the AWS Management Console and open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.

2. In the navigation bar of the Elastic Transcoder console, select the region corresponding to the pipeline that you want to use for the transcoding job. Pipelines are available only in the regions where they were created.
3. In the navigation (left) pane of the console, click **Pipelines**. (You create the job in the pipeline—the queue—that you want to use to transcode the file.)
4. On the **Pipelines** page, click **Create New Job**.
5. Enter the applicable values. For more information about each field, see Settings that You Specify When You Create an Elastic Transcoder Job (p. 20).
6. Click **Create Job**. Jobs start as soon as they are created.

   Note
   If a job fails with an `Access Denied` error, we recommend that you run the `Test Role` API action to determine what is causing the error. For more information, see Test Role (p. 177).

Listing Jobs and Viewing Job Settings in Elastic Transcoder

You can list the jobs in a specified pipeline or with a specified status either by using the Elastic Transcoder console or by using the applicable API action. You can also view the settings for an individual job. The following procedure explains how to list jobs and how to view settings for a job by using the console.

Note
When you list jobs by pipeline, Elastic Transcoder lists all of the jobs that you've ever created for that pipeline. When you list jobs by status, Elastic Transcoder lists all of the jobs that you've ever created that currently have the specified status.

For information about how to use the API to:

- List jobs in a specified pipeline, see List Jobs by Pipeline (p. 210).
- List jobs that have a specified status, see List Jobs by Status (p. 230).
- Get settings for a specified job, see Read Job (p. 251).

Note
If you have specified more than one output for your jobs (for example, one output for the Kindle Fire and another output for the Apple iPhone 4s), you currently must use the Elastic Transcoder API to list the jobs.

To list jobs and view job settings using the Elastic Transcoder console

1. Sign in to the AWS Management Console and open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. In the navigation bar of the Elastic Transcoder console, select the region in which you want to list jobs.
3. In the navigation (left) pane of the console, click **Jobs**.
4. On the **Jobs** page, specify the applicable values. For more information about a field, click the icon next to the field.
5. Click **Search**.
6. To display detailed information about a job that is listed in the search results, click the icon next to the job.

Canceling an Elastic Transcoder Job

You can cancel a job that still has a status of **Submitted**, which means that Elastic Transcoder hasn't started to transcode your file. The following procedure explains how to cancel a job using the Elastic Transcoder console.

To cancel a job using the API, pause the corresponding pipeline so Elastic Transcoder doesn't start processing the job, list jobs that have a status of **Submitted** to get the applicable job ID, then cancel the job using the job ID to identify which job you want to cancel. For more information, see:

- Update Pipeline Status (p. 167)
- List Jobs by Status (p. 230)
- Cancel Job (p. 271)

To cancel a job using the Elastic Transcoder console

1. Sign in to the AWS Management Console and open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. In the navigation bar of the Elastic Transcoder console, select the region in which you want to cancel a job.
3. **Optional but recommended:** Pause the pipeline to which you submitted the job, so Elastic Transcoder doesn't begin to process the job. You can't cancel a job after Elastic Transcoder begins to process it.

   a. In the navigation (left) pane, click **Pipelines**.
   b. Select the check box next to the pipeline that you want to pause.
   c. Click **Pause**.

4. In the navigation pane of the console, click **Jobs**.
5. On the **Jobs** page, specify the following values:

   Search By
   Click **Status**.

   Job Status
   Select **Submitted**.

   > **Note**
   > You can only cancel a job that has a status of **Submitted**.

   For **Order** and **Number of Jobs**, enter the applicable values.
6. Click **Search**.
7. In the search results, if you need to view more details about a job to determine whether it's the one you want to cancel, click the icon next to the job.
8. To cancel a job, select the check box next to the job, and click **Cancel**.
9. If you paused the pipeline in Step 3, reactivate it so it resumes processing jobs.

   a. In the navigation pane, click **Pipelines**.
   b. Select the check box next to the pipeline that you want to reactivate.
   c. Click **Activate**.

Settings that You Specify When You Create an Elastic Transcoder Job

When you create a job using the Elastic Transcoder console, you can specify the following values.

> **Note**
> You can configure Elastic Transcoder to notify you when the status of a job changes, including when Elastic Transcoder starts and finishes processing a job, and when Elastic Transcoder encounters a warning or error condition. For more information, see Notifications of Job Status (p. 66).

Topics

- Region (p. 21)
- General Settings (p. 21)
- Output Details, Part 1 (p. 23)
- Output Details, Part 2, Clip and Caption Settings (p. 25)
- Output Details, Part 3 (p. 27)
- (Optional) Output Encryption (p. 28)
- (Video/Thumbnails Only) Watermarks (p. 30)
- (AAC/MP3/MP4 Only) Album Art (p. 31)
- (Optional) User Metadata (p. 34)
- (Fragmented MP4/MPEG-TS Outputs Only) Playlist (p. 35)
- (Fragmented MP4/MPEG-TS Outputs Only, Optional) HLS Content Protection (p. 36)
- (Optional) Additional Job Details (p. 38)

Region

Elastic Transcoder creates your job in the region you are in.

General Settings

Pipeline
The name of the pipeline that you want Elastic Transcoder to use for transcoding. The pipeline determines several settings, including the Amazon S3 bucket from which Elastic Transcoder gets the files to transcode and the bucket into which Elastic Transcoder puts the transcoded files.

Input Key
The name of the file that you want to transcode. To determine from which Amazon S3 bucket to get the file, Elastic Transcoder refers to the **Input Bucket** field in the pipeline that you specified for this job.

If the file name includes a prefix—for example, `cooking/lasagna.mpg`—include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

Output Key Prefix
The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, and playlists. If you specify a value, it must contain a */* somewhere after the first character, which simplifies Amazon S3 file management.

Input Decryption
The encryption settings, if any, that are used for decrypting your input files. If your input file is encrypted, you must specify the mode that Elastic Transcoder will use to decrypt your file.

Decryption Mode (Required for Decryption)
The specific encryption mode that you want Elastic Transcoder to use when decrypting your files.

Elastic Transcoder supports the following options:

- **Amazon S3 Server-Side Encryption:** Amazon S3 handles the encryption and decryption of your files. As long as Elastic Transcoder has access permissions to your Amazon S3 bucket, you don't need to take any action.

  For more information, see Protecting Data Using Server-Side Encryption in the *Amazon Simple Storage Service Developer Guide*.
- **Server-Side Encryption Using Customer-Provided Keys:** Elastic Transcoder supports three types of encryption using customer-provided keys:
  - **AES-CBC-PKCS7:** A padded cipher-block mode of operation.
  - **AES-CTR:** AES Counter Mode.
  - **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

  If you chose one of the AES-encryption modes, you must also specify the following three values (all three must be base64-encoded):
  - **Encryption Key**
  - **Encryption Key MD5**
  - **Encryption Initialization Vector**

Decryption Key (Required for AES Decryption)
The data encryption key used to encrypt your file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

The key must also be encrypted by using AWS KMS. For more information, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

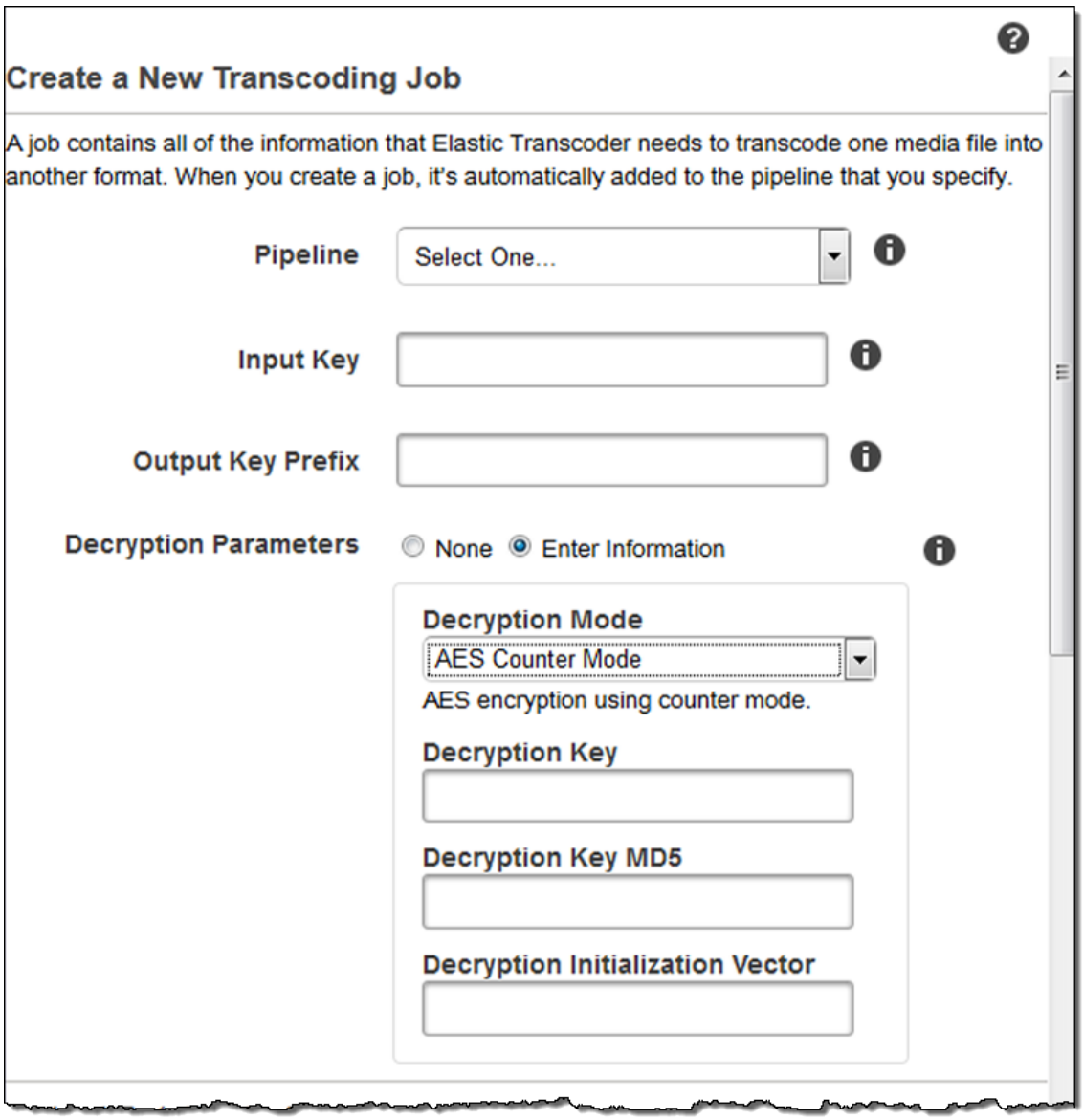
Decryption Key MD5 (Required for AES Decryption)

The MD5 digest of the key used to encrypt your input file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

Decryption Initialization Vector (Required for AES Decryption)

The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

For more information, go to Initialization Vector.



Output Details, Part 1

Use the settings in this section to specify information about the output files.

Preset
The preset that you want to use for this output. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding.

(Fragmented MP4/MPEG-TS Outputs Only) Segment Duration
If you specify a preset for the current output for which the value of **Container** is either **ts** (MPEG-TS) or **fmp4** (Fragmented MP4), **Segment Duration** is the target maximum duration of each segment in seconds. For **HLSv3** format playlists, each media segment is stored in a separate `.ts` file. For **HLSv4** and **Smooth** playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the **Segment Duration**, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by **Segment Duration**, the duration of the last segment is the remainder of:

```
total length/Segment Duration
```

Elastic Transcoder creates an output-specific playlist for each **HLS** output that you specify in **Output Keys**. To add an output to a master playlist for this job, include it in Outputs in Master Playlist.

Elastic Transcoder applies this segmenting to any captions associated with the output video.

For more information, see HTTP Live Streaming or Smooth Streaming Technical Overview.

Output Key
The name that you want Elastic Transcoder to assign to the transcoded file and playlist. Elastic Transcoder saves the file or files in the Amazon S3 bucket specified by the **Bucket** field in the pipeline that you specify in Pipeline (p. 21). If the bucket already contains a file that has the specified name, the output fails. However, other outputs in the same job might succeed.

The format for file names depends on the container type and whether the segment duration is set. If the container type is not `ts` or the segment duration is not provided, the name of the output file is a concatenation of **Output Key Prefix** and **Output Key**.

If the container type is `ts` and segment duration is provided, Elastic Transcoder uses the value of **Output Key** to name both the playlist for the output and the `.ts` files:

- **Playlist:**
  - **HLSv3:** The file name is a concatenation of **Output Key Prefix** and **Output Key** plus the file name extension **.m3u8**:

    Output Key Prefix**Output Key**.m3u8
  - **HLSv4:** The file name is a concatenation of **Output Key Prefix** and **Output Key** plus the file name extension **_v4.m3u8**. Video outputs create a second file with a file name that is a concatenation of **Output Key Prefix** and **Output Key** plus the file name extension **_iframe.m3u8**:

    Output Key Prefix**Output Key**_v4.m3u8

    Output Key Prefix**Output Key**_iframe.m3u8 (Video only)
- **Segment (.ts) files:**
  - **HLSv3:** The file name is a concatenation of **Output Key Prefix** and **Output Key**, plus a five-digit sequential counter beginning with **00000**, and the file name extension **.ts**:

    Output Key Prefix**Output Key**00000.ts
  - **HLSv4:** The file name is a concatenation of **Output Key Prefix** and **Output Key** plus the file name extension **.ts**:

Output Key Prefix**Output Key**.ts

If a segmented `ts` output is not included in a master playlist, Elastic Transcoder treats the output as **HLSv3**.

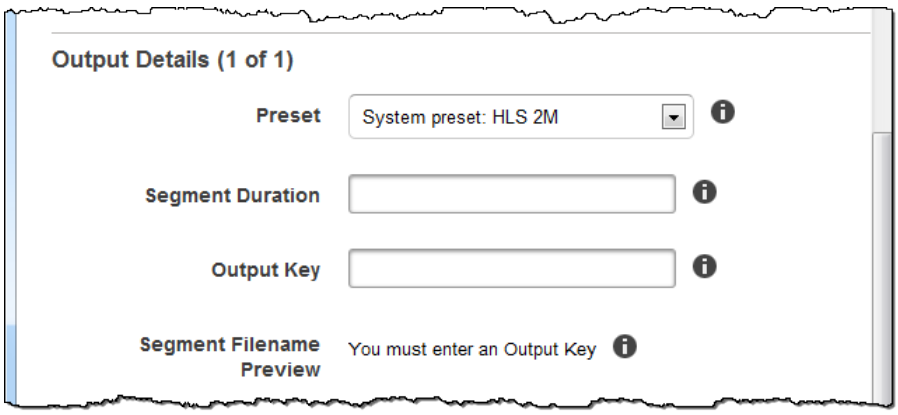
Note
Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for **HLSv3** playlists, **_v4.m3u8** and **_iframe.m3u8** for **HLSv4** playlists). If you include a file extension in **Output Key**, the file name will have two extensions.

Output Key Prefix groups all of the files for a job together in your Amazon S3 bucket. If you want to group the files for each output within a job, you can include a prefix in the value of **Output Key**, for example:

OutputKeyPrefix**iPhone/Key**00000.ts

OutputKeyPrefix**KindleFireHD/Key**00000.ts

Segment Filename Preview
The name of the first segment file, based on the value that you entered for Output Key.



If you are not generating clips or transcoding captions, you can skip to Output Details, Part 3 (p. 27).

Output Details, Part 2, Clip and Caption Settings

Use the settings in this section to specify information about clips and captions for the output files.

For more information on captions, see Captions (p. 68)

Clip Start Time
You can create an output file that contains an excerpt from the input file. **Clip Start Time** indicates the place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

Clip Duration
The duration of your excerpt clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from **Clip Start Time** to the end of the file.

If you specify a value longer than the duration of the input file, Elastic Transcoder transcodes from **Clip Start Time** to the end of the file and returns a warning message.

(Video Only) Captions Merge Policy
A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If **Caption Source** is empty, Elastic Transcoder omits all sidecar captions from the output files.
- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in **Caption Source**.

(Video Only, Optional) Input Key
The name of the sidecar caption file that you want Elastic Transcoder to include with the outputs.

(Video Only) Language
A string that specifies the language of the caption in one of the following formats:

- 2-character ISO 639-1 code, for example, **en** for English
- 3-character ISO 639-2 code, for example, **eng** for English

For more information on ISO language codes, see List of ISO 639-1 codes.

(Video Only, Optional) TimeOffset
For clip generation or captions that do not start at the same time as the associated video file, the **TimeOffset** tells Elastic Transcoder how much of the video to encode before including captions.

Specify the TimeOffset in the form [+-]SS.sss or [+-]HH:mm:SS.ss.

(Video Only, Optional) Label
The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

(Video Only) Caption Format
The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption for this output. If you leave this value blank, Elastic Transcoder returns an error.

- **Embedded Caption Formats:** For MP4 containers, mov-text is supported. For other container types, no embedded caption formats are supported.
- **Sidecar Caption Formats:** Elastic Transcoder supports dfxp, scc, srt, and webvtt. For fmp4 only dfxp is supported, and Elastic Transcoder creates a file with the extension `.ismt`. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

(Video Only) Captions Filename Pattern

The prefix for caption filenames, in the form *`description`*`-{language}`, where:

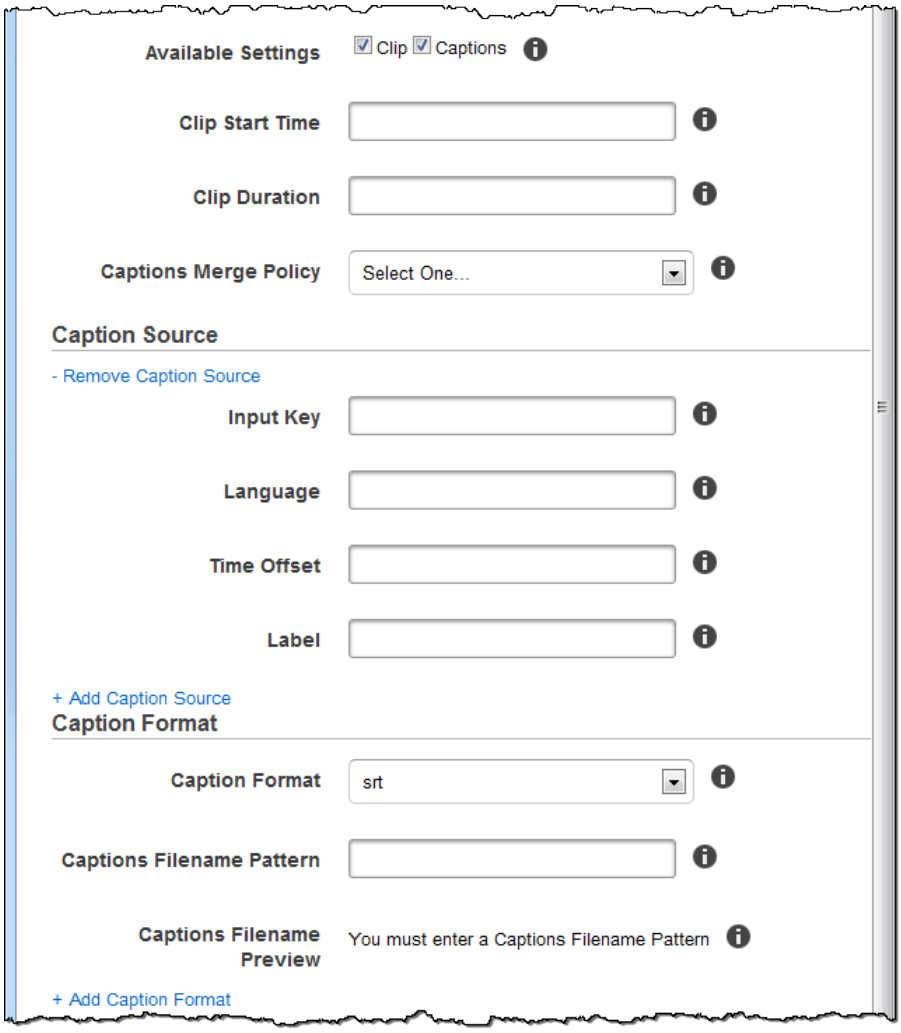
- *`description`* is a description of the video.
- `{language}` is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include `{language}` in the file name pattern, Elastic Transcoder automatically appends "`{language}`" to the value that you specify for the *`description`*. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When, you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be `Sydney-en-sunrise00000.srt`.

(Video Only) Captions Filename Preview

The name of the first caption file, based on the value that you entered for **Captions Filename Pattern**. The preview uses the English ISO code "en" to show how `{language}` will look in your file name.



Output Details, Part 3

Use the settings in this section to specify information about the output files.

(Video Only) Create Thumbnails
If you want Elastic Transcoder to create thumbnails for your videos, select **Yes**, and specify the format for the file names in the **Thumbnail Filename Pattern** field.

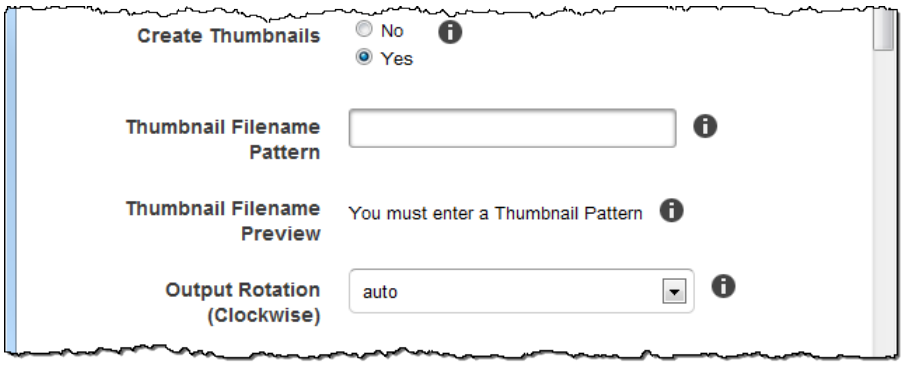
(Video Only) Thumbnail Filename Pattern
If you selected **Yes** for **Thumbnail Filename Pattern**, specify the format for the file names. You can specify the following values in any sequence:

- **{count} (Required):** A five-digit number beginning with **00001** that indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file. You must include **{count}** somewhere in the field. If you omit it, Elastic Transcoder automatically appends the count to the end of the file name, immediately before the file name extension (.jpg or .png).
- **(Optional) Literal values:** You can specify literal values anywhere in the field, for example, as a file name prefix or as a delimiter between `{resolution}` and `{count}`.
- (Optional) **{resolution}:** If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the field.

The **Thumbnail Filename Preview** field displays a sample of file names for thumbnails based on the value that you entered in **Thumbnail Filename Pattern**.

When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in Preset (p. 23). Elastic Transcoder also appends the applicable file name extension.

(Video/Thumbnails Only, Optional) Output Rotation
The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. If you want Elastic Transcoder to automatically detect whether the input file is rotated, select **auto**. Note, however, that Elastic Transcoder generally can only detect whether the output needs to be rotated if the file that you're transcoding contains rotation metadata.



(Optional) Output Encryption

You can specify encryption settings for any output files that you want to use for a transcoding job. This includes the output file and any watermarks, thumbnails, album art, or captions that you want to use. You must specify encryption settings for each file individually.

Output Encryption
The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

(Required for file-level Encryption) Encryption Mode
The specific server-side encryption mode that you want Elastic Transcoder to use when encrypting your output files individually. Elastic Transcoder supports the following **Encryption Mode** options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.

  For more information, see Protecting Data Using Server-Side Encryption in the *Amazon Simple Storage Service Developer Guide*.
- **S3-AWS-KMS:** Amazon S3 calls AWS KMS, which creates and manages the keys that are used for encrypting your files. If you specify **S3-AWS-KMS** and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

  For more information, see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys in the *Amazon Simple Storage Service Developer Guide*.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

If you chose one of the AES-encryption modes, you must also specify the following three values (all three must be base64-encoded):

- **Encryption Key**
- **Encryption Key MD5**
- **Encryption Initialization Vector**

If you chose one of the AES-encryption modes, and you want Elastic Transcoder to generate a 128-bit bit AES encryption key for you, do not specify values for the **Encryption Key**, **Encryption Key MD5**, or **Encryption Initialization Vector**.

> **Important**
> For the AES modes, your media-specific private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to decrypt your data.

(Optional) Encryption Key
If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

128, 192, or 256.

For more information about encrypting your key with AWS KMS, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Required if an Encryption Key is supplied) Encryption Key MD5
The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in

transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

(Required if an Encryption Key is supplied) Encryption Initialization Vector

The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

For more information, go to Initialization Vector.

(Video/Thumbnails Only) Watermarks

Information about the watermarks that you want Elastic Transcoder to add to the video during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in **Preset** for the current output.

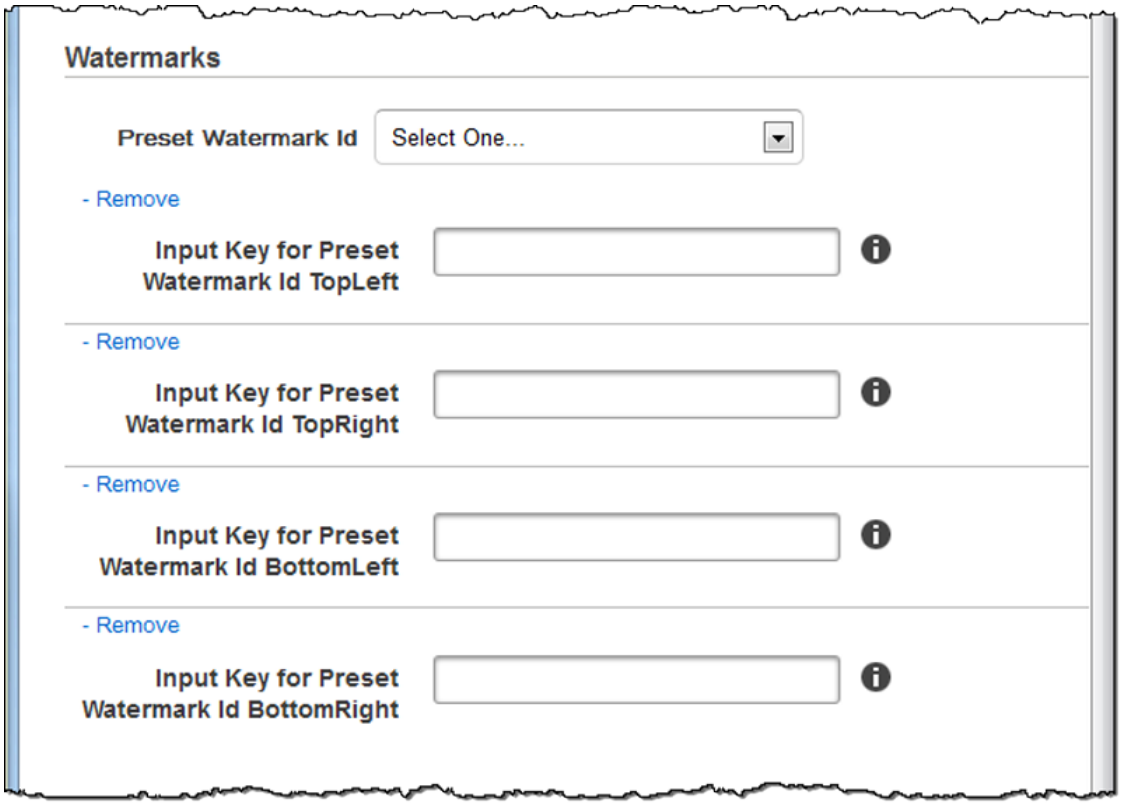
Watermarks are added to the output video in the sequence in which you list them in the job output—the first watermark in the list is added to the output video first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you add will cover the first one, the third one will cover the second, and the fourth one will cover the third.

For more information about watermarks, see Watermarks (p. 69).

Preset Watermark ID
: The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the video during transcoding. The settings are in the preset specified by **Preset** for the current output. In that preset, the value of **Watermarks Id** tells Elastic Transcoder which settings to use.

Input Key for Preset Watermark Id
: The name of the .png or .jpg file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by **Pipeline**; the **Input Bucket** object in that pipeline identifies the bucket.

  If the file name includes a prefix, for example, **logos/128x64.png**, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.



(AAC/MP3/MP4 Only) Album Art

The album art to be associated with the output file, if any.

To remove artwork or leave the artwork empty, you can either set **Artwork** to null, or set the **Merge Policy** to "Replace" and use an empty **Artwork** array.

To pass through existing artwork unchanged, set the **Merge Policy** to "Prepend", "Append", or "Fallback", and use an empty **Artwork** array.

> **Note**
> Album Art is only available for containers of type `AAC`, `mp3`, or `mp4`.

Album Art Merge Policy
A policy that determines how Elastic Transcoder will handle the existence of multiple album artwork files.

- **Replace:** The specified album art will replace any existing album art.
- **Prepend:** The specified album art will be placed in front of any existing album art.
- **Append:** The specified album art will be placed after any existing album art.
- **Fallback:** If the input file contains artwork, Elastic Transcoder will use that artwork for the output. If the input does not contain artwork, Elastic Transcoder will use the specified album art file.

Album Art Artwork
The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

Album Art Input Key
The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by **PipelineId**; the **InputBucket** object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

Album Art Format
The format of album art, if any. Valid formats are `jpg` and `png`.

Album Art Max Width
The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

Album Art Max Height
The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

Album Art Sizing Policy
A value that controls scaling of the output album art:

- **Fit:** Elastic Transcoder scales the output art so it matches the value that you specified in either **MaxWidth** or **MaxHeight** without exceeding the other value.
- **Fill:** Elastic Transcoder scales the output art so it matches the value that you specified in either **MaxWidth** or **MaxHeight** and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches the output art to match the values that you specified for **MaxWidth** and **MaxHeight**. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep:** Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for **MaxWidth** and **MaxHeight**, Elastic Transcoder crops the output art.

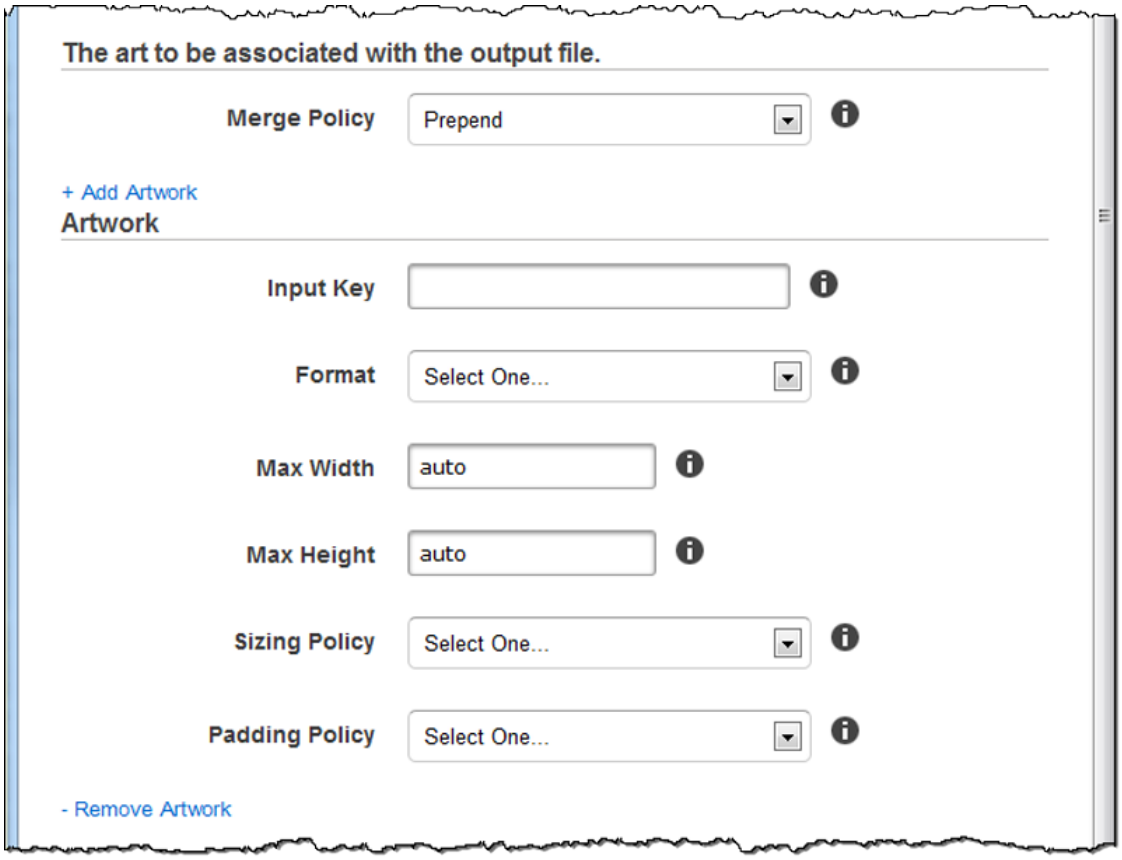
- **ShrinkToFit:** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of **MaxWidth** and **MaxHeight** without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **ShrinkToFill:** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of **MaxWidth** and **MaxHeight** without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.

The following table shows possible effects of **SizingPolicy** settings on the output album art:

SizingPolicy	Output Album Art May Be Scaled Up	Output Album Art May Be Padded When Padding Policy Is "Pad"	Output Album Art May Have a Different Pixel Aspect Ratio than Input Album Art	Output Album Art May Be Cropped
Fit	Yes	Yes		
Fill	Yes			Yes
Stretch	Yes		Yes	
Keep	Yes	Yes		Yes
ShrinkToFit		Yes		
ShrinkToFill		Yes		Yes

Album Art Padding Policy

When you set **PaddingPolicy** to `Pad`, Elastic Transcoder might add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for **MaxWidth** and **MaxHeight**. For more information, see the table at `AlbumArt:Art:SizingPolicy`.


The art to be associated with the output file.
Merge Policy
Prepend
+ Add Artwork
Artwork
Input Key
Format
Select One...
Max Width
auto
Max Height
auto
Sizing Policy
Select One...
Padding Policy
Select One...
- Remove Artwork

(Optional) User Metadata

User-defined metadata that you want to associate with an Elastic Transcoder job. You specify metadata in `key/value` pairs. You can use the `key/value` pairs to track details about a file, for example, `Season 1: Episode 3`.

You can add up to 10 key/value pairs to each job. Elastic Transcoder does not guarantee that `key/value` pairs are returned in the same order in which you specify them.

Metadata Key

The key of the metadata `key/value` pair that you want returned with the output file. Each key must be a unique string between `1-128` characters, and must use only characters from the following list:

- `0-9`
- `A-Z` and `a-z`
- `Space`
- The following symbols: `_.:/=+-%@`

You can use keys as a numbering system for organizing your metadata, for storing an extra 128 characters of metadata, or for labeling the metadata stored in the **value**. If you want to use only value metadata, you can put throw-away strings in your keys such as `key1`, and ignore the keys when you retrieve your metadata from Elastic Transcoder.

> **Important**
> You must specify unique strings for all of the keys in a job. If the same string is used for more than one key in a job, Elastic Transcoder returns only one of the key/value pairs that use that key. There is no way to guarantee which value is returned.

Metadata Value

The value of the metadata `key/value` pair that you want returned with your job. Each value must be a string between `0-256` characters, and must use only characters from the following list:

- `0-9`
- `A-Z` and `a-z`
- `Space`
- The following symbols: `_.:/=+-%@`

(Fragmented MP4/MPEG-TS Outputs Only) Playlist

If you choose a preset in the **Preset** list for which the value of **Container** is either **ts** (MPEG-TS) or **fmp4** (Fragmented MP4), use the settings in this section to specify information about the master playlists that you want Elastic Transcoder to create. We recommend that you create at most one master playlist per playlist format.

Master Playlist Name

The name that you want Elastic Transcoder to assign to a master playlist. If the name includes a `/` character, the section of the name before the last **/** must be identical for all **Playlist Names**. If you create more than one master playlist, each must have a unique name.

> **Note**
> Elastic Transcoder automatically appends the relevant file extension to the file name (**.m3u8** for **HLSv3** and **HLSv4** playlists, and **.ism** and **.ismc** for **Smooth** playlists). If you include a file extension in **Master Playlist Name**, the file name will have two extensions.

Any segment duration settings, clip settings, or caption settings must be the same for all outputs in the playlist. For **Smooth** playlists, the **Audio:Profile**, **Video:Profile**, and **Frame Rate** to **Maximum Number of Frames Between Keyframes** ratio must be the same for all outputs. For more information, see Maximum Number of Frames Between Keyframes.

Playlist Format

The format for the playlist. Valid formats include **HLSv3**, **HLSv4**, and **Smooth**.

Outputs in Master Playlist

For each output in this job that you want to include in a master playlist, the value of the **Output Key** field. If you include more than one output in a playlist, the value of **Segment Duration** for all of the outputs must be the same.

For **HLSv4** master playlists, Elastic Transcoder chooses which combinations of audio and video inputs will be linked in the output playlists. The first audio and video inputs will be linked and rendered as the default playback experience, allowing you to choose your preferred playback default. For other individual playlists in the master playlist, Elastic Transcoder chooses which audio and video bit rate combinations will provide the best playback.

(Fragmented MP4/MPEG-TS Outputs Only, Optional) HLS Content Protection

Note
HLS content protection encrypts each individual segment of a file, and Elastic Transcoder does not support HLS content protection combined with file-level encryption.

If you choose a preset in the **Preset** list for which the value of **Container** is either **ts** (MPEG-TS), you can specify encryption settings for streaming your file. You cannot combine data stream encryption with file or caption encryption. If you choose to encrypt your data stream, use the settings in this section to specify information about the data stream encryption.

In order to use HLS content protection, you must have a web server that can authenticate sessions (such as Amazon Elastic Compute Cloud), a way to distribute your streamed media files (such as Amazon CloudFront), and a way to play encrypted streamed media (such as a player-enabled browser).

Method
The content protection method for your output. The only valid value is:

`aes-128`.

This value will be written into the `method` attribute of the `EXT-X-KEY` metadata tag in the output playlist.

Key
If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

For more information about encrypting your key with AWS KMS, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

If you choose an HLS content protection method of `aes-128`, the key must be `128` bits. If you have a `relative` **KeyStoragePolicy** set, Elastic Transcoder writes your key to an Amazon S3 bucket with Amazon S3 server-side encryption.

Key MD5 (Required if an Encryption Key is supplied)
The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

Initialization Vector (Required if an Encryption Key is supplied)
The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

For more information, go to Initialization Vector.

License Acquisition URL
The location of the license key required to decrypt your HLS playlist. The URL must be an absolute path, and is referenced in the URI attribute of the EXT-X-KEY metadata tag in the playlist file. For example:

```
https://www.example.com/exampleKey/
```

Key Storage Policy

Specify whether you want Elastic Transcoder to write your HLS license key to an Amazon S3 bucket. If you choose `WithVariantPlaylists`, Elastic Transcoder will write your encrypted key into the same Amazon S3 bucket as the associated playlist.

Important

If you chose `NoStore`, Elastic Transcoder will not store your key. You are responsible for storing it and providing it to your users by giving them the **License Acquisition URL** where you are storing the key.

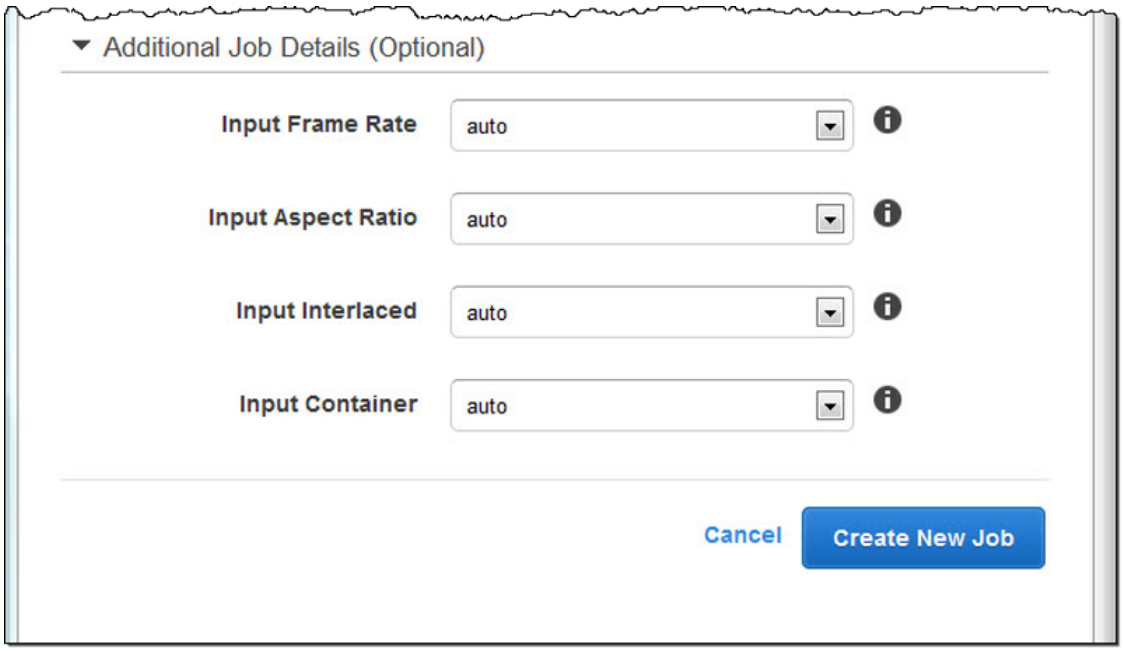
(Optional) Additional Job Details

Input Frame Rate
The frame rate of the input file. If you want Elastic Transcoder to automatically detect the frame rate of the input file, select **auto**. If you want to specify the frame rate for the input file, select the applicable value.

Input Aspect Ratio
The aspect ratio of the input file. If you want Elastic Transcoder to automatically detect the aspect ratio of the input file, select **auto**. If you want to specify the aspect ratio for the output file, select the applicable value.

Input Interlacing
Whether the input file is interlaced. If you want Elastic Transcoder to automatically detect whether the input file is interlaced, select **auto**. If you want to specify whether the input file is interlaced, select the applicable value.

Input Container
The container type for the input file. If you want Elastic Transcoder to automatically detect the container type of the input file, select **auto**. If you want to specify the container type for the input file, select the applicable value.



Working with Presets

Topics

- Creating a Preset in Elastic Transcoder (p. 39)
- Listing Presets and Viewing Preset Settings in Elastic Transcoder (p. 40)
- Changing Presets in Elastic Transcoder (p. 40)
- Deleting an Elastic Transcoder Preset (p. 40)
- Settings that You Specify When You Create an Elastic Transcoder Preset (p. 41)

A preset is a template that contains the settings that you want Elastic Transcoder to apply during the transcoding process, for example, the number of audio channels and the video resolution that you want in the transcoded file. When you create a job, you specify which preset you want to use. Elastic Transcoder includes default presets for a number of common output formats. You can also add presets for other output formats. For a list of system presets, see System Presets (p. 70).

Creating a Preset in Elastic Transcoder

You can create a preset using either the AWS Management Console or the Elastic Transcoder Create Preset API action. The following procedure explains how to create a preset using the console. For information about how to create a preset using the API, see Create Preset (p. 275).

> **Note**
> You cannot update an existing preset. If you need to change settings in a preset, create a new preset based on the preset that you want to change, update the applicable values, and save the new preset.

To create a preset using the Elastic Transcoder console

1. Sign in to the AWS Management Console and open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. In the navigation bar of the Elastic Transcoder console, select the region in which you want to create the preset.
3. In the left pane of the console, click **Presets**.
4. On the **Presets** page, click **Create New Preset**.
5. Enter the applicable values. For more information about each field, see Preset Settings (p. 41).
6. Click **Create Preset**.

Listing Presets and Viewing Preset Settings in Elastic Transcoder

You can list the default presets included with Elastic Transcoder and the presets that you have added in an AWS region by using the AWS Management Console or by using the Elastic Transcoder List Presets API. You can also view the settings for an individual preset. The following procedure explains how to list presets and how to view settings for a preset using the console.

For information about how to list presets using the API, see List Presets (p. 296). For information about how to view settings for a preset using the API, see Read Preset (p. 315). For the list of system presets, see System Presets (p. 70).

To list presets using the Elastic Transcoder console

1. Sign in to the AWS Management Console and open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. In the navigation bar of the Elastic Transcoder console, select the region in which you want to list presets.
3. In the left pane of the console, click **Presets**.
4. To display settings for an individual preset, click the icon next to the preset.

Changing Presets in Elastic Transcoder

Elastic Transcoder doesn't allow you to change the settings in an existing preset. This is true both for the default presets included with Elastic Transcoder and the presets you've added. However, you can easily achieve the same result by making a copy of the preset that you want to change, changing the applicable settings, saving the new preset, and deleting the old preset, as the following procedure explains.

To change a preset using the Elastic Transcoder console

1. Sign in to the AWS Management Console and open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. In the navigation bar of the Elastic Transcoder console, select the region that contains the preset that you want to change.
3. In the left pane of the console, click **Presets**.
4. Select the check box for the preset that you want to change.
5. Click **Copy**.
6. Change the applicable values in the copy of the preset that you want to change. For more information about each field, see Preset Settings (p. 41).
7. Click **Create Preset**.
8. Back on the **Presets** page, select the check box for the old version of the preset.
9. Click **Remove**.

Deleting an Elastic Transcoder Preset

You can delete the presets that you have added in an AWS region by using the AWS Management Console or by using the Elastic Transcoder Delete Preset API. The following procedure explains how to

delete presets using the console. For information about how to delete presets using the API, see Delete Preset (p. 334).

Note
You can't delete the default presets that are included with Elastic Transcoder.

To delete a preset using the Elastic Transcoder console

1. Sign in to the AWS Management Console and open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. In the navigation bar of the Elastic Transcoder console, select the region that contains the preset that you want to delete.
3. In the left pane of the console, click **Presets**.
4. Select the check box for the preset that you want to delete.
5. Click **Remove**.

Settings that You Specify When You Create an Elastic Transcoder Preset

When you create a preset, you specify the following values.

Topics

- General Settings (p. 41)
- Video Settings (p. 42)
- Watermark Settings (p. 51)
- Audio Settings (p. 54)
- Thumbnail Settings (p. 56)

General Settings

Start with Preset
The preset that you want to use as a starting point for your new preset. Elastic Transcoder copies the values from the preset that you select into the fields for the new preset. You can choose either one of the default Elastic Transcoder presets or a preset that you created.

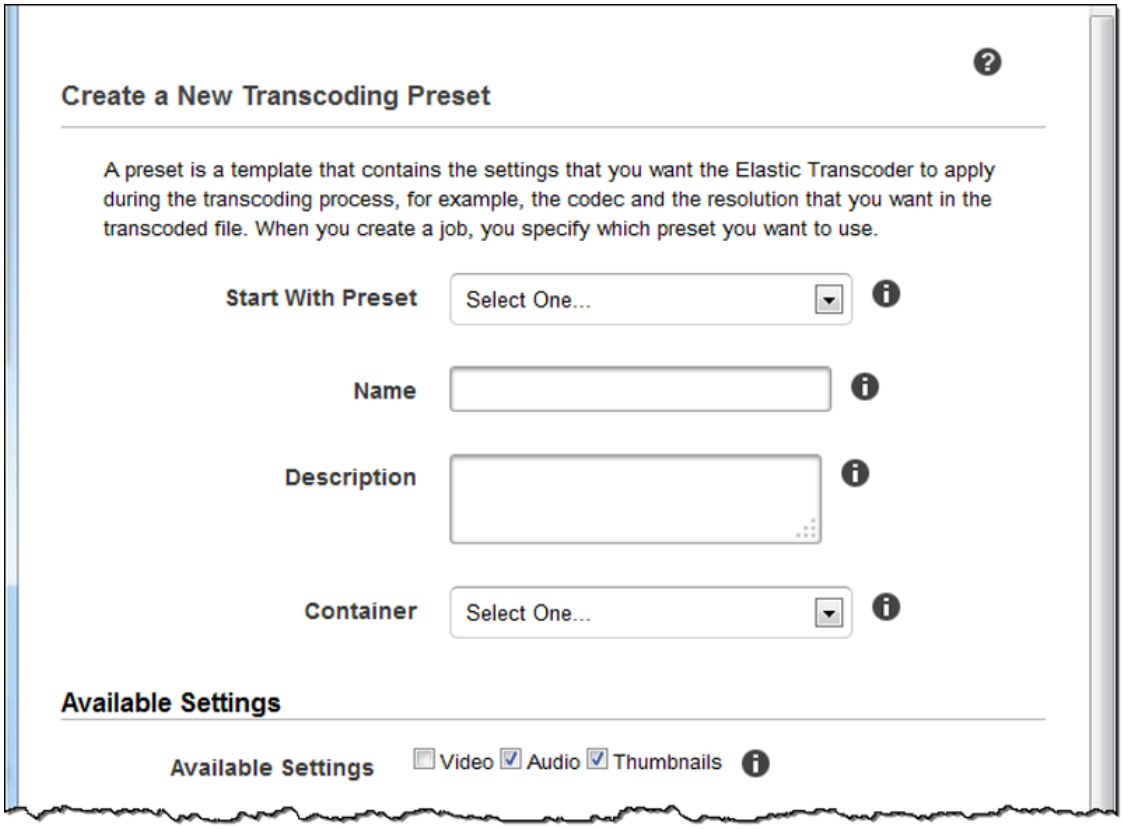
Name
The name of the preset. We recommend that the name be unique within the AWS account, but uniqueness is not enforced. The maximum length is 40 characters.

Description
A description of the preset. The maximum length is 255 characters.

Container
The container type for the output file. Valid values include **flv**, **fmp4**, **gif**, **mp3**, **mp4**, **mpg**, **ogg**, **ts**, and **webm**. The following restrictions apply:

Container	Audio Codecs	Video Codecs
FLV	AAC, mp3	H.264
FMP4	AAC	H.264
GIF	Not Allowed	gif

Container	Audio Codecs	Video Codecs
MP3	mp3	Not Allowed
MP4	AAC, mp3	H.264
MPG	mp2	mpeg2
OGG	vorbis	Not Allowed
TS	AAC, mp3	H.264
WEBM	vorbis	vp8



Video Settings

Codec

The video codec for the output file. Valid values include **gif**, **H.264**, **mpeg2**, and **vp8**. The following chart marks the allowed combinations:

Container & Video Codecs	GIF	H.264	MPEG2	VP8
FLV		x		
GIF	x			
FMP4		x		

Container & Video Codecs	GIF	H.264	MPEG2	VP8
MP3				
MP4		x		
MPG			x	
OGG				
TS		x		
WEBM				x

For more information about the H.264 video-compression format, go to the wikipedia page on H.264/MPEG-4 AVC.

For more information about VP8, go to the WebM project website.

(H.264/VP8 Only) Profile
The profile that you want to use for the output video. When the video codec is **H.264**, Elastic Transcoder supports the following profiles:

- **baseline:** The profile most commonly used for videoconferencing and for mobile applications.
- **main:** The profile used for standard-definition digital TV broadcasts.
- **high:** The profile used for high-definition digital TV broadcasts and for Blu-ray discs.

For more information about H.264 profiles, see Profiles in the Wikipedia entry "H.264/MPEG-4 AVC."

When the video codec is **VP8**, Elastic Transcoder supports values of **0**, **1**, **2**, and **3**.

(H.264 Only) Level
The H.264 level that you want to use for the output video. Select the applicable value.

For more information about levels, see Levels in the Wikipedia entry "H.264/MPEG-4 AVC."

(H.264 Only) Maximum Number of Reference Frames
The maximum number of previously decoded frames to use as a reference for decoding future frames. If you enter a value greater than the recommended value based on the values that you specified for Max Height, Max Height, and Level, Elastic Transcoder displays a message that contains the recommended value. For a detailed explanation, including the calculation that Elastic Transcoder performs, see MaxReferenceFrames in the topic Create Preset (p. 275).

(Optional, H.264/MPEG2/VP8 Only) Maximum Bit Rate
The maximum number of kilobits per second in the output video. Specify a value between 16 and 62,500, inclusive.

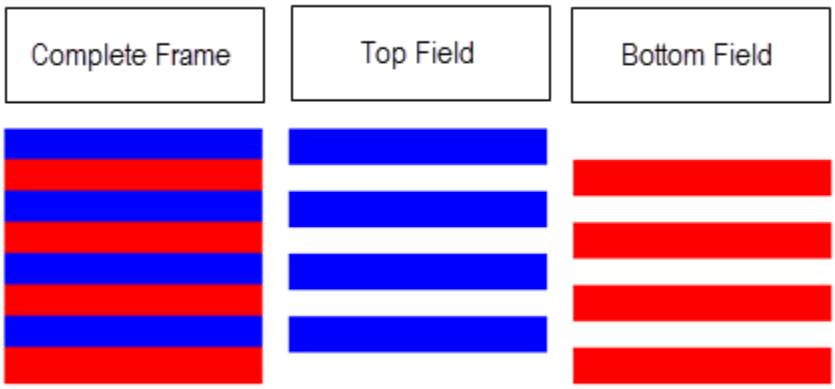
If you specify **auto** for **Bit Rate**, Elastic Transcoder uses the bit rate of the input video as the average bit rate of the output video. **Maximum Bit Rate** allows you to cap the bit rate of the output video, which is useful when the maximum bit rate supported by a target device is lower than the bit rate of the input video. Reducing the maximum bit rate might reduce the quality of the video.

(Optional, H.264/MPEG2/VP8 Only) Buffer Size
The maximum number of kilobits in any *x* seconds of the output video. This window is commonly 10 seconds, the standard segment duration when you're using MPEG-TS for the container type of the output video. Specify an integer greater than 0. If you specify **Maximum Bit Rate** and omit **Buffer Size**, Elastic Transcoder sets **Buffer Size** to 10 times the value of **Maximum Bit Rate**.

(Optional, H.264/MPEG2 Only) Interlaced Mode
The interlace mode for the output video.

Interlaced video is used to double the perceived frame rate for a video by interlacing two fields (one field on every other line, the other field on the other lines) so that the human eye registers multiple pictures per frame. Interlacing reduces the bandwidth required for transmitting a video, but can result in blurred images and flickering.

The two sets of lines are known as fields, and an interlaced frame splits two images across the fields:



Valid values include **Progressive** (no interlacing, top to bottom), **Top First** (top field first), **Bottom First** (bottom field first), and **Auto**.

If no **InterlaceMode** is specified, Elastic Transcoder uses **Progressive** for the output. If **Auto** is specified, Elastic Transcoder interlaces the output.

For more information, go to the Wikipedia page Interlaced video.

(Optional, H.264/MPEG2 Only) Color Space Conversion Mode

The color space conversion Elastic Transcoder applies to the output video. Color spaces are the algorithms used by the computer to store information about how to render color. **Bt.601** is the standard for standard definition video, while **Bt.709** is the standard for high definition video.

Valid values include **None**, **Bt709toBt601**, **Bt601toBt709**, and **Auto**.

If you chose **Auto** for **ColorSpaceConversionMode** and your output is interlaced, your frame rate is one of `23.97`, `24`, `25`, `29.97`, `50`, or `60`, your **SegmentDuration** is null, and you are using one of the resolution changes from the graph below, Elastic Transcoder applies the following color space conversions:

Input File	Input Resolution	Output Resolution	Applied Color Space Conversion
Standard to HD	720x480	1920x1080	Bt601ToBt709
Standard to HD	720x576	1920x1080	Bt601ToBt709
HD to Standard	1920x1080	720x576	Bt709ToBt601
HD to Standard	1920x1080	720x480	Bt709ToBt601

Note

Elastic Transcoder may change the behavior of the **ColorspaceConversionMode Auto** mode in the future. All outputs in a playlist must use the same **ColorSpaceConversionMode**.

If you do not specify a **ColorSpaceConversionMode**, Elastic Transcoder does not change the color space of a file.

If you are unsure what **ColorSpaceConversionMode** was applied to your output file, you can check the **AppliedColorSpaceConversion** parameter included in your job response. If your job does not have an **AppliedColorSpaceConversion** in its response, no **ColorSpaceConversionMode** was applied.

For more information about color space, go to the Wikipedia page Color space. For more information about **Bt.601** and **Bt.709**, go to the Wikipedia pages Rec. 601 and Rec. 709.

(MPEG2 Only) Chroma Subsampling

The sampling pattern for the chroma (color) channels of the output video. Valid values include **yuv420p** and **yuv422p**.

yuv420p samples the chroma information of every other horizontal and every other vertical line, **yuv422p** samples the color information of every horizontal line and every other vertical line.

To learn more about chroma subsampling, go to the wikidepia page Chroma subsampling.

(Gif Only) Loop Count

The number of times you want the output gif to loop.

Valid values include **Infinite** and integers between **0** and **100**, inclusive.

(H.264/MPEG2/VP8 Only) Maximum Number of Frames Between Keyframes

The maximum number of frames between key frames. Not applicable for containers of type **gif**. Key frames are fully encoded frames; the frames between key frames are encoded based, in part, on the content of the key frames. The value is an integer formatted as a string; valid values are between 1 (every frame is a key frame) and 100000, inclusive. A higher value results in higher compression but might also discernibly decrease video quality.

For Smooth outputs, the **Frame Rate** must have a constant ratio to the **Maximum Number of Frames Between Keyframes**. This allows Smooth playlists to switch between different quality levels while the file is being played.

For example, an input file can have a **Frame Rate** of 30 with a **Maximum Number of Frames Between Keyframes** of 90. The output file then needs to have a ratio of 1:3. Valid outputs would have **Frame Rates** of 30, 25, and 10, and **Maximum Number of Frames Between Keyframes** of 90, 75, and 30 respectively.

Alternately, this can be achieved by setting **Frame Rate** to auto and having the same values for **Video Max Frame Rate** and **Maximum Number of Frames Between Keyframes**.

For more information about key frames, see the Wikipedia entry Video compression picture types.

(H.264/MPEG2/VP8 Only) Fixed Number of Frames Between Keyframes

Whether to use a fixed value for Fixed Number of Frames Between Keyframes:

- **Yes:** Elastic Transcoder uses the value of Maximum Number of Frames Between Keyframes for the distance between key frames (the number of frames in a group of pictures, or GOP).
- **No:** The distance between key frames can vary.

> **Note**
> **Fixed Number of Frames Between Keyframes** must be set to `true` for **fmp4** containers.

Bit Rate

The bit rate of the video stream in the output video, in kilobits/second. You can configure variable bit rate or constant bit rate encoding:

- **Variable bit rate encoding:** Specify **auto.** Elastic Transcoder optimizes the bit rate and maintains a consistent quality for each frame of the output.
- **Constant bit rate encoding:** Specify the bit rate.

Valid values for the video bit rate depend on the value that you chose for Codec:

- **H.264:** Valid values depend on the values of Level and Profile. We recommend that you specify a value less than or equal to the maximum H.264-compliant value listed in the following table for your level and profile:
- **VP8:** do not use the following table; **Level** applies only when the video codec is H.264.

Level	Maximum Video Bit Rate in kilobits/Second: Baseline Profile and Main Profile	Maximum Video Bit Rate in kilobits/Second: High Profile
1	64	80
1b	128	160
1.1	192	240
1.2	384	480
1.3	768	960
2	2000	2500
2.1	4000	5000
2.2	4000	5000
3	10000	12500
3.1	14000	17500
3.2	20000	25000
4	20000	25000
4.1	50000	62500

Frame Rate

The frames per second for the video stream in the output video. Select the applicable value.

If you specify **auto**, Elastic Transcoder uses the detected frame rate of the input source. If you specify a frame rate, we recommend that you perform the following calculation:

Frame rate = maximum recommended decoding speed in luma samples/second / (Video Max Width) * (Video Max Height)

where **maximum recommended decoding speed in luma samples/second** is less than or equal to the maximum value listed in the following table, based on the value that you specified for Level.

> **Note**
> If you enter a value greater than the recommended value based on the values that you specified for Max Width, Max Height, and Level, Elastic Transcoder displays a message that contains the recommended value.

For Smooth outputs, the **Frame Rate** must have a constant ratio to the **Maximum Number of Frames Between Keyframes**. For example, if you specify an input file with a **Frame Rate** of 30 and a **Maximum Number of Frames Between Keyframes** of 90, the **Frame Rate** and **Maximum Number of Frames Between Keyframes** of the outputs must also have a ratio of 1:3.

Level	Maximum Recommended Decoding Speed in Luma Samples/Second
1	380160
1b	380160
1.1	768000
1.2	1536000
1.3	3041280
2	3041280
2.1	5068800
2.2	5184000
3	10368000
3.1	27648000
3.2	55296000
4	62914560
4.1	62914560

Video Max Frame Rate

If you specify `auto` for **Frame Rate**, Elastic Transcoder uses the frame rate of the input video for the frame rate of the output video, up to the maximum frame rate. If you do not specify a **Video Max Frame Rate**, Elastic Transcoder will use a default of `30`.

Specify the maximum frame rate that you want Elastic Transcoder to use when the frame rate of the input video is greater than either the desired maximum frame rate of the output video or the default maximum frame rate.

Elastic Transcoder uses the highest supported frame rate that meets both of the following criteria:

- The frame rate is less than or equal to the maximum frame rate.
- The frame rate divides into the input frame rate evenly, with no remainder.

For example, if you have an input file with a frame rate of 50 and specify a value of 30 for **Video Max Frame Rate**, Elastic Transcoder produces an output video for which the frame rate is 25 frames per second, because 25 is less than 30, and 50 divided by 25 is 2.

Max Width

The maximum width of the output video in pixels. If you specify **auto**, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 128 and 4096.

For more information, see Max Height.

Max Height

The maximum height of the output video in pixels. If you specify **auto**, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 96 and 3072.

If you specified H.264 for Codec We recommend that you specify values for **Max Width** and **Max Height** so the product of the two values is less than or equal to the applicable value in the following table.

> **Note**
> If you specified **VP8** for Codec, do not use the following table; **Level** applies only when the video codec is H.264.

Value of Level	Maximum Value of Max Width x Max Height
1	25344
1b	25344
1.1	101376
1.2	101376
1.3	101376
2	101376
2.1	202752
2.2	404720
3	404720
3.1	921600
3.2	1310720
4	2097152
4.1	2097152

Sizing Policy

Specify one of the following values to control scaling of the output video:

- **Fit:** Elastic Transcoder scales the output video so it matches the value that you specified in either **Max Width** or **Max Height** without exceeding the other value.
- **Fill:** Elastic Transcoder scales the output video so it matches the value that you specified in either **Max Width** or **Max Height** and matches or exceeds the other value. Elastic Transcoder centers the output video and then crops it in the dimension (if any) that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches the output video to match the values that you specified for **Max Width** and **Max Height**. If the relative proportions of the input video and the output video are different, the output video will be distorted.
- **Keep:** Elastic Transcoder does not scale the output video. If either dimension of the input video exceeds the values that you specified for **Max Width** and **Max Height**, Elastic Transcoder crops the output video.
- **ShrinkToFit:** Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of **Max Width** and **Max Height** without exceeding either value. If you specify this option, Elastic Transcoder does not scale the video up.
- **ShrinkToFill:** Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of **Max Width** and **Max Height** without dropping below either value. If you specify this option, Elastic Transcoder does not scale the video up.

The following table shows possible effects of **Sizing Policy** settings on the output video:

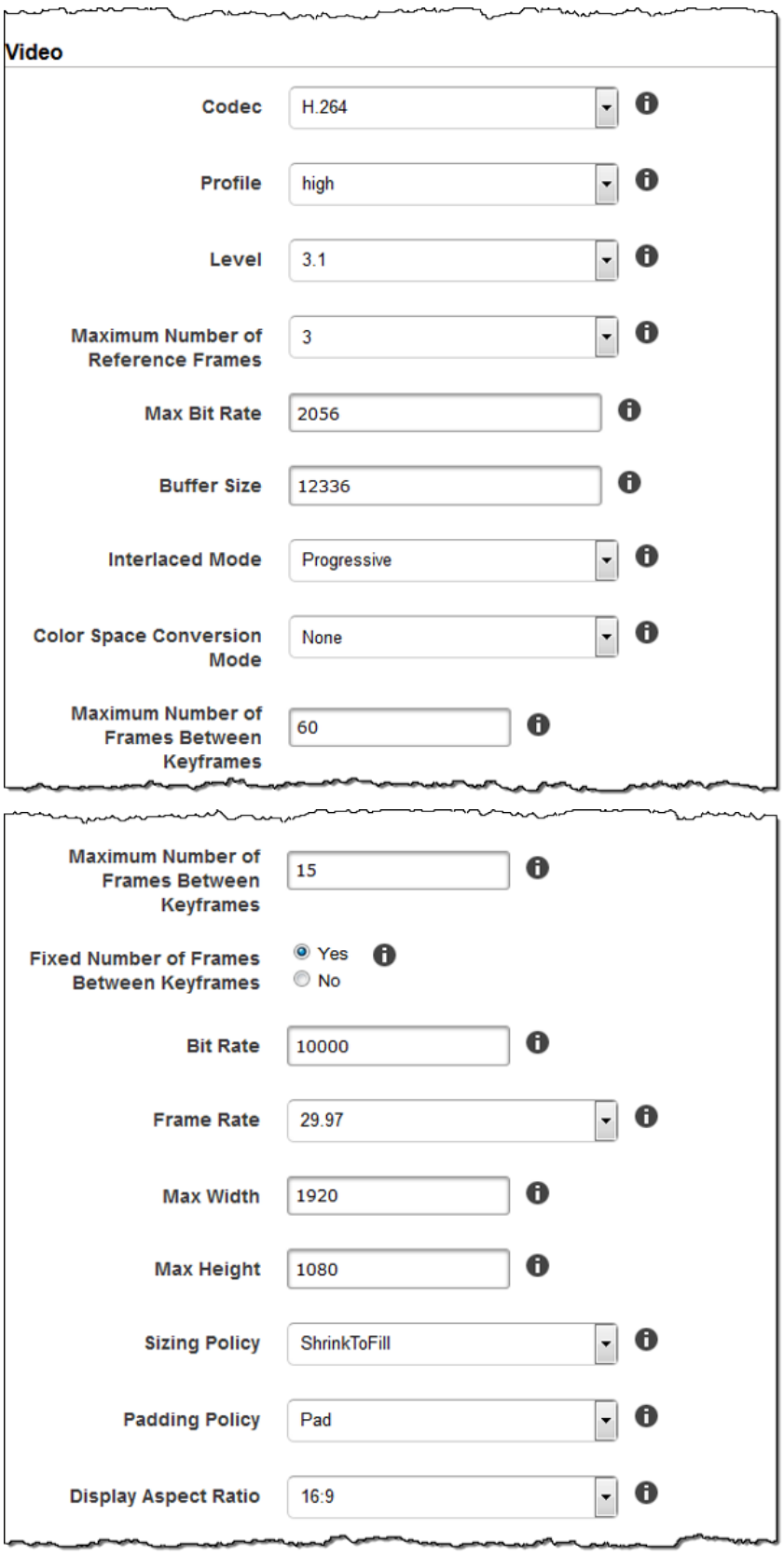
SizingPolicy	Output Video May Be Scaled Up	Output Video May Be Padded When Padding Policy Is "Pad"	Output Video May Have a Different Pixel Aspect Ratio than Input Video	Output Video May Be Cropped
Fit	Yes	Yes		
Fill	Yes			Yes
Stretch	Yes		Yes	
Keep	Yes	Yes		Yes
ShrinkToFit		Yes		
ShrinkToFill		Yes		Yes

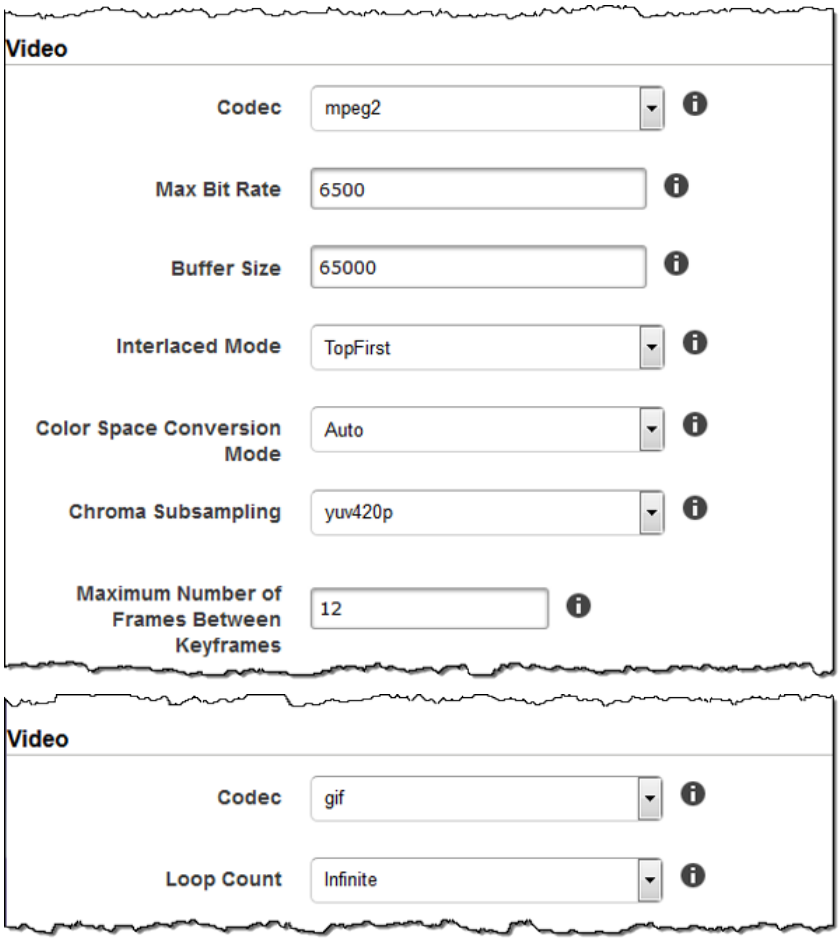
Padding Policy

When you set **PaddingPolicy** to **Pad**, Elastic Transcoder might add black bars to the top and bottom and/or left and right sides of the output video to make the total size of the output video match the values that you specified for **Max Width** and **Max Height**. For more information, see the table at Sizing Policy.

Display Aspect Ratio

The value that Elastic Transcoder adds to the metadata in the output video. If you set **Display Aspect Ratio** to **auto**, Elastic Transcoder chooses an aspect ratio that ensures square pixels. If you specify another option, Elastic Transcoder sets that value in the output video.


Video
Codec
H.264
Profile
high
Level
3.1
Maximum Number of Reference Frames
3
Max Bit Rate
2056
Buffer Size
12336
Interlaced Mode
Progressive
Color Space Conversion Mode
None
Maximum Number of Frames Between Keyframes
60
Maximum Number of Frames Between Keyframes
15
Fixed Number of Frames Between Keyframes
Yes
No
Bit Rate
10000
Frame Rate
29.97
Max Width
1920
Max Height
1080
Sizing Policy
ShrinkToFill
Padding Policy
Pad
Display Aspect Ratio
16:9



Watermark Settings

Use the settings in this section to specify the size, position, scale, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear for the duration of the transcoded video.

Watermarks can be in .png or .jpg format. If you want to display a watermark that is not rectangular, use the .png format, which supports transparency.

When you create a job that uses this preset, you specify the .png or .jpg graphics that you want Elastic Transcoder to include in the transcoded videos. Elastic Transcoder does not require you to specify as many watermarks in each job output as you specified in the corresponding preset. For example, you might specify settings for four watermarks in a preset and specify only one watermark in a job output.

To configure watermark settings so your graphic is not distorted, set the value of **Sizing Policy** to **Shrink to Fit**, and set the values of **Maximum Width** and **Maximum Height** to the same percentage. If you want the graphic to appear in the same size as the original, set **Maximum Width** and **Maximum Height** to 100%.

For more information, see Watermarks (p. 69).

Id

A unique identifier for the settings for one watermark. The value of **Id** can be up to 40 characters long.

Maximum Width

The maximum width of the watermark in one of the following formats:

- *number of pixels* **px**: The minimum value is 16 pixels, and the maximum value is the value of MaxWidth.
- *integer percentage* **%**: The range of valid values is 0 to 100. Use the value of **Target** to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

If you specify the value in pixels, it must be less than or equal to the value of MaxWidth.

Maximum Height

The maximum height of the watermark in one of the following formats:

- *number of pixels* **px**: The minimum value is 16 pixels, and the maximum value is the value of MaxHeight.
- *integer percentage* **%**: The range of valid values is 0 to 100. Use the value of **Target** to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

If you specify the value in pixels, it must be less than or equal to the value of MaxHeight.

Sizing Policy

A value that controls scaling of the watermark:

- **Fit:** Elastic Transcoder scales the watermark so it matches the value that you specified in either **Maximum Width** or **Maximum Height** without exceeding the other value.
- **Stretch:** Elastic Transcoder stretches the watermark to match the values that you specified for **Maximum Width** and **Maximum Height**. If the relative proportions of the watermark and the values of **Maximum Width** and **Maximum Height** are different, the watermark will be distorted.
- **Shrink to Fit:** Elastic Transcoder scales the watermark down so that its dimensions match the values that you specified for at least one of **Maximum Width** and **Maximum Height** without exceeding either value. If you specify this option, Elastic Transcoder does not scale the watermark up.

Horizontal Alignment

The horizontal position of the watermark. To position the watermark with respect to the left or right border, also specify a non-zero value for **Horizontal Offset**:

- **Left:** The left edge of the watermark is aligned with the left border of the video.
- **Right:** The right edge of the watermark is aligned with the right border of the video.
- **Center:** The watermark is centered between the left and right borders.

Horizontal Offset

The amount by which you want the horizontal position of the watermark to be offset from the position specified by **Horizontal Alignment**:

- *number of pixels* **px**: The minimum value is 0 pixels, and the maximum value is the value of MaxWidth.
- *integer percentage* **%**: The range of valid values is 0 to 100.

For example, if you specify **Left** for **Horizontal Alignment** and **5px** for **Horizontal Offset**, the left side of the watermark appears 5 pixels from the left border of the output video.

HorizontalOffset is only valid when the value of **Horizontal Alignment** is **Left** or **Right**.

If you specify an offset that causes the watermark to extend beyond the left or right border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of **Target** to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

Vertical Alignment
The vertical position of the watermark. To position the watermark with respect to the top or bottom border, also specify a non-zero value for **Vertical Offset**:

- **Top:** The top edge of the watermark is aligned with the top border of the video.
- **Bottom:** The bottom edge of the watermark is aligned with the bottom border of the video.
- **Center:** The watermark is centered between the top and bottom borders.

Vertical Offset
The amount by which you want the vertical position of the watermark to be offset from the position specified by **Vertical Alignment**:

- *number of pixels* **px**: The minimum value is 0 pixels, and the maximum value is the value of **Maximum Height**.
- *integer percentage* **%**: The range of valid values is 0 to 100.

For example, if you specify **Top** for **Vertical Alignment** and **5px** for **Vertical Offset**, the top of the watermark appears 5 pixels from the top border of the output video.

Vertical Offset is only valid when the value of **Vertical Alignment** is **Top** or **Bottom**.

If you specify an offset that causes the watermark to extend beyond the top or bottom border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of **Target** to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

Opacity
A percentage that indicates how much you want a watermark to obscure the video in the location where it appears. Valid values are 0 (the watermark is invisible) to 100 (the watermark completely obscures the video in the specified location).

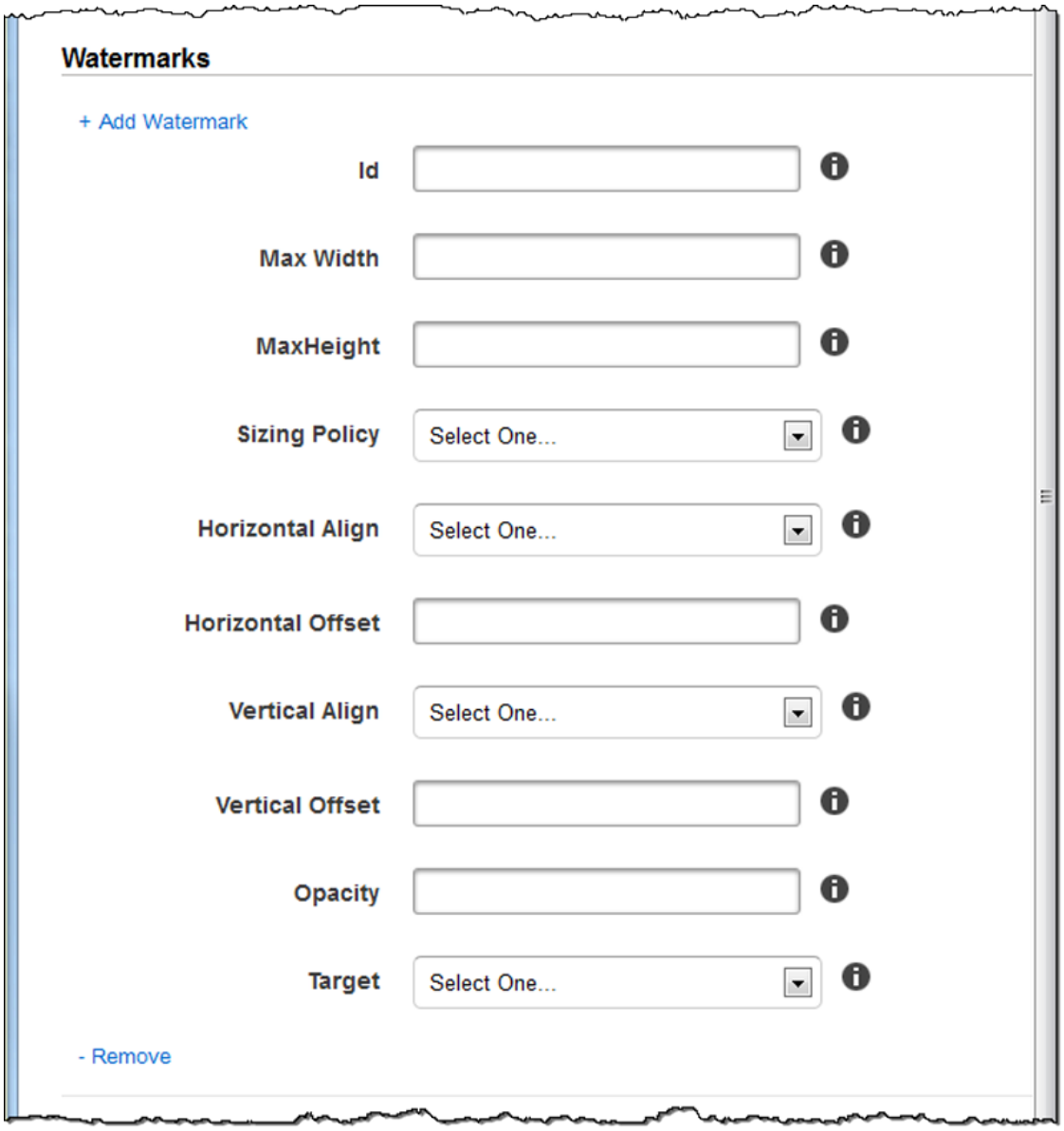
Elastic Transcoder supports transparent `.png` graphics. If you use a transparent `.png`, the transparent portion of the video appears as if you had specified a value of 0 for **Opacity**. The `.jpg` file format doesn't support transparency.

Target
A value that determines how Elastic Transcoder interprets values that you specified for the watermark settings **Horizontal Offset**, **Vertical Offset**, **Maximum Width**, and **Maximum Height**:

- **Content: Horizontal Offset** and **Vertical Offset** values are calculated based on the borders of the video **excluding** black bars added by Elastic Transcoder, if any.

  In addition, **Maximum Width** and **Maximum Height**, if specified as a percentage, are calculated based on the borders of the video **excluding** black bars added by Elastic Transcoder, if any.
- **Frame: Horizontal Offset** and **Vertical Offset** values are calculated based on the borders of the video **including** black bars added by Elastic Transcoder, if any.

  In addition, **Maximum Width** and **Maximum Height**, if specified as a percentage, are calculated based on the borders of the video **including** black bars added by Elastic Transcoder, if any.



Audio Settings

Codec

The audio codec for the output file. Valid values include **AAC**, **mp2**, **mp3**, and **vorbis**. The following chart marks the allowed combinations:

Container & Audio Codecs	AAC	MP2	MP3	Vorbis
FLV	x		x	
GIF				
FMP4	x			
MP3			x	

Container & Audio Codecs	AAC	MP2	MP3	Vorbis
MP4	x		x	
MPG		x		
OGG				x
TS	x		x	
WEBM				x

(AAC Only) Profile

If you specified **AAC** for **Audio:Codec**, choose the AAC profile for the output file. Elastic Transcoder supports the following profiles:

- **auto**: If you specify **auto**, Elastic Transcoder selects the profile based on the bit rate selected for the output file.
- **AAC-LC**: The most common AAC profile. Use for bit rates larger than 64 kbps. For more information, see Advanced Audio Coding.
- **HE-AAC**: Not supported on some older players and devices. Use for bit rates between 40 and 80 kbps. For more information, see High-Efficiency Advanced Audio Coding.
- **HE-AACv2**: Not supported on some players and devices. Use for bit rates less than 48 kbps. For more information, see High-Efficiency Advanced Audio Coding..

All outputs in a **Smooth** playlist must have the same value for **Profile**.

> **Note**
> If you created any presets before AAC profiles were added, Elastic Transcoder will use the AAC-LC profile for those presets.

For more information about AAC, see Audio Profiles in the Wikipedia entry "MPEG-4 Part 3."

Sample Rate

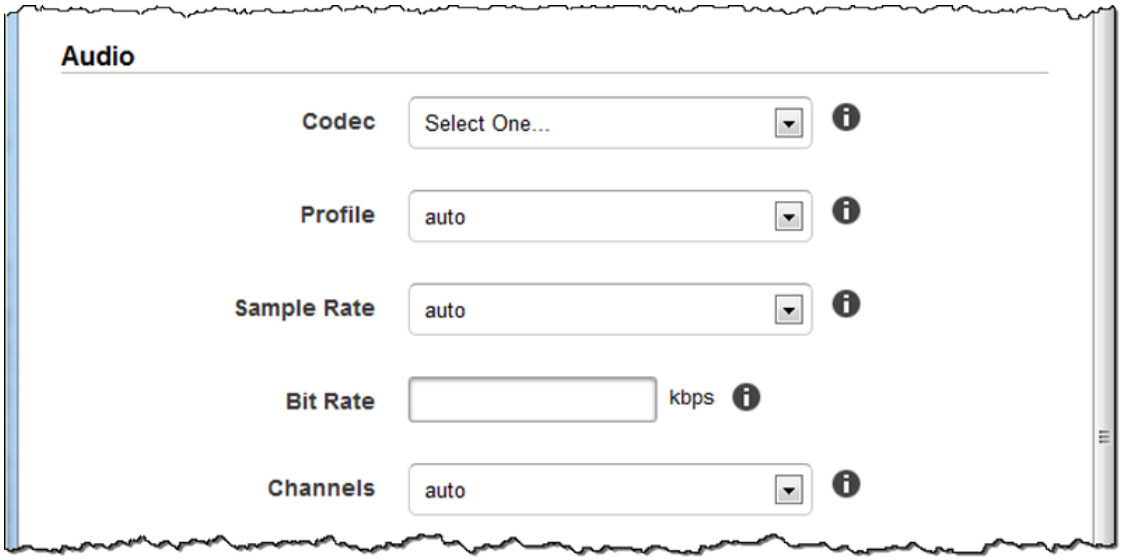
The sample rate of the audio stream in the output file, in Hz. If you want Elastic Transcoder to automatically detect the sample rate of the input file and use that value for the output file, select **auto**. If you want to specify the sample rate, select the applicable value.

Bit Rate

The bit rate of the audio stream in the output file, in kilobits/second. Enter an integer between 64 and 320, inclusive.

Channels

The number of audio channels in the output file. If you want Elastic Transcoder to automatically detect the number of audio channels in the input file and use that value for the output file, select **auto**. If you want Elastic Transcoder to omit audio from the output file, select **0**. If you want to specify the number of audio channels in the output file, select the applicable value.



Thumbnail Settings

Format
The format of thumbnails, if any. Valid formats are **jpg** and **png**.

You specify whether you want Elastic Transcoder to create thumbnails when you create a job. For more information, see Thumbnail Filename Pattern in the topic Settings that You Specify When You Create an Elastic Transcoder Job (p. 20).

Interval
The approximate number of seconds between thumbnails. Specify an integer value. The actual interval can vary by several seconds from one thumbnail to the next.

Max Height
The maximum height of thumbnails in pixels. If you specify **auto**, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 3072.

Max Width
The maximum width of thumbnails in pixels. If you specify **auto**, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 4096.

Sizing Policy
Specify one of the following values to control scaling of thumbnails:

- **Fit:** Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail **Max Width** or **Max Height** settings without exceeding the other value.
- **Fill:** Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail **Max Width** or **Max Height** settings and matches or exceeds the other value. Elastic Transcoder centers the image in thumbnails and then crops in the dimension (if any) that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches thumbnails to match the values that you specified for thumbnail **Max Width** and **Max Height** settings. If the relative proportions of the input video and thumbnails are different, the thumbnails will be distorted.
- **Keep:** Elastic Transcoder does not scale thumbnails. If either dimension of the input video exceeds the values that you specified for thumbnail **Max Width** and **Max Height** settings, Elastic Transcoder crops the thumbnails.
- **ShrinkToFit:** Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of thumbnail **MaxWidth** and **MaxHeight** without exceeding either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.

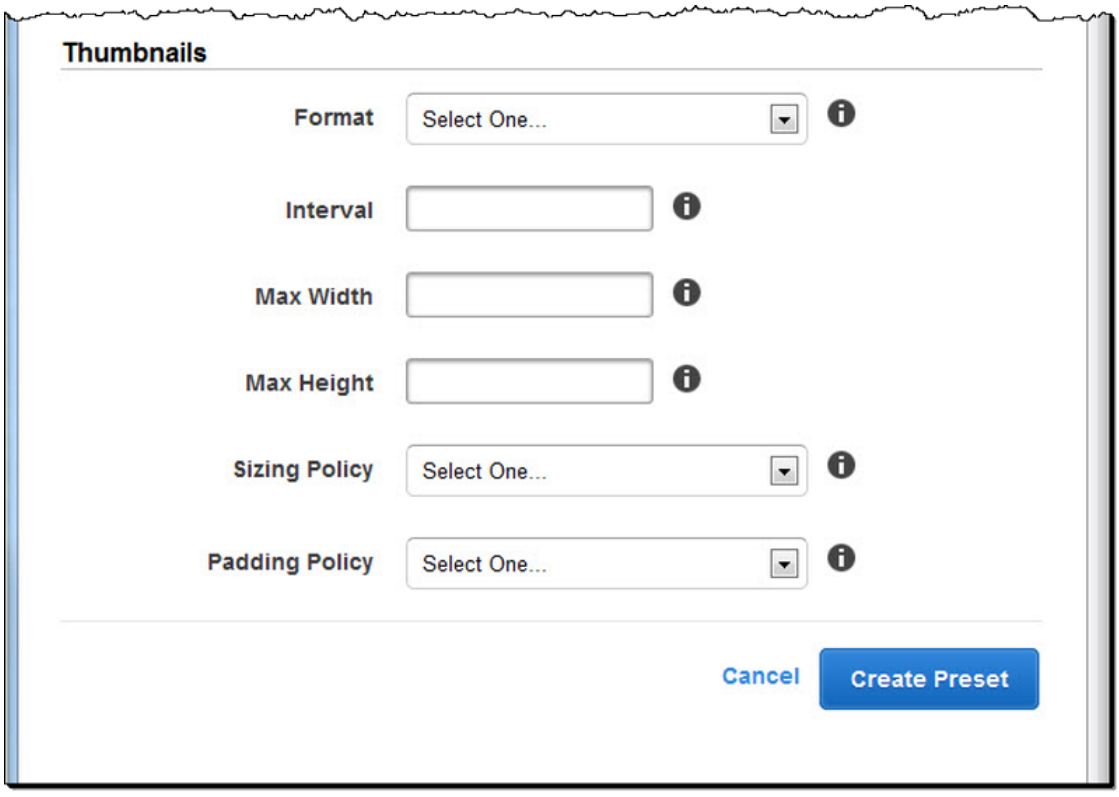
- **ShrinkToFill:** Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of **MaxWidth** and **MaxHeight** without dropping below either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.

The following table shows possible effects of **Sizing Policy** settings on thumbnails:

SizingPolicy	Thumbnails May Be Scaled Up	Thumbnails May Be Padded When Padding Policy Is "Pad"	Thumbnails May Have a Different Pixel Aspect Ratio than Input Video	Thumbnails May Be Cropped
Fit	Yes	Yes		
Fill	Yes			Yes
Stretch	Yes		Yes	
Keep	Yes	Yes		Yes
ShrinkToFit		Yes		
ShrinkToFill		Yes		Yes

Padding Policy

When you set **PaddingPolicy** to **Pad**, Elastic Transcoder might add black bars to the top and bottom and/or left and right sides of thumbnails to make the total size of the thumbnails match the values that you specified for thumbnail **Max Width** and **Max Height** settings. For more information, see the table at Sizing Policy.



Organizing Files in Your Amazon S3 Bucket

Amazon Elastic Transcoder integrates with Amazon S3 to store input and output files. The 100-bucket storage limit in Amazon S3 affects how you organize and manage your Elastic Transcoder files.

When you work with Amazon S3, we recommend that you mimic a regular file system. You can do this by using the OutputKeyPrefix in Elastic Transcoder to add file paths to your output file. Mimicking a file system allows you to use the prefix and delimiter parameters in the Amazon S3 API to find your files.

For example, suppose you are creating a user-generated content (UGC) site that takes in videos for many customers and transcodes them for use on several common devices. To store the videos, you could organize your input and output buckets like this:

```
Input:  /<your bucket>/users/<user account>/input/movie.mp4
Output:  /<your bucket>/users/<user account>/output/movie/<format>/movie.mp4
```

This enables you to store separate formats so you can locate them easily. For example, say you have two customers using your service to transcode different movies into multiple formats. You can organize your bucket like this:

```
Customer 1:
/<your bucket>/users/<user account 1>/output/movie/avi/movie.avi
/<your bucket>/users/<user account 1>/output/movie/mkv/movie.mkv
/<your bucket>/users/<user account 1>/output/movie/hls400k/movie.m3u8
  - playlist file for the hls400k version of movie
/<your bucket>/users/<user account 1>/output/movie/hls400k/movie.ts
  - video file for the hls400k version of movie
/<your bucket>/users/<user account 1>/output/movie/hls400k/movie.png
  - thumbnail file for the hls400k version of movie
/<your bucket>/users/<user account 1>/output/myOthermovie/hls1000k/movie.m3u8
  - playlist file for the hls1000k version of movie
/<your bucket>/users/<user account 1>/output/myOthermovie/hls1000k/movie.ts
  - video file for the hls1000k version of movie
/<your bucket>/users/<user account 1>/output/myOthermovie/hls1000k/movie.png
  - thumbnail file for the hls1000k version of movie
```

```
Customer 2:
/<your bucket>/users/<user account 2>/output/somemovie/avi/somemovie.avi
/<your bucket>/users/<user account 2>/output/somemovie/mkv/somemovie.mkv
/<your bucket>/users/<user account 2>/output/somemovie/hls400k/somemovie.m3u8
  - playlist file for the hls400k version of somemovie
/<your bucket>/users/<user account 2>/output/somemovie/hls400k/somemovie.ts
  - video file for the hls400k version of somemovie
/<your bucket>/users/<user account 2>/output/somemovie/hls400k/somemovie.png
  - thumbnail file for the hls400k version of somemovie
/<your bucket>/users/<user account 2>/output/myOthermovie/hls1000k/movie.m3u8
  - playlist file for the hls1000k version of movie
/<your bucket>/users/<user account 2>/output/myOthermovie/hls1000k/movie.ts
  - video file for the hls1000k version of movie
/<your bucket>/users/<user account 2>/output/myOthermovie/hls1000k/movie.png
  - thumbnail file for the hls1000k version of movie
```

We recommend that you end your OutputKeyPrefix with a '/' so that the last part of the `OutputKeyPrefix` and the `OutputKey` don't run together. Otherwise, your files will look like this:

```
/users/<user account>/output/<movie title>/mkvmovie.mkv
```

We also recommend that you keep your input media in a single bucket per region, and that you keep transcoded media and thumbnails in a separate bucket in the same region. This approach helps you avoid cross-region transfer fees between your Amazon S3 bucket and Elastic Transcoder.

Note

If you expect to have more than a hundred requests per second accessing your Amazon S3 buckets, we recommend you follow Amazon S3 performance considerations when designing your system.

Data Encryption Options

You can protect your Elastic Transcoder data by encrypting any input and output files that you want to use for a transcoding job while the files are stored, or at rest, in Amazon S3. This includes the input file, the output file, and any thumbnails, captions, input watermarks, or input album art. Playlists and metadata are not encrypted.

All resources for a job — including the pipeline, Amazon S3 buckets, and AWS Key Management Service key — should be in the same AWS region.

Topics

- Encryption Options (p. 60)
- Using AWS KMS with Elastic Transcoder (p. 62)
- HLS Content Protection (p. 64)

Encryption Options

Elastic Transcoder supports two main encryption options:

- **Amazon S3 Server-Side Encryption:** AWS manages the encryption process for you. For example, Elastic Transcoder calls Amazon S3, and Amazon S3 encrypts your data, saves it on disks in data centers, and decrypts the data when you download it.

  By default, Amazon S3 buckets accept both encrypted and unencrypted files, but you can set up your Amazon S3 bucket to accept only encrypted files. You don't need to make permission changes as long as Elastic Transcoder has access to your Amazon S3 bucket.

  For more information about Amazon S3 server-side encryption, see Protecting Data Using Server-Side Encryption in the *Amazon Simple Storage Service Developer Guide*. For more information about AWS KMS keys, see What is the AWS Key Management Service? in the *AWS Key Management Service Developer Guide*.

  Note
  There are additional charges for using AWS-KMS keys. For more information, see AWS Key Management Service Pricing.

- **Server-Side Encryption Using Customer-Provided Keys:** Elastic Transcoder can also use a client-provided encryption key to decrypt input files or encrypt your output file before storing it in Amazon S3. In this case, you manage the encryption keys and related tools.

If you want to transcode a file using client-provided keys, your job request must include the AWS KMS-encrypted key that you used to encrypt the file, the MD5 of the key that will be used as a checksum, and the initialization vector (or series of random bits created by a random bit generator) that you want Elastic Transcoder to use when encrypting your output files.

Elastic Transcoder can only use customer-provided keys that are encrypted with an AWS KMS master key, and Elastic Transcoder must be given permissions to use the master key. To encrypt your key, you must call AWS KMS programmatically with an encryption call that contains the following information:

```
{
    "EncryptionContext": {
        "service" : "elastictranscoder.amazonaws.com"
    },
    "KeyId": "The ARN of the key associated with your pipeline",
    "Plaintext": blob that is your AES key
}
```

Important
Your private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to decrypt your data.

To give Elastic Transcoder permission to use your key, see Using AWS KMS with Elastic Transcoder (p. 62).

For more information on encrypting data, see the AWS KMS API Reference and Encrypting and Decrypting Data. For more information on contexts, see Encryption Context in the *AWS Key Management Service Developer Guide*.

For more information about client-provided keys, see Protecting Data Using Server-Side Encryption with Customer-Provided Encryption Keys in the *Amazon Simple Storage Service Developer Guide*.

For information about the settings required when decrypting and encrypting files using the Elastic Transcoder console, see (Optional) Output Encryption (p. 28). For information about the settings required when decrypting and encrypting files using the Elastic Transcoder API, see the Create Job (p. 181) API action beginning with the Encryption (p. 184) element.

Using AWS KMS with Elastic Transcoder

You can use the AWS Key Management Service (AWS KMS) with Elastic Transcoder to create and manage the encryption keys that are used to encrypt your data. Before you can set up Elastic Transcoder to use AWS KMS, you must have the following:

- Elastic Transcoder pipeline
- IAM role associated with the Elastic Transcoder pipeline
- AWS KMS key
- ARN of the AWS KMS key

The following procedures show how to identify your existing resources or create new ones.

Getting Ready to Use AWS KMS with Elastic Transcoder

To create a pipeline

- Follow the steps in Creating a Pipeline in Elastic Transcoder (p. 9).

To identify the IAM role associated with your pipeline

1. Sign in to the AWS Management Console and open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. In the navigation pane, click **Pipelines**.
3. Click the magnifying glass icon next to the pipeline name.
4. Click the **Permissions** section to expand it.
5. Take note of the IAM role. If you are using the default role created by Elastic Transcoder, the role is **Elastic_Transcoder_Default_Role**.

To create an AWS KMS key

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. Follow the steps in **Creating Keys**.

To identify the ARN of an AWS KMS key

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click **Encryption Keys**.
3. In the region drop-down list, select the region that your key and pipeline are in.
4. Click the key you want to use.
5. Take note of the ARN.

To use your AWS KMS key to encrypt data

You can use the console to create an AWS KMS key, but you must use the encryption and decryption APIs to encrypt or decrypt data with an AWS KMS key. For more information, see Encrypting and Decrypting Data.

Connecting Elastic Transcoder and AWS KMS

Once you have your pipeline, IAM role, and AWS KMS key, you must tell the pipeline which key to use, and tell the key which IAM role can use it.

To add the AWS KMS key to your pipeline

1. Open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. Select the pipeline that you want to use the AWS KMS key with, and click **Edit**.
3. Click the **Encryption** section to expand it, and in the **AWS KMS Key ARN** section, select **Custom**.
4. Type in the ARN of your AWS KMS key, and click **Save**.

To add an IAM role to your AWS KMS key

If you did not create your AWS KMS key with the IAM role associated with your pipeline, you can add it by following this procedure:

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click **Encryption Keys**.
3. In the region drop-down list, select the region that your key and pipeline are in.
4. Click the name of the key that you want to use.
5. Click the **Key Usage** section to expand it, and then click **Add**.
6. On the **Attach** page, search for the role associated with your pipeline, select it from the results, and click **Attach**.

You can now use your AWS KMS key with your Elastic Transcoder pipeline.

HLS Content Protection

HTTP Live Streaming (HLS) is a protocol that segments media files for optimization during streaming. HLS enables media players to play segments with the highest quality resolution that is supported by their network connection during playback.

You can use Elastic Transcoder to encrypt segments of a streamed media file, send the encrypted segments over the Internet, and decrypt them upon playback. This protects your media content and ensures that only authorized users can view the encrypted segments of your media files.

The following is a summary of the playback process of a media file that has HLS content protection:

1. A user visiting your web page authenticates with your web server, which sets a session cookie in the user's browser.
2. The user loads a player from your web server.
3. The player fetches the master playlist from your content delivery network (CDN). The master playlist provides the available bit rates and resolutions for the media file.
4. The player calls your web server, which validates the session cookie, checks that the user is authorized to view the content, and returns the data decryption key.
5. The player chooses a variant playlist and fetches the associated media segments from the CDN.
6. The player uses the data key to decrypt the segments, and begins playing the media.

Note
You can use HLS content protection to encrypt segments of a streamed file, or you can encrypt entire files. You can't do both, so don't select both HLS content protection and individual file protection.

Keys for HLS Content Protection

To use HLS content protection with Elastic Transcoder, you need two types of keys:

- **Customer master key (CMK)** — The key associated with your Elastic Transcoder pipeline
- **Data key** — The key associated with your Elastic Transcoder job

You must have a customer master key (CMK) to use HLS content protection. The CMK is used to encrypt your data key before it is sent it over the Internet. We recommend that you create one CMK key to use with all your transcoding jobs. For more information about creating and setting up a CMK, see Using AWS KMS with Elastic Transcoder (p. 62).

The data key is used to encrypt your media file. All variations and segments of the same content are encrypted using the same data key. If you do not specify a data key, Elastic Transcoder generates one for you.

Streaming HLS Protected Content

To deliver HLS protected content, you must have the following:

- A location for storing your encrypted media files and data keys. We recommend that you store your files in Amazon S3 and secure your keys in a database, such as DynamoDB. For more information on DynamoDB, see What is Amazon DynamoDB? in the *Amazon DynamoDB Developer Guide*.
- (Optional) A content distribution network (CDN) to stream your files. For more information about CDNs, see Getting Started with CloudFront in the *Amazon CloudFront Developer Guide*.

- An application capable of authenticating and authorizing your users, and securely serving the data encryption key. You can use Amazon EC2 to run this application. For more information, see Setting Up with Amazon EC2 in the *Amazon EC2 User Guide for Microsoft Windows Instances* (for Windows users) or Setting Up with Amazon EC2 in the *Amazon EC2 User Guide for Linux Instances* (for Linux users).
- A player capable of decrypting an encrypted HLS file. For more information, go to Http Live Streaming.

Creating Encrypted Streamed Content

To prepare your files for HLS content protection, you must associate a CMK with a new or existing pipeline.

To set up a pipeline with a CMK that you specify, see Using AWS KMS with Elastic Transcoder (p. 62).

The following steps show how to encrypt your files for HLS content protection by using the Elastic Transcoder console:

To use HLS content protection for your files

1. Open the Elastic Transcoder console at https://console.aws.amazon.com/elastictranscoder/.
2. In the navigation pane, click **Jobs** and create a new job. For more information, see Creating a Job in Elastic Transcoder (p. 17).
3. In **Output Details**, in the **Preset** dropdown list, select an `HLS` preset.
4. Leave **Encryption Parameters** set to `None`.
5. In **Playlists**, click **Add Playlist** and select either `HLSv3` or `HLSv4` as your playlist type.
6. In **Content Protection**, select `Enter Information`.

   a. To manage your own key, in **Key Storage Policy**, select `No Store`. In **License Acquisition Url**, type in the absolute path to the location where you will store your data key. For example:

   ```
   https://www.example.com/datakey
   ```

   We recommend that you select `No Store` and store your key in a secure Amazon S3 bucket or a database such as DynamoDB.

   b. To store your key in a public Amazon S3 bucket, in **Key Storage Policy**, select `With Variant Playlists`. Elastic Transcoder writes your data key into the same bucket as the playlist files.

   Important
   Keys stored using `With Variant Playlists` are written to a public bucket. Use `No Store` for your actual keys.

   Note
   Elastic Transcoder returns your data key as part of the job object, but does not store it. You are responsible for storing the data key.

Notifications of Job Status

Elastic Transcoder can notify you when the status of a job changes. You can configure Elastic Transcoder to send you notifications for any combination (or none) of the following changes in status:

- **Progressing:** Elastic Transcoder has started to process a job in the pipeline.
- **Completed:** Elastic Transcoder has finished processing a job in the pipeline.
- **Warning:** Elastic Transcoder encountered a warning condition while processing a job in the pipeline.
- **Error:** Elastic Transcoder encountered an error condition while processing a job in the pipeline.

Elastic Transcoder sends notifications by using Amazon Simple Notification Service (Amazon SNS). Amazon SNS offers a variety of notification options, including the ability to send messages to HTTP endpoints, email addresses, and Amazon Simple Queue Service (Amazon SQS) queues. For more information about these and other options, see the Amazon Simple Notification Service Developer Guide.

Notifications are useful for designing event-driven applications. If you use notifications to determine when jobs have completed, you can eliminate polling, and you won't encounter the `Limit Exceeded` exceptions (HTTP status code 429) that sometimes result from polling.

You configure notifications when you create or update a pipeline. For every job that you submit to that pipeline, Elastic Transcoder sends the associated notifications.

> **Important**
> When you change notifications, your changes take effect immediately. Jobs that you have already submitted and that Elastic Transcoder has not started to process are affected in addition to jobs that you submit after you change notifications.

To configure Elastic Transcoder to notify you of changes in job status when you're using the Elastic Transcoder API:

1. Create one or more Amazon SNS topics, and subscribe to each topic. For more information, see the Amazon Simple Notification Service documentation.

   If you already have Amazon SNS topics that you want to use, you can skip this step.
2. Create a pipeline or update an existing pipeline. For each change in status for which you want notification (Progressing, Completed, Warning, Error), specify the applicable Amazon SNS topic that you created in Step 1. For more information, see Create Pipeline (p. 128), Update Pipeline (p. 156), or Update Pipeline Notifications (p. 170).
3. Test notifications for `Progressing` and `Completed` statuses by submitting a job to Elastic Transcoder.

You can also use the Elastic Transcoder console to configure notifications. If you use the console, you can create Amazon SNS topics as you create or update the pipeline. However, you still need to use the Amazon SNS console, API, or CLI to subscribe to the new topics.

When Elastic Transcoder sends you notification of a change in status, the message returned in the notification is in the following JSON format. Note that the message includes the status for each output as well as a state for the job:

```
{
   "state" : "PROGRESSING|COMPLETED|WARNING|ERROR",
   "errorCode" : "the code of any error that occurred",
   "messageDetails" : "the notification message you created in Amazon SNS",
   "version" : "API version that you used to create the job",
   "jobId" : "value of Job:Id object that Elastic Transcoder
              returns in the response to a Create Job request",
   "pipelineId" : "value of PipelineId object
                   in the Create Job request",
   "input" : {
      job Input settings
   },
   "outputKeyPrefix" : "prefix for file names in Amazon S3 bucket",
   "outputs": [
      {
         applicable job Outputs settings,
         "status" : "Progressing|Completed|Warning|Error"
      },
      {...}
   ],
   "playlists": [
      {
         applicable job playlists settings
      }
   ],
   "userMetadata": {
      "metadata key": "metadata value"
   }
}
```

Captions

You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language.

  Supported input formats include CEA-608 (EIA-608, first non-empty channel only), CEA-708 (EIA-708, first non-empty channel only), and mov-text.

  The supported output format is mov-text (only for MP4).

  Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions tracks per file.

  Valid input formats include: DFXP (first div element only), EBU-TT, SCC, SMPT, SRT, TTML, (first div element only), and WebVTT.

  Valid output formats include: DFXP (first div element only), SCC, SRT, and WebVTT.

If you want TTML or SMPTE-TT compatible captions, specify DFXP as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

For information about transcoding captions when you're using the Elastic Transcoder console, see Output Details, Part 2, Clip and Caption Settings (p. 25). For information about transcoding captions when you're using the Elastic Transcoder API, see the documentation about the Create Job (p. 181) API action beginning with the Captions (p. 191) element.

For more information on embedded files, see the Subtitle (caption) Wikipedia page.

For more information on sidecar files, see the Metadata Platform and Sidecar file Wikipedia pages.

Watermarks

You can configure Elastic Transcoder to overlay up to four graphics, known as watermarks, over a video during transcoding. For example, the logo that a television channel sometimes displays over its broadcast is a watermark.

Watermarks appear for the duration of the transcoded video in the size, position, scale, and opacity that you specify. Opacity controls the density of a watermark, from completely transparent (invisible) to completely obscuring the portion of the video that it overlays.

The graphics that you use for watermarks can be in .png or .jpg format; if you want to display a watermark that is not rectangular, use the .png format, which supports transparency. You save the graphics in the same Amazon S3 bucket in which you save the videos that you want to transcode. You specify this bucket, the input bucket, in pipeline settings.

To add watermarks, you specify settings both in presets and in jobs:

- **Presets** include settings for the size, the location, and the opacity of up to four watermarks. You assign an identifier to each group of settings. When you create a job, this identifier allows you to control which preset settings are used for which graphics.

  For information about specifying watermark settings in presets by using the Elastic Transcoder console, see Creating a Preset in Elastic Transcoder (p. 39). For information about specifying settings by using the Elastic Transcoder API, see Create Preset (p. 275).
- **Jobs** identify the .png or .jpg images that you want to use as watermarks and the group of settings from the preset that you want to use for each watermark. When you create a job and specify a preset that includes watermark settings, you can choose to add graphics for none, some, or all of the watermark settings that you defined in the preset.

  For information about specifying watermark settings in jobs by using the Elastic Transcoder console, see Creating a Job in Elastic Transcoder (p. 17). For information about specifying settings by using the Elastic Transcoder API, see Create Job (p. 181).

System Presets

The value of the `Id` object for the preset that you want to use for a job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset.

Note
If you created any presets before AAC profiles were added, Elastic Transcoder will use the AAC-LC profile for those presets.

To use one of the system presets that are included with Elastic Transcoder, use the following IDs. (You can also get these IDs using List Presets (p. 296).)

Description	PresetId
Audio AAC - 256k	1351620000001-100110
Audio AAC - 160k	1351620000001-100120
Audio AAC - 120k	1351620000001-100130
Audio AAC - 64k (Uses auto for `Audio:CodecOptions:Profile`)	1351620000001-100141
Audio MP3 - 320k	1351620000001-300010
Audio MP3 - 192k	1351620000001-300020
Audio MP3 - 160k	1351620000001-300030
Audio MP3 - 128k	1351620000001-300040
Amazon Kindle Fire HDX	1351620000001-100150
Amazon Kindle Fire HD 8.9	1351620000001-100090
Amazon Kindle Fire HD	1351620000001-100080
Amazon Kindle Fire	1351620000001-100100
Apple TV 3G, Roku HD/2 XD	1351620000001-100060
Apple TV 2G	1351620000001-100050

Description	PresetId
Full HD 1080i50	1351620000001-100180
Full HD 1080i60	1351620000001-100190
Generic 1080p	1351620000001-000001
Generic 720p	1351620000001-000010
Generic 480p 16:9	1351620000001-000020
Generic 480p 4:3	1351620000001-000030
Generic 360p 16:9	1351620000001-000040
Generic 360p 4:3	1351620000001-000050
Generic 320x240 (Uses auto for `Audio:CodecOptions:Profile`)	1351620000001-000061
Gif (Animated)	1351620000001-100200
HLS v3 (Apple HTTP Live Streaming), 2 megabits/second	1351620000001-200010
HLS v3 and v4 (Apple HTTP Live Streaming), 2 megabits/second, Video-only	1351620000001-200015
HLS v3 (Apple HTTP Live Streaming), 1.5 megabits/second	1351620000001-200020
HLS v3 and v4 (Apple HTTP Live Streaming), 1.5 megabits/second, Video-only	1351620000001-200025
HLS v3 (Apple HTTP Live Streaming), 1 megabit/second	1351620000001-200030
HLS v3 and v4 (Apple HTTP Live Streaming), 1 megabit/second, Video-only	1351620000001-200035
HLS v3 (Apple HTTP Live Streaming), 600 kilobits/second	1351620000001-200040
HLS v3 and v4 (Apple HTTP Live Streaming), 600 kilobits/second, Video-only	1351620000001-200045
HLS v3 (Apple HTTP Live Streaming), 400 kilobits/second	1351620000001-200050
HLS v3 and v4 (Apple HTTP Live Streaming), 400 kilobits/second, Video-only	1351620000001-200055
HLS v3 and v4 Audio, 160k	1351620000001-200060
HLS v3 and v4 Audio, 64k (Uses auto for `Audio:CodecOptions:Profile`)	1351620000001-200071
iPhone 5, iPhone 4S, iPad 4G and 3G, iPad mini, Samsung Galaxy S2/S3/Tab 2	1351620000001-100020
iPhone 4, iPod touch 5G and 4G, iPad 2G and 1G	1351620000001-100010

Description	PresetId
iPhone 3GS	1351620000001-100030
iPod touch, iPhone 3 and 1, iPod classic	1351620000001-100040
NTSC	1351620000001-100160
PAL	1351620000001-100170
Smooth Streaming 3 megabits/second	1351620000001-400010
Smooth Streaming 2 megabits/second	1351620000001-400020
Smooth Streaming 1.5 megabits/second	1351620000001-400030
Smooth Streaming 1 megabits/second	1351620000001-400040
Smooth Streaming 800 kilobits/second	1351620000001-400050
Smooth Streaming 600 kilobits/second	1351620000001-400060
Smooth Streaming 500 kilobits/second	1351620000001-400070
Smooth Streaming 400 kilobits/second	1351620000001-400080
Web: Facebook, SmugMug, Vimeo, YouTube	1351620000001-100070
Web: Flash Video	1351620000001-100210

The following system presets are no longer supported, and are not recommended for use:

Description	Deprecated PresetId	Recommended PresetId
Audio AAC - 64k (Uses AAC-LC for `Audio:CodecOptions:Profile`)	1351620000001-100140	1351620000001-100141
Amazon Kindle Fire HD 8.9	1351620000000-100090	1351620000001-100090
Amazon Kindle Fire HD	1351620000000-100080	1351620000001-100080
Amazon Kindle Fire	1351620000000-100100	1351620000001-100100
Apple TV 3G	1351620000000-100060	1351620000001-100060
Apple TV 2G	1351620000000-100050	1351620000001-100050
Generic 1080p	1351620000000-000001	1351620000001-000001
Generic 720p	1351620000000-000010	1351620000001-000010
Generic 480p 16:9	1351620000000-000020	1351620000001-000020
Generic 480p 4:3	1351620000000-000030	1351620000001-000030
Generic 360p 16:9	1351620000000-000040	1351620000001-000040
Generic 360p 4:3	1351620000000-000050	1351620000001-000050

Description	Deprecated PresetId	Recommended PresetId
Generic 320x240 (Uses AAC-LC for `Audio:CodecOptions:Profile`)	1351620000001-000060	1351620000001-000061
Generic 320x240	1351620000000-000060	1351620000001-000060
HLS Audio, 64k (Uses AAC-LC for `Audio:CodecOptions:Profile`)	1351620000001-200070	1351620000001-200071
iPhone4S	1351620000000-100020	1351620000001-100020
iPhone4	1351620000000-100010	1351620000001-100010
iPhone3GS	1351620000000-100030	1351620000001-100030
iPod Touch	1351620000000-100040	1351620000001-100040
Web	1351620000000-100070	1351620000001-100070

Controlling Access to Elastic Transcoder

Elastic Transcoder lets you use AWS Identity and Access Management (IAM) to control what users can do with Elastic Transcoder, and to control Elastic Transcoder's access to other AWS services that Elastic Transcoder requires. You control access using IAM policies, which are a collection of permissions that can be associated with a user, an IAM group, or a role.

To control what users can do with Elastic Transcoder (for example, who is allowed to create and manage pipelines and jobs), you can create policies for users. For IAM users in your account, you can attach the policy directly to the IAM user or to an IAM group. If you are granting permissions to an IAM user in another AWS account, known as delegation, or to users signing in from an external identity system, known as federation, you can attach the policy to a role and allow the user to assume that role. For more information on delegation and federation, see Roles (Delegation and Federation) in the Using IAM guide.

To control Elastic Transcoder's access to other AWS services, you can create service roles. These are IAM roles that you assign when you create a pipeline, and that give Elastic Transcoder itself permissions to perform the tasks associated with transcoding.

As an example of how user and service roles are both important during the transcoding process, Elastic Transcoder needs a service role in order to get files from an Amazon S3 bucket and store the transcoded files in another Amazon S3 bucket, while a user needs an IAM role that allows them to create a job in Elastic Transcoder.

For more information about IAM, see Using IAM. For more information about service roles, see Creating a Role for an AWS Service.

Topics

- Controlling User Access to Elastic Transcoder (p. 75)
- Service Roles for Elastic Transcoder Pipelines (p. 79)
- Security Best Practices (p. 80)

Controlling User Access to Elastic Transcoder

Example User Policies for Elastic Transcoder

To allow users to perform Elastic Transcoder administrative functions, such as creating pipelines and running jobs, you must have a policy that you can associate with the user. This section shows three policies for controlling access to Elastic Transcoder operations and to the operations of related services that Elastic Transcoder relies on. You can give users of your AWS account access to all Elastic Transcoder operations or to only a subset of them.

To use these policies with an IAM user, you attach them directly to the IAM user or to an IAM group that the user belongs to. To use these policies with a delegated or federated user, you attach them to an IAM role that the delegated or federated user will assume.

For more information on managing policies, see Managing IAM Policies in the Using IAM guide.

Give Users Read-only Access to Elastic Transcoder and Amazon S3

The following policy grants users read-only access to Elastic Transcoder resources and access to the list operation of Amazon S3. This policy is useful for users who need permissions to find and watch transcoded files and to see what buckets are available to the IAM account, but who don't need the ability to update, create, or delete resources or files. This policy also allows a user to list all available pipelines, presets, and jobs for the IAM account. To restrict access to a particular bucket, see Restricting Users to Certain Resources (p. 77).

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "elastictranscoder:Read*",
            "elastictranscoder:List*",
            "s3:List*"
         ],
         "Resource": "*"
      }
   ]
}
```

Give Users Permission to Create Jobs

The following policy grants users the permissions to list and get all Elastic Transcoder resources associated with the account, create or modify jobs and presets, and use the list operations of Amazon S3 and Amazon SNS.

This policy is useful for users who need the ability to modify transcoding settings, and the ability to create or delete presets or jobs. It does not allow the user to create, update, or delete pipelines, Amazon S3 buckets, or Amazon SNS notifications.

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "elastictranscoder:Read*",
               "elastictranscoder:List*",
               "elastictranscoder:*Job",
               "elastictranscoder:*Preset",
               "s3:List*",
               "sns:List*"
           ],
           "Resource": "*"
       }
    ]
}
```

Give Users Broad Access to Elastic Transcoder and Related Services

The following policy lets users perform any Elastic Transcoder or CloudFront action, manage Amazon S3 buckets, create and list IAM roles, and create and list Amazon SNS notifications.

This policy is useful for users who are administrators for Elastic Transcoder.

We strongly recommend that for extra security, users who have administrative permissions use multi-factor authentication (MFA). For more information, see Using Multi-Factor Authentication (MFA) Devices with AWS in the Using IAM guide.

Important
We recommend that you do not add policies to IAM roles that include permissions for IAM actions. The policy illustrated here is useful for IAM users in your own account, but should not be assigned to a role that can be assumed by users in another account or by a federated user.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action": [
            "elastictranscoder:*",
            "cloudfront:*",
            "s3:List*",
            "s3:Put*",
            "s3:Get*",
            "s3:*MultipartUpload*",
            "iam:CreateRole",
            "iam:PutRolePolicy",
            "iam:GetRolePolicy",
            "iam:List*",
            "sns:CreateTopic",
            "sns:List*"
         ],
         "Resource":[
            "*"
         ]
      }
```

```
    ]
}
```

List of Elastic Transcoder Operations with Controllable Access

The following is the full list of Elastic Transcoder operations.

```
    elastictranscoder:CancelJob
    elastictranscoder:CreateJob
    elastictranscoder:CreatePipeline
    elastictranscoder:CreatePreset
    elastictranscoder:DeletePipeline
    elastictranscoder:DeletePreset
    elastictranscoder:ListJobsByPipeline
    elastictranscoder:ListJobsByStatus
    elastictranscoder:ListPipelines
    elastictranscoder:ListPresets
    elastictranscoder:ReadJob
    elastictranscoder:ReadPipeline
    elastictranscoder:ReadPreset
    elastictranscoder:TestRole
    elastictranscoder:UpdatePipeline
    elastictranscoder:UpdatePipelineNotifications
    elastictranscoder:UpdatePipelineStatus
```

Restricting Users to Certain Resources

In addition to restricting access to operations (actions), you can further restrict access to specific jobs, pipelines, and presets, which is referred to as granting resource-level permissions.

To restrict or grant access to a subset of Elastic Transcoder resources, put the ARN of the resource in the resource element of your policy. Elastic Transcoder ARNs have the following general format:

```
arn:aws:elastictranscoder:region:account:resource/ID
```

Replace the *region*, *account*, *resource*, and *ID* variables with valid values. Valid values can be the following:

- *region*: The name of the region. A list of regions is available here. To indicate all regions, use a wildcard (*). You must specify a value.
- *account*: The ID of the AWS account. You must specify a value.
- *resource*: The type of Elastic Transcoder resource; `preset`, `pipeline`, or `job`.
- *ID*: The ID of the specific preset, pipeline, or job, or * to indicate all resources of the specified type that are associated with the current AWS account.

For example, the following ARN specifies all preset resources in the `us-west-2` region for the account `111122223333`:

```
arn:aws:elastictranscoder:us-west-2:111122223333:preset/*
```

You can find the ARN of a resource by clicking the magnifying-glass icon ( ) next to the resource name in the pipeline, preset, or job console pages.

For more information, see Resources in *Using IAM*.

Example Policy for Restricting Resources for Users

The following policy grants permissions to the bucket named `example_bucket` in Amazon S3, list and read permissions for everything in Elastic Transcoder, and permission to create jobs in the pipeline named `example_pipeline`.

This policy is useful for SDK and CLI users who need to be able to see what files and resources are available, and use those resources to create their own transcoding jobs. It does not allow for updating or deleting resources, creating resources other than jobs, or for working with resources other than the ones specified here, and will not work for console users.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"1",
         "Effect":"Allow",
         "Action":[
            "s3:ListAllMyBuckets",
            "s3:GetBucketLocation",
            "s3:ListBucket",
            "s3:GetObject",
            "s3:PutObject"
            ],
         "Resource":[
             "arn:aws:s3:::example_bucket",
             "arn:aws:s3:::example_bucket/*"
         ]
      },
      {
         "Sid":"2",
         "Effect":"Allow",
         "Action":[
            "elastictranscoder:List*",
            "elastictranscoder:Read*",
            "elastictranscoder:CreateJob"
         ],
         "Resource":[
          "arn:aws:elastictranscoder:region:account:pipeline/example_pipeline",

            "arn:aws:elastictranscoder:region:account:job/*"
         ]
      }
   ]
}
```

Service Roles for Elastic Transcoder Pipelines

When you create a pipeline that manages your transcoding jobs, you must specify an IAM service role. The IAM service role has a policy that specifies the permissions used by that pipeline for transcoding.

You have two options when you specify a role for a pipeline:

- Use the default role, which includes only the permissions that Elastic Transcoder needs for transcoding. If you use the Elastic Transcoder console to create your pipelines, when you create your first pipeline the console gives you the option to create the default role automatically. You must have administrative permissions to create IAM service roles, including the default role.
- Choose an existing role. In this case, you must have previously created the role in IAM and attached a policy to the role that gives Elastic Transcoder sufficient permissions to transcode your files. This is useful if you want to use the role for other AWS services as well.

The Default IAM Role that the Elastic Transcoder Console Creates for Pipelines

The default role created by Elastic Transcoder lets Elastic Transcoder perform the following operations:

- Get a file from an Amazon S3 bucket for transcoding.
- List the contents of any Amazon S3 bucket.
- Save a transcoded file to an Amazon S3 bucket.
- Create an Amazon S3 multipart upload.
- Publish notification to any SNS topic.

The policy prevents Elastic Transcoder from performing any of the following operations:

- Perform any Amazon SNS delete operations, or add or remove a policy statement in a topic.
- Perform any Amazon S3 bucket or item delete operations, or add, remove, or modify a bucket policy.

The access (permission) policy definition for the default role looks like:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"1",
         "Effect":"Allow",
         "Action":[
            "s3:Get*",
            "s3:ListBucket",
            "s3:Put*",
            "s3:*MultipartUpload*"
         ],
         "Resource":"*"
      },
      {
         "Sid":"2",
         "Effect":"Allow",
         "Action":"sns:Publish",
```

```
         "Resource":"*"
      },
      {
         "Sid":"3",
         "Effect":"Deny",
         "Action":[
            "sns:*Permission*",
            "sns:*Delete*",
            "sns:*Remove*",
            "s3:*Policy*",
            "s3:*Delete*"
         ],
         "Resource":"*"
      }
   ]
}
```

Security Best Practices

For more information on security best practices, see the IAM Best Practices guide.

Logging Elastic Transcoder API Calls Using CloudTrail

Elastic Transcoder is integrated with CloudTrail, an AWS service that captures information about every request that is sent to the Elastic Transcoder API by your AWS account, including your IAM users. CloudTrail periodically saves log files of these requests to an Amazon S3 bucket that you specify. CloudTrail captures information about all requests, whether they were made using the Elastic Transcoder console, the Elastic Transcoder API, the AWS SDKs, the Elastic Transcoder CLI, or another service, for example, CloudFront.

You can use information in the CloudTrail log files to determine which requests were made to Elastic Transcoder, the source IP address from which each request was made, who made the request, when it was made, and so on. To learn more about CloudTrail, including how to configure and enable it, see the *AWS CloudTrail User Guide*.

Topics

- Elastic Transcoder Information in CloudTrail Log Files (p. 81)
- Understanding Elastic Transcoder Log File Entries (p. 82)

Elastic Transcoder Information in CloudTrail Log Files

When you enable CloudTrail, CloudTrail captures every request that you make to every AWS service that CloudTrail supports. (For a list of supported services, see Supported Services in the *AWS CloudTrail User Guide*.) CloudTrail saves the captured requests in log files in each region separately, and stores them in an Amazon S3 bucket. The log files aren't organized or sorted by service; each log file might contain records from more than one service. CloudTrail determines when to create a new log file.

> **Note**
> CloudTrail supports all Elastic Transcoder API actions.

Every log file entry contains information about who made the request. The user identity information in the log file helps you determine whether the request was made using root or IAM user credentials, using temporary security credentials for a role or federated user, or by another AWS service. For more information, see userIdentity Element in the *AWS CloudTrail User Guide*.

You can store log files for as long as you want. You can also define Amazon S3 lifecycle rules to archive or delete log files automatically.

By default, your log files are encrypted by using Amazon S3 server-side encryption (SSE).

You can choose to have CloudTrail publish Amazon SNS notifications when new log files are delivered if you want to take quick action upon log file delivery. For more information, see Configuring Amazon SNS Notifications in the *AWS CloudTrail User Guide*.

You can also aggregate log files from multiple AWS regions and multiple AWS accounts into a single Amazon S3 bucket. For more information, see Aggregating CloudTrail Log Files to a Single Amazon S3 Bucket in the *AWS CloudTrail User Guide*.

There is no cost to use the CloudTrail service. However, standard rates for Amazon S3 usage apply as well as rates for Amazon SNS usage should you include that option. For pricing details, see the Amazon S3 and Amazon SNS pricing pages.

Understanding Elastic Transcoder Log File Entries

Each JSON-formatted CloudTrail log file can contain one or more log entries. A log entry represents a single request from any source and includes information about the requested action, including any parameters, the date and time of the action, and so on. The log entries are not guaranteed to be in any particular order; they are not an ordered stack trace of API calls.

The `eventSource` element identifies the source of the action that occurred. For example, the following `eventSource` value indicates that Elastic Transcoder was called:

```
elastictranscoder.amazonaws.com
```

The `eventName` element identifies the action that occurred. For example, the following `eventName` value indicates that a job was created:

```
CreateJob
```

The following example shows a CloudTrail log entry that demonstrates five actions:

- Creating a job. The value of `eventName` is `CreateJob`.
- Listing jobs by status. The value of `eventName` is `ListJobsByStatus`.
- Getting a job. The value of `eventName` is `ReadJob`.
- Deleting a Preset. The value of `eventName` is `DeletePreset`.
- Deleting a pipeline. The value of `eventName` is `DeletePipeline`.

```
{
    "Records": [
        {
            "eventVersion": "1.02",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "A1B2C3D4E5F6G7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/smithj",
                "accountId": "111122223333",
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
```

```
            },
            "eventTime": "2014-09-29T19:29:02Z",
            "eventSource": "elastictranscoder.amazonaws.com",
            "eventName": "CreateJob",
            "awsRegion": "us-west-2",
            "sourceIPAddress": "192.0.2.17",
            "userAgent": "aws-sdk-ruby/1.39.0 ruby/1.9.3 x86_64-linux",
            "requestParameters": {
                "input": {
                    "interlaced": "auto",
                    "resolution": "auto",
                    "frameRate": "auto",
                    "aspectRatio": "auto",
                    "container": "auto",
                    "key": "source/audio/cheesytoast.wav"
                },
                "output": {
                    "presetId": "1234-preset-example",
                    "key": "output/testing-toast.mp4",
                    "thumbnailPattern": "",
                    "rotate": "auto"
                },
                "pipelineId": "1234-pipeline-example"
            },
            "responseElements": {
                "job": {
                    "output": {
                        "rotate": "auto",
                        "presetId": "1234-preset-example",
                        "thumbnailPattern": "",
                        "watermarks": [],
                        "id": "1",
                        "key": "output/testing-toast.mp4",
                        "status": "Submitted"
                    },
                    "status": "Submitted",
                    "playlists": [],
                    "arn": "arn:aws:elastictranscoder:us-west-
2:111122223333:job/1234-job-example",
                    "id": "1234-job-example",
                    "outputs": [
                        {
                            "rotate": "auto",
                            "presetId": "1234-preset-example",
                            "thumbnailPattern": "",
                            "watermarks": [],
                            "id": "1",
                            "key": "output/testing-toast.mp4",
                            "status": "Submitted"
                        }
                    ],
                    "pipelineId": "1234-pipeline-example",
                    "input": {
                        "interlaced": "auto",
                        "resolution": "auto",
                        "frameRate": "auto",
                        "aspectRatio": "auto",
                        "container": "auto",
```

```
                    "key": "source/audio/cheesytoast.wav"
                }
            }
        },
        "requestID": "4e6b66f9-d548-11e3-a8a9-73e33example",
        "eventID": "5ab02562-0fc5-43d0-b7b6-90293example",
        "eventType": "AwsApiCall",
        "recipientAccountId": "111122223333"
    },
    {
        "eventVersion": "1.02",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
        },
        "eventTime": "2014-09-29T19:29:18Z",
        "eventSource": "elastictranscoder.amazonaws.com",
        "eventName": "ListJobsByStatus",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "192.0.2.17",
        "userAgent": "aws-sdk-ruby/1.39.0 ruby/1.9.3 x86_64-linux",
        "requestParameters": {
            "status": "Submitted",
            "ascending": "false"
        },
        "responseElements": null,
        "requestID": "52de9f97-d548-11e3-8fb9-4dad0example",
        "eventID": "eb91f423-6dd3-4bb0-a148-3cdfbexample",
        "eventType": "AwsApiCall",
        "recipientAccountId": "111122223333"
    },
    {
        "eventVersion": "1.02",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
        },
        "eventTime": "2014-09-29T19:28:50Z",
        "eventSource": "elastictranscoder.amazonaws.com",
        "eventName": "ReadJob",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "192.0.2.17",
        "userAgent": "aws-sdk-ruby/1.39.0 ruby/1.9.3 x86_64-linux",
        "requestParameters": {
            "id": "1412018849233-f2czlr"
        },
        "responseElements": null,
        "requestID": "497b3622-d548-11e3-8fb9-4dad0example",
        "eventID": "c32289c7-005a-46f7-9801-cba41example",
        "eventType": "AwsApiCall",
        "recipientAccountId": "111122223333"
    },
```

```
        {
            "eventVersion": "1.02",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "A1B2C3D4E5F6G7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/smithj",
                "accountId": "111122223333",
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
            },
            "eventTime": "2014-09-29T19:29:18Z",
            "eventSource": "elastictranscoder.amazonaws.com",
            "eventName": "DeletePreset",
            "awsRegion": "us-west-2",
            "sourceIPAddress": "192.0.2.17",
            "userAgent": "aws-sdk-ruby/1.39.0 ruby/1.9.3 x86_64-linux",
            "requestParameters": {
                "id": "1234-preset-example"
            },
            "responseElements": null,
            "requestID": "4e200613-d548-11e3-a8a9-73e33example",
            "eventID": "191ebb93-66b7-4517-a741-92b0eexample",
            "eventType": "AwsApiCall",
            "recipientAccountId": "111122223333"
        },
        {
            "eventVersion": "1.02",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "A1B2C3D4E5F6G7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/smithj",
                "accountId": "111122223333",
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
            },
            "eventTime": "2014-09-29T19:29:01Z",
            "eventSource": "elastictranscoder.amazonaws.com",
            "eventName": "DeletePipeline",
            "awsRegion": "us-west-2",
            "sourceIPAddress": "192.0.2.17",
            "userAgent": "aws-sdk-ruby/1.39.0 ruby/1.9.3 x86_64-linux",
            "requestParameters": {
                "id": "1412018848038-nkomx0"
            },
            "responseElements": null,
            "requestID": "42ca4299-d548-11e3-8fb9-4dad0example",
            "eventID": "7aeb434f-eb55-4e2a-82d8-417d5example",
            "eventType": "AwsApiCall",
            "recipientAccountId": "111122223333"
        },
    ]
}
```

AWS SDK Sample Code

This section contains sample code that shows how to use the AWS SDKs for Java, Ruby, PHP, and Python to perform the following operations:

- Create an HLS playlist in Amazon Elastic Transcoder
- Create a job in Elastic Transcoder
- Poll an Amazon Simple Queue Service (Amazon SQS) queue
- Parse a notification from the Amazon Simple Notification Service (Amazon SNS)
- Handle the notification from Amazon SNS

These operations represent Elastic Transcoder best practices. Polling an Amazon SQS queue allows for long polling, which scales more efficiently than regular polling, and setting up Amazon SNS notifications allows Elastic Transcoder to deliver messages to the Amazon SQS queue.

The sample code shows how to transcode for HLS (HTTP Live Streaming) and how to handle the Amazon SNS notification once it has been removed from the Amazon SQS queue.

If you are using a language for which an SDK exists, we recommend that you use the SDK rather than try to work your way through the APIs. You'll find that using the SDKs makes authentication simpler, integrates easily with your development environment, and provides easy access to related commands.

Topics

- Background for AWS SDK Sample Code (p. 87)
- Setting Up Your Environment (p. 87)
- Java Samples (p. 101)
- Ruby Samples (p. 104)
- PHP Samples (p. 107)
- Python Samples (p. 110)

Background for AWS SDK Sample Code

HTTP Live Streaming (HLS)

You use HTTP Live Streaming (HLS) outputs in order to deliver adaptive-bit rate content to HLS-enabled players. Examples of HLS-enabled devices are Android 4+ devices, iOS devices, desktop media players such as QuickTime or VLC, and browser players such as jwplayer. Adaptive bit rate allows you to automatically adjust the quality of delivered content based on the quality of the client connection.

Notifications

If you poll the Elastic Transcoder's `ReadJob` API to track job status, you need to continuously call `ReadJob` on every submitted job. This methodology cannot scale as the number of transcode jobs increases. To solve this problem, Elastic Transcoder can publish notifications to Amazon SNS which provides an event-driven mechanism for tracking job status.

Each Elastic Transcoder notification is sent as a JSON object in the `Message` field. Because notifications are themselves sent in JSON format, you need to escape the Elastic Transcoder notification in the message string. For information about the format and content of Elastic Transcoder notifications, see the Notifications section.

When you are writing custom code to handle job status notifications, follow these best practices:

- **Handlers must be idempotent.** It is possible that a notification will be delivered more than once.
- **Handlers must support out of order notifications.** It is possible that notifications will be delivered out of order.
- **Handlers must be able to process a notification for any job.** There is no way to guarantee that a notification for a particular job will be delivered to a particular worker.
- **Handlers should be short operations.** All of the messages must be handled and deleted before the visibility timeout. If the visibility timeout is 15 seconds and a maximum of 5 messages will be returned from Amazon SQS, each message must be handled and deleted from the queue in less than 3 seconds. If handling takes longer than this, unprocessed messages will time out and be delivered to a different worker.

For Java, Python, and Ruby, we recommend that you consume notifications by polling an Amazon SQS queue that is subscribed to your notification topic. Since Amazon SQS uses a long-poll mechanism, polling the Amazon SQS queue gives a scalable method for consuming job notifications. Amazon SQS also simplifies availability and scaling for when hosts fail or during times of high load, and generally requires no special ACL setup.

For PHP running in Apache, we recommend that you subscribe your endpoint directly to the Amazon SNS topic. This requires that your endpoint be publicly available since Amazon SNS will need to be able to push notifications directly to you.

Setting Up Your Environment

To run the sample code, you need to have an AWS environment set up. This section walks you through the setup process, and shows how to create the AWS resources that Elastic Transcoder needs in order to work optimally.

Topics

- Setting up your Java environment (p. 88)

- Setting up your Ruby environment (p. 92)
- Setting up your PHP environment (p. 95)
- Setting up your Python environment (p. 98)

Setting up your Java environment

This section walks you through setting up your Java environment.

Topics

- Setting up an AWS account (p. 88)
- Setting up the AWS SDK for Java (p. 88)
- Creating Amazon S3 input and output buckets (p. 89)
- Creating an Amazon SNS topic to receive job status notifications (p. 89)
- Creating an Amazon SQS queue to poll for job status notifications (p. 89)
- Subscribing your Amazon SQS queue to your Amazon SNS topic (p. 90)
- Creating an Elastic Transcoder pipeline (p. 90)
- Creating an Amazon CloudFront distribution to deliver content in a scalable manner (p. 90)
- Modifying the Amazon S3 bucket policy (p. 91)

The sample code makes two major assumptions:

- Samples are written to work with Java version 1.6 or higher.
- Samples are run using Eclipse with the AWS Toolkit for Eclipse.

Setting up an AWS account

To sign up for an AWS account

1. Open http://aws.amazon.com/, and then click **Sign Up**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Setting up the AWS SDK for Java

These samples assume you are using the AWS Toolkit for Eclipse. You need both the AWS SDK for Java and the following Jackson JSON processor JAR files :

- Jackson Core
- Jackson Databind
- Jackson Annotations

The Jackson JSON processor handles job status notifications.

Alternatively, if you are using Maven to manage your dependencies, you can add the following snippets to your `pom.xml` file:

Version property:

```
    <jackson-2-version>2.2.3</jackson-2-version>
```

Dependencies:

```
    <dependency>
        <groupId>com.amazonaws</groupId>
        <artifactId>aws-java-sdk</artifactId>
        <version>LATEST</version>
    </dependency>

    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-core</artifactId>
        <version>${jackson-2-version}</version>
    </dependency>

    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-annotations</artifactId>
        <version>${jackson-2-version}</version>
    </dependency>

    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-databind</artifactId>
        <version>${jackson-2-version}</version>
    </dependency>
```

For more information, see the AWS SDK for Java Documentation.

Creating Amazon S3 input and output buckets

1. Open the Amazon S3 console.
2. In the console, click **Create Bucket**.
3. In the **Bucket Name** field, enter the desired name for your input bucket.
4. Still in the Amazon S3 console, click **Create Bucket** again.
5. In the **Bucket Name** field, enter the desired name for your output bucket.

Creating an Amazon SNS topic to receive job status notifications

1. Open the Amazon SNS console.
2. In the console, click **Create and Add** and select **Create New Topic**.
3. In the **Topic Name** field, enter **`ets-sample-topic`** and then click **Create Topic**.
4. Record the ARN of `ets-sample-topic`.

Creating an Amazon SQS queue to poll for job status notifications

1. Open the Amazon SQS console.

2. In the console, click **Create New Queue**.
3. In the **Queue Name** field, enter **`ets-sample-queue`** and then click **Create Queue**.
4. Record the Amazon SQS queue URL.

Subscribing your Amazon SQS queue to your Amazon SNS topic

1. Open the Amazon SQS console.
2. In the console, select **ets-sample-queue** from the list of queues.
3. Under **Queue Actions**, select **Subscribe Queue to Amazon SNS Topic**.
4. Under **Choose a Topic**, select **ets-sample-topic** and then click **Subscribe**.

You should see a confirmation that your queue has been successfully subscribed to your topic.

Creating an Elastic Transcoder pipeline

1. Open the Amazon Elastic Transcoder console.
2. In the console, click **Create New Pipeline**.
3. In the **Pipeline Name** field, enter **`ets-sample-pipeline`**.
4. In the **Input Bucket** field, enter the name of your input bucket.
5. Leave **IAM Role** as **`Console Default Role`**.
6. In **Configure Amazon S3 Bucket for Transcoded Files and Playlists**, enter the name of your output bucket.
7. Still in **Configure Amazon S3 Bucket for Transcoded Files and Playlists**, select the **standard storage class**.
8. Expand the **Notifications** drop-down menu.
9. For all four event types, select **Use an existing SNS topic**, and under **Select a Topic**, select **`ets-sample-topic`**.
10. Click **Create Pipeline**.
11. Record the Elastic Transcoder pipeline ID.

Creating an Amazon CloudFront distribution to deliver content in a scalable manner

1. Open the Amazon CloudFront console.
2. In the navigation pane of the console, expand **Private Content** and then click **Origin Access Identity**.
3. Click **Create Origin Access Identity**.
4. Click **Create**.
5. Record the Amazon S3 canonical user ID that is generated when you create your origin access identity.
6. In the navigation pane, click **Distribution**.
7. Click **Create Distribution**.
8. Select **Web** and then click **Continue**.
9. Under **Origin Settings**, enter your Amazon S3 output bucket as the **Origin Domain Name**.
10. In the **Origin ID** field, enter **`S3-transcoder-sample-output`**.
11. For **Restrict Bucket Access**, select **Yes**.
12. Click **Create Distribution**, and record the domain name of the distribution.

Modifying the Amazon S3 bucket policy

1. Open the Amazon S3 console.
2. Next to the output bucket, click the **magnifying glass icon** to bring up the bucket properties.
3. Expand **Permissions** and click **Add bucket policy**.
4. Enter the following policy statement, replacing **CloudFront-oai-s3-canonical-user-id** with **`your recorded Amazon S3 canonical user ID`** and **s3-output-bucket-name** with **`the name of your output bucket`**.
5. Click **Save**.

```
{
  "Version":"2012-10-17",
  "Id":"PolicyForCloudFrontPrivateContent",
  "Statement":[{
     "Sid":" Granting CloudFront OAI access to private content",
     "Effect":"Allow",
     "Principal":{ "CanonicalUser":"<CloudFront-oai-s3-canonical-user-id>" },
     "Action":"s3:GetObject",
     "Resource":"arn:aws:s3:::<s3-output-bucket-name>/*"
   }]
}
```

Skip to the Java Samples (p. 101).

Setting up your Ruby environment

This section walks you through setting up your Ruby environment.

Topics

- Setting up an AWS account (p. 92)
- Setting up the AWS SDK for Ruby (p. 92)
- Creating Amazon S3 input and output buckets (p. 92)
- Creating an Amazon SNS topic to receive job status notifications (p. 93)
- Creating an Amazon SQS queue to poll for job status notifications (p. 93)
- Subscribing your Amazon SQS queue to your Amazon SNS topic (p. 93)
- Creating an Elastic Transcoder pipeline (p. 93)
- Creating an Amazon CloudFront distribution to deliver content in a scalable manner (p. 94)
- Modifying the Amazon S3 bucket policy (p. 94)

The sample code makes one major assumption:

- Samples are written to be compatible with Ruby version 1.9.

Setting up an AWS account

To sign up for an AWS account

1. Open http://aws.amazon.com/, and then click **Sign Up**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Setting up the AWS SDK for Ruby

The AWS SDK for Ruby is available here. You can install it easily using Ruby gems with the following command:

```
gem install aws-sdk
```

For more information, see the AWS SDK for Ruby Documentation.

Creating Amazon S3 input and output buckets

1. Open the Amazon S3 console.
2. In the console, click **Create Bucket**.
3. In the **Bucket Name** field, enter the desired name for your input bucket.
4. Still in the Amazon S3 console, click **Create Bucket** again.
5. In the **Bucket Name** field, enter the desired name for your output bucket.

Creating an Amazon SNS topic to receive job status notifications

1. Open the Amazon SNS console.
2. In the console, click **Create and Add** and select **Create New Topic**.
3. In the **Topic Name** field, enter **`ets-sample-topic`** and then click **Create Topic**.
4. Record the ARN of `ets-sample-topic`.

Creating an Amazon SQS queue to poll for job status notifications

1. Open the Amazon SQS console.
2. In the console, click **Create New Queue**.
3. In the **Queue Name** field, enter **`ets-sample-queue`** and then click **Create Queue**.
4. Record the Amazon SQS queue URL.

Subscribing your Amazon SQS queue to your Amazon SNS topic

1. Open the Amazon SQS console.
2. In the console, select **ets-sample-queue** from the list of queues.
3. Under **Queue Actions**, select **Subscribe Queue to Amazon SNS Topic**.
4. Under **Choose a Topic**, select **ets-sample-topic** and then click **Subscribe**.

You should see a confirmation that your queue has been successfully subscribed to your topic.

Creating an Elastic Transcoder pipeline

1. Open the Amazon Elastic Transcoder console.
2. In the console, click **Create New Pipeline**.
3. In the **Pipeline Name** field, enter **`ets-sample-pipeline`**.
4. In the **Input Bucket** field, enter the name of your input bucket.
5. Leave **IAM Role** as **`Console Default Role`**.
6. In **Configure Amazon S3 Bucket for Transcoded Files and Playlists**, enter the name of your output bucket.
7. Still in **Configure Amazon S3 Bucket for Transcoded Files and Playlists**, select the **standard storage class**.
8. Expand the **Notifications** drop-down menu.
9. For all four event types, select **Use an existing SNS topic**, and under **Select a Topic**, select **`ets-sample-topic`**.
10. Click **Create Pipeline**.
11. Record the Elastic Transcoder pipeline ID.

Creating an Amazon CloudFront distribution to deliver content in a scalable manner

1. Open the Amazon CloudFront console.
2. In the navigation pane of the console, expand **Private Content** and then click **Origin Access Identity**.
3. Click **Create Origin Access Identity**.
4. Click **Create**.
5. Record the Amazon S3 canonical user ID that is generated when you create your origin access identity.
6. In the navigation pane, click **Distribution**.
7. Click **Create Distribution**.
8. Select **Web** and then click **Continue**.
9. Under **Origin Settings**, enter your Amazon S3 output bucket as the **Origin Domain Name**.
10. In the **Origin ID** field, enter **`S3-transcoder-sample-output`**.
11. For **Restrict Bucket Access**, select **Yes**.
12. Click **Create Distribution**, and record the domain name of the distribution.

Modifying the Amazon S3 bucket policy

1. Open the Amazon S3 console.
2. Next to the output bucket, click the **magnifying glass icon** to bring up the bucket properties.
3. Expand **Permissions** and click **Add bucket policy**.
4. Enter the following policy statement, replacing **CloudFront-oai-s3-canonical-user-id** with **`your recorded Amazon S3 canonical user ID`** and **s3-output-bucket-name** with **`the name of your output bucket`**.
5. Click **Save**.

```
{
  "Version":"2012-10-17",
  "Id":"PolicyForCloudFrontPrivateContent",
  "Statement":[{
     "Sid":" Granting CloudFront OAI access to private content",
     "Effect":"Allow",
     "Principal":{ "CanonicalUser":"<CloudFront-oai-s3-canonical-user-id>" },
     "Action":"s3:GetObject",
     "Resource":"arn:aws:s3:::<s3-output-bucket-name>/*"
   }]
}
```

Skip to the Ruby Samples (p. 104).

Setting up your PHP environment

This section walks you through setting up your PHP environment.

Topics

- Setting up an AWS account (p. 95)
- Installing the sample code (p. 95)
- Setting up the AWS SDK for PHP (p. 96)
- Creating Amazon S3 input and output buckets (p. 96)
- Creating Amazon SNS topic to receive job status notifications (p. 96)
- Subscribing your server to Amazon SNS (p. 96)
- Creating an Elastic Transcoder pipeline (p. 96)
- Creating Amazon CloudFront distribution to deliver content in a scalable manner (p. 97)
- Modifying the Amazon S3 bucket policy (p. 97)

The sample code makes three major assumptions:

- PHP samples are run inside an Apache server.
- Apache server must be able to take input from the internet to receive Amazon SNS notifications.
- Samples are written to work with PHP version 5.4 or higher.

Setting up an AWS account

To sign up for an AWS account

1. Open http://aws.amazon.com/, and then click **Sign Up**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Installing the sample code

To use the PHP sample code, you must first download and install it.

- Download the sample code:
  - HLS Sample Code.
  - Notification Sample Code.
- Unzip the sample code.
- Place the code under your Apache server's DocumentRoot.
- Remove the downloaded sample code .zip file.
- Update the path to your AWS SDK Installation.

> **Note**
> In `HlsJobCreationSample.php` and `JobStatusNotificationsSample.php`, you need to update the path to your `autoload.php`. If you use the phar installation, you can use the path to the `aws.phar` file that you downloaded, which includes all necessary dependencies.

Setting up the AWS SDK for PHP

You can find the AWS SDK for PHP here. For this tutorial, we recommend the phar installation; however, long-term projects are better managed using composer.

For more information, see the AWS SDK for PHP Documentation.

Creating Amazon S3 input and output buckets

1. Open the Amazon S3 console.
2. In the console, click **Create Bucket**.
3. In the **Bucket Name** field, enter the desired name for your input bucket.
4. Still in the Amazon S3 console, click **Create Bucket** again.
5. In the **Bucket Name** field, enter the desired name for your output bucket.

Creating Amazon SNS topic to receive job status notifications

1. Open the Amazon SNS console.
2. In the console, click **Create and Add** and select **Create New Topic**.
3. In the **Topic Name** field, enter **`ets-sample-topic`** and then click **Create Topic**.
4. Record the ARN of `ets-sample-topic`.

Subscribing your server to Amazon SNS

1. Verify that the PHP sample code has been installed.
2. Open the Amazon SNS console.
3. Under **AdditionalActions**, click **Create Subscription**.
4. Select HTTP or HTTPS based on your server's configuration.
5. In **Endpoint**, enter the endpoint of your server. The path of the endpoint should point to `JobStatusNotificationsSampleNotificationHandler.php`.
6. Click **Subscribe**. This sends a subscription request to your PHP endpoint.

The PHP sample code automatically handles the subscription request and confirms a subscription. The subscription request and response are written to `/tmp/subscribe_requests.txt`.

Creating an Elastic Transcoder pipeline

1. Open the Amazon Elastic Transcoder console.
2. In the console, click **Create New Pipeline**.
3. In the **Pipeline Name** field, enter **`ets-sample-pipeline`**.
4. In the **Input Bucket** field, enter the name of your input bucket.
5. Leave **IAM Role** as **`Console Default Role`**.
6. In **Configure Amazon S3 Bucket for Transcoded Files and Playlists**, enter the name of your output bucket.
7. Still in **Configure Amazon S3 Bucket for Transcoded Files and Playlists**, select the **standard storage class**.
8. Expand the **Notifications** drop-down menu.
9. For all four event types, select **Use an existing SNS topic**, and under **Select a Topic**, select **`ets-sample-topic`**.

10. Click **Create Pipeline**.
11. Record the Elastic Transcoder pipeline ID.

Creating Amazon CloudFront distribution to deliver content in a scalable manner

1. Open the Amazon CloudFront console.
2. In the navigation pane of the console, expand **Private Content** and then click **Origin Access Identity**.
3. Click **Create Origin Access Identity**.
4. Click **Create**.
5. Record the Amazon S3 canonical user ID that is generated when you create your origin access identity.
6. In the navigation pane, click **Distribution**.
7. Click **Create Distribution**.
8. Select **Web** and then click **Continue**.
9. Under **Origin Settings**, enter your Amazon S3 output bucket as the **Origin Domain Name**.
10. In the **Origin ID** field, enter **`S3-transcoder-sample-output`**.
11. For **Restrict Bucket Access**, select **Yes**.
12. Click **Create Distribution**, and record the domain name of the distribution.

Modifying the Amazon S3 bucket policy

1. Open the Amazon S3 console.
2. Next to the output bucket, click the **magnifying glass icon** to bring up the bucket properties.
3. Expand **Permissions** and click **Add bucket policy**.
4. Enter the following policy statement, replacing **CloudFront-oai-s3-canonical-user-id** with **`your recorded Amazon S3 canonical user ID`** and **s3-output-bucket-name** with **`the name of your output bucket`**.
5. Click **Save**.

```
{
  "Version":"2012-10-17",
  "Id":"PolicyForCloudFrontPrivateContent",
  "Statement":[{
     "Sid":" Granting CloudFront OAI access to private content",
     "Effect":"Allow",
     "Principal":{ "CanonicalUser":"<CloudFront-oai-s3-canonical-user-id>" },
     "Action":"s3:GetObject",
     "Resource":"arn:aws:s3:::<s3-output-bucket-name>/*"
   }]
}
```

Skip to the PHP Samples (p. 107).

Setting up your Python environment

This section walks you through setting up your Python environment.

Topics

- Setting up an AWS account (p. 98)
- Setting up the AWS SDK for Python (p. 98)
- Creating Amazon S3 input and output buckets (p. 98)
- Creating Amazon SNS topic to receive job status notifications (p. 98)
- Creating Amazon SQS queue to poll for job status notifications (p. 99)
- Subscribing your Amazon SQS queue to your Amazon SNS topic (p. 99)
- Creating an Elastic Transcoder pipeline (p. 99)
- Creating Amazon CloudFront distribution to deliver content in a scalable manner (p. 99)
- Modifying the Amazon S3 bucket policy (p. 100)

It makes one major assumption:

- Samples are written to be compatible with Python version 2.7.

Setting up an AWS account

To sign up for an AWS account

1. Open http://aws.amazon.com/, and then click **Sign Up**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Setting up the AWS SDK for Python

You can find the AWS SDK for Python here. You can also install it easily by using PIP with the following command:

```
pip-2.7 install boto
```

For more information, see the AWS SDK for Python Documentation.

Creating Amazon S3 input and output buckets

1. Open the Amazon S3 console.
2. In the console, click **Create Bucket**.
3. In the **Bucket Name** field, enter the desired name for your input bucket.
4. Still in the Amazon S3 console, click **Create Bucket** again.
5. In the **Bucket Name** field, enter the desired name for your output bucket.

Creating Amazon SNS topic to receive job status notifications

1. Open the Amazon SNS console.

2. In the console, click **Create and Add** and select **Create New Topic**.
3. In the **Topic Name** field, enter **`ets-sample-topic`** and then click **Create Topic**.
4. Record the ARN of `ets-sample-topic`.

Creating Amazon SQS queue to poll for job status notifications

1. Open the Amazon SQS console.
2. In the console, click **Create New Queue**.
3. In the **Queue Name** field, enter **`ets-sample-queue`** and then click **Create Queue**.
4. Record the Amazon SQS queue URL.

Subscribing your Amazon SQS queue to your Amazon SNS topic

1. Open the Amazon SQS console.
2. In the console, select **ets-sample-queue** from the list of queues.
3. Under **Queue Actions**, select **Subscribe Queue to Amazon SNS Topic**.
4. Under **Choose a Topic**, select **ets-sample-topic** and then click **Subscribe**.

You should see a confirmation that your queue has been successfully subscribed to your topic.

Creating an Elastic Transcoder pipeline

1. Open the Amazon Elastic Transcoder console.
2. In the console, click **Create New Pipeline**.
3. In the **Pipeline Name** field, enter **`ets-sample-pipeline`**.
4. In the **Input Bucket** field, enter the name of your input bucket.
5. Leave **IAM Role** as **`Console Default Role`**.
6. In **Configure Amazon S3 Bucket for Transcoded Files and Playlists**, enter the name of your output bucket.
7. Still in **Configure Amazon S3 Bucket for Transcoded Files and Playlists**, select the **standard storage class**.
8. Expand the **Notifications** drop-down menu.
9. For all four event types, select **Use an existing SNS topic**, and under **Select a Topic**, select **`ets-sample-topic`**.
10. Click **Create Pipeline**.
11. Record the Elastic Transcoder pipeline ID.

Creating Amazon CloudFront distribution to deliver content in a scalable manner

1. Open the Amazon CloudFront console.
2. In the navigation pane of the console, expand **Private Content** and then click **Origin Access Identity**.
3. Click **Create Origin Access Identity**.
4. Click **Create**.

5. Record the Amazon S3 canonical user ID that is generated when you create your origin access identity.
6. In the navigation pane, click **Distribution**.
7. Click **Create Distribution**.
8. Select **Web** and then click **Continue**.
9. Under **Origin Settings**, enter your Amazon S3 output bucket as the **Origin Domain Name**.
10. In the **Origin ID** field, enter **`S3-transcoder-sample-output`**.
11. For **Restrict Bucket Access**, select **Yes**.
12. Click **Create Distribution**, and record the domain name of the distribution.

Modifying the Amazon S3 bucket policy

1. Open the Amazon S3 console.
2. Next to the output bucket, click the **magnifying glass icon** to bring up the bucket properties.
3. Expand **Permissions** and click **Add bucket policy**.
4. Enter the following policy statement, replacing **CloudFront-oai-s3-canonical-user-id** with **`your recorded Amazon S3 canonical user ID`** and **s3-output-bucket-name** with **`the name of your output bucket`**.
5. Click **Save**.

```
{
  "Version":"2012-10-17",
  "Id":"PolicyForCloudFrontPrivateContent",
  "Statement":[{
     "Sid":" Granting CloudFront OAI access to private content",
     "Effect":"Allow",
     "Principal":{ "CanonicalUser":"<CloudFront-oai-s3-canonical-user-id>" },
     "Action":"s3:GetObject",
     "Resource":"arn:aws:s3:::<s3-output-bucket-name>/*"
   }]
}
```

Skip to the Python Samples (p. 110).

Java Samples

All Java samples make two assumptions:

- Samples are written to work with Java version 1.6 or higher.
- Samples are run using Eclipse with the AWS Toolkit for Eclipse.

HLS Samples

Topics

- Java HLS Sample Download (p. 101)
- Sample Code Pieces (p. 101)
- Tasks (p. 101)

This sample shows you how to create an HLS job and an HLS playlist file that can be used to play an adaptive bit rate stream.

Java HLS Sample Download

You can download the sample code here.

Sample Code Pieces

The Java code sample includes:

- The class `HlsJobCreationSample`

Tasks

To run the sample, follow these steps:

1. Set up your environment according to these instructions: Setting up your Java environment (p. 88)
2. Download the sample code.
3. In Eclipse, open `com/amazonaws/services/elastictranscoder/samples/HlsJobCreationSample.java`.
4. Replace PIPELINE_ID and INPUT_KEY with the appropriate values.
5. Run the samples in Eclipse.

Notification Samples

Topics

- Example of Notification Syntax (p. 102)
- Java Notification Sample Download (p. 102)
- Sample Code Pieces (p. 102)
- Tasks (p. 103)

Example of Notification Syntax

```
{
   "Type" : "Notification",
   "MessageId" : "341527b6-9081-5f3d-b933-6c8472c3be40",
   "TopicArn" : "arn:aws:sns:us-east-1:123456789012:ets-sample-topic",
   "Subject" : "Amazon Elastic Transcoder has scheduled job 1387847681009
      -abcdef for transcoding.",
   "Message" : "{\n  \"state\" : \"PROGRESSING\",\n
      \"version\" : \"2012-09-25\",\n  \"jobId\" : \"1387847681009-abcdef\",
      \n \"pipelineId\" : \"1387847629327-fedcba\",\n  \"input\" : {\n
      \"key\" : \"input/example/key.mp4\",\n \"frameRate\" : \"auto\",\n
      \"resolution\" : \"auto\",\n  \"aspectRatio\" : \"auto\",\n
      \"interlaced\" : \"auto\",\n  \"container\" : \"auto\"\n  },\n
      \"outputKeyPrefix\" : \"elastic-transcoder-samples/\",\n
      \"outputs\" : [ {\n \"id\" : \"1\",\n \"presetId\" :
      \"1351620000001-000020\",\n \"key\" : \"output/example/key.mp4\",\n
      \"thumbnailPattern\" : \"\",\n \"rotate\" : \"auto\",\n \"status\" :
      \"Progressing\"\n  } ]\n}",
   "Timestamp" : "2013-12-24T01:14:42.493Z",
   "SignatureVersion" : "1",
   "Signature" : "ElSqJW3ZksCPViY
GTayI/p+LjdF2HB42iJlIJRJ+jWzWwygXdiJXvZXl94qhd/tLln1lxPqijjivb5RWu7n5yzdZwbod6lpL
wyZ2TfWM6dZt57OzsG3GbdTxgqwVsztVSyWCYhcV8f+CrT3IQrfrU3Me/SRYVUqr
SUXXsu4Ls7A2q9mosG7v0Sn+3Z1rAa9+Rf6QmkfAg05UX0sLyA+I2hFqTu5oAGDK4Cm6FHuIwV+oYJXN
BbGWuS7tD6mRNwrYvPBlUvBLXx9m3CbqSXv5Yoj39vglv+1djtaLA3GpwX+B1hHx8QT373lgmmsmG
DRWhSQretpOTWDYb81PV2K0bg==",
   "SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotification
Service-e372f8ca30337fdb084e8ac449342c77.pem",
   "UnsubscribeURL" : "https://sns.us-east-1.amazonaws.com/?Action=
      Unsubscribe&SubscriptionArn=arn:aws:sns:us-east-1:123456789012:ets-
      sample-topic:b3ec47e5-e1f0-401f-a0a5-98c7fe405c2b"
 }
```

Java Notification Sample Download

You can download the sample code here.

Sample Code Pieces

The Java code sample includes:

- The class `Notification`: To consume notifications from Java, we use the Jackson JSON library to deserialize job status notifications into POJOs. The `Notification` class models the message we get back from the Amazon SQS queue.
- The class `JobStatusNotification`: The `JobStatusNotification` class models an Elastic Transcoder notification.
- The class `SqsQueueNotificationWorker`: The `SqsQueueNotificationWorker` class can be started in a separate thread to poll Amazon SQS and handle job status notifications. This class receives messages, calls all registered handlers for each notification received, and deletes the message from the queue.
- The interface `JobStatusNotificationHandler`: The `JobStatusNotificationHandler` interface can be implemented to allow custom handling for notifications.

- The class `JobStatusNotificationsSample`: The `JobStatusNotificationsSample` class creates a job and waits for it to complete. When the job goes into a terminal state, the queue worker shuts down and the application exits.

Tasks

To run the sample, follow these steps:

1. Set up your environment according to these instructions: Setting up your Java environment (p. 88).
2. Download the sample code.
3. Unzip the sample code into your JAVA project's source directory.
4. Open `com/amazonaws/services/elastictranscoder/samples/JobStatusNotificationsSample.java` in Eclipse.
5. Replace PIPELINE_ID, SQS_QUEUE_URL, and INPUT_KEY with the appropriate values.
6. Run the sample in Eclipse.

Ruby Samples

All Ruby samples make one major assumption:

- Samples are written to be compatible with Ruby version 1.9.

HLS Samples

Topics

- Ruby HLS Sample Download (p. 104)
- Sample Code Pieces (p. 104)
- Tasks (p. 104)

This sample shows you how to create an HLS job and an HLS playlist file that can be used to play an adaptive bit rate stream.

Ruby HLS Sample Download

You can download the sample code here.

Sample Code Pieces

The Ruby sample code includes:

- The `HlsJobCreationSample.rb` file

Tasks

To run the sample, follow these steps:

1. Set up your environment according to these instructions: Setting up your Ruby environment (p. 92)
2. Download the sample code.
3. Unzip the sample code into your Ruby project's source directory.
4. Edit `HlsJobCreationSample.rb` and replace pipeline_id and input_key with the appropriate values.
5. From a terminal, navigate to the directory where you unzipped the sample code and run:

```
$ruby HlsJobCreationSample.rb
```

Notification Samples

Topics

- Example of Notification Syntax (p. 105)
- Ruby Notification Sample Download (p. 105)
- Sample Code Pieces (p. 105)
- Tasks (p. 106)

Example of Notification Syntax

```
{
   "Type" : "Notification",
   "MessageId" : "341527b6-9081-5f3d-b933-6c8472c3be40",
   "TopicArn" : "arn:aws:sns:us-east-1:123456789012:ets-sample-topic",
   "Subject" : "Amazon Elastic Transcoder has scheduled job 1387847681009
      -abcdef for transcoding.",
   "Message" : "{\n  \"state\" : \"PROGRESSING\",\n
      \"version\" : \"2012-09-25\",\n  \"jobId\" : \"1387847681009-abcdef\",
      \n \"pipelineId\" : \"1387847629327-fedcba\",\n  \"input\" : {\n
      \"key\" : \"input/example/key.mp4\",\n \"frameRate\" : \"auto\",\n
      \"resolution\" : \"auto\",\n  \"aspectRatio\" : \"auto\",\n
      \"interlaced\" : \"auto\",\n  \"container\" : \"auto\"\n  },\n
      \"outputKeyPrefix\" : \"elastic-transcoder-samples/\",\n
      \"outputs\" : [ {\n \"id\" : \"1\",\n \"presetId\" :
      \"1351620000001-000020\",\n \"key\" : \"output/example/key.mp4\",\n
      \"thumbnailPattern\" : \"\",\n \"rotate\" : \"auto\",\n \"status\" :
      \"Progressing\"\n  } ]\n}",
   "Timestamp" : "2013-12-24T01:14:42.493Z",
   "SignatureVersion" : "1",
   "Signature" : "ElSqJW3ZksCPViY
GTayI/p+LjdF2HB42iJlIJRJ+jWzWwygXdiJXvZXl94qhd/tLln1lxPqijjivb5RWu7n5yzdZwbod6lpL
wyZ2TfWM6dZt57OzsG3GbdTxgqwVsztVSyWCYhcV8f+CrT3IQrfrU3Me/SRYVUqr
SUXXsu4Ls7A2q9mosG7v0Sn+3Z1rAa9+Rf6QmkfAg05UX0sLyA+I2hFqTu5oAGDK4Cm6FHuIwV+oYJXN
BbGWuS7tD6mRNwrYvPBlUvBLXx9m3CbqSXv5Yoj39vglv+1djtaLA3GpwX+B1hHx8QT373lgmmsmG
DRWhSQretpOTWDYb81PV2K0bg==",
   "SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotification
Service-e372f8ca30337fdb084e8ac449342c77.pem",
   "UnsubscribeURL" : "https://sns.us-east-1.amazonaws.com/?Action=
      Unsubscribe&SubscriptionArn=arn:aws:sns:us-east-1:123456789012:ets-
      sample-topic:b3ec47e5-e1f0-401f-a0a5-98c7fe405c2b"
 }
```

Ruby Notification Sample Download

You can download the sample code here.

Sample Code Pieces

The Ruby sample code includes:

- The class `SqsQueueNotificationWorker`: The `SqsQueueNotificationWorker` class polls Amazon SQS for notifications, calls all registered handlers for each notification, and deletes the message from the queue. Note that the worker runs in a separate thread. If the Ruby implementation used has "green" threads, then only a single thread runs at a time (no true multiprocessing).
- The script `JobStatusNotificationsSample.rb`: This script creates an Elastic Transcoder job, starts an Amazon SQS worker and waits for the job to complete. The provided handler tells the Amazon SQS worker to stop when the created job's processing is complete. Handling notifications is done by calling the add_handlers method and providing a Proc or lambda as a handler that takes a single argument. For each notification received, all registered handlers are called with the notification provided as the input argument to the handler.

Tasks

To run the sample, follow these steps:

1. Set up your environment according to these instructions: Setting up your Ruby environment (p. 92)
2. Download the sample code.
3. Unzip the sample code into your Ruby project's source directory.
4. Edit `JobStatusNotificationsSample.rb` and replace pipeline_id, sqs_queue_url, and input_key with the appropriate values.
5. From a terminal, navigate to the directory where you unzipped the sample code and run:

```
$ruby JobStatusNotificationsSample.rb
```

PHP Samples

All PHP samples make three major assumptions:

- PHP samples are run inside an Apache server.
- Apache server must be able to take input from the Internet to receive Amazon SNS notifications.
- Samples are written to work with PHP version 5.4 or higher.

HLS Samples

Topics

- PHP HLS Sample Download (p. 107)
- Sample Code Pieces (p. 107)
- Tasks (p. 107)

This sample shows you how to create an HLS job and an HLS playlist file that can be used to play an adaptive bit rate stream.

PHP HLS Sample Download

You can download the sample code here.

Sample Code Pieces

The PHP sample code includes:

- The `HlsJobCreationSample.php` file

Tasks

To run the sample, follow these steps:

1. Set up your environment according to these instructions: Setting up your PHP environment (p. 95)
2. Download the sample code.
3. In your browser, load the page http://<your-endpoint>/transcoder-samples/HlsJobCreationSample.php.
4. Fill in the pipeline ID and input key and submit the form to create a job.

Notification Samples

Topics

- Example of Notification Syntax (p. 108)
- PHP Notification Sample Download (p. 108)
- Sample Code Pieces (p. 108)
- Tasks (p. 108)

Example of Notification Syntax

```
{
   "Type" : "Notification",
   "MessageId" : "341527b6-9081-5f3d-b933-6c8472c3be40",
   "TopicArn" : "arn:aws:sns:us-east-1:123456789012:ets-sample-topic",
   "Subject" : "Amazon Elastic Transcoder has scheduled job 1387847681009
      -abcdef for transcoding.",
   "Message" : "{\n  \"state\" : \"PROGRESSING\",\n
      \"version\" : \"2012-09-25\",\n  \"jobId\" : \"1387847681009-abcdef\",
      \n \"pipelineId\" : \"1387847629327-fedcba\",\n  \"input\" : {\n
      \"key\" : \"input/example/key.mp4\",\n \"frameRate\" : \"auto\",\n
      \"resolution\" : \"auto\",\n  \"aspectRatio\" : \"auto\",\n
      \"interlaced\" : \"auto\",\n  \"container\" : \"auto\"\n  },\n
      \"outputKeyPrefix\" : \"elastic-transcoder-samples/\",\n
      \"outputs\" : [ {\n \"id\" : \"1\",\n \"presetId\" :
      \"1351620000001-000020\",\n \"key\" : \"output/example/key.mp4\",\n
      \"thumbnailPattern\" : \"\",\n \"rotate\" : \"auto\",\n \"status\" :
      \"Progressing\"\n  } ]\n}",
   "Timestamp" : "2013-12-24T01:14:42.493Z",
   "SignatureVersion" : "1",
   "Signature" : "ElSqJW3ZksCPViY
GTayI/p+LjdF2HB42iJlIJRJ+jWzWwygXdiJXvZXl94qhd/tLln1lxPqijjivb5RWu7n5yzdZwbod6lpL
wyZ2TfWM6dZt57OzsG3GbdTxgqwVsztVSyWCYhcV8f+CrT3IQrfrU3Me/SRYVUqr
SUXXsu4Ls7A2q9mosG7v0Sn+3Z1rAa9+Rf6QmkfAg05UX0sLyA+I2hFqTu5oAGDK4Cm6FHuIwV+oYJXN
BbGWuS7tD6mRNwrYvPBlUvBLXx9m3CbqSXv5Yoj39vglv+1djtaLA3GpwX+B1hHx8QT373lgmmsmG
DRWhSQretpOTWDYb81PV2K0bg==",
   "SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotification
Service-e372f8ca30337fdb084e8ac449342c77.pem",
   "UnsubscribeURL" : "https://sns.us-east-1.amazonaws.com/?Action=
      Unsubscribe&SubscriptionArn=arn:aws:sns:us-east-1:123456789012:ets-
      sample-topic:b3ec47e5-e1f0-401f-a0a5-98c7fe405c2b"
 }
```

PHP Notification Sample Download

You can download the sample code here.

Sample Code Pieces

The PHP sample code includes:

- The file `JobStatusNotificationsSampleNotificationHandler.php`: When a notification is POSTed to your server, the notification handler writes the status back to a file under /tmp/<job-id>.
- The file `JobStatusNotificationsSample.php`: Once the notification handler writes the status back to a file under /tmp/<job-id>, `JobStatusNotificationsSample.php` loads the /tmp/<job-id> status file specified by the Id in the query string given to it.

Tasks

To run the sample, follow these steps:

1. Set up your environment according to these instructions: Setting up your PHP environment (p. 95)
2. Download the sample code.

3. Unzip the sample code into your PHP project's source directory.
4. In your browser, load the page http://<your-endpoint>/transcoder-samples/JobStatusNotificationsSample.php.
5. Fill in the pipeline ID and input key you wish to transcode and press the **Create Job** button.

Python Samples

All Python samples make one major assumption:

- Samples are written to be compatible with Python version 2.7.

HLS Samples

Topics

- Python HLS Sample Download (p. 110)
- Sample Code Pieces (p. 110)
- Tasks (p. 110)

This sample shows you how to create an HLS job and an HLS playlist file that can be used to play an adaptive bit rate stream.

Python HLS Sample Download

You can download the sample code here.

Sample Code Pieces

The Python sample code includes:

- The `HlsJobCreationSample.py` file

Tasks

To run the sample, follow these steps:

1. Set up your environment according to these instructions: Setting up your Python environment (p. 98)
2. Download the sample code.
3. Unzip the sample code.
4. Edit `HlsJobCreationSample.py` and replace pipeline_id and input_key with the appropriate values.
5. From a terminal, navigate to the directory where you unzipped the sample code and run:

```
$python HlsJobCreationSample.py
```

Notification Samples

Topics

- Example of Notification Syntax (p. 111)
- Python Notification Sample Download (p. 111)
- Sample Code Pieces (p. 111)
- Tasks (p. 112)

Example of Notification Syntax

```
{
   "Type" : "Notification",
   "MessageId" : "341527b6-9081-5f3d-b933-6c8472c3be40",
   "TopicArn" : "arn:aws:sns:us-east-1:123456789012:ets-sample-topic",
   "Subject" : "Amazon Elastic Transcoder has scheduled job 1387847681009
      -abcdef for transcoding.",
   "Message" : "{\n  \"state\" : \"PROGRESSING\",\n
      \"version\" : \"2012-09-25\",\n  \"jobId\" : \"1387847681009-abcdef\",
      \n \"pipelineId\" : \"1387847629327-fedcba\",\n  \"input\" : {\n
      \"key\" : \"input/example/key.mp4\",\n \"frameRate\" : \"auto\",\n
      \"resolution\" : \"auto\",\n  \"aspectRatio\" : \"auto\",\n
      \"interlaced\" : \"auto\",\n  \"container\" : \"auto\"\n  },\n
      \"outputKeyPrefix\" : \"elastic-transcoder-samples/\",\n
      \"outputs\" : [ {\n \"id\" : \"1\",\n \"presetId\" :
      \"1351620000001-000020\",\n \"key\" : \"output/example/key.mp4\",\n
      \"thumbnailPattern\" : \"\",\n \"rotate\" : \"auto\",\n \"status\" :
      \"Progressing\"\n  } ]\n}",
   "Timestamp" : "2013-12-24T01:14:42.493Z",
   "SignatureVersion" : "1",
   "Signature" : "ElSqJW3ZksCPViY
GTayI/p+LjdF2HB42iJlIJRJ+jWzWwygXdiJXvZXl94qhd/tLln1lxPqijjivb5RWu7n5yzdZwbod6lpL
wyZ2TfWM6dZt57OzsG3GbdTxgqwVsztVSyWCYhcV8f+CrT3IQrfrU3Me/SRYVUqr
SUXXsu4Ls7A2q9mosG7v0Sn+3Z1rAa9+Rf6QmkfAg05UX0sLyA+I2hFqTu5oAGDK4Cm6FHuIwV+oYJXN
BbGWuS7tD6mRNwrYvPBlUvBLXx9m3CbqSXv5Yoj39vglv+1djtaLA3GpwX+B1hHx8QT373lgmmsmG
DRWhSQretpOTWDYb81PV2K0bg==",
   "SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotification
Service-e372f8ca30337fdb084e8ac449342c77.pem",
   "UnsubscribeURL" : "https://sns.us-east-1.amazonaws.com/?Action=
      Unsubscribe&SubscriptionArn=arn:aws:sns:us-east-1:123456789012:ets-
      sample-topic:b3ec47e5-e1f0-401f-a0a5-98c7fe405c2b"
 }
```

Python Notification Sample Download

You can download the sample code here.

Sample Code Pieces

The Python sample code includes:

- The class `SqsQueueNotificationWorker.py`: The `SqsQueueNotificationWorker` class polls Amazon SQS and handles notifications in a separate process. This allows for true multiprocessing in Python. This class receives messages, calls all registered handlers for each notification received, and deletes the message from the queue. This class also includes the definition for the `JobStatusNotificationHandler` class, which is provided to give a framework for handling Elastic Transcoder notifications. This class can be extended and the handle method overwritten to provide custom job handling. Note that this does not follow Pythons standards of duck typing, but does give a formal definition of handlers for the purposes of this sample.
- The script `JobStatusNotificationSample.py`: The `JobStatusNotificationSample.py` script creates a job in Elastic Transcoder and waits for it to complete. When the job completes, it stops the queue polling process and exits. Cross-process synchronization using the multiprocessing value object is necessary, because the handler runs in a separate process from the queue it polls.

Tasks

To run the sample, follow these steps:

1. Set up your environment according to these instructions: Setting up your Python environment (p. 98)
2. Download the sample code.
3. Unzip the sample code to a local folder.
4. Edit `JobStatusNotificationsSample.py` and replace pipeline_id, sqs_queue_url, and input_key with the appropriate values.
5. From a terminal, navigate to the directory where you unzipped the sample code and run:

```
$python JobStatusNotificationsSample.py
```

API Reference

If you are using a language for which an AWS SDK exists, use the SDK rather than trying to work your way through the APIs. The SDKs make authentication simpler, integrate easily with your development environment, and provide easy access to Elastic Transcoder commands. For more information about the AWS SDKs, including how to set up your environment, links to the SDK documentation, and sample code, see AWS SDK Sample Code (p. 86).

Topics

- Using the AWS SDKs (p. 113)
- Making HTTP Requests to Elastic Transcoder (p. 114)
- Signing Requests (p. 117)
- Handling Errors in Elastic Transcoder (p. 118)
- Pipeline Operations (p. 127)
- Job Operations (p. 180)
- Preset Operations (p. 274)

Using the AWS SDKs

If you are using a language for which an AWS SDK exists, use the SDK rather than trying to work your way through the APIs. The SDKs make authentication simpler, integrate easily with your development environment, and provide easy access to Elastic Transcoder commands. For more information about the AWS SDKs, including how to set up your environment, links to the SDK documentation, and sample code, see AWS SDK Sample Code (p. 86)

Making HTTP Requests to Elastic Transcoder

Elastic Transcoder REST requests are HTTPS requests as defined in RFC 2616. (For more information, go to http://www.ietf.org/rfc/rfc2616.txt.) This section describes the structure of an Elastic Transcoder REST request. For detailed descriptions of the actions that you can perform, see Pipeline Operations (p. 127), Job Operations (p. 180), and Preset Operations (p. 274).

A typical REST action consists of sending an HTTPS request to Elastic Transcoder and waiting for the response. Like any HTTP request, a REST request to Elastic Transcoder contains a request method, a URI, request headers, and sometimes a query string or request body. The response contains an HTTP status code, response headers, and sometimes a response body.

Topics

- HTTP Header Contents (p. 114)
- HTTP Request Body (p. 115)
- HTTP Responses (p. 116)

HTTP Header Contents

Elastic Transcoder requires the following information in the header of an HTTP request:

Host (Required)
The Elastic Transcoder endpoint that specifies where your resources are created. The value must be a named regional endpoint. We recommend that you use the same endpoint for your Amazon S3 buckets and for your Elastic Transcoder pipelines and jobs. If you use different endpoints, you'll incur additional charges for data transferred between the region that contains your Amazon S3 buckets and the region in which Elastic Transcoder does the encoding. In addition, the time required for the data transfer will delay access to the transcoded file.

For a list of supported Elastic Transcoder endpoints, go to the Regions and Endpoints section in the *Amazon Web Services General Reference*.

For more information on cross-regional fees, see Data Transfer Pricing in Amazon S3 Pricing.

x-amz-date or Date (Required)
The date used to create the signature contained in the `Authorization` header. Specify the date in ISO 8601 standard format, in UTC time, as in the following example: `X-Amz-Date: 20130613T203622Z`.

You must include either `x-amz-date` or `Date`. (Some HTTP client libraries don't let you set the `Date` header). When an `x-amz-date` header is present, the system ignores any `Date` header when authenticating the request.

The time stamp must be within 15 minutes of the AWS system time when the request is received. If it isn't, the request fails with the `RequestExpired` error code to prevent someone else from replaying your requests.

Authorization (Required)
The information required for request authentication. For more information about constructing this header, see Signing Requests (p. 117).

Content-Type (Conditional)
Specifies JSON and the version, for example, `Content-Type: application/x-amz-json-1.0`.

Condition: Required for POST requests.

Content-Length (Conditional)
Length of the message (without the headers) according to RFC 2616.

Condition: Required if the request body itself contains information (most toolkits add this header automatically).

The following is an example header for an HTTP request to create a pipeline.

```
POST /2012-09-25/pipelines HTTP/1.1
host: elastictranscoder.us-east-1.amazonaws.com:443
x-amz-date: 20120116T174952Z
Authorization: AWS4-HMAC-SHA256 Credential=AccessKeyID/20120116/us-east-
1/ets/aws4_request,SignedHeaders=host;x-amz-date;x-amz-target,Signa
ture=145b1567ab3c50d929412f28f52c45dbf1e63ec5c66023d232a539a4afd11fd9
content-type: application/x-amz-json-1.0
content-length: 231
connection: Keep-Alive
```

HTTP Request Body

Many Elastic Transcoder API actions require you to include JSON-formatted data in the body of the request. The JSON conforms to the Elastic Transcoder schema.

Note
JSON values in the request body are strings.

Example Request

The following example request uses a simple JSON statement to create a job that transcodes a file named `sample.mp4` and saves it as `sams-birthday.mp4`.

```
POST /2012-09-25/jobs HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.us-east-1.amazonaws.com:443
Content-Length: 300

{
   "Input":{
      "Key":"sample.mp4",
      "FrameRate":"auto",
      "Resolution":"auto",
      "AspectRatio":"auto",
      "Interlaced":"auto",
      "Container":"mp4"
   },
   "OutputKeyPrefix":"family-videos/",
   "Outputs":[
      {
         "Key":"sams-birthday.mp4",
         "ThumbnailPattern":"thumbnails/sams-birthday-{count}",
         "Rotate":"0",
         "PresetId":"1351620000000-100080"
      }
   ],
   "PipelineId":"1111111111111-abcde1"
}
```

HTTP Responses

All Elastic Transcoder API actions include JSON-formatted data in the response. The JSON conforms to the Elastic Transcoder schema.

Note
JSON values in the response are strings.

Here are some important headers in the HTTP response and how you should handle them in your application, if applicable:

HTTP/1.1
This header is followed by a status code. Status code `200` indicates a successful operation. For information about error codes, see API Error Codes (Client and Server Errors) (p. 118).

Type: String

x-amzn-RequestId
A value created by Elastic Transcoder that uniquely identifies your request, for example, K2QH8DNOU907N97FNA2GDLL8OBVV4KQNSO5AEMVJF66Q9ASUAAJG. If you have a problem with Elastic Transcoder, AWS can use this value to troubleshoot the problem. We recommend that you log these values.

Type: String

Content-Length
The length of the response body in bytes.

Type: String

Date
The date and time that Elastic Transcoder responded, for example, `Sun, 25 Mar 2012 12:00:00 GMT`. The format of the date must be one of the full date formats specified by RFC 2616, section 3.3.

Type: String

Signing Requests

If you're using a language for which AWS provides an SDK, we recommend that you use the SDK. All of the AWS SDKs greatly simplify the process of signing requests and save you a significant amount of time when compared with using the Elastic Transcoder API. In addition, the SDKs integrate easily with your development environment and provide easy access to related commands.

Elastic Transcoder requires that you authenticate every request you send by signing the request. To sign a request, you calculate a digital signature using a cryptographic hash function, which returns a hash value based on the input. The input includes the text of your request and your secret access key. The hash function returns a hash value that you include in the request as your signature. The signature is part of the `Authorization` header of your request.

After receiving your request, Elastic Transcoder recalculates the signature using the same hash function and input that you used to sign the request. If the resulting signature matches the signature in the request, Elastic Transcoder processes the request. Otherwise, the request is rejected.

Elastic Transcoder supports authentication using AWS Signature Version 4. The process for calculating a signature can be broken into three tasks:

- Task 1: Create a Canonical Request

  Create your HTTP request in canonical format as described in Task 1: Create a Canonical Request For Signature Version 4 in the *Amazon Web Services General Reference*.
- Task 2: Create a String to Sign

  Create a string that you will use as one of the input values to your cryptographic hash function. The string, called the *string to sign*, is a concatenation of the name of the hash algorithm, the request date, a *credential scope* string, and the canonicalized request from the previous task. The *credential scope* string itself is a concatenation of date, region, and service information.

  For the `X-Amz-Credential` parameter, specify:
  - The code for the endpoint to which you're sending the request, for example, `us-east-1`. For a list of regions and endpoints for Elastic Transcoder, see the Regions and Endpoints chapter of the *Amazon Web Services General Reference*. When specifying the code for the endpoint, include only the part between `elastictranscoder.` and `.amazonaws.com`
  - `elastictranscoder` for the service abbreviation

  For example:

  ```
  X-Amz-Credential=AKIAIOSFODNN7EXAMPLE/20130501/us-east-1/elastictranscoder/aws4_request
  ```
- Task 3: Create a Signature

  Create a signature for your request by using a cryptographic hash function that accepts two input strings: your *string to sign* and a *derived key*. The *derived key* is calculated by starting with your secret access key and using the *credential scope* string to create a series of hash-based message authentication codes (HMACs).

Handling Errors in Elastic Transcoder

Topics

- API Error Codes (Client and Server Errors) (p. 118)
- Errors During Job Processing (p. 121)
- Catching Errors (p. 124)
- Error Retries and Exponential Backoff (p. 125)

When you send requests to and get responses from the Elastic Transcoder API, you might encounter two types of API errors:

- **Client errors:** Client errors are indicated by a 4xx HTTP response code. Client errors indicate that Elastic Transcoder found a problem with the client request, such as an authentication failure or missing required parameters. Fix the issue in the client application before submitting the request again.
- **Server errors:** Server errors are indicated by a 5xx HTTP response code, and need to be resolved by Amazon. You can resubmit/retry the request until it succeeds.

For each API error, Elastic Transcoder returns the following values:

- A status code, for example, `400`
- An error code, for example, `ValidationException`
- An error message, for example, `Supplied AttributeValue is empty, must contain exactly one of the supported datatypes`

For a list of error codes that Elastic Transcoder returns for client and server errors, see API Error Codes (Client and Server Errors) (p. 118).

In addition, you might encounter errors while Elastic Transcoder is processing your job. For more information, see Errors During Job Processing (p. 121).

API Error Codes (Client and Server Errors)

HTTP status codes indicate whether an operation is successful or not.

A response code of `200` indicates the operation was successful. Other error codes indicate either a client error (4xx) or a server error (5xx).

The following table lists the errors returned by Elastic Transcoder. Some errors are resolved if you simply retry the same request. The table indicates which errors are likely to be resolved with successive retries. If the value of the Retry column is:

- **Yes:** Submit the same request again.
- **No:** Fix the problem on the client side before submitting a new request.

For more information about retrying requests, see Error Retries and Exponential Backoff (p. 125).

HTTP Status Code	Error code	Message	Cause	Retry
400	Conditional Check Failed Exception	The conditional request failed.	Example: The expected value did not match what was stored in the system.	No
400	Incomplete Signature Exception	The request signature does not conform to AWS standards.	The signature in the request did not include all of the required components. See HTTP Header Contents (p. 114).	No
400	Missing Authentication Token Exception	The request must contain a valid (registered) AWS Access Key ID.	The request did not include the required `x-amz-security-token`. See Making HTTP Requests to Elastic Transcoder (p. 114).	No
400	Validation Exception	Various.	One or more values in a request were missing or invalid; for example, a value was empty or was greater than the maximum valid value.	No
403	AccessDenied Exception	• Deleting a system preset is not allowed: account=<accountId>, presetId=<presetId>. • General authentication failure. The client did not correctly sign the request. See Signing Requests (p. 117).	You attempted to delete a system preset, the signature in a call to the Elastic Transcoder API was invalid, or an IAM user is not authorized to perform the operation.	No
404	ResourceNot Found Exception	• The specified <resource> could not be found: <resourceId>. • The specified job was not found: account=<accountId>, jobId=<jobId>. • The specified pipeline was not found: account=<accountId>, pipelineId=<pipelineId> • The specified preset was not found: account=<accountId>, presetId=<presetId>	Example: The pipeline to which you're trying to add a job doesn't exist or is still being created.	No
409	Resource InUse Exception	• The <resource> was already in use: accountId=<accountId>, resourceId=<resourceId>. • The pipeline contains active jobs: account=<accountId>, pipeline=<pipelineId>.	Example: You attempted to delete a pipeline that is currently in use.	No

HTTP Status Code	Error code	Message	Cause	Retry
429	Limit Exceeded Exception	• The account already has the maximum number of pipelines allowed: account=<accountId>, maximum number of pipelines=<maximum> • The account already has the maximum number of presets allowed: account=<accountId>, maximum number of presets=<maximum> • The account already has the maximum number of jobs per pipeline in the backlog: account=<accountId>, maximum number of jobs in backlog for pipeline=<maximum>	The current AWS account has exceeded limits on Elastic Transcoder objects. For more information, see Limits on the Number of Elastic Transcoder Pipelines, Jobs, and Presets (p. 3).	
429	Provisioned Throughput Exceeded Exception	You exceeded your maximum allowed provisioned throughput.	Example: Your request rate is too high. The AWS SDKs for Elastic Transcoder automatically retry requests that receive this exception. Your request is eventually successful unless your retry queue is too large to finish. Reduce the frequency of requests. For more information, see Error Retries and Exponential Backoff (p. 125). If you're polling to determine the status of a request, consider using notifications to determine status. For more information, see Notifications of Job Status (p. 66).	Yes
429	Throttling Exception	Rate of requests exceeds the allowed throughput.	You are submitting requests too rapidly; for example, requests to create new jobs. If you're polling to determine the status of a request, consider using notifications to determine status. For more information, see Notifications of Job Status (p. 66).	Yes
500	Internal Failure	The server encountered an internal error trying to fulfill the request.	The server encountered an error while processing your request.	Yes
500	Internal Server Error	The server encountered an internal error trying to fulfill the request.	The server encountered an error while processing your request.	Yes

HTTP Status Code	Error code	Message	Cause	Retry
500	Internal Service Exception		The service encountered an unexpected exception while trying to fulfill the request.	Yes
500	Service Unavailable Exception	The service is currently unavailable or busy.	There was an unexpected error on the server while processing your request.	Yes

Sample Error Response

The following is an HTTP response indicating that the value for `inputBucket` was null, which is not a valid value.

```
HTTP/1.1 400 Bad Request
x-amzn-RequestId: b0e91dc8-3807-11e2-83c6-5912bf8ad066
x-amzn-ErrorType: ValidationException
Content-Type: application/json
Content-Length: 124
Date: Mon, 26 Nov 2012 20:27:25 GMT

{"message":"1 validation error detected: Value null at 'inputBucket' failed to
 satisfy constraint: Member must not be null"}
```

Errors During Job Processing

When Elastic Transcoder encounters an error while processing your job, it reports the error in two ways:

- **Job Status and Output Status:** Elastic Transcoder sets the `Job:Status` object and the `Outputs:Status` object for the failed output to `Error`. In addition, Elastic Transcoder sets the `Outputs:StatusDetail` JSON object for the failed output to a value that explains the failure.
- **SNS Notification:** If you configured the pipeline to send an SNS notification when Elastic Transcoder encounters an error during processing, Elastic Transcoder includes a JSON object in the notification in the following format:

```
{
   "state" : "PROGRESSING|COMPLETED|WARNING|ERROR",
   "errorCode" : "the code of any error that occurred",
   "messageDetails" : "the notification message you created in Amazon SNS",
   "version" : "API version that you used to create the job",
   "jobId" : "value of Job:Id object that Elastic Transcoder
              returns in the response to a Create Job request",
   "pipelineId" : "value of PipelineId object
                   in the Create Job request",
   "input" : {
      job Input settings
   },
   "outputKeyPrefix" : "prefix for file names in Amazon S3 bucket",
   "outputs": [
      {
         applicable job Outputs settings,
```

```
         "status" : "Progressing|Completed|Warning|Error"
      },
      {...}
   ],
   "playlists": [
      {
         applicable job playlists settings
      }
   ],
   "userMetadata": {
      "metadata key": "metadata value"
   }
}
```

Value of `errorCode`	Value of `messageDetails`	Cause
1000	Validation Error	While processing the job, Elastic Transcoder determined that one or more values in the request were invalid.
1001	Dependency Error	Elastic Transcoder could not generate the playlist because it encountered an error with one or more of the playlist's dependencies.
2000	Cannot Assume Role	Elastic Transcoder cannot assume the AWS Identity and Access Management role that is specified in the `Role` object in the pipeline for this job.
3000	Unclassified Storage Error	
3001	Input Does Not Exist	No file exists with the name that you specified in the `Input:Key` object for this job. The file must exist in the Amazon S3 bucket that is specified in the `InputBucket` object in the pipeline for this job.
3002	Output Already Exists	A file already exists with the name that you specified in the `Outputs:Key` (or `Output:Key`) object for this job. The file cannot exist in the Amazon S3 bucket that is specified in the `OutputBucket` object in the pipeline for this job.
3003	Does Not Have Read Permission	The IAM role specified in the `Role` object in the pipeline that you used for this job doesn't have permission to read from the Amazon S3 bucket that contains the file you want to transcode.
3004	Does Not Have Write Permission	The IAM role specified in the `Role` object in the pipeline that you used for this job doesn't have permission to write to the Amazon S3 bucket in which you want to save either transcoded files or thumbnail files.
3005	Bucket Does Not Exist	The specified S3 bucket does not exist: bucket={1}.
3006	Does Not Have Write Permission	Elastic Transcoder was unable to write the key={1} to bucket={2}, as the key is not in the same region as the bucket

Value of `errorCode`	Value of `messageDetails`	Cause
4000	Bad Input File	The file that you specified in the `Input:Key` object for this job is in a format that is currently not supported by Elastic Transcoder.
4001	Bad Input File	The width x height of the file that you specified in the `Input:Key` object for this job exceeds the maximum allowed width x height.
4002	Bad Input File	The file size of the file that you specified in the `Input:Key` object for this job exceeds the maximum allowed size.
4003	Bad Input File	Elastic Transcoder couldn't interpret the file that you specified in one of the `Outputs:Watermarks:InputKey` objects for this job.
4004	Bad Input File	The width x height of a file that you specified in one of the `Outputs:Watermarks:InputKey` objects for this job exceeds the maximum allowed width x height.
4005	Bad Input File	The size of a file that you specified for one of the {1} objects exceeds the maximum allowed size: bucket={2}, key={3}, size{4}, max size={5}.
4006	Bad Input File	Elastic Transcoder could not transcode the input file because the format is not supported.
4007	Unhandled Input File	Elastic Transcoder encountered a file type that is generally supported, but was unable to process the file correctly. This error automatically opened a support case, and we have started to research the cause of the problem.
4008	Bad Input File	The underlying cause of this is a mismatch between the preset and the input file. Examples include: • The preset includes audio settings, but the input file lacks audio. • The preset includes video settings, but the input file lacks video.
4009	Bad Input File	Elastic Transcoder was unable to insert all of your album art into the output file because you exceeded the maximum number of artwork streams.
4010	Bad Input File	Elastic Transcoder could not interpret the graphic file you specified for `AlbumArt:Artwork:InputKey`.
4011	Bad Input File	Elastic Transcoder detected an embedded artwork stream, but could not interpret it.
4012	Bad Input File	The image that you specified for `AlbumArt:Artwork` exceeds the maximum allowed width x height: 4096 x 3072.
4013	Bad Input File	The width x height of the embedded artwork exceeds the maximum allowed width x height: 4096 x 3072.

Value of `errorCode`	Value of `messageDetails`	Cause
4014	Bad Input	The value that you specified for starting time of a clip is after the end of the input file. Elastic Transcoder could not create an output file.
4015	Bad Input	Elastic Transcoder could not generate a manifest file because the generated segments did not match.
4016	Bad Input	Elastic Transcoder could not decrypt the input file from {1} using {2}.
4017	Bad Input	The AES key was encrypted with a {2}-bit encryption key. AES supports only 128-, 192-, and 256-bit encryption keys. MD5={1}.
4018	Bad Input	Elastic Transcoder was unable to decrypt the ciphered key with MD5={1}
4019	Bad Input	Elastic Transcoder was unable to generate a data key using the KMS key ARN {0}.
4020	Bad Input	Your key must be 128 bits for AES-128 encryption. MD5={1}, {2} bits.
4100	Bad Input File	Elastic Transcoder detected an embedded caption track but could not interpret it.
4101	Bad Input File	Elastic Transcoder could not interpret the specified caption file for Amazon S3 bucket={1}, key={2}.
4102	Bad Input File	Elastic Transcoder could not interpret the specified caption file since it was not UTF-8 encoded: Amazon S3 bucket={1}, key={2}.
4103	Bad Input File	Elastic Transcoder was unable to process all of your caption tracks because you exceeded {1}, the maximum number of caption tracks.
4104	Bad Input File	4104: Elastic Transcoder could not generate a master playlist because the desired output contains {1} embedded captions, when the maximum is 4.
9000	Internal Service Error	
9001	Internal Service Error	
9999	Internal Service Error	

Catching Errors

For your application to run smoothly, you need to build in logic to catch and respond to errors. One typical approach is to implement your request within a `try` block or `if-then` statement.

The AWS SDKs perform their own retries and error checking. If you encounter an error while using one of the AWS SDKs, you should see the error code and description. You should also see a `Request ID` value. The `Request ID` value can help troubleshoot problems with Elastic Transcoder support.

The following example uses the AWS SDK for Java to delete an item within a `try` block and uses a `catch` block to respond to the error. In this case, it warns the user that the request failed. The example uses the `AmazonServiceException` class to retrieve information about any operation errors, including the `Request ID`. The example also uses the `AmazonClientException` class in case the request is not successful for other reasons.

```
try {
   DeleteJobRequest request = new DeleteJobRequest(jobId);
   DeleteJobResult result = ET.deleteJob(request);
   System.out.println("Result: " + result);
   // Get error information from the service while trying to run the operation

   }  catch (AmazonServiceException ase) {
      System.err.println("Failed to delete job " + jobId);
      // Get specific error information
      System.out.println("Error Message:    " + ase.getMessage());
      System.out.println("HTTP Status Code: " + ase.getStatusCode());
      System.out.println("AWS Error Code:   " + ase.getErrorCode());
      System.out.println("Error Type:       " + ase.getErrorType());
      System.out.println("Request ID:       " + ase.getRequestId());
   // Get information in case the operation is not successful for other reasons

   }  catch (AmazonClientException ace) {
      System.out.println("Caught an AmazonClientException, which means"+
      " the client encountered " +
      "an internal error while trying to " +
      "communicate with Elastic Transcoder, " +
      "such as not being able to access the network.");
      System.out.println("Error Message: " + ace.getMessage());
   }
```

Error Retries and Exponential Backoff

Numerous components on a network, such as DNS servers, switches, load balancers, and others can generate errors anywhere in the life of a given request.

The usual technique for dealing with these error responses in a networked environment is to implement retries in the client application. This technique increases the reliability of the application and reduces operational costs for the developer.

Each AWS SDK supporting Elastic Transcoder implements automatic retry logic. The AWS SDK for Java automatically retries requests, and you can configure the retry settings using the `ClientConfiguration` class. For example, in some cases, such as a web page making a request with minimal latency and no retries, you might want to turn off the retry logic. Use the `ClientConfiguration` class and provide a `maxErrorRetry` value of `0` to turn off the retries.

If you're not using an AWS SDK, you should retry original requests that receive server errors (5xx). However, client errors (4xx, other than a `ThrottlingException` or a `ProvisionedThroughputExceededException`) indicate you need to revise the request itself to correct the problem before trying again.

Note

If you're polling to determine the status of a request, and if Elastic Transcoder is returning HTTP status code 429 with an error code of `Provisioned Throughput Exceeded Exception` or `Throttling Exception`, consider using notifications instead of polling to determine status. For more information, see Notifications of Job Status (p. 66).

In addition to simple retries, we recommend using an exponential backoff algorithm for better flow control. The idea behind exponential backoff is to use progressively longer waits between retries for consecutive error responses. For example, you might let one second elapse before the first retry, four seconds before the second retry, 16 seconds before the third retry, and so on. However, if the request has not succeeded after a minute, the problem might be a hard limit and not the request rate. For example, you may have reached the maximum number of pipelines allowed. Set the maximum number of retries to stop around one minute.

Following is a workflow showing retry logic. The workflow logic first determines if the error is a server error (5xx). Then, if the error is a server error, the code retries the original request.

```
currentRetry = 0
DO
  set retry to false

  execute Elastic Transcoder request

  IF Exception.errorCode = ProvisionedThroughputExceededException
    set retry to true
  ELSE IF Exception.httpStatusCode = 500
    set retry to true
  ELSE IF Exception.httpStatusCode = 400
    set retry to false
    fix client error (4xx)

  IF retry = true
    wait for (2^currentRetry * 50) milliseconds
    currentRetry = currentRetry + 1

WHILE (retry = true AND currentRetry < MaxNumberOfRetries)  // limit retries
```

Pipeline Operations

Topics

- Create Pipeline (p. 128)
- List Pipelines (p. 138)
- Read Pipeline (p. 147)
- Update Pipeline (p. 156)
- Update Pipeline Status (p. 167)
- Update Pipeline Notifications (p. 170)
- Delete Pipeline (p. 174)
- Test Role (p. 177)

Pipelines are queues that manage your transcoding jobs. When you create a job, you specify which pipeline you want to add the job to. Elastic Transcoder starts processing the jobs in a pipeline in the order in which you added them.

This section describes operations that you can perform on pipelines using the Elastic Transcoder API. For more information about pipelines, including how to perform the same operations using the Elastic Transcoder console, see Working with Pipelines (p. 9).

Create Pipeline

Topics

- Description (p. 128)
- Requests (p. 128)
- Responses (p. 133)
- Errors (p. 135)
- Examples (p. 135)

Description

To create a pipeline, send a POST request to the `/2012-09-25/pipelines/` resource.

Requests

Syntax

```
POST /2012-09-25/pipelines HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
   "Name":"pipeline name",
   "InputBucket":"Amazon S3 bucket that contains files to transcode
      and graphics to use as watermarks",
   "OutputBucket":"Use this, or use ContentConfig:Bucket plus
      ThumbnailConfig:Bucket",
   "Role":"IAM role ARN",
   "AwsKmsKeyArn":"AWS-KMS key arn of the AWS-KMS key you want to
      use with this pipeline",
   "Notifications":{
      "Progressing":"SNS topic to notify when
         Elastic Transcoder has started to process the job",
      "Completed":"SNS topic to notify when
         Elastic Transcoder has finished processing the job",
      "Warning":"SNS topic to notify when
         Elastic Transcoder encounters a warning condition"
      "Error":"SNS topic to notify when
         Elastic Transcoder encounters an error condition"
   },
   "ContentConfig":{
      "Bucket":"Use this plus ThumbnailConfig:Bucket,
         or use OutputBucket",
      "Permissions":[
         {
            "GranteeType":"Canonical|Email|Group",
            "Grantee":"AWS user ID or CloudFront origin access identity"|
```

```
                    "registered email address for AWS account|
                    AllUsers|AuthenticatedUsers|LogDelivery",
            "Access":[
               "Read|ReadAcp|WriteAcp|FullControl",
               ...
            ]
         },
         {...}
      ],
      "StorageClass":"Standard|ReducedRedundancy"
   },
   "ThumbnailConfig":{
      "Bucket":"Use this plus ContentConfig:Bucket,
         or use OutputBucket",
      "Permissions":[
         {
            "GranteeType":"Canonical|Email|Group",
            "Grantee":"AWS user ID or CloudFront origin access identity"|
                    "registered email address for AWS account|
                    AllUsers|AuthenticatedUsers|LogDelivery",
            "Access":[
               "Read|ReadAcp|WriteAcp|FullControl",
               ...
            ]
         },
         {...}
      ],
      "StorageClass":"Standard|ReducedRedundancy"
   }
}
```

Request Parameters

This operation does not use request parameters.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

The JSON string in the request body contains the following objects.

Name
The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

InputBucket
The Amazon S3 bucket in which you saved the media files that you want to transcode and the graphics that you want to use as watermarks.

OutputBucket (Use this, or use ContentConfig:Bucket plus ThumbnailConfig:Bucket)
The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.

- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.

  Note
  When Elastic Transcoder saves files in `OutputBucket`, it grants full control over the files only to the AWS account that owns the role that is specified by `Role`.
- You want to associate the transcoded files and thumbnails with the Amazon S3 `Standard` storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

See also: ContentConfig:Bucket, ThumbnailConfig:Bucket

Role
The IAM Amazon Resource Name (ARN) for the role that you want Elastic Transcoder to use to transcode jobs for this pipeline.

(Optional) AwsKmsKeyArn
The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

If you use either **S3** or **S3-AWS-KMS** as your **Encryption:Mode**, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an **Encryption:Mode** of **AES-PKCS7**, **AES-CTR**, or **AES-GCM**.

Notifications:Progressing
The topic ARN for the Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic. For more information, see Create a Topic in the *Amazon Simple Notification Service Developer Guide*.

Important
To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

Amazon SNS offers a variety of notification options, including the ability to send Amazon SNS messages to Amazon Simple Queue Service queues. For more information, see the Amazon Simple Notification Service Developer Guide.

Notifications:Completed
The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

Notifications:Warning
The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

Notifications:Error
The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

ContentConfig (Use this plus ThumbnailConfig, or use OutputBucket)
The `ContentConfig` object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists: which bucket to use, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for `ContentConfig`, you must also specify values for `ThumbnailConfig>`.

If you specify values for `ContentConfig` and `ThumbnailConfig`, omit the `OutputBucket` object.

ContentConfig:Bucket
The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.

(Optional) ContentConfig:Permissions
The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

ContentConfig:Permissions:GranteeType
Specify the type of value that appears in the `ContentConfig:Permissions:Grantee` object:

- **Canonical:** The value in the `Grantee` object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see Access Control List (ACL) Overview in the *Amazon Simple Storage Service Developer Guide*. For more information about using CloudFront origin access identities to require that users use CloudFront URLs instead of Amazon S3 URLs, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content.

  Important
  A canonical user ID is not the same as an AWS account number.

- **Email:** The value in the `Grantee` object is the registered email address of an AWS account.
- **Group:** The value in the `Grantee` object is one of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.

ContentConfig:Permissions:Grantee
The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group. For more information, see `ContentConfig:Permissions:GranteeType`.

ContentConfig:Permissions:Access
The permission that you want to give to the AWS user that you specified in `ContentConfig:Permissions:Grantee`. Permissions are granted on the files that Elastic Transcoder adds to the bucket, including playlists, video files, and audio files. Valid values include:

- **`READ`:** The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **`READ_ACP`:** The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **`WRITE_ACP`:** The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **`FULL_CONTROL`:** The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.

ContentConfig:StorageClass
The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the files and playlists that it stores in your Amazon S3 bucket. For more information, see Reduced Redundancy Storage in the *Amazon Simple Storage Service Developer Guide*.

ThumbnailConfig (Use this plus ContentConfig, or use OutputBucket)
The `ThumbnailConfig` object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files: which bucket to use, which users you want to have access

to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for `ContentConfig`, you must also specify values for `ThumbnailConfig` even if you don't want to create thumbnails. (You control whether to create thumbnails when you create a job. For more information, see ThumbnailPattern in the topic Create Job (p. 181).)

If you specify values for `ContentConfig` and `ThumbnailConfig`, omit the `OutputBucket` object.

ThumbnailConfig:Bucket
The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.

(Optional) ThumbnailConfig:Permissions
The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the thumbnails to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

ThumbnailConfig:Permissions:GranteeType
Specify the type of value that appears in the `ThumbnailConfig:Permissions:Grantee` object:

- **Canonical:** The value in the `Grantee` object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see Access Control List (ACL) Overview in the *Amazon Simple Storage Service Developer Guide*. For more information about using CloudFront origin access identities to require that users use CloudFront URLs instead of Amazon S3 URLs, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content.

  Important
  A canonical user ID is not the same as an AWS account number.

- **Email:** The value in the `Grantee` object is the registered email address of an AWS account.
- **Group:** The value in the `Grantee` object is one of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.

ThumbnailConfig:Permissions:Grantee
The AWS user or group that you want to have access to thumbnail files. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group. For more information, see `ThumbnailConfig:Permissions:GranteeType`.

ThumbnailConfig:Permissions:Access
The permission that you want to give to the AWS user that you specified in `ThumbnailConfig:Permissions:Grantee`. Permissions are granted on the thumbnail files that Elastic Transcoder adds to the bucket. Valid values include:

- **`READ`:** The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **`READ_ACP`:** The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **`WRITE_ACP`:** The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **`FULL_CONTROL`:** The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.

ThumbnailConfig:StorageClass

The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket. For more information, see Reduced Redundancy Storage in the *Amazon Simple Storage Service Developer Guide*.

Responses

Syntax

```
Status: 201 Created
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Pipeline":{
      "Id":"Id for the new pipeline",
      "Name":"pipeline name",
      "InputBucket":"Amazon S3 bucket that contains files to transcode
         and graphics to use as watermarks",
      "OutputBucket":"Use this, or use ContentConfig:Bucket plus
         ThumbnailConfig:Bucket",
      "Role":"IAM role ARN",
      "AwsKmsKeyArn":"AWS-KMS key arn of the AWS-KMS key you want to
            use with this pipeline",
      "Notifications":{
         "Progressing":"SNS topic to notify when
            Elastic Transcoder has started to process the job",
         "Completed":"SNS topic to notify when
            Elastic Transcoder has finished processing the job",
         "Warning":"SNS topic to notify when
            Elastic Transcoder encounters a warning condition",
         "Error":"SNS topic to notify when
            Elastic Transcoder encounters an error condition"
      },
      "ContentConfig":{
         "Bucket":"Use this plus ThumbnailConfig:Bucket,
            or use OutputBucket",
         "Permissions":[
            {
               "GranteeType":"Canonical|Email|Group",
               "Grantee":"AWS user ID or CloudFront origin access identity"|
                         "registered email address for AWS account|
                         AllUsers|AuthenticatedUsers|LogDelivery",
               "Access":[
                  "Read|ReadAcp|WriteAcp|FullControl",
                  ...
               ]
            },
            {...}
         ],
         "StorageClass":"Standard|ReducedRedundancy"
      },
      "ThumbnailConfig":{
         "Bucket":"Use this plus ContentConfig:Bucket,
```

```
            or use OutputBucket",
         "Permissions":[
            {
               "Access":[
                  "Read|ReadAcp|WriteAcp|FullControl",
                  ...
               ],
               "Grantee":"AWS user ID or CloudFront origin access identity"|
                         "registered email address for AWS account"|
                         AllUsers|AuthenticatedUsers|LogDelivery",
               "GranteeType":"Canonical|Email|Group"
            },
            {...}
         ],
         "StorageClass":"Standard|ReducedRedundancy"
      },
      "Status":"Active|Paused"
   },
   "Warnings (p. 134)": [
      {
         "Code": "6000|6001|6002|6003|6004|6005|6006|6007|6008",
         "Message": "The code message"
      },
      {...}
   ]
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

When you create a pipeline, Elastic Transcoder returns the values that you specified in the request. For more information, see Request Body (p. 129).

In addition, Elastic Transcoder returns the following values.

Id
Identifier for the pipeline. You use this value to identify the pipeline in which you want to perform a variety of operations, for example, creating a job or a preset.

Status
The current status of the pipeline:

- `Active`: The pipeline is processing jobs.
- `Paused`: The pipeline is not currently processing jobs.

Warnings

When you create a pipeline that uses resources in other regions, Elastic Transcoder returns one or more warnings. Your pipeline is still created, but might have increased processing times and incur cross-regional charges. The warnings are in the following format:

Code — The warning code.
Message — the message associated with the warning code.

The warning codes and messages that can be returned are as follows:

6000
The input bucket and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the input bucket and the pipeline.

6001
The ContentConfig bucket and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the ContentConfig bucket and the pipeline.

6002
The ThumbnailConfig bucket and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the ThumbnailConfig bucket and the pipeline.

6003
The SNS notification topic for progressing events and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the SNS notification topic and the pipeline.

6004
The SNS notification topic for warning events and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the SNS notification topic and the pipeline.

6005
The SNS notification topic for completion events and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the SNS notification topic and the pipeline.

6006
The SNS notification topic for error events and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the SNS notification topic and the pipeline.

6007
The AWS KMS key and ContentConfig bucket specified for this pipeline are in different regions, which causes outputs using S3-AWS-KMS encryption mode to fail. To use S3-AWS-KMS encryption mode, use the same region for the KMS key and the ContentConfig bucket.

6008
The AWS KMS key and ThumbnailConfig bucket specified for this pipeline are in different regions, which causes outputs using S3-AWS-KMS encryption mode to fail. To use S3-AWS-KMS encryption mode, use the same region for the KMS key and the ThumbnailConfig bucket.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

The following example request creates a pipeline named `Default`.

Sample Request

```
POST /2012-09-25/pipelines HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
   "Name":"Default",
   "InputBucket":"salesoffice.example.com-source",
   "Role":"arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
   "AwsKmsKeyArn":"base64-encoded key from KMS",
   "Notifications":{
      "Progressing":"",
      "Completed":"",
      "Warning":"",
      "Error":"arn:aws:sns:us-east-1:111222333444:ET_Errors"
   },
   "ContentConfig":{
      "Bucket":"salesoffice.example.com-public-promos",
      "Permissions":[
         {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
               "FullControl"
            ]
         }
      ],
      "StorageClass":"Standard"
   },
   "ThumbnailConfig":{
      "Bucket":"salesoffice.example.com-public-promos-thumbnails",
      "Permissions":[
         {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
               "FullControl"
            ]
         }
      ],
      "StorageClass":"ReducedRedundancy"
   }
}
```

Sample Response

```
Status: 201 Created
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
```

```
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT
{
   "Pipeline":{
     "Id":"1111111111111-abcde1",
     "Name":"Default",
     "InputBucket":"salesoffice.example.com-source",
     "Role":"arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
     "AwsKmsKeyArn":"base64-encoded key from KMS",
     "Notifications":{
         "Completed":"",
         "Error":"arn:aws:sns:us-east-1:111222333444:ET_Errors",
         "Progressing":"",
         "Warning":""
      },
      "ContentConfig":{
         "Bucket":"salesoffice.example.com-public-promos",
         "Permissions":[
            {
               "GranteeType":"Email",
               "Grantee":"marketing-promos@example.com",
               "Access":[
                  "FullControl"
               ]
            }
         ],
         "StorageClass":"Standard"
      },
      "ThumbnailConfig":{
         "Bucket":"salesoffice.example.com-public-promos-thumbnails",
         "Permissions":[
            {
               "GranteeType":"Email",
               "Grantee":"marketing-promos@example.com",
               "Access":[
                  "FullControl"
               ]
            }
         ],
         "StorageClass":"ReducedRedundancy"
      },
      "Status":"Active"
   },
   "Warnings": [
      {
         "Code": "6000",
         "Message": "The input bucket and the pipeline are in different
            regions, which increases processing time for jobs in the
            pipeline and can incur additional charges. To decrease
            processing time and prevent cross-regional charges, use the
            same region for the input bucket and the pipeline."
      },
      {...}
   ]
}
```

List Pipelines

Topics

- Description (p. 138)
- Requests (p. 138)
- Responses (p. 139)
- Errors (p. 144)
- Examples (p. 144)

Description

To get a list of the pipelines associated with the current AWS account, send a GET request to the `/2012-09-25/pipelines/` resource.

Requests

Syntax

To get information about all of the pipelines associated with the current AWS account, send the following GET request.

```
GET /2012-09-25/pipelines/Ascending=true|false&
PageToken=value for accessing the next page of results HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Request Parameters

This operation takes the following request parameters. Elastic Transcoder returns all of the pipelines.

Ascending
To list pipelines in chronological order by the date and time that they were submitted, enter `true`. To list pipelines in reverse chronological order, enter `false`.

PageToken
When Elastic Transcoder returns more than one page of results, use `PageToken` in subsequent `GET` requests to get each successive page of results.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

This operation does not have a request body.

Responses

Syntax

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT
{
   "Pipelines":[
      {
         "Id":"Id for the new pipeline",
         "Name":"pipeline name",
         "InputBucket":"Amazon S3 bucket that contains files to transcode
            and graphics to use as watermarks",
         "OutputBucket":"Use this, or use ContentConfig:Bucket plus
            ThumbnailConfig:Bucket",
         "Role":"IAM role ARN",
         "AwsKmsKeyArn":"AWS-KMS key ARN of the AWS-KMS key you want to
            use with this pipeline",
         "Notifications":{
            "Progressing":"SNS topic to notify when
               Elastic Transcoder has started to create the pipeline",
            "Completed":"SNS topic to notify when
               Elastic Transcoder has created the pipeline",
            "Warning":"SNS topic to notify when
               Elastic Transcoder returns a warning",
            "Error":"SNS topic to notify when
               Elastic Transcoder returns an error"
         },
         "ContentConfig":{
            "Bucket":"Use this plus ThumbnailConfig:Bucket,
               or use OutputBucket",
            "Permissions":[
               {
                  "GranteeType":"Canonical|Email|Group",
                  "Grantee":"AWS user ID or CloudFront origin access
                     identity|registered email address for AWS
                     account|AllUsers|AuthenticatedUsers|LogDelivery",
                  "Access":[
                     "Read|ReadAcp|WriteAcp|FullControl",
                     ...
                  ]
               },
               {...}
            ],
            "StorageClass":"Standard|ReducedRedundancy"
         },
         "ThumbnailConfig":{
            "Bucket":"Use this plus ContentConfig:Bucket,
               or use OutputBucket",
            "Permissions":[
               {
                  "GranteeType":"Canonical|Email|Group",
                  "Grantee":"AWS user ID or CloudFront origin access
                     identity|registered email address for AWS
```

```
                    account|AllUsers|AuthenticatedUsers|LogDelivery",
                  "Access":[
                     "Read|ReadAcp|WriteAcp|FullControl",
                     ...
                  ]
               },
               {...}
            ],
            "StorageClass":"Standard|ReducedRedundancy"
         }
         "Status":"Active|Paused"
      },
      {...}
   ],
   "NextPageToken":value for accessing the next page of results|null
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

Id

Identifier for the pipeline. You use this value to identify the pipeline in which you want to perform a variety of operations, for example, creating a job or a preset.

Name

The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

InputBucket

The Amazon S3 bucket in which you saved the media files that you want to transcode and the graphics that you want to use as watermarks.

OutputBucket (Use this, or use ContentConfig:Bucket plus ThumbnailConfig:Bucket)

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.

  Note
  When Elastic Transcoder saves files in `OutputBucket`, it grants full control over the files only to the AWS account that owns the role that is specified by `Role`.

- You want to associate the transcoded files and thumbnails with the Amazon S3 `Standard` storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

Role
The IAM Amazon Resource Name (ARN) for the role that you want Elastic Transcoder to use to transcode jobs for this pipeline.

(Optional) AwsKmsKeyArn
The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

If you use either **S3** or **S3-AWS-KMS** as your **Encryption:Mode**, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an **Encryption:Mode** of **AES-PKCS7**, **AES-CTR**, or **AES-GCM**.

Notifications:Progressing
The topic ARN for the Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic. For more information, see Create a Topic in the *Amazon Simple Notification Service Developer Guide*.

> **Important**
> To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

Amazon SNS offers a variety of notification options, including the ability to send Amazon SNS messages to Amazon Simple Queue Service queues. For more information, see the Amazon Simple Notification Service Developer Guide.

Notifications:Completed
The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

Notifications:Warning
The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

Notifications:Error
The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

ContentConfig (Use this plus ThumbnailConfig, or use OutputBucket)
The `ContentConfig` object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists: which bucket to use, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for `ContentConfig`, you must also specify values for `ThumbnailConfig`>.

If you specify values for `ContentConfig` and `ThumbnailConfig`, omit the `OutputBucket` object.

ContentConfig:Bucket
The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.

(Optional) ContentConfig:Permissions
The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

ContentConfig:Permissions:GranteeType

Specify the type of value that appears in the `ContentConfig:Permissions:Grantee` object:

- **Canonical:** The value in the `Grantee` object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see Access Control List (ACL) Overview in the *Amazon Simple Storage Service Developer Guide*. For more information about using CloudFront origin access identities to require that users use CloudFront URLs instead of Amazon S3 URLs, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content.

  Important
  A canonical user ID is not the same as an AWS account number.

- **Email:** The value in the `Grantee` object is the registered email address of an AWS account.
- **Group:** The value in the `Grantee` object is one of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.

ContentConfig:Permissions:Grantee

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group. For more information, see `ContentConfig:Permissions:GranteeType`.

ContentConfig:Permissions:Access

The permission that you want to give to the AWS user that you specified in `ContentConfig:Permissions:Grantee`. Permissions are granted on the files that Elastic Transcoder adds to the bucket, including playlists, video files, and audio files. Valid values include:

- **`READ`:** The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **`READ_ACP`:** The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **`WRITE_ACP`:** The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **`FULL_CONTROL`:** The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.

ContentConfig:StorageClass

The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the files and playlists that it stores in your Amazon S3 bucket. For more information, see Reduced Redundancy Storage in the *Amazon Simple Storage Service Developer Guide*.

ThumbnailConfig (Use this plus ContentConfig, or use OutputBucket)

The `ThumbnailConfig` object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files: which bucket to use, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for `ContentConfig`, you must also specify values for `ThumbnailConfig` even if you don't want to create thumbnails. (You control whether to create thumbnails when you create a job. For more information, see ThumbnailPattern in the topic Create Job (p. 181).)

If you specify values for `ContentConfig` and `ThumbnailConfig`, omit the `OutputBucket` object.

ThumbnailConfig:Bucket

The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.

(Optional) ThumbnailConfig:Permissions
The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the thumbnails to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

ThumbnailConfig:Permissions:GranteeType
Specify the type of value that appears in the `ThumbnailConfig:Permissions:Grantee` object:

- **Canonical:** The value in the `Grantee` object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see Access Control List (ACL) Overview in the *Amazon Simple Storage Service Developer Guide*. For more information about using CloudFront origin access identities to require that users use CloudFront URLs instead of Amazon S3 URLs, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content.

  Important
  A canonical user ID is not the same as an AWS account number.
- **Email:** The value in the `Grantee` object is the registered email address of an AWS account.
- **Group:** The value in the `Grantee` object is one of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.

ThumbnailConfig:Permissions:Grantee
The AWS user or group that you want to have access to thumbnail files. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group. For more information, see `ThumbnailConfig:Permissions:GranteeType`.

ThumbnailConfig:Permissions:Access
The permission that you want to give to the AWS user that you specified in `ThumbnailConfig:Permissions:Grantee`. Permissions are granted on the thumbnail files that Elastic Transcoder adds to the bucket. Valid values include:

- **`READ`:** The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **`READ_ACP`:** The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **`WRITE_ACP`:** The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **`FULL_CONTROL`:** The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.

ThumbnailConfig:StorageClass
The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket. For more information, see Reduced Redundancy Storage in the *Amazon Simple Storage Service Developer Guide*.

Status
The current status of the pipeline:

- `Active`: The pipeline is processing jobs.
- `Paused`: The pipeline is not currently processing jobs.

NextPageToken
A value that you use to access the second and subsequent pages of results, if any. When the pipelines fit on one page or when you've reached the last page of results, the value of `NextPageToken` is `null`.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

The following example request gets a list of the pipelines associated with the current AWS account.

Sample Request

```
GET /2012-09-25/pipelines HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Sample Response

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT
{
   "Pipelines":[
      {
         "Id":"1111111111111-abcde1",
         "Name":"Tokyo-Default",
         "InputBucket":"salesoffice-tokyo.example.com-source",
         "OutputBucket":"salesoffice-tokyo.example.com-output",
         "Role":"arn:aws:iam::123456789012:role/Elastic_Transcoder_Default
            _Role",
         "AwsKmsKeyArn":"base64 encoded key from KMS",
         "Notifications":{
            "Progressing":"",
            "Completed":"",
            "Warning":"",
            "Error":"arn:aws:sns:us-east-1:111222333444:ETS_Errors"
         },
         "ContentConfig":{
            "Bucket":"salesoffice-tokyo.example.com-public-promos",
            "Permissions":[
               {
                  "GranteeType":"Email",
                  "Grantee":"marketing-promos-tokyo@example.com",
                  "Access":[
                     "FullControl"
                  ]
               }
```

```
            ],
            "StorageClass":"Standard"
         },
         "ThumbnailConfig":{
            "Bucket":"salesoffice-tokyo.example.com-public-promos-
               thumbnails",
            "Permissions":[
               {
                  "GranteeType":"Email",
                  "Grantee":"marketing-promos-tokyo@example.com",
                  "Access":[
                     "FullControl"
                  ]
               }
            ],
            "StorageClass":"ReducedRedundancy"
         },
         "Status":"Active"
      },
      {
         "Id":"2222222222222-abcde2",
         "Name":"Amsterdam-Default",
         "InputBucket":"salesoffice-amsterdam.example.com-source",
         "OutputBucket":"salesoffice-amsterdam.example.com-output",
         "Role":"arn:aws:iam::123456789012:role/Elastic_Transcoder_Default
            _Role",
         "AwsKmsKeyArn":"base64 encoded key from KMS",
         "Notifications":{
            "Progressing":"",
            "Completed":"",
            "Warning":"",
            "Error":"arn:aws:sns:us-east-1:111222333444:ETS_Errors"
         },
         "ContentConfig":{
            "Bucket":"salesoffice-amsterdam.example.com-public-promos",
            "Permissions":[
               {
                  "GranteeType":"Email",
                  "Grantee":"marketing-promos-amsterdam@example.com",
                  "Access":[
                     "FullControl"
                  ]
               }
            ],
            "StorageClass":"Standard"
         },
         "ThumbnailConfig":{
            "Bucket":"salesoffice-amsterdam.example.com-public-promos-
               thumbnails",
            "Permissions":[
               {
                  "GranteeType":"Email",
                  "Grantee":"marketing-promos-amsterdam@example.com",
                  "Access":[
                     "FullControl"
                  ]
               }
            ],
```

```
            "StorageClass":"ReducedRedundancy"
         },
         "Status":"Active"
      }
   ]
}
```

Read Pipeline

Topics

- Description (p. 147)
- Requests (p. 147)
- Responses (p. 147)
- Errors (p. 153)
- Examples (p. 153)

Description

To get detailed information about a pipeline, send a GET request to the `/2012-09-25/pipelines/pipelineId` resource.

Requests

Syntax

```
GET /2012-09-25/pipelines/pipelineId HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Request Parameters

This operation takes the following request parameter.

pipelineId
The identifier of the pipeline for which you want to get detailed information.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

This operation does not have a request body.

Responses

Syntax

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
```

```
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Pipeline":{
      "Id":"Id for the new pipeline",
      "Name":"pipeline name",
      "InputBucket":"Amazon S3 bucket that contains files to transcode
         and graphics to use as watermarks",
      "OutputBucket":"Use this, or use ContentConfig:Bucket plus
            ThumbnailConfig:Bucket",
      "Role":"IAM role ARN",
      "AwsKmsKeyArn":"AWS-KMS key ARN of the AWS-KMS key you want to
         use with this pipeline",
      "Notifications":{
         "Progressing":"SNS topic to notify when
            Elastic Transcoder has started to create the pipeline",
         "Completed":"SNS topic to notify when
            Elastic Transcoder has created the pipeline",
         "Warning":"SNS topic to notify when
            Elastic Transcoder returns a warning",
         "Error":"SNS topic to notify when
            Elastic Transcoder returns an error"
      },
      "ContentConfig":{
         "Bucket":"Use this plus ThumbnailConfig:Bucket,
            or use OutputBucket",
         "Permissions":[
            {
               "GranteeType":"Canonical|Email|Group",
               "Grantee":"AWS user ID or CloudFront origin access identity"|
                         "registered email address for AWS account|
                         AllUsers|AuthenticatedUsers|LogDelivery",
               "Access":[
                  "Read|ReadAcp|WriteAcp|FullControl",
                  ...
               ]
            },
            {...}
         ],
         "StorageClass":"Standard|ReducedRedundancy"
      },
      "ThumbnailConfig":{
         "Bucket":"Use this plus ContentConfig:Bucket,
               or use OutputBucket",
         "Permissions":[
            {
               "GranteeType":"Canonical|Email|Group",
               "Grantee":"AWS user ID or CloudFront origin access identity"|
                         "registered email address for AWS account|
                         AllUsers|AuthenticatedUsers|LogDelivery",
               "Access":[
                  "Read|ReadAcp|WriteAcp|FullControl",
                  ...
               ]
            },
            {...}
```

```
         ],
         "StorageClass":"Standard|ReducedRedundancy"
      }
      "Status":"Active|Paused"
   },
   "Warnings (p. 152)": [
      {
         "Code": "6000|6001|6002|6003|6004|6005|6006|6007|6008",
         "Message": "The code message"
      },
      {...}
   ]
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

The response body contains the following JSON objects.

Id

Identifier for the pipeline. You use this value to identify the pipeline in which you want to perform a variety of operations, for example, creating a job or a preset.

Name

The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

InputBucket

The Amazon S3 bucket in which you saved the media files that you want to transcode and the graphics that you want to use as watermarks.

(Optional) OutputBucket

The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.

  Note
  When Elastic Transcoder saves files in `OutputBucket`, it grants full control over the files only to the AWS account that owns the role that is specified by `Role`.

- You want to associate the transcoded files and thumbnails with the Amazon S3 `Standard` storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

Role

The IAM Amazon Resource Name (ARN) for the role that you want Elastic Transcoder to use to transcode jobs for this pipeline.

(Optional) AwsKmsKeyArn
The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

If you use either **S3** or **S3-AWS-KMS** as your **Encryption:Mode**, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an **Encryption:Mode** of **AES-PKCS7**, **AES-CTR**, or **AES-GCM**.

Notifications:Progressing
The topic ARN for the Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic. For more information, see Create a Topic in the *Amazon Simple Notification Service Developer Guide*.

Important
To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

Amazon SNS offers a variety of notification options, including the ability to send Amazon SNS messages to Amazon Simple Queue Service queues. For more information, see the Amazon Simple Notification Service Developer Guide.

Notifications:Completed
The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

Notifications:Warning
The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

Notifications:Error
The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

ContentConfig (Use this plus ThumbnailConfig, or use OutputBucket)
The `ContentConfig` object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists: which bucket to use, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for `ContentConfig`, you must also specify values for `ThumbnailConfig`>.

If you specify values for `ContentConfig` and `ThumbnailConfig`, omit the `OutputBucket` object.

ContentConfig:Bucket
The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.

(Optional) ContentConfig:Permissions
The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

ContentConfig:Permissions:GranteeType
Specify the type of value that appears in the `ContentConfig:Permissions:Grantee` object:

- **Canonical:** The value in the `Grantee` object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see Access Control List (ACL) Overview in the *Amazon Simple Storage Service Developer Guide*. For more information about using CloudFront origin access identities to require that users use CloudFront URLs instead of Amazon S3 URLs, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content.

  Important
  A canonical user ID is not the same as an AWS account number.
- **Email:** The value in the `Grantee` object is the registered email address of an AWS account.
- **Group:** The value in the `Grantee` object is one of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.

ContentConfig:Permissions:Grantee
The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group. For more information, see `ContentConfig:Permissions:GranteeType`.

ContentConfig:Permissions:Access
The permission that you want to give to the AWS user that you specified in `ContentConfig:Permissions:Grantee`. Permissions are granted on the files that Elastic Transcoder adds to the bucket, including playlists, video files, and audio files. Valid values include:

- **`READ`:** The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **`READ_ACP`:** The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **`WRITE_ACP`:** The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **`FULL_CONTROL`:** The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.

ContentConfig:StorageClass
The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the files and playlists that it stores in your Amazon S3 bucket. For more information, see Reduced Redundancy Storage in the *Amazon Simple Storage Service Developer Guide*.

ThumbnailConfig (Use this plus ContentConfig, or use OutputBucket)
The `ThumbnailConfig` object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files: which bucket to use, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for `ContentConfig`, you must also specify values for `ThumbnailConfig` even if you don't want to create thumbnails. (You control whether to create thumbnails when you create a job. For more information, see ThumbnailPattern in the topic Create Job (p. 181).)

If you specify values for `ContentConfig` and `ThumbnailConfig`, omit the `OutputBucket` object.

ThumbnailConfig:Bucket
The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.

(Optional) ThumbnailConfig:Permissions
The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the thumbnails to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

ThumbnailConfig:Permissions:GranteeType

Specify the type of value that appears in the `ThumbnailConfig:Permissions:Grantee` object:

- **Canonical:** The value in the `Grantee` object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see Access Control List (ACL) Overview in the *Amazon Simple Storage Service Developer Guide*. For more information about using CloudFront origin access identities to require that users use CloudFront URLs instead of Amazon S3 URLs, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content.

  Important
  A canonical user ID is not the same as an AWS account number.

- **Email:** The value in the `Grantee` object is the registered email address of an AWS account.
- **Group:** The value in the `Grantee` object is one of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.

ThumbnailConfig:Permissions:Grantee

The AWS user or group that you want to have access to thumbnail files. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group. For more information, see `ThumbnailConfig:Permissions:GranteeType`.

ThumbnailConfig:Permissions:Access

The permission that you want to give to the AWS user that you specified in `ThumbnailConfig:Permissions:Grantee`. Permissions are granted on the thumbnail files that Elastic Transcoder adds to the bucket. Valid values include:

- **`READ`:** The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **`READ_ACP`:** The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **`WRITE_ACP`:** The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **`FULL_CONTROL`:** The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.

ThumbnailConfig:StorageClass

The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket. For more information, see Reduced Redundancy Storage in the *Amazon Simple Storage Service Developer Guide*.

Status

The current status of the pipeline:

- `Active`: The pipeline is processing jobs.
- `Paused`: The pipeline is not currently processing jobs.

Warnings

When you create a pipeline that uses resources in other regions, Elastic Transcoder returns one or more warnings. Your pipeline is still created, but might have increased processing times and incur cross-regional charges. The warnings are in the following format:

Code

Message — the message associated with the warning code.

The following is a list of valid warning codes and their messages:

6000
The input bucket and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the input bucket and the pipeline.

6001
The ContentConfig bucket and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the ContentConfig bucket and the pipeline.

6002
The ThumbnailConfig bucket and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the ThumbnailConfig bucket and the pipeline.

6003
The SNS notification topic for progressing events and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the SNS notification topic and the pipeline.

6004
The SNS notification topic for warning events and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the SNS notification topic and the pipeline.

6005
The SNS notification topic for completion events and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the SNS notification topic and the pipeline.

6006
The SNS notification topic for error events and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the SNS notification topic and the pipeline.

6007
The AWS KMS key and ContentConfig bucket specified for this pipeline are in different regions, which causes outputs using S3-AWS-KMS encryption mode to fail. To use S3-AWS-KMS encryption mode, use the same region for the KMS key and the ContentConfig bucket.

6008
The AWS KMS key and ThumbnailConfig bucket specified for this pipeline are in different regions, which causes outputs using S3-AWS-KMS encryption mode to fail. To use S3-AWS-KMS encryption mode, use the same region for the KMS key and the ThumbnailConfig bucket.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

The following example request gets the pipeline that has the ID `1111111111111-abcde1`.

Sample Request

```
GET /2012-09-25/pipelines/1111111111111-abcde1 HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Sample Response

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Pipeline":{
      "Id":"1111111111111-abcde1",
      "Name":"Default",
      "InputBucket":"salesoffice.example.com-source",
      "OutputBucket":"salesoffice.example.com-output",
     "Role":"arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
      "AwsKmsKeyArn":"base64 encoded key from KMS",
      "Notifications":{
         "Progressing":"",
         "Completed":"",
         "Warning":"",
         "Error":"arn:aws:sns:us-east-1:111222333444:ET_Errors"
      },
      "ContentConfig":{
         "Bucket":"salesoffice.example.com-public-promos",
         "Permissions":[
            {
               "GranteeType":"Email",
               "Grantee":"marketing-promos@example.com",
               "Access":[
                  "FullControl"
               ]
            }
         ],
         "StorageClass":"Standard"
      },
      "ThumbnailConfig":{
         "Bucket":"salesoffice.example.com-public-promos-thumbnails",
         "Permissions":[
            {
               "GranteeType":"Email",
               "Grantee":"marketing-promos@example.com",
               "Access":[
                  "FullControl"
```

```
                ]
            }
         ],
         "StorageClass":"ReducedRedundancy"
      },
      "Status":"Active"
   },
   "Warnings": [
      {
         "Code": "6000",
         "Message": "The input bucket and the pipeline are in different
            regions, which increases processing time for jobs in the
            pipeline and can incur additional charges. To decrease
            processing time and prevent cross-regional charges, use the
            same region for the input bucket and the pipeline."
      },
      {...}
   ]
}
```

Update Pipeline

Topics

- Description (p. 156)
- How to Update and Delete Values (p. 156)
- Requests (p. 157)
- Responses (p. 162)
- Errors (p. 164)
- Examples (p. 164)

Description

To update settings for a pipeline, send a PUT request to the `/2012-09-25/pipelines/`*`pipelineId`* resource.

> **Important**
> When you change pipeline settings, your changes take effect immediately. Jobs that you have already submitted and that Elastic Transcoder has not started to process are affected in addition to jobs that you submit after you change settings.

How to Update and Delete Values

To update a value, specify the new value, for example:

```
"GranteeType":"Canonical"
```

To leave a value unchanged, do one of the following:

- Specify the current value.
- Specify a null value. This applies both to individual objects (for example, `"Role":null`) and to arrays (for example, `"ContentConfig":null`).
- Omit the JSON object from the request. This applies both to individual objects, for example, `Role`, and to arrays, for example, `ContentConfig`.

To delete the current value, specify an empty value. For example, to delete a notification, specify an empty string: `"Progressing":""`. To delete all of the values in an array, specify an empty array: `"Permissions":[]`.

For example, if you no longer want to be notified when Elastic Transcoder encounters a warning while processing a job, and you don't want to change any of the other notification settings, you would specify the following for `Notifications` in the request:

```
...
"Notifications":{
   "Warning":""
},
...
```

Requests

Syntax

```
PUT /2012-09-25/pipelines/pipelineId HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
   "Id":"pipelineId",
   "Name":"pipeline name",
   "InputBucket":"Amazon S3 bucket that contains files to transcode
      and graphics to use as watermarks",
   "OutputBucket":"Use this, or use ContentConfig:Bucket plus
      ThumbnailConfig:Bucket",
   "Role":"IAM role ARN",
   "AwsKmsKeyArn":"AWS-KMS key ARN of the AWS-KMS key you want to
      use with this pipeline",
   "Notifications":{
      "Progressing":"SNS topic to notify when
         Elastic Transcoder has started to process the job",
      "Completed":"SNS topic to notify when
         Elastic Transcoder has finished processing the job",
      "Warning":"SNS topic to notify when
         Elastic Transcoder encounters a warning condition"
      "Error":"SNS topic to notify when
         Elastic Transcoder encounters an error condition"
   },
   "ContentConfig":{
      "Bucket":"Use this plus ThumbnailConfig:Bucket,
         or use OutputBucket",
      "Permissions":[
         {
            "GranteeType":"Canonical|Email|Group",
            "Grantee":"AWS user ID or CloudFront origin access identity"|
                      "registered email address for AWS account|
                      AllUsers|AuthenticatedUsers|LogDelivery",
            "Access":[
               "Read|ReadAcp|WriteAcp|FullControl",
               ...
            ]
         },
         {...}
      ],
      "StorageClass":"Standard|ReducedRedundancy"
   },
   "ThumbnailConfig":{
      "Bucket":"Use this plus ContentConfig:Bucket,
         or use OutputBucket",
      "Permissions":[
```

```
         {
            "GranteeType":"Canonical|Email|Group",
            "Grantee":"AWS user ID or CloudFront origin access identity"|
                      "registered email address for AWS account|
                      AllUsers|AuthenticatedUsers|LogDelivery",
            "Access":[
               "Read|ReadAcp|WriteAcp|FullControl",
               ...
            ]
         },
         {...}
      ],
      "StorageClass":"Standard|ReducedRedundancy"
   },
   "Status":"Active|Paused"
}
```

Request Parameters

This operation does not use request parameters.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

The JSON string in the request body contains the following objects.

Id
Identifier for the pipeline. You use this value to identify the pipeline in which you want to perform a variety of operations, for example, creating a job or a preset.

Name
The name of the pipeline. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

InputBucket
The Amazon S3 bucket in which you saved the media files that you want to transcode and the graphics that you want to use as watermarks.

OutputBucket (Use this, or use ContentConfig:Bucket plus ThumbnailConfig:Bucket)
The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. Specify this value when all of the following are true:

- You want to save transcoded files, thumbnails (if any), and playlists (if any) together in one bucket.
- You do not want to specify the users or groups who have access to the transcoded files, thumbnails, and playlists.
- You do not want to specify the permissions that Elastic Transcoder grants to the files.

  Note
  When Elastic Transcoder saves files in `OutputBucket`, it grants full control over the files only to the AWS account that owns the role that is specified by `Role`.

- You want to associate the transcoded files and thumbnails with the Amazon S3 `Standard` storage class.

If you want to save transcoded files and playlists in one bucket and thumbnails in another bucket, specify which users can access the transcoded files or the permissions the users have, or change the Amazon S3 storage class, omit `OutputBucket` and specify values for `ContentConfig` and `ThumbnailConfig` instead.

Role

The IAM Amazon Resource Name (ARN) for the role that you want Elastic Transcoder to use to transcode jobs for this pipeline.

(Optional) AwsKmsKeyArn

The AWS Key Management Service (AWS KMS) key that you want to use with this pipeline.

If you use either **S3** or **S3-AWS-KMS** as your **Encryption:Mode**, you don't need to provide a key with your job because a default key, known as an AWS-KMS key, is created for you automatically. You need to provide an AWS-KMS key only if you want to use a non-default AWS-KMS key, or if you are using an **Encryption:Mode** of **AES-PKCS7**, **AES-CTR**, or **AES-GCM**.

Notifications:Progressing

The topic ARN for the Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic. For more information, see Create a Topic in the *Amazon Simple Notification Service Developer Guide*.

> **Important**
> To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

Amazon SNS offers a variety of notification options, including the ability to send Amazon SNS messages to Amazon Simple Queue Service queues. For more information, see the Amazon Simple Notification Service Developer Guide.

Notifications:Completed

The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

Notifications:Warning

The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

Notifications:Error

The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

ContentConfig (Use this plus ThumbnailConfig, or use OutputBucket)

The `ContentConfig` object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists: which bucket to use, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for `ContentConfig`, you must also specify values for `ThumbnailConfig>`.

If you specify values for `ContentConfig` and `ThumbnailConfig`, omit the `OutputBucket` object.

ContentConfig:Bucket

The Amazon S3 bucket in which you want Elastic Transcoder to save transcoded files and playlists.

(Optional) ContentConfig:Permissions

The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to transcoded files and playlists, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the transcoded files and playlists to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

ContentConfig:Permissions:GranteeType

Specify the type of value that appears in the `ContentConfig:Permissions:Grantee` object:

- **Canonical:** The value in the `Grantee` object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see Access Control List (ACL) Overview in the *Amazon Simple Storage Service Developer Guide*. For more information about using CloudFront origin access identities to require that users use CloudFront URLs instead of Amazon S3 URLs, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content.

  Important
  A canonical user ID is not the same as an AWS account number.

- **Email:** The value in the `Grantee` object is the registered email address of an AWS account.
- **Group:** The value in the `Grantee` object is one of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.

ContentConfig:Permissions:Grantee

The AWS user or group that you want to have access to transcoded files and playlists. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group. For more information, see `ContentConfig:Permissions:GranteeType`.

ContentConfig:Permissions:Access

The permission that you want to give to the AWS user that you specified in `ContentConfig:Permissions:Grantee`. Permissions are granted on the files that Elastic Transcoder adds to the bucket, including playlists, video files, and audio files. Valid values include:

- **`READ`:** The grantee can read the objects and metadata for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **`READ_ACP`:** The grantee can read the object ACL for objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **`WRITE_ACP`:** The grantee can write the ACL for the objects that Elastic Transcoder adds to the Amazon S3 bucket.
- **`FULL_CONTROL`:** The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the objects that Elastic Transcoder adds to the Amazon S3 bucket.

ContentConfig:StorageClass

The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the files and playlists that it stores in your Amazon S3 bucket. For more information, see Reduced Redundancy Storage in the *Amazon Simple Storage Service Developer Guide*.

ThumbnailConfig (Use this plus ContentConfig, or use OutputBucket)

The `ThumbnailConfig` object specifies information about the Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files: which bucket to use, which users you want to have access to the files, the type of access you want users to have, and the storage class that you want to assign to the files.

If you specify values for `ContentConfig`, you must also specify values for `ThumbnailConfig` even if you don't want to create thumbnails. (You control whether to create thumbnails when you create a job. For more information, see ThumbnailPattern in the topic Create Job (p. 181).)

If you specify values for `ContentConfig` and `ThumbnailConfig`, omit the `OutputBucket` object.

ThumbnailConfig:Bucket

The Amazon S3 bucket in which you want Elastic Transcoder to save thumbnail files.

(Optional) ThumbnailConfig:Permissions

The `Permissions` object specifies which users and/or predefined Amazon S3 groups you want to have access to thumbnail files, and the type of access you want them to have. You can grant permissions to a maximum of 30 users and/or predefined Amazon S3 groups.

If you include `Permissions`, Elastic Transcoder grants only the permissions that you specify. It does not grant full permissions to the owner of the role specified by `Role`. If you want that user to have full control, you must explicitly grant full control to the user.

If you omit `Permissions`, Elastic Transcoder grants full control over the thumbnails to the owner of the role specified by `Role`, and grants no other permissions to any other user or group.

ThumbnailConfig:Permissions:GranteeType

Specify the type of value that appears in the `ThumbnailConfig:Permissions:Grantee` object:

- **Canonical:** The value in the `Grantee` object is either the canonical user ID for an AWS account or an origin access identity for an Amazon CloudFront distribution. For more information about canonical user IDs, see Access Control List (ACL) Overview in the *Amazon Simple Storage Service Developer Guide*. For more information about using CloudFront origin access identities to require that users use CloudFront URLs instead of Amazon S3 URLs, see Using an Origin Access Identity to Restrict Access to Your Amazon S3 Content.

  Important
  A canonical user ID is not the same as an AWS account number.

- **Email:** The value in the `Grantee` object is the registered email address of an AWS account.
- **Group:** The value in the `Grantee` object is one of the following predefined Amazon S3 groups: `AllUsers`, `AuthenticatedUsers`, or `LogDelivery`.

ThumbnailConfig:Permissions:Grantee

The AWS user or group that you want to have access to thumbnail files. To identify the user or group, you can specify the canonical user ID for an AWS account, an origin access identity for a CloudFront distribution, the registered email address of an AWS account, or a predefined Amazon S3 group. For more information, see `ThumbnailConfig:Permissions:GranteeType`.

ThumbnailConfig:Permissions:Access

The permission that you want to give to the AWS user that you specified in `ThumbnailConfig:Permissions:Grantee`. Permissions are granted on the thumbnail files that Elastic Transcoder adds to the bucket. Valid values include:

- **`READ`:** The grantee can read the thumbnails and metadata for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **`READ_ACP`:** The grantee can read the object ACL for thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **`WRITE_ACP`:** The grantee can write the ACL for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.
- **`FULL_CONTROL`:** The grantee has `READ`, `READ_ACP`, and `WRITE_ACP` permissions for the thumbnails that Elastic Transcoder adds to the Amazon S3 bucket.

ThumbnailConfig:StorageClass

The Amazon S3 storage class, `Standard` or `ReducedRedundancy`, that you want Elastic Transcoder to assign to the thumbnails that it stores in your Amazon S3 bucket. For more information, see Reduced Redundancy Storage in the *Amazon Simple Storage Service Developer Guide*.

Status

The current status of the pipeline:

- `Active`: The pipeline is processing jobs.
- `Paused`: The pipeline is not currently processing jobs.

Responses

Syntax

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Id":"Id for the new pipeline",
   "Name":"pipeline name",
   "InputBucket":"Amazon S3 bucket that contains files to transcode
      and graphics to use as watermarks",
   "OutputBucket":"Use this, or use ContentConfig:Bucket plus
      ThumbnailConfig:Bucket",
   "Role":"IAM role ARN",
   "AwsKmsKeyArn":"AWS-KMS key id",
   "Notifications":{
      "Progressing":"SNS topic to notify when
         when Elastic Transcoder has started to process the job",
      "Completed":"SNS topic to notify when
         Elastic Transcoder has finished processing the job",
      "Warning":"SNS topic to notify when
         when Elastic Transcoder encounters a warning condition",
      "Error":"SNS topic to notify when
         Elastic Transcoder encounters an error condition"
   },
   "ContentConfig":{
      "Bucket":"Use this plus ThumbnailConfig:Bucket,
         or use OutputBucket",
      "Permissions":[
         {
            "GranteeType":"Canonical|Email|Group",
            "Grantee":"AWS user ID or CloudFront origin access identity"|
                      "registered email address for AWS account|
                      AllUsers|AuthenticatedUsers|LogDelivery",
            "Access":[
               "Read|ReadAcp|WriteAcp|FullControl",
               ...
            ]
         },
         {...}
      ],
      "StorageClass":"Standard|ReducedRedundancy"
   },
   "ThumbnailConfig":{
      "Bucket":"Use this plus ContentConfig:Bucket,
         or use OutputBucket",
      "Permissions":[
         {
            "Access":[
               "Read|ReadAcp|WriteAcp|FullControl",
               ...
            ],
            "Grantee":"AWS user ID or CloudFront origin access identity"|
```

```
                    "registered email address for AWS account|
                    AllUsers|AuthenticatedUsers|LogDelivery",
            "GranteeType":"Canonical|Email|Group"
         },
         {...}
      ],
      "StorageClass":"Standard|ReducedRedundancy"
   },
   "Status":"Active|Paused",
   "Warnings (p. 163)": [
      {
         "Code": "6000|6001|6002|6003|6004|6005|6006|6007|6008",
         "Message": "The code message"
      },
      {...}
   ]
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

When you create a pipeline, Elastic Transcoder returns the values that you specified in the request. For more information, see Request Body (p. 158).

Elastic Transcoder will also include the following warnings, if applicable:

Warnings

When you create a pipeline that uses resources in other regions, Elastic Transcoder returns one or more warnings. Your pipeline is still created, but might have increased processing times and incur cross-regional charges. The warnings are in the following format:

Code
Message — the message associated with the warning code.

The following is a list of valid warning codes and their messages:

6000
The input bucket and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the input bucket and the pipeline.

6001
The ContentConfig bucket and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the ContentConfig bucket and the pipeline.

6002
The ThumbnailConfig bucket and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the ThumbnailConfig bucket and the pipeline.

6003
The SNS notification topic for progressing events and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease

processing time and prevent cross-regional charges, use the same region for the SNS notification topic and the pipeline.

6004
The SNS notification topic for warning events and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the SNS notification topic and the pipeline.

6005
The SNS notification topic for completion events and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the SNS notification topic and the pipeline.

6006
The SNS notification topic for error events and the pipeline are in different regions, which increases processing time for jobs in the pipeline and can incur additional charges. To decrease processing time and prevent cross-regional charges, use the same region for the SNS notification topic and the pipeline.

6007
The AWS KMS key and ContentConfig bucket specified for this pipeline are in different regions, which causes outputs using S3-AWS-KMS encryption mode to fail. To use S3-AWS-KMS encryption mode, use the same region for the KMS key and the ContentConfig bucket.

6008
The AWS KMS key and ThumbnailConfig bucket specified for this pipeline are in different regions, which causes outputs using S3-AWS-KMS encryption mode to fail. To use S3-AWS-KMS encryption mode, use the same region for the KMS key and the ThumbnailConfig bucket.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

The following example request updates the pipeline that has the Id `1111111111111-abcde1`.

Sample Request

```
PUT /2012-09-25/pipelines/1111111111111-abcde1 HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
   "Id":"1111111111111-abcde1"
   "Name":"Default",
   "InputBucket":"salesoffice.example.com-source",
   "OutputBucket":"salesoffice.example.com-output",
   "Role":"arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
```

```
   "AwsKmsKeyArn":"base64 encoded key from KMS",
   "Notifications":{
      "Progressing":"",
      "Completed":"",
      "Warning":"",
      "Error":"arn:aws:sns:us-east-1:111222333444:ET_Errors"
   },
   "ContentConfig":{
      "Bucket":"salesoffice.example.com-public-promos",
      "Permissions":[
         {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
               "FullControl"
            ]
         }
      ],
      "StorageClass":"Standard"
   },
   "ThumbnailConfig":{
      "Bucket":"salesoffice.example.com-public-promos-thumbnails",
      "Permissions":[
         {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
               "FullControl"
            ]
         }
      ],
      "StorageClass":"ReducedRedundancy"
   },
   "Status":"Active"
}
```

Sample Response

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT
{
   "Id":"1111111111111-abcde1",
   "Name":"Default",
   "InputBucket":"salesoffice.example.com-source",
   "OutputBucket":"salesoffice.example.com-output",
   "Role":"arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
   "AwsKmsKeyArn":"base64 encoded key from KMS",
   "Notifications":{
      "Progressing":"",
      "Completed":"",
      "Warning":"",
      "Error":"arn:aws:sns:us-east-1:111222333444:ET_Errors"
   },
```

```
   "ContentConfig":{
      "Bucket":"salesoffice.example.com-public-promos",
      "Permissions":[
         {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
               "FullControl"
            ]
         }
      ],
      "StorageClass":"Standard"
   },
   "ThumbnailConfig":{
      "Bucket":"salesoffice.example.com-public-promos-thumbnails",
      "Permissions":[
         {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
               "FullControl"
            ]
         }
      ],
      "StorageClass":"ReducedRedundancy"
   },
   "Status":"Active",
   "Warnings": [
      {
         "Code": "6000",
         "Message": "The input bucket and the pipeline are in different
            regions, which increases processing time for jobs in the
            pipeline and can incur additional charges. To decrease
            processing time and prevent cross-regional charges, use the
            same region for the input bucket and the pipeline."
      },
      {...}
   ]
}
```

Update Pipeline Status

Topics

- Description (p. 167)
- Requests (p. 167)
- Responses (p. 168)
- Errors (p. 168)
- Example (p. 168)

Description

To pause or reactivate a pipeline, so the pipeline stops or restarts processing jobs, update the status for the pipeline. Send a POST request to the `/2012-09-25/pipelines/`*`pipelineId`*`/status` resource.

Changing the pipeline status is useful if you want to cancel one or more jobs. You can't cancel jobs after Elastic Transcoder has started processing them; if you pause the pipeline to which you submitted the jobs, you have more time to get the job IDs for the jobs that you want to cancel, and to send a `Delete Job` request.

Requests

Syntax

```
POST /2012-09-25/pipelines/pipelineId/status HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
   "Status":"new status for the pipeline"
}
```

Request Parameters

This operation takes the following request parameter.

pipelineId
The identifier of the pipeline that you want to pause or reactivate.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

The JSON string in the request body contains the following object.

Status

The new status of the pipeline:

- `Active`: Enable the pipeline, so it starts processing jobs.
- `Paused`: Disable the pipeline, so it stops processing jobs.

Responses

Syntax

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Id":"ID for the pipeline",
   "Status":"new status for the pipeline"
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

When you update status for a pipeline, Elastic Transcoder returns the values that you specified in the request. For more information, see Request Body (p. 167).

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Example

The following example request enables the pipeline that has the ID `1111111111111-abcde1`.

Sample Request

```
POST /2012-09-25/pipelines/1111111111111-abcde1/status HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
```

```
   "Status":"Active"
}
```

Sample Response

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT
{
   "Id":"1111111111111-abcde1",
   "Status":"Active"
}
```

Update Pipeline Notifications

Topics

- Description (p. 170)
- Requests (p. 170)
- Responses (p. 171)
- Errors (p. 172)
- Examples (p. 172)

Description

To update only Amazon Simple Notification Service (Amazon SNS) notifications for a pipeline, send a POST request to the `/2012-09-25/pipelines/pipelineId/notifications` resource.

> **Important**
> When you change notifications, your changes take effect immediately. Jobs that you have already submitted and that Elastic Transcoder has not started to process are affected in addition to jobs that you submit after you change notifications.

Requests

Syntax

```
POST /2012-09-25/pipelines/pipelineId/notifications HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
   "Id":"pipelineId",
   "Notifications":{
      "Progressing":"SNS topic to notify when
         Elastic Transcoder has started to process the job",
      "Completed":"SNS topic to notify when
         Elastic Transcoder has finished processing the job",
      "Warning":"SNS topic to notify when
       Elastic Transcoder returns a warning",
      "Error":"SNS topic to notify when
         Elastic Transcoder returns an error"
   }
}
```

Request Parameters

This operation takes the following request parameter.

pipelineId
The identifier of the pipeline for which you want to change notification settings.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

The JSON string in the request body contains the following objects.

Id
The ID of the pipeline that you want to update.

Notifications:Progressing
The topic ARN for the Amazon Simple Notification Service (Amazon SNS) topic that you want to notify when Elastic Transcoder has started to process a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic. For more information, see Create a Topic in the *Amazon Simple Notification Service Developer Guide*.

> **Important**
> To receive notifications, you must also subscribe to the new topic in the Amazon SNS console.

Amazon SNS offers a variety of notification options, including the ability to send Amazon SNS messages to Amazon Simple Queue Service queues. For more information, see the Amazon Simple Notification Service Developer Guide.

Notifications:Completed
The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder has finished processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

Notifications:Warning
The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters a warning condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

Notifications:Error
The topic ARN for the Amazon SNS topic that you want to notify when Elastic Transcoder encounters an error condition while processing a job in this pipeline. This is the ARN that Amazon SNS returned when you created the topic.

Responses

Syntax

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT
{
   "Pipeline":{
      "Id":"ID for the new pipeline",
      "Notifications":{
        "Progressing":"SNS topic to notify when Elastic Transcoder has started
 to process the job",
         "Completed":"SNS topic to notify when Elastic Transcoder has finished
 processing the job",
         "Warning":"SNS topic to notify when Elastic Transcoder returns a
```

```
warning",
        "Error":"SNS topic to notify when Elastic Transcoder returns an error"

      }
   }
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

When you update notifications for a pipeline, Elastic Transcoder returns the values that you specified in the request. For more information, see Request Body (p. 171).

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

The following example request updates the notifications for a pipeline.

Sample Request

```
POST /2012-09-25/pipelines/1111111111111-abcde1/notifications HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
   "Id":"1111111111111-abcde1",
   "Notifications":{
      "Progressing":"",
      "Completed":"",
      "Warning":"",
      "Error":"arn:aws:sns:us-east-1:111222333444:ETS_Errors"
   }
}
```

Sample Response

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
```

```
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Id":"1111111111111-abcde1",
   "Notifications":{
      "Progressing":"",
      "Completed":"",
      "Warning":"",
      "Error":"arn:aws:sns:us-east-1:111222333444:ETS_Errors"
   }
}
```

Delete Pipeline

Topics

- Description (p. 174)
- Requests (p. 174)
- Responses (p. 175)
- Errors (p. 175)
- Example (p. 175)

Description

To delete a pipeline, send a DELETE request to the `/2012-09-25/pipelines/`*`pipelineId`* resource. You can only delete a pipeline that has never been used or that is not currently in use (doesn't contain any active jobs). If the pipeline is currently in use, `Delete Pipeline` returns an error.

Requests

Syntax

```
DELETE /2012-09-25/pipelines/pipelineId HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Request Parameters

This operation takes the following request parameter.

pipelineId
The identifier of the pipeline that you want to delete.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

This operation does not have a request body.

Responses

Syntax

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Success":"true"
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

The response body contains the following JSON object.

Success
If the pipeline is successfully deleted, the value of `Success` is `true`.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Example

The following example request deletes the pipeline 1111111111111-abcde1.

Sample Request

```
DELETE /2012-09-25/pipelines/1111111111111-abcde1 HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Sample Response

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
```

```
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Success":"true"
}
```

Test Role

Topics

- Description (p. 177)
- Requests (p. 177)
- Responses (p. 178)
- Errors (p. 178)
- Examples (p. 179)

Description

To test the settings for a pipeline to ensure that Elastic Transcoder can create and process jobs, send a POST request to the `/2012-09-25/roleTests` resource.

Requests

Syntax

```
POST /2012-09-25/roleTests HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
   "InputBucket":"Amazon S3 bucket that contains files to transcode",
   "OutputBucket":"Amazon S3 bucket in which to save transcoded files",
   "Role":"IAM ARN for the role to test",
   "Topics": [
      "ARN of SNS topic to test"
   ]
}
```

Request Parameters

This operation does not use request parameters.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

The JSON string in the request body contains the following objects.

InputBucket
The Amazon S3 bucket in which you saved the media files that you want to transcode. `Test Role` tries to read from this bucket.

OutputBucket
The Amazon S3 bucket in which you want Elastic Transcoder to save the transcoded files. `Test Role` tries to read from this bucket.

Role
The IAM Amazon Resource Name (ARN) for the role that you want Elastic Transcoder to use to transcode jobs. `Test Role` tries to assume the specified role.

Topics
The ARNs of one or more Amazon Simple Notification Service (Amazon SNS) topics to which you want `Test Role` to send test notifications. If you aren't using Amazon SNS notifications, you can specify an empty list.

Responses

Syntax

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Messages": [
      "error messages, if any"
   ],
   "Success": "true | false"
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

When you test settings for a pipeline, Elastic Transcoder returns the following values.

Messages
If the value of `Success` is `false`, `Messages` contains an array of one or more messages that explain which tests failed.

Success
If the operation is successful, this value is `true`; otherwise, the value is `false`.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

Sample Request

```
POST /2012-09-25/roleTests HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
   "InputBucket":"salesoffice.example.com-source",
   "OutputBucket":"salesoffice.example.com-public-promos",
   "Role":"arn:aws:iam::123456789012:role/transcode-service",
   "Topics":
      ["arn:aws:sns:us-east-1:111222333444:ETS_Errors",
       "arn:aws:sns:us-east-1:111222333444:ETS_Progressing"]
}
```

Sample Response

```
Status: 201 Created
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT
{
   "Messages":[
      "The role arn:aws:iam::123456789012:role/transcode-service does not have
 access to the bucket: salesoffice.example.com-source",
      "The role arn:aws:iam::123456789012:role/transcode-service does not have
 access to the topic: arn:aws:sns:us-east-1:111222333444:ETS_Errors"
   ],
   "Success": "false"
}
```

Job Operations

Topics

- Create Job (p. 181)
- List Jobs by Pipeline (p. 210)
- List Jobs by Status (p. 230)
- Read Job (p. 251)
- Cancel Job (p. 271)

Jobs do the work of transcoding. Each job converts one file into one format. For example, if you want to convert a media file into six different formats, you create six jobs.

This section describes operations that you can perform on jobs using the Elastic Transcoder API. For more information about jobs, including how to perform the same operations using the Elastic Transcoder console, see Working with Jobs (p. 17).

Create Job

Topics

- Description (p. 181)
- Requests (p. 181)
- Responses (p. 196)
- Errors (p. 204)
- Examples (p. 205)

Description

To create a job, send a POST request to the `/2012-09-25/jobs` resource. Jobs start as soon as you create them.

> **Note**
> You can configure Elastic Transcoder to notify you when the status of a job changes, including when Elastic Transcoder starts and finishes processing a job, and when Elastic Transcoder encounters a warning or error condition. For more information, see Create Pipeline (p. 128).

Requests

Syntax

```
POST /2012-09-25/jobs HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
   "Input":{
      "Key":"name of the file to transcode",
      "Encryption":{
         "Mode":"AES-CBC-PKCS7|AES-CTR|AES-GCM",
         "Key":"encrypted and base64-encoded decryption key",
         "KeyMd5":"base64-encoded key digest",
         "InitializationVector":"base64-encoded initialization vector"
      },
      "FrameRate":"auto|10|15|23.97|24|25|29.97|30|50|60",
      "Resolution":"auto",
      "AspectRatio":"auto|1:1|4:3|3:2|16:9",
      "Interlaced":"auto|true|false",
      "Container":"auto|3gp|aac|asf|avi|divx|flv|m4a|mkv|mov|mp3|
         mp4|mpeg|mpeg-ps|mpeg-ts|mxf|ogg|vob|wav|webm"
   },
   "OutputKeyPrefix":"prefix for file names in Amazon S3 bucket",
   "Outputs":[
      {
         "Key":"name of the transcoded file",
```

```
"Encryption":{
   "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|AES-CTR|
      AES-GCM",
   "Key":"encrypted and base64-encoded encryption key",
   "KeyMd5":"base64-encoded key digest",
   "InitializationVector":"base64-encoded initialization vector"
},
"ThumbnailPattern":""|"pattern",
"Rotate":"auto|0|90|180|270",
"PresetId":"preset to use for the job",
"SegmentDuration":"[1,60]",
"Watermarks":[
   {
      "InputKey":"name of the .png or .jpg file",
      "Encryption":{
         "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
            AES-CTR|AES-GCM",
         "Key":"encrypted and base64-encoded encryption key",
         "KeyMd5":"base64-encoded key digest",
         "InitializationVector":"base64-encoded initialization
            vector"
      },
      "PresetWatermarkId":"value of Video:Watermarks:Id in preset"
   },
   {...}
],
"AlbumArt":[
   {
      "AlbumArtMerge":"Replace|Prepend|Append|Fallback",
      "AlbumArtArtwork":"can be empty, but not null":[
         {
            "AlbumArtInputKey":"name of the file to use as album
               art",
            "Encryption":{
               "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                  AES-CTR|AES-GCM",
               "Key":"encrypted and base64-encoded encryption key",
               "KeyMd5":"base64-encoded key digest",
               "InitializationVector":"base64-encoded
                  initialization vector"
            },
            "AlbumArtMaxWidth":"maximum width of output album art
               in pixels",
            "AlbumArtMaxHeight":"maximum height of output album
               art in pixels",
            "AlbumArtSizingPolicy":"Fit|Fill|Stretch|Keep|
               ShrinkToFit|ShrinkToFill",
            "AlbumArtPaddingPolicy":"Pad|NoPad",
            "AlbumArtFormat":"jpg|png"
         },
         {...}
      ]
   },
   {...}
],
"Composition":[
   {
      "TimeSpan":
```

```
                {
                   "StartTime":"starting place of the clip, in
                      HH:mm:ss.SSS or sssss.SSS",
                   "Duration":"duration of the clip, in HH:mm:ss.SSS
                      or sssss.SSS"
                }
            }
         ],
         "Captions":{
            "MergePolicy":"MergeOverride|MergeRetain|Override",
            "CaptionSources":[
               {
                  "Key":"name of the input caption file",
                  "Encryption":{
                     "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                        AES-CTR|AES-GCM",
                     "Key":"encrypted and base64-encoded encryption key",
                     "KeyMd5":"base64-encoded key digest",
                     "InitializationVector":"base64-encoded
                        initialization vector"
                  },
                  "Language":"language of the input caption file",
                  "TimeOffset":"starting place of the captions, in
                     either [-+]SS.sss or [-+]HH:mm:SS.ss",
                  "Label":"label for the caption"
               },
               {...}
            ],
            "CaptionFormats":[
               {
                  "Format":"dfxp|mov-text|scc|srt|webvtt",
                  "Pattern":"myCaption/file-language",
                  "Encryption":{
                     "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                        AES-CTR|AES-GCM",
                     "Key":"encrypted and base64-encoded encryption key",
                     "KeyMd5":"base64-encoded key digest",
                     "InitializationVector":"base64-encoded
                        initialization vector"
                  }
               },
               {...}
            ]
         }
      },
      {...}
   ],
   "Playlists":[
      {
         "Format":"HLSv3|HLSv4|Smooth",
         "Name":"name",
         "OutputKeys":[
            "Outputs:Key to include in this playlist",
            ...
         ],
         "HlsContentProtection":{
             "Method":"aes-128",
             "Key":"encrypted and base64-encoded protection key",
```

```
            "KeyMd5":"base64-encoded key digest",
            "InitializationVector":"base64-encoded
                      initialization vector",
            "LicenseAcquisitionUrl":"license aquisition url",
            "KeyStoragePolicy":"NoStore|WithVariantPlaylists"
         }
      },
      {...}
   ],
   "UserMetadata":
      {
         "Key":"Value",
         "Second user metadata key":"Second user metadata value"
      },
   "PipelineId":"pipeline to use for transcoding"
}
```

Request Parameters

This operation does not use request parameters.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

The JSON string in the request body contains the following objects.

Input
Information about the file that you're transcoding.

Input:Key
The name of the file that you want to transcode. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/lasagna.mpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

(Optional) Input:Encryption
The encryption settings, if any, that are used for decrypting your input files. If your input file is encrypted, you must specify the mode that Elastic Transcoder will use to decrypt your file.

(Required for Encryption) Input:Encryption:Mode
The specific encryption mode that you want Elastic Transcoder to use when decrypting your files.

Elastic Transcoder supports the following options:

- **Amazon S3 Server-Side Encryption:** Amazon S3 handles the encryption and decryption of your files. As long as Elastic Transcoder has access permissions to your Amazon S3 bucket, you don't need to take any action.

  For more information, see Protecting Data Using Server-Side Encryption in the *Amazon Simple Storage Service Developer Guide*.
- **Server-Side Encryption Using Customer-Provided Keys:** Elastic Transcoder supports three types of encryption using customer-provided keys:
  - **AES-CBC-PKCS7:** A padded cipher-block mode of operation.
  - **AES-CTR:** AES Counter Mode.

- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

If you chose one of the AES-encryption modes, you must also specify the following three values (all three must be base64-encoded):

- **Encryption Key**
- **Encryption Key MD5**
- **Encryption Initialization Vector**

(Optional) Input:Encryption:Key
The data encryption key used to encrypt your file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

The key must also be encrypted by using AWS KMS. For more information, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Input:Encryption:KeyMd5
The MD5 digest of the key used to encrypt your input file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

(Optional) Input:Encryption:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

For more information, go to Initialization Vector.

(Optional) Input:FrameRate
The frame rate of the input file. If you want Elastic Transcoder to automatically detect the frame rate of the input file, specify `auto`. If you want to specify the frame rate for the input file, enter one of the following values:

`10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `50`, `60`

The default value is `auto`.

(Optional) Input:Resolution
The resolution, in pixels, of the input file. This value must be `auto`, which causes Elastic Transcoder to automatically detect the resolution of the input file.

(Optional) Input:AspectRatio
The aspect ratio of the input file. If you want Elastic Transcoder to automatically detect the aspect ratio of the input file, specify `auto`. If you want to specify the aspect ratio for the output file, enter one of the following values:

`1:1`, `4:3`, `3:2`, `16:9`

The default value is `auto`.

(Optional) Input:Interlaced
Whether the input file is interlaced. If you want Elastic Transcoder to automatically detect whether the input file is interlaced, specify `auto`. If you want to specify whether the input file is interlaced, enter one of the following values:

`true`, `false`

The default value is `auto`.

(Optional) Input:Container

The container type for the input file. If you want Elastic Transcoder to automatically detect the container type of the input file, specify `auto`. If you want to specify the container type for the input file, enter one of the following values:

`3gp`, `aac`, `asf`, `avi`, `divx`, `flv`, `m4a`, `mkv`, `mov`, `mp3`, `mp4`, `mpeg`, `mpeg-ps`, `mpeg-ts`, `mxf`, `ogg`, `vob`, `wav`, `webm`

OutputKeyPrefix

The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, and playlists. If you specify a value, it must contain a `/` somewhere after the first character, which simplifies Amazon S3 file management.

Outputs

Information about the output files. You can create a maximum of 30 outputs per job. If you specify more than one output for a job, Elastic Transcoder creates the files for each output in the order in which you specify them in the job.

> **Note**
> In early versions of Elastic Transcoder you could create just one output per job, so the object name was `Output`. The `Output` syntax still works, but we recommend that you use the `Outputs` syntax for all jobs, even when you want Elastic Transcoder to transcode a file into only one format. Do not use both the `Outputs` and `Output` syntaxes in the same request.

Outputs:Key

The name that you want Elastic Transcoder to assign to the transcoded file and playlist. Elastic Transcoder saves the file or files in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that you specify in `PipelineId`.

If the bucket already contains a file that has the specified name, the output fails. In the `Create Job` response, the value of `Outputs:Status` for that output will be `Error`, as will the final value of Status for the job. However, other outputs in the same job may succeed.

The format for file names depends the container type and whether the segment duration is set. If the container type is not `ts` or the segment duration is not provided, the name of the output file is a concatenation of `OutputKeyPrefix` and `Key`.

If the container type is `ts` and segment duration is provided, Elastic Transcoder uses the value of `Key` to name both the playlist for the output and the `.ts` files:

- **Playlist:**
  - **HLSv3:** The file name is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `.m3u8`:

    `OutputKeyPrefix``Key`*`.m3u8`*
  - **HLSv4:** The file name is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `_v4.m3u8`. Video outputs create a second file with a file name that is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `_iframe.m3u8`:

    `OutputKeyPrefix``Key`*`_v4.m3u8`*

    `OutputKeyPrefix``Key`*`_iframe.m3u8`*
- **Segment (.ts) files:**
  - **HLSv3:** The file name is a concatenation of `OutputKeyPrefix` and `Key`, plus a five-digit sequential counter beginning with `00000`, and the file name extension `.ts`:

    `OutputKeyPrefix``Key`*`00000.ts`*
  - **HLSv4:** The file name is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `.ts`:

    `OutputKeyPrefix``Key`*`.ts`*

If a segmented ts output is not included in a master playlist, Elastic Transcoder treats the output as `HLSv3`.

> **Note**
> Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for `HLSv3` playlists, `_v4.m3u8` and `_iframe.m3u8` for `HLSv4` playlists). If you include a file extension in `Output Key`, the file name will have two extensions.

`OutputKeyPrefix` groups all of the files for a job together in your Amazon S3 bucket. If you want to group the files for each output within a job, you can include a prefix in the value of `Key`, for example:

`OutputKeyPrefix``iPhone/Key`*`00000.ts`*

`OutputKeyPrefix``KindleFireHD/Key`*`00000.ts`*

(Optional) Outputs:Encryption
The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

(Required for Encryption) Outputs:Encryption:Mode
The specific server-side encryption mode that you want Elastic Transcoder to use when encrypting your output files individually. Elastic Transcoder supports the following **Encryption Mode** options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.

  For more information, see Protecting Data Using Server-Side Encryption in the *Amazon Simple Storage Service Developer Guide*.
- **S3-AWS-KMS:** Amazon S3 calls AWS KMS, which creates and manages the keys that are used for encrypting your files. If you specify **S3-AWS-KMS** and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

  For more information, see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys in the *Amazon Simple Storage Service Developer Guide*.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

If you chose one of the AES-encryption modes, you must also specify the following three values (all three must be base64-encoded):

- **Encryption Key**
- **Encryption Key MD5**
- **Encryption Initialization Vector**

If you chose one of the AES-encryption modes, and you want Elastic Transcoder to generate a `128`-bit bit AES encryption key for you, do not specify values for the **Encryption Key**, **Encryption Key MD5**, or **Encryption Initialization Vector**.

> **Important**
> For the AES modes, your media-specific private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to decrypt your data.

(Optional) Outputs:Encryption:Key
If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

For more information about encrypting your key with AWS KMS, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Outputs:Encryption:KeyMd5
The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

(Optional) Outputs:Encryption:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

For more information, go to Initialization Vector.

(Optional, Sidecar only) Outputs:ThumbnailPattern
Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify `""`.

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **`{count}` (Required):** If you want to create thumbnails, you must include `{count}` in the `ThumbnailPattern` object. Wherever you specify `{count}`, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file.

  Important
  If you specify a literal value and/or `{resolution}` but you omit `{count}`, Elastic Transcoder returns a validation error and does not create the job.

- **(Optional) Literal values:** You can specify literal values anywhere in the `ThumbnailPattern` object, for example, as a file name prefix or as a delimiter between `{resolution}` and `{count}`.
- **(Optional) `{resolution}`:** If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object.

When creating thumbnails, Elastic Transcoder automatically saves the files in the format (`.jpg` or `.png`) that appears in the preset that you specified in `PresetId`. Elastic Transcoder also appends the applicable file name extension.

As with `Outputs:Key`, you can include a prefix in `ThumbnailPattern` that groups the applicable files together, for example, all of the thumbnails for one video in one format, or all of the thumbnails with the corresponding output file.

(Optional) Outputs:Rotate
The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Valid values include:

`auto`, `0`, `90`, `180`, `270`

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

Outputs:PresetId
The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset

that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset.

Note
If you created any presets before AAC profiles were added, Elastic Transcoder uses the AAC-LC profile for those presets.

For a list of system presets, see System Presets (p. 70) (You can also get these IDs using List Presets (p. 296).)

(Fragmented MP4/MPEG-TS Outputs Only) Outputs:SegmentDuration
If you specify a preset for the current output for which the value of `Container` is either **ts** (MPEG-TS) or **fmp4** (Fragmented MP4), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of:

```
total length/SegmentDuration
```

Elastic Transcoder creates an output-specific playlist for each `HLS` output that you specify in `OutputKeys`. To add an output to a master playlist for this job, include it in Outputs in Master Playlist.

Elastic Transcoder applies this segmenting to any captions associated with the output video.

For more information, see HTTP Live Streaming.

(Video Only) Outputs:Watermarks
Information about the watermarks that you want Elastic Transcoder to add to the video or artwork during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Outputs:PresetId` for the current output.

Watermarks are added to the output file in the sequence in which you list them in the job output—the first watermark in the list is added to the output file first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you list in `Outputs:Watermarks` will cover the first one, the third one will cover the second, and the fourth one will cover the third.

For more information about watermarks, see Watermarks (p. 69).

(Video Only) Outputs:Watermarks:InputKey
The name of the `.png` or `.jpg` file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `logos/128x64.png`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

(Video Only) Outputs:Watermarks:PresetWatermarkId
The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the file during transcoding. The settings are in the preset specified by `Outputs:PresetId` for the current output. In that preset, the value of `Watermarks:Id` tells Elastic Transcoder which settings to use.

(MP3/MP4 Only) Outputs:AlbumArt
The album art to be associated with the output file, if any.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `MergePolicy` to `Replace` and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `MergePolicy` to `Prepend`, `Append`, or `Fallback`, and use an empty `Artwork` array.

Note
Album Art is available only for containers of type `mp3` or `mp4`.

(MP3/MP4 Only) Outputs:AlbumArt:MergePolicy
A policy that determines how Elastic Transcoder handles the existence of multiple album artwork files.

- **Replace:** The specified album art replaces any existing album art.
- **Prepend:** The specified album art is placed in front of any existing album art.
- **Append:** The specified album art is placed after any existing album art.
- **Fallback:** If the input file contains artwork, Elastic Transcoder uses that artwork for the output. If the input does not contain artwork, Elastic Transcoder uses the specified album art file.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork
The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:InputKey
The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:MaxWidth
The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:MaxHeight
The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:SizingPolicy
A value that controls scaling of the output album art:

- **Fit:** Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill:** Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it to the dimension (if any) that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep:** Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit:** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **ShrinkToFill:** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.

The following table shows possible effects of `SizingPolicy` settings on the output album art:

SizingPolicy	Output Album Art May Be Scaled Up	Output Album Art May Be Padded When Padding Policy Is "Pad"	Output Album Art May Have a Different Pixel Aspect Ratio than Input Album Art	Output Album Art May Be Cropped
Fit	Yes	Yes		
Fill	Yes			Yes
Stretch	Yes		Yes	
Keep	Yes	Yes		Yes
ShrinkToFit		Yes		
ShrinkToFill		Yes		Yes

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:PaddingPolicy
When you set `PaddingPolicy` to `Pad`, Elastic Transcoder might add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`. For more information, see the table at `AlbumArt:Art:SizingPolicy`.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:AlbumArtFormat
The format of album art, if any. Valid formats are `jpg` and `png`.

Outputs:Composition
Using a composition, you can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The `Composition` object contains settings for the clips that make up an output file.

Currently, you can specify settings for only a single clip per output file. The `Composition` object cannot be null. All jobs in a playlist must have the same clip settings.

Outputs:Composition:TimeSpan
Settings that determine when a clip begins and how long it lasts.

Outputs:Composition:TimeSpan:StartTime
The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

Outputs:Composition:TimeSpan:Duration
The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file , Elastic Transcoder transcodes the file and returns a warning message.

(Video Only) Outputs:Captions
You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file.

  Valid input values include: `CEA-608 (EIA-608`, first non-empty channel only), `CEA-708 (EIA-708`, first non-empty channel only), and `mov-text`

Valid outputs include: `mov-text`

Elastic Transcoder supports a maximum of one embedded format per output.

- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file.

  Valid input values include: `dfxp` (first div element only), `ebu-tt`, `scc`, `smpt`, `srt`, `ttml` (first div element only), and `webvtt`

  Valid outputs include: `dfxp` (first div element only), `scc`, `srt`, and `webvtt`.

If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to null. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a null `CaptionSources` array.

For more information about embedded files, see the Subtitle (caption) Wikipedia page.

For more information about sidecar files, see the Metadata Platform and Sidecar file Wikipedia pages.

(Video Only) Outputs:Captions:MergePolicy
A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If **CaptionSources** is empty, Elastic Transcoder omits all sidecar captions from the output files.
- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`.

`MergePolicy` cannot be null.

(Video/Sidecar Only, Optional) Outputs:Captions:CaptionSources
Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

(Video Only) Outputs:Captions:CaptionSources:Key
The name of the sidecar caption file that you want Elastic Transcoder to include with the outputs.

(Video Only) Outputs:Captions:CaptionSources:Language
A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information about ISO language codes, see List of ISO 639-1 codes.

(Video Only, Optional) Outputs:Captions:CaptionSources:TimeOffset
For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the TimeOffset in the form [+-]SS.sss or [+-]HH:mm:SS.ss.

(Video Only, Optional) Outputs:Captions:CaptionSources:Label
The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

(Video Only) Outputs:Captions:CaptionFormats
The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

(Video Only) Outputs:Captions:CaptionFormats:Format
The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption for this output.

- **Embedded Caption Formats:**

Container Type	Supported Embedded types
MP4	mov-text
Non-MP4 containers	Embedded captions are not supported

Elastic Transcoder supports a maximum of one embedded format per output.

- **Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

Container Type	Supported Sidecar types
fmp4	dfxp
Non-fmp4 containers	All sidecar types

Note
`fmp4` captions have an extension of `.ismt`

(Video/Sidecar Only) Outputs:Captions:CaptionFormats:Pattern
The prefix for caption filenames, in the form *`description`*`-{language}`, where:

- *`description`* is a description of the video.
- `{language}` is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include `{language}` in the file name pattern, Elastic Transcoder automatically appends "`{language}`" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be `Sydney-en-sunrise00000.srt`.

(Optional) Outputs:UserMetadata
User-defined metadata that you want to associate with an Elastic Transcoder job. You specify metadata in `key/value` pairs. You can use the `key/value` pairs to track details about a file, for example, `Season 1: Episode 3`.

You can add up to 10 key/value pairs to each job. Elastic Transcoder does not guarantee that `key/value` pairs are returned in the same order in which you specify them.

Outputs:UserMetadata:Key
The key of the metadata `key/value` pair that you want returned with the output file. Each key must be a unique string between `1-128` characters, and must use only characters from the following list:

- `0-9`

- A-Z and a-z
- Space
- The following symbols: _.:/=+-%@

You can use keys as a numbering system for organizing your metadata, for storing an extra 128 characters of metadata, or for labeling the metadata stored in the **value**. If you want to use only value metadata, you can put throw-away strings in your keys such as key1, and ignore the keys when you retrieve your metadata from Elastic Transcoder.

> **Important**
> You must specify unique strings for all of the keys in a job. If the same string is used for more than one key in a job, Elastic Transcoder returns only one of the key/value pairs that use that key. There is no way to guarantee which value is returned.

Outputs:UserMetadata:Value

The value of the metadata key/value pair that you want returned with your job. Each value must be a string between 0-256 characters, and must use only characters from the following list:

- 0-9
- A-Z and a-z
- Space
- The following symbols: _.:/=+-%@

(Fragmented MP4/MPEG-TS Outputs Only) Playlists

If you specify a preset in PresetId for which the value of Container is either **ts** (MPEG-TS) or **fmp4** (Fragmented MP4), Playlists contains information about the master playlists that you want Elastic Transcoder to create.

We recommend that you create at most one master playlist per playlist format. The maximum number of master playlists in a job is 30.

Playlists:Format

The format of the output playlist. Valid formats include HLSv3, HLSv4, and Smooth.

Playlists:Name

The name that you want Elastic Transcoder to assign to a master playlist, for example, nyc-vacation.m3u8. If the name includes a / character, the section of the name before the / must be identical for all Name objects. If you create more than one master playlist, the values of all Name objects must be unique.

> **Note**
> Elastic Transcoder automatically appends the relevant file extension to the file name (.m3u8 for HLSv3 playlists, _v4.m3u8 and _iframe.m3u8 for HLSv4 playlists, and .ism and .ismc for Smooth playlists). If you include a file extension in Name, the file name will have two extensions.

Any segment duration settings, clip settings, or caption settings must be the same for all outputs in the playlist. For Smooth playlists, the Audio:Profile, Video:Profile, and Video:FrameRate to Video:KeyframesMaxDist ratio must be the same for all outputs. For more information, see KeyframesMaxDist (p. 282).

Playlists:OutputKeys

For each output in this job that you want to include in a master playlist, the value of the Outputs:Key object. If you include more than one output in a playlist, the value of SegmentDuration for all of the outputs must be the same.

For HLSv4 master playlists, Elastic Transcoder chooses which combinations of audio and video inputs will be linked in the output playlists. The first audio and video inputs will be linked and rendered as the default playback experience, allowing you to choose your preferred playback default. For other individual playlists in the master playlist, Elastic Transcoder chooses which audio and video bit rate combinations will provide the best playback.

(Optional) Outputs:Playlists:HlsContentProtection
The HLS content protection settings, if any, that you want Elastic Transcoder to apply to your output files. If you want to use HLS content protection do not specify encryption settings for the output file or captions. HLS content protection encrypts each segment of a file so that they can be streamed encrypted and only decrypted on playback, while the output file and caption encryptions encrypt the file all at once. Elastic Transcoder does not support files that are encrypted both ways.

Outputs:Playlists:HlsContentProtection:Method
The content protection method for your output. The only valid value is:

`aes-128`.

This value will be written into the `method` attribute of the `EXT-X-KEY` metadata tag in the output playlist.

(Optional) Outputs:Playlists:HlsContentProtection:Key
If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

For more information about encrypting your key with AWS KMS, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Outputs:Playlists:HlsContentProtection:KeyMd5
The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

(Optional) Outputs:Playlists:HlsContentProtection:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

For more information, go to Initialization Vector.

Outputs:Playlists:HlsContentProtection:LicenseAcquisitionUrl
The location of the license key required to decrypt your HLS playlist. The URL must be an absolute path, and is referenced in the URI attribute of the EXT-X-KEY metadata tag in the playlist file. For example:

```
https://www.example.com/exampleKey/
```

Outputs:Playlists:HlsContentProtection:KeyStoragePolicy
Specify whether you want Elastic Transcoder to write your HLS license key to an Amazon S3 bucket. If you choose `WithVariantPlaylists`, Elastic Transcoder will write your encrypted key into the same Amazon S3 bucket as the associated playlist.

> **Important**
> If you chose `NoStore`, Elastic Transcoder will not store your key. You are responsible for storing it and providing it to your users by giving them the **License Acquisition URL** where you are storing the key.

PipelineId
The value of the `Id` object for the pipeline that you want Elastic Transcoder to use for transcoding. The pipeline determines several settings, including the Amazon S3 bucket from which Elastic

Transcoder gets the files to transcode and the bucket into which Elastic Transcoder puts the transcoded files.

Responses

Syntax

```
Status: 201 Created
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Job":{
      "Id":"Id that Elastic Transcoder assigns to the job",
      "Input":{
         "Key":"name of the file to transcode",
         "Encryption":{
            "Mode":"AES-CBC-PKCS7|AES-CTR|AES-GCM",
            "Key":"encrypted and base64-encoded decryption key",
            "KeyMd5":"base64-encoded key digest",
            "InitializationVector":"base64-encoded initialization vector"
         },
         "FrameRate":"auto|10|15|23.97|24|25|29.97|30|50|60",
         "Resolution":"auto|width in pixelsxheight in pixels",
         "AspectRatio":"auto|1:1|4:3|3:2|16:9",
         "Interlaced":"auto|true|false",
         "Container":"auto|3gp|aac|asf|avi|divx|flv|m4a|mkv|mov|mp3|
            mp4|mpeg|mpeg-ps|mpeg-ts|mxf|ogg|vob|wav|webm"
      },
      "OutputKeyPrefix":"prefix for file names in Amazon S3 bucket",
      "Outputs":[
         {
            "Id":"sequential counter",
            "Key":"name of the transcoded file",
            "Encryption":{
               "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                  AES-CTR|AES-GCM",
               "Key":"encrypted and base64-encoded encryption key",
               "KeyMd5":"base64-encoded key digest",
               "InitializationVector":"base64-encoded initialization
                  vector"
            },
            "ThumbnailPattern":""|"pattern",
            "Rotate":"auto|0|90|180|270",
            "PresetId":"preset to use for the job",
            "SegmentDuration":"[1,60]",
            "Watermarks":[
               {
                  "InputKey":"name of the .png or .jpg file",
                  "Encryption":{
                     "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                        AES-CTR|AES-GCM",
                     "Key":"encrypted and base64-encoded encryption key",
                     "KeyMd5":"base64-encoded key digest",
```

```
            "InitializationVector":"base64-encoded initialization
               vector"
         },
         "PresetWatermarkId":"value of Video:Watermarks:Id in preset"

      },
      {...}
   ],
   "AlbumArt":[
      {
         "AlbumArtMerge":"Replace|Prepend|Append|Fallback",
         "AlbumArtArtwork":"can be empty, but not null":[
            {
               "AlbumArtInputKey":"name of the file to use as album
                  art",
               "Encryption":{
                  "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                     AES-CTR|AES-GCM",
                  "Key":"encrypted and base64-encoded encryption
                     key",
                  "KeyMd5":"base64-encoded key digest",
                  "InitializationVector":"base64-encoded
                     initialization vector"
               },
               "AlbumArtMaxWidth":"maximum width of output album art
                  in pixels",
               "AlbumArtMaxHeight":"maximum height of output album
                  art in pixels",
               "AlbumArtSizingPolicy":"Fit|Fill|Stretch|Keep|
                  ShrinkToFit|ShrinkToFill",
               "AlbumArtPaddingPolicy":"Pad|NoPad",
               "AlbumArtFormat":"jpg|png"
            },
            {...}
         ]
      },
      {...}
   ],
   "Composition":[
      {
         "TimeSpan":
            {
               "StartTime":"starting place of the clip, in
                  HH:mm:ss.SSS or sssss.SSS",
               "Duration":"duration of the clip, in HH:mm:ss.SSS
                  or sssss.SSS"
            }
      }
   ],
   "Duration":"duration in seconds",
   "Width":"width in pixels",
   "Height":"height in pixels",
   "Status":"Submitted|Progressing|Complete|Canceled|Error",
   "StatusDetail":"additional information about job status"
   "Captions":
      {
         "MergePolicy":"MergeOverride|MergeRetain|Override",
         "CaptionSources":[
```

```
                {
                   "Key":"name of the input caption file",
                   "Encryption":{
                      "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                         AES-CTR|AES-GCM",
                      "Key":"encrypted and base64-encoded encryption
                         key",
                      "KeyMd5":"base64-encoded key digest",
                      "InitializationVector":"base64-encoded
                         initialization vector"
                   },
                   "Language":"language of the input caption file",
                   "TimeOffset":"starting place of the captions, in
                      either [-+]SS.sss or [-+]HH:mm:SS.ss",
                   "Label":"label for the caption"
                },
                {...}
             ],
             "CaptionFormats":[
                {
                   "Format":"dfxp|mov-text|scc|srt|webvtt",
                   "Pattern":"myCaption/file-language",
                   "Encryption":{
                      "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                         AES-CTR|AES-GCM",
                      "Key":"encrypted and base64-encoded encryption
                         key",
                      "KeyMd5":"base64-encoded key digest",
                      "InitializationVector":"base64-encoded
                         initialization vector"
                   }
                },
                {...}
             ]
          },
          "AppliedColorSpaceConversion":"None|Bt601ToBt709|
             Bt709ToBt601"
       },
       {...}
    ],
    "Playlists":[
       {
          "Format":"HLSv3|HLSv4|Smooth",
          "Name":"name",
          "OutputKeys":[
             "Outputs:Key to include in this playlist",
             ...
          ],
          "HlsContentProtection":{
              "Method":"aes-128",
              "Key":"encrypted and base64-encoded protection key",
              "KeyMd5":"base64-encoded key digest",
              "InitializationVector":"base64-encoded
                     initialization vector",
              "LicenseAcquisitionUrl":"license aquisition url",
              "KeyStoragePolicy":"NoStore|WithVariantPlaylists"
          }
       },
```

```
         {...}
      ],
      "UserMetadata":
         {
            "key1":"First user metadata value",
            "key2":"Second user metadata value"
         },
      "PipelineId":"pipeline to add the job to",
      "Status":"Submitted|Progressing|Complete|Canceled|Error"
   }
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

When you create a job, Elastic Transcoder returns the values that you specified in the request. For more information, see Request Body (p. 184).

In addition, Elastic Transcoder returns the following values.

Id
The identifier that Elastic Transcoder assigned to the job. You use this value to get settings for the job or to delete the job.

Input
Information about the file that Elastic Transcoder transcoded. These are values that you specified when you created the job.

Outputs
Information about the output files. You can create a maximum of 30 outputs per job. If you specify more than one output for a job, Elastic Transcoder creates the files for each output in the order in which you specify them in the job. The `Outputs:Id` object identifies the position of an output in the sequence.

> **Note**
> In early versions of Elastic Transcoder, you could create just one output per job, so the object name was `Output`. The `Output` syntax still works, but we recommend that you use the `Outputs` syntax for all jobs, even when you want Elastic Transcoder to transcode a file into only one format. Do not use both the `Outputs` and `Output` syntaxes in the same request.

Outputs:Id
A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the `Output` syntax, this value is always `1`.

(Optional) Outputs:Encryption
The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

(Required for Encryption) Outputs:Encryption:Mode
The specific server-side encryption mode that you want Elastic Transcoder to use when encrypting your output files individually. Elastic Transcoder supports the following **Encryption Mode** options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.

  For more information, see Protecting Data Using Server-Side Encryption in the *Amazon Simple Storage Service Developer Guide*.

- **S3-AWS-KMS:** Amazon S3 calls AWS KMS, which creates and manages the keys that are used for encrypting your files. If you specify **S3-AWS-KMS** and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

  For more information, see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys in the *Amazon Simple Storage Service Developer Guide*.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

If you chose one of the AES-encryption modes, you must also specify the following three values (all three must be base64-encoded):

- **Encryption Key**
- **Encryption Key MD5**
- **Encryption Initialization Vector**

If you chose one of the AES-encryption modes, and you want Elastic Transcoder to generate a 128-bit bit AES encryption key for you, do not specify values for the **Encryption Key**, **Encryption Key MD5**, or **Encryption Initialization Vector**.

Important
For the AES modes, your media-specific private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to decrypt your data.

(Optional) Outputs:Encryption:Key
If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

128, 192, or 256.

For more information about encrypting your key with AWS KMS, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Outputs:Encryption:KeyMd5
The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

(Optional) Outputs:Encryption:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

For more information, go to Initialization Vector.

Outputs:Duration
Duration of the output file, in seconds.

Outputs:Width
Width of the output file, in pixels.

Outputs:Height
Height of the output file, in pixels.

Outputs:Status
The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
- When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.
- `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
- When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the terminal status for `Job:Status` is also `Error`.

The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

Outputs:StatusDetail
Information that further explains `Outputs:Status`.

Outputs:Composition
Using a composition, you can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The `Composition` object contains settings for the clips that make up an output file.

Currently, you can specify settings for only a single clip per output file. The `Composition` object cannot be null. All jobs in a playlist must have the same clip settings.

Outputs:Composition:TimeSpan
Settings that determine when a clip begins and how long it lasts.

Outputs:Composition:TimeSpan:StartTime
The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

Outputs:Composition:TimeSpan:Duration
The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file , Elastic Transcoder transcodes the file and returns a warning message.

(Video Only) Outputs:Captions
You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file.

  Valid input values include: `CEA-608 (EIA-608`, first non-empty channel only), `CEA-708 (EIA-708`, first non-empty channel only), and `mov-text`

  Valid outputs include: `mov-text`

  Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file.

Valid input values include: `dfxp` (first div element only), `ebu-tt`, `scc`, `smpt`, `srt`, `ttml` (first div element only), and `webvtt`

Valid outputs include: `dfxp` (first div element only), `scc`, `srt`, and `webvtt`.

If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to null. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a null `CaptionSources` array.

For more information about embedded files, see the Subtitle (caption) Wikipedia page.

For more information about sidecar files, see the Metadata Platform and Sidecar file Wikipedia pages.

(Video Only) Outputs:Captions:MergePolicy
A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If **CaptionSources** is empty, Elastic Transcoder omits all sidecar captions from the output files.
- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`.

`MergePolicy` cannot be null.

(Video Only, Optional) Outputs:Captions:CaptionSources
Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

(Video Only) Outputs:Captions:CaptionSources:Key
The name of the sidecar caption file that you want Elastic Transcoder to include with the outputs.

(Video Only) Outputs:Captions:CaptionSources:Language
A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information about ISO language codes, see List of ISO 639-1 codes.

(Video Only) Outputs:Captions:CaptionSources:TimeOffset
For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the TimeOffset in the form [+-]SS.sss or [+-]HH:mm:SS.ss.

(Video Only) Outputs:Captions:CaptionSources:Label
The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

(Video Only) Outputs:Captions:CaptionFormats
The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

(Video Only) Outputs:Captions:CaptionFormats:Format
The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption for this output.

- **Embedded Caption Formats:**

Container Type	Supported Embedded types
MP4	mov-text
Non-MP4 containers	Embedded captions are not supported

  Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

Container Type	Supported Sidecar types
fmp4	dfxp
Non-fmp4 containers	All sidecar types

Note
`fmp4` captions have an extension of `.ismt`

(Video Only) Outputs:Captions:CaptionFormats:Pattern
The prefix for caption filenames, in the form *`description`*`-{language}`, where:

- *`description`* is a description of the video.
- `{language}` is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include `{language}` in the file name pattern, Elastic Transcoder automatically appends "`{language}`" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be `Sydney-en-sunrise00000.srt`.

(Optional) Outputs:Playlists:HlsContentProtection
The HLS content protection settings, if any, that you want Elastic Transcoder to apply to your output files. If you want to use HLS content protection do not specify encryption settings for the output file or captions. HLS content protection encrypts each segment of a file so that they can be streamed encrypted and only decrypted on playback, while the output file and caption encryptions encrypt the file all at once. Elastic Transcoder does not support files that are encrypted both ways.

Outputs:Playlists:HlsContentProtection:Method
The content protection method for your output. The only valid value is:

`aes-128`.

This value will be written into the `method` attribute of the `EXT-X-KEY` metadata tag in the output playlist.

(Optional) Outputs:Playlists:HlsContentProtection:Key
If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

For more information about encrypting your key with AWS KMS, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Outputs:Playlists:HlsContentProtection:KeyMd5
The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

(Optional) Outputs:Playlists:HlsContentProtection:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

For more information, go to Initialization Vector.

Outputs:Playlists:HlsContentProtection:LicenseAcquisitionUrl
The location of the license key required to decrypt your HLS playlist. The URL must be an absolute path, and is referenced in the URI attribute of the EXT-X-KEY metadata tag in the playlist file. For example:

```
https://www.example.com/exampleKey/
```

Outputs:Playlists:HlsContentProtection:KeyStoragePolicy
Specify whether you want Elastic Transcoder to write your HLS license key to an Amazon S3 bucket. If you choose `WithVariantPlaylists`, Elastic Transcoder will write your encrypted key into the same Amazon S3 bucket as the associated playlist.

> **Important**
> If you chose `NoStore`, Elastic Transcoder will not store your key. You are responsible for storing it and providing it to your users by giving them the **License Acquisition URL** where you are storing the key.

Status
If you specified more than one output for the job, the status of the entire job. When Elastic Transcoder starts processing a job, the value of `Job:Status` changes to `Progressing` and doesn't change until Elastic Transcoder has finished processing all outputs. When processing is complete, `Job:Status` changes either to `Complete` or, if any of the outputs failed, to `Error`.

If you specified only one output for the job, `Job:Status` is the same as `Outputs:Status`.

The value of `Job:Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

(Automatic) Outputs:AppliedColorSpaceConversion
If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

For more information about `ColorSpaceConversionMode`, see ColorSpaceConversion.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Note
If a job fails with an `Access Denied` error, we recommend that you run the `Test Role` API action to determine what is causing the error. For more information, see Test Role (p. 177).

Examples

The following example request creates a job that has two outputs.

Sample Request

```
POST /2012-09-25/jobs HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
   "Input":{
      "Key":"recipes/lasagna.mp4",

      "FrameRate":"auto",
      "Resolution":"auto",
      "AspectRatio":"auto",
      "Interlaced":"auto",
      "Container":"mp4"
   },
   "OutputKeyPrefix":"recipes/",
   "Outputs":[
      {
         "Key":"mp4/lasagna-kindlefirehd.mp4",

         "ThumbnailPattern":"mp4/thumbnails/lasagna-{count}",
         "Rotate":"0",
         "PresetId":"1351620000000-100080",
         "Watermarks":[
            {
               "InputKey":"logo/128x64.png",
               "PresetWatermarkId":"company logo 128x64"

            }
         ],
         "Captions":{
            "MergePolicy":"MergeOverride",
            "CaptionSources":[
               {
                  "Key":"scc/lasagna-kindlefirehd.scc",

                  "Language":"en",
                  "Label":"English"
               },
               {
                  "Key":"srt/lasagna-kindlefirehd.srt",
```

```
                  "Language":"fr",
                  "TimeOffset":"1:00:00",
                  "Label":"French"
               }
            ],
            "CaptionFormats":[
               {
                  "Format":"scc",
                  "Pattern":"scc/lasagna-{language}"

               },
               {
                  "Format":"srt",
                  "Pattern":"srt/lasagna-{language}",

               }
            ]
         }
      },
      {
         "Key":"iphone/lasagna-1024k",
         "ThumbnailPattern":"iphone/th1024k/lasagna-{count}",

         "Rotate":"0",
         "PresetId":"1351620000000-987654",
         "SegmentDuration":"5"
      },
      {
         "Key":"iphone/lasagna-512k",
         "ThumbnailPattern":"iphone/th512k/lasagna-{count}",
         "Rotate":"0",
         "PresetId":"1351620000000-456789",
         "SegmentDuration":"5"
      },
   ],
   "Playlists": [
      {
         "Format": "HLSv3",
         "Name": "playlist-iPhone-lasagna.m3u8",
         "OutputKeys": [
            "iphone/lasagna-1024k",
            "iphone/lasagna-512k"
         ]
      }
   ],
   "UserMetadata":
      {
         "Food type":"Italian",
         "Cook book":"recipe notebook"
      },
   "PipelineId":"1111111111111-abcde1"
}
```

Sample Response

```
Status: 201 Created
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Job":{
      "Id":"3333333333333-abcde3",
      "Input":{
         "Key":"cooking/lasagna.mp4",

         "FrameRate":"auto",
         "Resolution":"auto",
         "AspectRatio":"auto",
         "Interlaced":"auto",
         "Container":"mp4"
      },
      "Outputs":[
         {
            "Id":"1",
            "Key":"mp4/lasagna-kindlefirehd.mp4",

            "ThumbnailPattern":"mp4/thumbnails/lasagna-{count}",
            "Rotate":"0",
            "PresetId":"1351620000000-100080",
            "Watermarks":[
               {
                  "InputKey":"logo/128x64.png",
                  "PresetWatermarkId":"company logo 128x64",

               }
            ],
            "Duration":"1003",
            "Width":"1280",
            "Height":"720",
            "Status":"Progressing",
            "StatusDetail":"",
            "Captions":{
               "MergePolicy":"MergeOverride",
               "CaptionSources":[
                  {
                     "Key":"scc/lasagna-kindlefirehd.scc",

                     "Language":"en",
                     "Label":"English"
                  },
                  {
                     "Key":"srt/lasagna-kindlefirehd.srt",

                     "Language":"fr",
                     "TimeOffset":"1:00:00",
                     "Label":"French"
                  }
               ],
```

```
               "CaptionFormats":[
                  {
                     "Format":"scc",
                     "Pattern":"scc/lasagna-{language}",

                  },
                  {
                     "Format":"srt",
                     "Pattern":"srt/lasagna-{language}",

                  },
                  {
                     "Format":"mov-text"
                  }
               ]
            }
         },
         {
            "Id":"2",
            "Key":"iphone/lasagna-1024k","ThumbnailPat
tern":"iphone/th1024k/lasagna-{count}",
            "Rotate":"0",
            "PresetId":"1351620000000-987654",
            "SegmentDuration":"5",
            "Duration":"1003",
            "Width":"1136",
            "Height":"640",
            "Status":"Progressing",
            "StatusDetail":""
         },
         {
            "Id":"3",
            "Key":"iphone/lasagna-512k",
            "ThumbnailPattern":"iphone/th512k/lasagna-{count}",
            "Rotate":"0",
            "PresetId":"1351620000000-456789",
            "SegmentDuration":"5",
            "Duration":"1003",
            "Width":"1136",
            "Height":"640",
            "Status":"Complete",
            "StatusDetail":"",
            "AppliedColorSpaceConversion":"None"}
      ],
      "Playlists":[
         {
            "Format":"HLSv3",
            "Name":"playlist-iPhone-lasagna.m3u8",
            "OutputKeys": [
               "iphone/lasagna-1024k",
               "iphone/lasagna-512k"
            ]
         }
      ],
      "UserMetadata":
         {
            "Food type":"Italian",
            "Cook book":"recipe notebook"
```

```
        },
      "PipelineId":"1111111111111-abcde1",
      "Status":"Progressing"
   }
}
```

List Jobs by Pipeline

Topics

- Description (p. 210)
- Requests (p. 210)
- Responses (p. 211)
- Errors (p. 227)
- Examples (p. 227)

Description

To get a list of the jobs that you assigned to a pipeline, send a GET request to the `/2012-09-25/jobsByPipeline/pipelineId` resource. When you list jobs by pipeline, Elastic Transcoder lists the jobs that you've created recently for the specified pipeline.

Requests

Syntax

To get information about the jobs currently in a pipeline, send the following GET request.

```
GET /2012-09-25/jobsByPipeline/pipeline Id?
Ascending=true|false&
PageToken=value for accessing the next page of results HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Request Parameters

This operation takes the following request parameters. Elastic Transcoder returns all of the jobs currently in the specified pipeline.

PipelineId
To get information about jobs that you added to a given pipeline, specify the value of the `Id` object for that pipeline.

Ascending
To list jobs in chronological order by the date and time that they were submitted, enter `true`. To list jobs in reverse chronological order, enter `false`.

PageToken
When Elastic Transcoder returns more than one page of results, use `PageToken` in subsequent `GET` requests to get each successive page of results.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

This operation does not use a request body.

Responses

Syntax

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Jobs":[
      {
         "Id":"Id that Elastic Transcoder assigned to the job",
         "Input":{
            "Key":"name of the file to transcode",
            "Encryption":{
               "Mode":"AES-CBC-PKCS7|AES-CTR|AES-GCM",
               "Key":"encrypted and base64-encoded decryption key",
               "KeyMd5":"base64-encoded key digest",
               "InitializationVector":"base64-encoded initialization
                  vector"
            },
            "FrameRate":"auto|10|15|23.97|24|25|29.97|30|50|60",
            "Resolution":"auto",
            "AspectRatio":"auto|1:1|4:3|3:2|16:9",
            "Interlaced":"auto|true|false",
            "Container":"auto|3gp|aac|asf|avi|divx|flv|m4a|mkv|mov|mp3|
               mp4|mpeg|mpeg-ps|mpeg-ts|mxf|ogg|vob|wav|webm"
         },
         "OutputKeyPrefix":"prefix for file names in Amazon S3 bucket",
         "Outputs":[
            {
               "Id":"sequential counter",
               "Key":"name of the transcoded file",
               "Encryption":{
                  "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                     AES-CTR|AES-GCM",
                  "Key":"encrypted and base64-encoded encryption key",
                  "KeyMd5":"base64-encoded key digest",
                  "InitializationVector":"base64-encoded initialization
                     vector"
               },
               "ThumbnailPattern":""|"pattern",
               "Rotate":"auto|0|90|180|270",
               "PresetId":"PresetId for the job",
               "SegmentDuration":"[1,60]",
               "Watermarks":[
```

```
      {
         "InputKey":"name of the .png or .jpg file",
         "Encryption":{
            "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
               AES-CTR|AES-GCM",
            "Key":"encrypted and base64-encoded encryption key",
            "KeyMd5":"base64-encoded key digest",
            "InitializationVector":"base64-encoded
               initialization vector"
         },
       "PresetWatermarkId":"value of Video:Watermarks:Id in preset"

      },
      {...}
   ],
   "AlbumArt":[
      {
         "AlbumArtMerge":"Replace|Prepend|Append|Fallback",
         "AlbumArtArtwork":"can be empty, but not null":[
            {
               "AlbumArtInputKey":"name of the file to use as
                  album art",
               "Encryption":{
                  "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                     AES-CTR|AES-GCM",
                  "Key":"encrypted and base64-encoded encryption
                     key",
                  "KeyMd5":"base64-encoded key digest",
                  "InitializationVector":"base64-encoded
                     initialization vector"
               },
               "AlbumArtMaxWidth":"maximum width of output album
                  art in pixels",
               "AlbumArtMaxHeight":"maximum height of output
                  album art in pixels",
               "AlbumArtSizingPolicy":"Fit|Fill|Stretch|Keep|
                  ShrinkToFit|ShrinkToFill",
               "AlbumArtPaddingPolicy":"Pad|NoPad",
               "AlbumArtFormat":"jpg|png"
            },
            {...}
         ]
       },
      {...}
   ],
   "Composition":[
      {
         "TimeSpan":
            {
               "StartTime":"starting place of the clip, in
                  HH:mm:ss.SSS or sssss.SSS",
               "Duration":"duration of the clip, in HH:mm:ss.SSS
                  or sssss.SSS"
            }
      }
   ],
   "Duration":"duration in seconds",
   "Width":"width in pixels",
```

```
            "Height":"height in pixels",
            "Status":"Submitted|Progressing|Complete|Canceled|Error",
            "StatusDetail":"detail associated with Status",
            "Captions":{
               "MergePolicy":"MergeOverride|MergeRetain|Override",
               "CaptionSources":[
                  {
                     "Key":"name of the input caption file",
                     "Encryption":{
                        "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                           AES-CTR|AES-GCM",
                        "Key":"encrypted and base64-encoded encryption
                           key",
                        "KeyMd5":"base64-encoded key digest",
                        "InitializationVector":"base64-encoded
                           initialization vector"
                     },
                     "Language":"language of the input caption file",
                     "TimeOffset":"starting place of the captions, in
                        either [-+]SS.sss or [-+]HH:mm:SS.ss",
                     "Label":"label for the caption"
                  },
                  {...}
               ],
               "CaptionFormats":[
                  {
                     "Format":"dfxp|mov-text|scc|srt|webvtt",
                     "Pattern":"myCaption/file-language"
                     "Encryption":{
                        "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                           AES-CTR|AES-GCM",
                        "Key":"encrypted and base64-encoded encryption
                           key",
                        "KeyMd5":"base64-encoded key digest",
                        "InitializationVector":"base64-encoded
                           initialization vector"
                     }
                  },
                  {...}
               ]
            },
            "AppliedColorSpaceConversion":"None|Bt601ToBt709|
               Bt709ToBt601"
         },
         {...}
      ],
      "Playlists":[
         {
            "Format":"HLSv3|HLSv4|Smooth",
            "Name":"name",
            "OutputKeys":[
               "Outputs:Key to include in this playlist",
               ...
            ],
            "HlsContentProtection":{
               "Method":"aes-128",
               "Key":"encrypted and base64-encoded protection key",
               "KeyMd5":"base64-encoded key digest",
```

```
                "InitializationVector":"base64-encoded
                      initialization vector",
                "LicenseAcquisitionUrl":"license aquisition url",
                "KeyStoragePolicy":"NoStore|WithVariantPlaylists"
             }
          },
          {...}
       ],
       "UserMetadata":
          {
              "Key":"Value",
              "Second user metadata key":"Second user metadata value"
          },
       "PipelineId":"PipelineId for the job",
       "Status":"Submitted|Progressing|Complete|Canceled|Error"
    },
    {...}
 ],
 "NextPageToken":value for accessing the next page of results|null
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

The response body contains one element for each job that satisfies the search criteria. Each element contains the following JSON objects.

Id
The identifier that Elastic Transcoder assigned to the job. You use this value to get settings for the job or to delete the job.

Input
Information about the file that Elastic Transcoder transcoded. These are values that you specified when you created the job.

Input:Key
The name of the file that you want to transcode. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/lasagna.mpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

(Optional) Input:Encryption
The encryption settings, if any, that are used for decrypting your input files. If your input file is encrypted, you must specify the mode that Elastic Transcoder will use to decrypt your file.

Input:Encryption:Mode (Required for Encryption)
The specific encryption mode that you want Elastic Transcoder to use when decrypting your files.

Elastic Transcoder supports the following options:

- **Amazon S3 Server-Side Encryption:** Amazon S3 handles the encryption and decryption of your files. As long as Elastic Transcoder has access permissions to your Amazon S3 bucket, you don't need to take any action.

For more information, see Protecting Data Using Server-Side Encryption in the *Amazon Simple Storage Service Developer Guide*.

- **Server-Side Encryption Using Customer-Provided Keys:** Elastic Transcoder supports three types of encryption using customer-provided keys:
  - **AES-CBC-PKCS7:** A padded cipher-block mode of operation.
  - **AES-CTR:** AES Counter Mode.
  - **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

  If you chose one of the AES-encryption modes, you must also specify the following three values (all three must be base64-encoded):
  - **Encryption Key**
  - **Encryption Key MD5**
  - **Encryption Initialization Vector**

(Optional) Input:Encryption:Key
The data encryption key used to encrypt your file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

The key must also be encrypted by using AWS KMS. For more information, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Input:Encryption:KeyMd5
The MD5 digest of the key used to encrypt your input file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

(Optional) Input:Encryption:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

For more information, go to Initialization Vector.

(Optional) Input:FrameRate
The frame rate of the input file. If you want Elastic Transcoder to automatically detect the frame rate of the input file, specify `auto`. If you want to specify the frame rate for the input file, enter one of the following values:

`10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `50`, `60`

The default value is `auto`.

(Optional) Input:Resolution
The resolution, in pixels, of the input file. This value must be `auto`, which causes Elastic Transcoder to automatically detect the resolution of the input file.

(Optional) Input:AspectRatio
The aspect ratio of the input file. If you want Elastic Transcoder to automatically detect the aspect ratio of the input file, specify `auto`. If you want to specify the aspect ratio for the output file, enter one of the following values:

`1:1`, `4:3`, `3:2`, `16:9`

The default value is `auto`.

(Optional) Input:Interlaced
Whether the input file is interlaced. If you want Elastic Transcoder to automatically detect whether the input file is interlaced, specify `auto`. If you want to specify whether the input file is interlaced, enter one of the following values:

`true`, `false`

The default value is `auto`.

(Optional) Input:Container
The container type for the input file. If you want Elastic Transcoder to automatically detect the container type of the input file, specify `auto`. If you want to specify the container type for the input file, enter one of the following values:

`3gp`, `aac`, `asf`, `avi`, `divx`, `flv`, `m4a`, `mkv`, `mov`, `mp3`, `mp4`, `mpeg`, `mpeg-ps`, `mpeg-ts`, `mxf`, `ogg`, `vob`, `wav`, `webm`

OutputKeyPrefix
The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, and playlists. If you specify a value, it must contain a `/` somewhere after the first character, which simplifies Amazon S3 file management.

Outputs
Information about the output files. You can create a maximum of 30 outputs per job. If you specify more than one output for a job, Elastic Transcoder creates the files for each output in the order in which you specify them in the job. The `Outputs:Id` object identifies the position of an output in the sequence.

> **Note**
> In early versions of Elastic Transcoder, you could create just one output per job, so the object name was `Output`. The `Output` syntax still works, but we recommend that you use the `Outputs` syntax for all jobs, even when you want Elastic Transcoder to transcode a file into only one format. Do not use both the `Outputs` and `Output` syntaxes in the same request.

Outputs:Id
A sequential counter, starting with `1`, that identifies an output among the outputs from the current job. In the `Output` syntax, this value is always `1`.

Outputs:Key
The name that you want Elastic Transcoder to assign to the transcoded file and playlist. Elastic Transcoder saves the file or files in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that you specify in `PipelineId`.

If the bucket already contains a file that has the specified name, the output fails. In the `Create Job` response, the value of `Outputs:Status` for that output will be `Error`, as will the final value of Status for the job. However, other outputs in the same job may succeed.

The format for file names depends the container type and whether the segment duration is set. If the container type is not `ts` or the segment duration is not provided, the name of the output file is a concatenation of `OutputKeyPrefix` and `Key`.

If the container type is `ts` and segment duration is provided, Elastic Transcoder uses the value of `Key` to name both the playlist for the output and the `.ts` files:

- **Playlist:**
  - **HLSv3:** The file name is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `.m3u8`:

    `OutputKeyPrefix``Key`*`.m3u8`*
  - **HLSv4:** The file name is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `_v4.m3u8`. Video outputs create a second file with a file name that is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `_iframe.m3u8`:

*OutputKeyPrefix*Key*_v4.m3u8*

*OutputKeyPrefix*Key*_iframe.m3u8*

- **Segment (.ts) files:**
  - **HLSv3:** The file name is a concatenation of `OutputKeyPrefix` and `Key`, plus a five-digit sequential counter beginning with `00000`, and the file name extension `.ts`:

    *OutputKeyPrefix*Key*00000.ts*
  - **HLSv4:** The file name is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `.ts`:

    *OutputKeyPrefix*Key*.ts*

If a segmented ts output is not included in a master playlist, Elastic Transcoder treats the output as `HLSv3`.

> **Note**
> Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for `HLSv3` playlists, `_v4.m3u8` and `_iframe.m3u8` for `HLSv4` playlists). If you include a file extension in `Output Key`, the file name will have two extensions.

`OutputKeyPrefix` groups all of the files for a job together in your Amazon S3 bucket. If you want to group the files for each output within a job, you can include a prefix in the value of `Key`, for example:

*OutputKeyPrefix*iPhone/Key*00000.ts*

*OutputKeyPrefix*KindleFireHD/Key*00000.ts*

(Optional) Outputs:Encryption

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

(Required for Encryption) Outputs:Encryption:Mode

The specific server-side encryption mode that you want Elastic Transcoder to use when encrypting your output files individually. Elastic Transcoder supports the following **Encryption Mode** options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.

  For more information, see Protecting Data Using Server-Side Encryption in the *Amazon Simple Storage Service Developer Guide*.
- **S3-AWS-KMS:** Amazon S3 calls AWS KMS, which creates and manages the keys that are used for encrypting your files. If you specify **S3-AWS-KMS** and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

  For more information, see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys in the *Amazon Simple Storage Service Developer Guide*.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

If you chose one of the AES-encryption modes, you must also specify the following three values (all three must be base64-encoded):

- **Encryption Key**
- **Encryption Key MD5**
- **Encryption Initialization Vector**

If you chose one of the AES-encryption modes, and you want Elastic Transcoder to generate a 128-bit bit AES encryption key for you, do not specify values for the **Encryption Key**, **Encryption Key MD5**, or **Encryption Initialization Vector**.

Important
For the AES modes, your media-specific private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to decrypt your data.

(Optional) Outputs:Encryption:Key
If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

128, 192, or 256.

For more information about encrypting your key with AWS KMS, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Outputs:Encryption:KeyMd5
The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

(Optional) Outputs:Encryption:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

For more information, go to Initialization Vector.

(Optional) Outputs:ThumbnailPattern
Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify "".

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **{count} (Required):** If you want to create thumbnails, you must include {count} in the ThumbnailPattern object. Wherever you specify {count}, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file.

  Important
  If you specify a literal value and/or {resolution} but you omit {count}, Elastic Transcoder returns a validation error and does not create the job.

- **(Optional) Literal values:** You can specify literal values anywhere in the ThumbnailPattern object, for example, as a file name prefix or as a delimiter between {resolution} and {count}.
- **(Optional) {resolution}:** If you want Elastic Transcoder to include the resolution in the file name, include {resolution} in the ThumbnailPattern object.

When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in PresetId. Elastic Transcoder also appends the applicable file name extension.

As with `Outputs:Key`, you can include a prefix in `ThumbnailPattern` that groups the applicable files together, for example, all of the thumbnails for one video in one format, or all of the thumbnails with the corresponding output file.

(Optional) Outputs:Rotate

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Valid values include:

`auto`, `0`, `90`, `180`, `270`

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

Outputs:PresetId

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset.

> **Note**
> If you created any presets before AAC profiles were added, Elastic Transcoder uses the AAC-LC profile for those presets.

For a list of system presets, see System Presets (p. 70) (You can also get these IDs using List Presets (p. 296).)

(Fragmented MP4/MPEG-TS Outputs Only) Outputs:SegmentDuration

If you specify a preset for the current output for which the value of `Container` is either **ts** (MPEG-TS) or **fmp4** (Fragmented MP4), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of:

```
total length/SegmentDuration
```

Elastic Transcoder creates an output-specific playlist for each `HLS` output that you specify in `OutputKeys`. To add an output to a master playlist for this job, include it in Outputs in Master Playlist.

Elastic Transcoder applies this segmenting to any captions associated with the output video.

For more information, see HTTP Live Streaming.

(Video Only) Outputs:Watermarks

Information about the watermarks that you want Elastic Transcoder to add to the video or artwork during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Outputs:PresetId` for the current output.

Watermarks are added to the output file in the sequence in which you list them in the job output—the first watermark in the list is added to the output file first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you list in `Outputs:Watermarks` will cover the first one, the third one will cover the second, and the fourth one will cover the third.

For more information about watermarks, see Watermarks (p. 69).

(Video Only) Outputs:Watermarks:InputKey

The name of the `.png` or `.jpg` file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `logos/128x64.png`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

(Video Only) Outputs:Watermarks:PresetWatermarkId
The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the file during transcoding. The settings are in the preset specified by `Outputs:PresetId` for the current output. In that preset, the value of `Watermarks:Id` tells Elastic Transcoder which settings to use.

(MP3/MP4 Only) Outputs:AlbumArt
The album art to be associated with the output file, if any.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `MergePolicy` to `Replace` and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `MergePolicy` to `Prepend`, `Append`, or `Fallback`, and use an empty `Artwork` array.

> **Note**
> Album Art is available only for containers of type `mp3` or `mp4`.

(MP3/MP4 Only) Outputs:AlbumArt:MergePolicy
A policy that determines how Elastic Transcoder handles the existence of multiple album artwork files.

- **Replace:** The specified album art replaces any existing album art.
- **Prepend:** The specified album art is placed in front of any existing album art.
- **Append:** The specified album art is placed after any existing album art.
- **Fallback:** If the input file contains artwork, Elastic Transcoder uses that artwork for the output. If the input does not contain artwork, Elastic Transcoder uses the specified album art file.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork
The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:InputKey
The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:MaxWidth
The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:MaxHeight
The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:SizingPolicy
A value that controls scaling of the output album art:

- **Fit:** Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill:** Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it to the dimension (if any) that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.

- **Keep:** Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit:** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **ShrinkToFill:** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.

The following table shows possible effects of `SizingPolicy` settings on the output album art:

SizingPolicy	Output Album Art May Be Scaled Up	Output Album Art May Be Padded When Padding Policy Is "Pad"	Output Album Art May Have a Different Pixel Aspect Ratio than Input Album Art	Output Album Art May Be Cropped
Fit	Yes	Yes		
Fill	Yes			Yes
Stretch	Yes		Yes	
Keep	Yes	Yes		Yes
ShrinkToFit		Yes		
ShrinkToFill		Yes		Yes

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:PaddingPolicy
When you set `PaddingPolicy` to `Pad`, Elastic Transcoder might add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`. For more information, see the table at `AlbumArt:Art:SizingPolicy`.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:AlbumArtFormat
The format of album art, if any. Valid formats are `jpg` and `png`.

Outputs:Composition
Using a composition, you can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The `Composition` object contains settings for the clips that make up an output file.

Currently, you can specify settings for only a single clip per output file. The `Composition` object cannot be null. All jobs in a playlist must have the same clip settings.

Outputs:Composition:TimeSpan
Settings that determine when a clip begins and how long it lasts.

Outputs:Composition:TimeSpan:StartTime
The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

Outputs:Composition:TimeSpan:Duration
The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file , Elastic Transcoder transcodes the file and returns a warning message.

Outputs:Duration
Duration of the output file, in seconds.

Outputs:Width
Width of the output file, in pixels.

Outputs:Height
Height of the output file, in pixels.

Outputs:Status
The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
- When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.
- `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
- When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the terminal status for `Job:Status` is also `Error`.

The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

Outputs:StatusDetail
Information that further explains `Outputs:Status`.

(Video Only) Outputs:Captions
You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file.

  Valid input values include: `CEA-608 (EIA-608`, first non-empty channel only), `CEA-708 (EIA-708`, first non-empty channel only), and `mov-text`

  Valid outputs include: `mov-text`

  Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file.

  Valid input values include: `dfxp` (first div element only), `ebu-tt`, `scc`, `smpt`, `srt`, `ttml` (first div element only), and `webvtt`

  Valid outputs include: `dfxp` (first div element only), `scc`, `srt`, and `webvtt`.

If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to null. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a null `CaptionSources` array.

For more information about embedded files, see the Subtitle (caption) Wikipedia page.

For more information about sidecar files, see the Metadata Platform and Sidecar file Wikipedia pages.

(Video Only) Outputs:Captions:MergePolicy
A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If **CaptionSources** is empty, Elastic Transcoder omits all sidecar captions from the output files.
- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`.

`MergePolicy` cannot be null.

(Video/Sidecar Only, Optional) Outputs:Captions:CaptionSources
Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

(Video Only) Outputs:Captions:CaptionSources:Key
The name of the sidecar caption file that you want Elastic Transcoder to include with the outputs.

(Video Only) Outputs:Captions:CaptionSources:Language
A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information about ISO language codes, see List of ISO 639-1 codes.

(Video Only, Optional) Outputs:Captions:CaptionSources:TimeOffset
For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the TimeOffset in the form [+-]SS.sss or [+-]HH:mm:SS.ss.

(Video Only, Optional) Outputs:Captions:CaptionSources:Label
The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

(Video Only) Outputs:Captions:CaptionFormats
The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

(Video Only) Outputs:Captions:CaptionFormats:Format
The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption for this output.

- **Embedded Caption Formats:**

Container Type	Supported Embedded types
MP4	mov-text
Non-MP4 containers	Embedded captions are not supported

Elastic Transcoder supports a maximum of one embedded format per output.

- **Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

Container Type	Supported Sidecar types
fmp4	dfxp
Non-fmp4 containers	All sidecar types

Note
`fmp4` captions have an extension of `.ismt`

(Video/Sidecar Only) Outputs:Captions:CaptionFormats:Pattern
The prefix for caption filenames, in the form *`description`*`-{language}`, where:

- *`description`* is a description of the video.
- `{language}` is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include `{language}` in the file name pattern, Elastic Transcoder automatically appends "`{language}`" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be `Sydney-en-sunrise00000.srt`.

(Automatic) Outputs:AppliedColorSpaceConversion
If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

For more information about `ColorSpaceConversionMode`, see ColorSpaceConversion.

(Optional) Outputs:UserMetadata
User-defined metadata that you want to associate with an Elastic Transcoder job. You specify metadata in `key/value` pairs. You can use the `key/value` pairs to track details about a file, for example, `Season 1: Episode 3`.

You can add up to 10 key/value pairs to each job. Elastic Transcoder does not guarantee that `key/value` pairs are returned in the same order in which you specify them.

Outputs:UserMetadata:Key
The key of the metadata `key/value` pair that you want returned with the output file. Each key must be a unique string between `1-128` characters, and must use only characters from the following list:

- `0-9`
- `A-Z` and `a-z`
- `Space`
- The following symbols: `_.:/=+-%@`

You can use keys as a numbering system for organizing your metadata, for storing an extra 128 characters of metadata, or for labeling the metadata stored in the **value**. If you want to use only value metadata, you can put throw-away strings in your keys such as `key1`, and ignore the keys when you retrieve your metadata from Elastic Transcoder.

Important
You must specify unique strings for all of the keys in a job. If the same string is used for more than one key in a job, Elastic Transcoder returns only one of the key/value pairs that use that key. There is no way to guarantee which value is returned.

Outputs:UserMetadata:Value
The value of the metadata `key/value` pair that you want returned with your job. Each value must be a string between `0-256` characters, and must use only characters from the following list:

- `0-9`
- `A-Z` and `a-z`
- `Space`
- The following symbols: `_.:/=+-%@`

(Fragmented MP4/MPEG-TS Outputs Only) Playlists
If you specify a preset in `PresetId` for which the value of `Container` is either **ts** (MPEG-TS) or **fmp4** (Fragmented MP4), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create.

We recommend that you create at most one master playlist per playlist format. The maximum number of master playlists in a job is 30.

Playlists:Format
The format of the output playlist. Valid formats include `HLSv3`, `HLSv4`, and `Smooth`.

Playlists:Name
The name that you want Elastic Transcoder to assign to a master playlist, for example, `nyc-vacation.m3u8`. If the name includes a `/` character, the section of the name before the `/` must be identical for all `Name` objects. If you create more than one master playlist, the values of all `Name` objects must be unique.

Note
Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for `HLSv3` playlists, `_v4.m3u8` and `_iframe.m3u8` for `HLSv4` playlists, and `.ism` and `.ismc` for `Smooth` playlists). If you include a file extension in `Name`, the file name will have two extensions.

Any segment duration settings, clip settings, or caption settings must be the same for all outputs in the playlist. For `Smooth` playlists, the `Audio:Profile`, `Video:Profile`, and `Video:FrameRate` to `Video:KeyframesMaxDist` ratio must be the same for all outputs. For more information, see KeyframesMaxDist (p. 282).

Playlists:OutputKeys
For each output in this job that you want to include in a master playlist, the value of the `Outputs:Key` object. If you include more than one output in a playlist, the value of `SegmentDuration` for all of the outputs must be the same.

For `HLSv4` master playlists, Elastic Transcoder chooses which combinations of audio and video inputs will be linked in the output playlists. The first audio and video inputs will be linked and rendered as the default playback experience, allowing you to choose your preferred playback default. For other individual playlists in the master playlist, Elastic Transcoder chooses which audio and video bit rate combinations will provide the best playback.

(Optional) Outputs:Playlists:HlsContentProtection
The HLS content protection settings, if any, that you want Elastic Transcoder to apply to your output files. If you want to use HLS content protection do not specify encryption settings for the output file or captions. HLS content protection encrypts each segment of a file so that they can be streamed encrypted and only decrypted on playback, while the output file and caption encryptions encrypt the file all at once. Elastic Transcoder does not support files that are encrypted both ways.

Outputs:Playlists:HlsContentProtection:Method
The content protection method for your output. The only valid value is:

`aes-128`.

This value will be written into the `method` attribute of the `EXT-X-KEY` metadata tag in the output playlist.

(Optional) Outputs:Playlists:HlsContentProtection:Key
If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

For more information about encrypting your key with AWS KMS, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Outputs:Playlists:HlsContentProtection:KeyMd5
The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

(Optional) Outputs:Playlists:HlsContentProtection:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

For more information, go to Initialization Vector.

Outputs:Playlists:HlsContentProtection:LicenseAcquisitionUrl
The location of the license key required to decrypt your HLS playlist. The URL must be an absolute path, and is referenced in the URI attribute of the EXT-X-KEY metadata tag in the playlist file. For example:

```
https://www.example.com/exampleKey/
```

Outputs:Playlists:HlsContentProtection:KeyStoragePolicy
Specify whether you want Elastic Transcoder to write your HLS license key to an Amazon S3 bucket. If you choose `WithVariantPlaylists`, Elastic Transcoder will write your encrypted key into the same Amazon S3 bucket as the associated playlist.

> **Important**
> If you chose `NoStore`, Elastic Transcoder will not store your key. You are responsible for storing it and providing it to your users by giving them the **License Acquisition URL** where you are storing the key.

PipelineId
The value of the `Id` object for the pipeline that you want Elastic Transcoder to use for transcoding. The pipeline determines several settings, including the Amazon S3 bucket from which Elastic Transcoder gets the files to transcode and the bucket into which Elastic Transcoder puts the transcoded files.

Status
If you specified more than one output for the job, the status of the entire job. When Elastic Transcoder starts processing a job, the value of `Job:Status` changes to `Progressing` and doesn't change until Elastic Transcoder has finished processing all outputs. When processing is complete, `Job:Status` changes either to `Complete` or, if any of the outputs failed, to `Error`.

If you specified only one output for the job, `Job:Status` is the same as `Outputs:Status`.

The value of `Job:Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

NextPageToken

A value that you use to access the second and subsequent pages of results, if any. When the jobs in the specified pipeline fit on one page or when you've reached the last page of results, the value of `NextPageToken` is `null`.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

The following example request creates a job.

Sample Request

The following example request gets a list of all of the jobs that you have ever assigned to the pipeline ID `1111111111111-abcde1`.

```
GET /2012-09-25/jobsByPipeline/1111111111111-abcde1?Ascending=true HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Sample Response

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Jobs":[
      {
         "Id":"3333333333333-abcde3",
         "Input":{
            "Key":"cooking/lasagna.mp4",

            "FrameRate":"auto",
            "Resolution":"auto",
            "AspectRatio":"auto",
            "Interlaced":"auto",
            "Container":"mp4"
         },
         "Outputs":[
```

```
        {
           "Id":"1",
           "Key":"cooking/lasagna-KindleFireHD.mp4",

           "ThumbnailPattern":"cooking/lasagna-{count}-KindleFireHD",
           "Rotate":"0",
           "PresetId":"1351620000000-100080",
           "Watermarks":[
              {
                 "InputKey":"logo/128x64.png",
                 "PresetWatermarkId":"company logo 128x64",

              }
           ],
           "Duration":"1003",
           "Width":"1280",
           "Height":"720",
           "Status":"Submitted",
           "StatusDetail":"Job has been received.",
           "Captions":{
              "MergePolicy":"MergeOverride",
              "CaptionSources":[
                 {
                    "Key":"scc/lasagna-kindlefirehd.scc",

                    "Language":"en",
                    "Label":"English"
                 },
                 {
                    "Key":"srt/lasagna-kindlefirehd.srt",

                    "Language":"fr",
                    "TimeOffset":"1:00:00",
                    "Label":"French"
                 }
              ],
              "CaptionFormats":[
                 {
                    "Format":"scc",
                    "Pattern":"scc/lasagna-{language}",

                 },
                 {
                    "Format":"srt",
                    "Pattern":"srt/lasagna-{language}"
                 },
                 {
                    "Format":"mov-text"
                 }
              ]
           }
        },
        {
           "Id":"2",
           "Key":"cooking/lasagna-iPhone4s.mp4",
           "ThumbnailPattern":"cooking/lasagna-{count}-iPhone4s",
           "Rotate":"0",
           "PresetId":"1351620000000-100020",
```

```
            "Watermarks":[
               {
                  "InputKey":"logo/128x64.png",
                  "PresetWatermarkId":"company logo 128x64"
               }
            ],
            "Duration":"1003",
            "Width":"1920",
            "Height":"1080",
            "Status":"Submitted",
            "StatusDetail":"Job has been received.",
            "AppliedColorSpaceConversion":"None"
         }
      ],
      "PipelineId":"1111111111111-abcde1"
   },
   {
      "Id":"4444444444444-abcde4",
      "Input":{
         "Key":"cooking/baked-ziti.mp4",
         "FrameRate":"auto",
         "Resolution":"auto",
         "AspectRatio":"auto",
         "Interlaced":"auto",
         "Container":"mp4"
      },
      "Outputs":[
         {
            "Id":"1",
            "Key":"cooking/baked-ziti-KindleFireHD.mp4",
            "ThumbnailPattern":"cooking/baked-ziti-{count}-KindleFireHD",
            "Rotate":"0",
            "PresetId":"1351620000000-100080",
            "Watermarks":[
               {
                  "InputKey":"logo/128x64.png",
                  "PresetWatermarkId":"company logo 128x64"
               }
            ],
            "Duration":"596",
            "Width":"1280",
            "Height":"720",
            "Status":"Complete",
            "StatusDetail":""
         }
      ],
      "UserMetadata":
         {
            "Food type":"Italian",
            "Cook book":"recipe notebook"
         },
      "PipelineId":"1111111111111-abcde1"
   }
],
"NextPageToken":null
}
```

List Jobs by Status

Topics

- Description (p. 230)
- Requests (p. 230)
- Responses (p. 231)
- Errors (p. 247)
- Examples (p. 247)

Description

To get a list of the jobs that have a specified status, send a GET request to the `/2012-09-25/jobsByStatus/Submitted` resource. Elastic Transcoder lists the jobs that you've created recently and that currently have the specified status.

Requests

Syntax

To get information about all of the jobs associated with the current AWS account that have a specified status, send the following GET request.

```
GET /2012-09-25/jobsByStatus/Status?
Ascending=true|false&
PageToken=value for accessing the next page of results HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Request Parameters

This operation takes the following request parameters. Elastic Transcoder returns all of the jobs that have the specified status.

Status
: To get information about all of the jobs associated with the current AWS account that have a given status, specify the status: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

Ascending
: To list jobs in chronological order by the date and time that they were submitted, enter `true`. To list jobs in reverse chronological order, enter `false`.

PageToken
: When Elastic Transcoder returns more than one page of results, use `PageToken` in subsequent `GET` requests to get each successive page of results.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

This operation does not use a request body.

Responses

Syntax

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Jobs":[
      {
         "Id":"Id that Elastic Transcoder assigned to the job",
         "Input":{
            "Key":"name of the file to transcode",
            "Encryption":{
               "Mode":"AES-CBC-PKCS7|AES-CTR|AES-GCM",
               "Key":"encrypted and base64-encoded decryption key",
               "KeyMd5":"base64-encoded key digest",
               "InitializationVector":"base64-encoded initialization
                  vector"
            },
            "FrameRate":"auto|10|15|23.97|24|25|29.97|30|50|60",
            "Resolution":"auto",
            "AspectRatio":"auto|1:1|4:3|3:2|16:9",
            "Interlaced":"auto|true|false",
            "Container":"auto|3gp|aac|asf|avi|divx|flv|m4a|mkv|mov|mp3|
               mp4|mpeg|mpeg-ps|mpeg-ts|mxf|ogg|vob|wav|webm"
         },
         "OutputKeyPrefix":"prefix for file names in Amazon S3 bucket",
         "Outputs":[
            {
               "Id":"sequential counter",
               "Key":"name of the transcoded file",
               "Encryption":{
                  "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                     AES-CTR|AES-GCM",
                  "Key":"encrypted and base64-encoded encryption key",
                  "KeyMd5":"base64-encoded key digest",
                  "InitializationVector":"base64-encoded initialization
                     vector"
               },
               "ThumbnailPattern":""|"pattern",
               "Rotate":"auto|0|90|180|270",
               "PresetId":"PresetId for the job",
               "SegmentDuration":"[1,60]",
               "Watermarks":[
```

```
      {
         "InputKey":"name of the .png or .jpg file",
         "PresetWatermarkId":"value of Video:Watermarks:Id in
            preset",
         "Encryption":{
            "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
               AES-CTR|AES-GCM",
            "Key":"encrypted and base64-encoded encryption key",
            "KeyMd5":"base64-encoded key digest",
            "InitializationVector":"base64-encoded
               initialization vector"
         }
      },
      {...}
   ],
   "AlbumArt":[
      {
         "AlbumArtMerge":"Replace|Prepend|Append|Fallback",
         "AlbumArtArtwork":"can be empty, but not null":[
            {
               "AlbumArtInputKey":"name of the file to use as
                  album art",
               "Encryption":{
                  "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                     AES-CTR|AES-GCM",
                  "Key":"encrypted and base64-encoded encryption
                     key",
                  "KeyMd5":"base64-encoded key digest",
                  "InitializationVector":"base64-encoded
                     initialization vector"
               },
               "AlbumArtMaxWidth":"maximum width of output album
                  art in pixels",
               "AlbumArtMaxHeight":"maximum height of output
                  album art in pixels",
               "AlbumArtSizingPolicy":"Fit|Fill|Stretch|Keep|
                  ShrinkToFit|ShrinkToFill",
               "AlbumArtPaddingPolicy":"Pad|NoPad",
               "AlbumArtFormat":"jpg|png"
            },
            {...}
         ]
       },
      {...}
   ],
   "Composition":[
      {
         "TimeSpan":
            {
               "StartTime":"starting place of the clip, in
                  HH:mm:ss.SSS or sssss.SSS",
               "Duration":"duration of the clip, in HH:mm:ss.SSS
                  or sssss.SSS"
            }
      }
   ],
   "Duration":"duration in seconds",
   "Height":"height in pixels",
```

```
            "Width":"width in pixels",
            "Status":"Submitted|In Progress|Complete|Error",
            "StatusDetail":"detail associated with Status",
            "Captions":{
               "MergePolicy":"MergeOverride|MergeRetain|Override",
               "CaptionSources":[
                  {
                     "Key":"name of the input caption file",
                     "Encryption":{
                        "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                           AES-CTR|AES-GCM",
                        "Key":"encrypted and base64-encoded encryption
                           key",
                        "KeyMd5":"base64-encoded key digest",
                        "InitializationVector":"base64-encoded
                           initialization vector"
                     },
                     "Language":"language of the input caption file",
                     "TimeOffset":"starting place of the captions, in
                        either [-+]SS.sss or [-+]HH:mm:SS.ss",
                     "Label":"label for the caption"
                  },
                  {...}
               ],
               "CaptionFormats":[
                  {
                     "Format":"dfxp|mov-text|scc|srt|webvtt",
                     "Pattern":"myCaption/file-language",
                     "Encryption":{
                        "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                           AES-CTR|AES-GCM",
                        "Key":"encrypted and base64-encoded encryption
                           key",
                        "KeyMd5":"base64-encoded key digest",
                        "InitializationVector":"base64-encoded
                           initialization vector"
                     }
                  },
                  {...}
               ]
            },
            "AppliedColorSpaceConversion":"None|Bt601ToBt709|
               Bt709ToBt601"
         },
         {...}
      ],
      "Playlists":[
         {
            "Format":"HLSv3|HLSv4|Smooth",
            "Name":"name",
            "OutputKeys":[
               "Outputs:Key to include in this playlist",
               ...
            ],
            "HlsContentProtection":{
               "Method":"aes-128",
               "Key":"encrypted and base64-encoded protection key",
               "KeyMd5":"base64-encoded key digest",
```

```
                "InitializationVector":"base64-encoded
                      initialization vector",
                "LicenseAcquisitionUrl":"license aquisition url",
                "KeyStoragePolicy":"NoStore|WithVariantPlaylists"
             }
          },
          {...}
       ],
       "UserMetadata":
          {
             "Key":"Value",
             "Second user metadata key":"Second user metadata value"
          },
       "PipelineId":"PipelineId for the job",
       "Status":"Submitted|Progressing|Complete|Canceled|Error"
    },
    {...}
 ],
 "NextPageToken":value for accessing the next page of results|null
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

The response body contains one element for each job that satisfies the search criteria. Each element contains the following JSON objects.

Id
The identifier that Elastic Transcoder assigned to the job. You use this value to get settings for the job or to delete the job.

Input
Information about the file that Elastic Transcoder transcoded. These are values that you specified when you created the job.

Input:Key
The name of the file that you want to transcode. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/lasagna.mpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

(Optional) Input:Encryption
The encryption settings, if any, that are used for decrypting your input files. If your input file is encrypted, you must specify the mode that Elastic Transcoder will use to decrypt your file.

(Required for Encryption) Input:Encryption:Mode
The specific encryption mode that you want Elastic Transcoder to use when decrypting your files.

Elastic Transcoder supports the following options:

- **Amazon S3 Server-Side Encryption:** Amazon S3 handles the encryption and decryption of your files. As long as Elastic Transcoder has access permissions to your Amazon S3 bucket, you don't need to take any action.

For more information, see Protecting Data Using Server-Side Encryption in the *Amazon Simple Storage Service Developer Guide*.

- **Server-Side Encryption Using Customer-Provided Keys:** Elastic Transcoder supports three types of encryption using customer-provided keys:
  - **AES-CBC-PKCS7:** A padded cipher-block mode of operation.
  - **AES-CTR:** AES Counter Mode.
  - **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

  If you chose one of the AES-encryption modes, you must also specify the following three values (all three must be base64-encoded):
  - **Encryption Key**
  - **Encryption Key MD5**
  - **Encryption Initialization Vector**

(Optional) Input:Encryption:Key
The data encryption key used to encrypt your file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

The key must also be encrypted by using AWS KMS. For more information, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Input:Encryption:KeyMd5
The MD5 digest of the key used to encrypt your input file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

(Optional) Input:Encryption:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

For more information, go to Initialization Vector.

(Optional) Input:FrameRate
The frame rate of the input file. If you want Elastic Transcoder to automatically detect the frame rate of the input file, specify `auto`. If you want to specify the frame rate for the input file, enter one of the following values:

`10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `50`, `60`

The default value is `auto`.

(Optional) Input:Resolution
The resolution, in pixels, of the input file. This value must be `auto`, which causes Elastic Transcoder to automatically detect the resolution of the input file.

(Optional) Input:AspectRatio
The aspect ratio of the input file. If you want Elastic Transcoder to automatically detect the aspect ratio of the input file, specify `auto`. If you want to specify the aspect ratio for the output file, enter one of the following values:

`1:1`, `4:3`, `3:2`, `16:9`

The default value is `auto`.

(Optional) Input:Interlaced
Whether the input file is interlaced. If you want Elastic Transcoder to automatically detect whether the input file is interlaced, specify `auto`. If you want to specify whether the input file is interlaced, enter one of the following values:

`true`, `false`

The default value is `auto`.

(Optional) Input:Container
The container type for the input file. If you want Elastic Transcoder to automatically detect the container type of the input file, specify `auto`. If you want to specify the container type for the input file, enter one of the following values:

`3gp`, `aac`, `asf`, `avi`, `divx`, `flv`, `m4a`, `mkv`, `mov`, `mp3`, `mp4`, `mpeg`, `mpeg-ps`, `mpeg-ts`, `mxf`, `ogg`, `vob`, `wav`, `webm`

OutputKeyPrefix
The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, and playlists. If you specify a value, it must contain a `/` somewhere after the first character, which simplifies Amazon S3 file management.

Outputs
Information about the output files. You can create a maximum of 30 outputs per job. If you specify more than one output for a job, Elastic Transcoder creates the files for each output in the order in which you specify them in the job. The `Outputs:Id` object identifies the position of an output in the sequence.

> **Note**
> In early versions of Elastic Transcoder, you could create just one output per job, so the object name was `Output`. The `Output` syntax still works, but we recommend that you use the `Outputs` syntax for all jobs, even when you want Elastic Transcoder to transcode a file into only one format. Do not use both the `Outputs` and `Output` syntaxes in the same request.

Outputs:Id
A sequential counter, starting with `1`, that identifies an output among the outputs from the current job. In the `Output` syntax, this value is always `1`.

Outputs:Key
The name that you want Elastic Transcoder to assign to the transcoded file and playlist. Elastic Transcoder saves the file or files in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that you specify in `PipelineId`.

If the bucket already contains a file that has the specified name, the output fails. In the `Create Job` response, the value of `Outputs:Status` for that output will be `Error`, as will the final value of Status for the job. However, other outputs in the same job may succeed.

The format for file names depends the container type and whether the segment duration is set. If the container type is not `ts` or the segment duration is not provided, the name of the output file is a concatenation of `OutputKeyPrefix` and `Key`.

If the container type is `ts` and segment duration is provided, Elastic Transcoder uses the value of `Key` to name both the playlist for the output and the `.ts` files:

- **Playlist:**
  - **HLSv3:** The file name is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `.m3u8`:

    `OutputKeyPrefix``Key`*`.m3u8`*
  - **HLSv4:** The file name is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `_v4.m3u8`. Video outputs create a second file with a file name that is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `_iframe.m3u8`:

*OutputKeyPrefix*Key*_v4.m3u8*

*OutputKeyPrefix*Key*_iframe.m3u8*

- **Segment (.ts) files:**
  - **HLSv3:** The file name is a concatenation of `OutputKeyPrefix` and `Key`, plus a five-digit sequential counter beginning with `00000`, and the file name extension `.ts`:

    *OutputKeyPrefix*Key*00000.ts*
  - **HLSv4:** The file name is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `.ts`:

    *OutputKeyPrefix*Key*.ts*

If a segmented ts output is not included in a master playlist, Elastic Transcoder treats the output as `HLSv3`.

> **Note**
> Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for `HLSv3` playlists, `_v4.m3u8` and `_iframe.m3u8` for `HLSv4` playlists). If you include a file extension in `Output Key`, the file name will have two extensions.

`OutputKeyPrefix` groups all of the files for a job together in your Amazon S3 bucket. If you want to group the files for each output within a job, you can include a prefix in the value of `Key`, for example:

*OutputKeyPrefix*iPhone/Key*00000.ts*

*OutputKeyPrefix*KindleFireHD/Key*00000.ts*

(Optional) Outputs:Encryption

The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

(Required for Encryption) Outputs:Encryption:Mode

The specific server-side encryption mode that you want Elastic Transcoder to use when encrypting your output files individually. Elastic Transcoder supports the following **Encryption Mode** options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.

  For more information, see Protecting Data Using Server-Side Encryption in the *Amazon Simple Storage Service Developer Guide*.
- **S3-AWS-KMS:** Amazon S3 calls AWS KMS, which creates and manages the keys that are used for encrypting your files. If you specify **S3-AWS-KMS** and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

  For more information, see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys in the *Amazon Simple Storage Service Developer Guide*.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

If you chose one of the AES-encryption modes, you must also specify the following three values (all three must be base64-encoded):

- **Encryption Key**
- **Encryption Key MD5**
- **Encryption Initialization Vector**

If you chose one of the AES-encryption modes, and you want Elastic Transcoder to generate a 128-bit bit AES encryption key for you, do not specify values for the **Encryption Key**, **Encryption Key MD5**, or **Encryption Initialization Vector**.

Important
For the AES modes, your media-specific private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to decrypt your data.

(Optional) Outputs:Encryption:Key
If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

128, 192, or 256.

For more information about encrypting your key with AWS KMS, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Outputs:Encryption:KeyMd5
The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

(Optional) Outputs:Encryption:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

For more information, go to Initialization Vector.

(Optional) Outputs:ThumbnailPattern
Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify "".

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **{count} (Required):** If you want to create thumbnails, you must include {count} in the ThumbnailPattern object. Wherever you specify {count}, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file.

  Important
  If you specify a literal value and/or {resolution} but you omit {count}, Elastic Transcoder returns a validation error and does not create the job.

- **(Optional) Literal values:** You can specify literal values anywhere in the ThumbnailPattern object, for example, as a file name prefix or as a delimiter between {resolution} and {count}.
- **(Optional) {resolution}:** If you want Elastic Transcoder to include the resolution in the file name, include {resolution} in the ThumbnailPattern object.

When creating thumbnails, Elastic Transcoder automatically saves the files in the format (.jpg or .png) that appears in the preset that you specified in PresetId. Elastic Transcoder also appends the applicable file name extension.

As with `Outputs:Key`, you can include a prefix in `ThumbnailPattern` that groups the applicable files together, for example, all of the thumbnails for one video in one format, or all of the thumbnails with the corresponding output file.

(Optional) Outputs:Rotate

The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Valid values include:

`auto`, `0`, `90`, `180`, `270`

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

Outputs:PresetId

The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset.

> **Note**
> If you created any presets before AAC profiles were added, Elastic Transcoder uses the AAC-LC profile for those presets.

For a list of system presets, see System Presets (p. 70) (You can also get these IDs using List Presets (p. 296).)

(Fragmented MP4/MPEG-TS Outputs Only) Outputs:SegmentDuration

If you specify a preset for the current output for which the value of `Container` is either **ts** (MPEG-TS) or **fmp4** (Fragmented MP4), `SegmentDuration` is the target maximum duration of each segment in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of:

```
total length/SegmentDuration
```

Elastic Transcoder creates an output-specific playlist for each `HLS` output that you specify in `OutputKeys`. To add an output to a master playlist for this job, include it in Outputs in Master Playlist.

Elastic Transcoder applies this segmenting to any captions associated with the output video.

For more information, see HTTP Live Streaming.

(Video Only) Outputs:Watermarks

Information about the watermarks that you want Elastic Transcoder to add to the video or artwork during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Outputs:PresetId` for the current output.

Watermarks are added to the output file in the sequence in which you list them in the job output—the first watermark in the list is added to the output file first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you list in `Outputs:Watermarks` will cover the first one, the third one will cover the second, and the fourth one will cover the third.

For more information about watermarks, see Watermarks (p. 69).

(Video Only) Outputs:Watermarks:InputKey

The name of the `.png` or `.jpg` file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `logos/128x64.png`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

(Video Only) Outputs:Watermarks:PresetWatermarkId
The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the file during transcoding. The settings are in the preset specified by `Outputs:PresetId` for the current output. In that preset, the value of `Watermarks:Id` tells Elastic Transcoder which settings to use.

(MP3/MP4 Only) Outputs:AlbumArt
The album art to be associated with the output file, if any.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `MergePolicy` to `Replace` and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `MergePolicy` to `Prepend`, `Append`, or `Fallback`, and use an empty `Artwork` array.

Note
Album Art is available only for containers of type `mp3` or `mp4`.

(MP3/MP4 Only) Outputs:AlbumArt:MergePolicy
A policy that determines how Elastic Transcoder handles the existence of multiple album artwork files.

- **Replace:** The specified album art replaces any existing album art.
- **Prepend:** The specified album art is placed in front of any existing album art.
- **Append:** The specified album art is placed after any existing album art.
- **Fallback:** If the input file contains artwork, Elastic Transcoder uses that artwork for the output. If the input does not contain artwork, Elastic Transcoder uses the specified album art file.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork
The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:InputKey
The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:MaxWidth
The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:MaxHeight
The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:SizingPolicy
A value that controls scaling of the output album art:

- **Fit:** Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill:** Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it to the dimension (if any) that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.

- **Keep:** Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit:** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **ShrinkToFill:** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.

The following table shows possible effects of `SizingPolicy` settings on the output album art:

SizingPolicy	Output Album Art May Be Scaled Up	Output Album Art May Be Padded When Padding Policy Is "Pad"	Output Album Art May Have a Different Pixel Aspect Ratio than Input Album Art	Output Album Art May Be Cropped
Fit	Yes	Yes		
Fill	Yes			Yes
Stretch	Yes		Yes	
Keep	Yes	Yes		Yes
ShrinkToFit		Yes		
ShrinkToFill		Yes		Yes

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:PaddingPolicy
When you set `PaddingPolicy` to `Pad`, Elastic Transcoder might add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`. For more information, see the table at `AlbumArt:Art:SizingPolicy`.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:AlbumArtFormat
The format of album art, if any. Valid formats are `jpg` and `png`.

Outputs:Composition
Using a composition, you can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The `Composition` object contains settings for the clips that make up an output file.

Currently, you can specify settings for only a single clip per output file. The `Composition` object cannot be null. All jobs in a playlist must have the same clip settings.

Outputs:Composition:TimeSpan
Settings that determine when a clip begins and how long it lasts.

Outputs:Composition:TimeSpan:StartTime
The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

Outputs:Composition:TimeSpan:Duration
The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file , Elastic Transcoder transcodes the file and returns a warning message.

Outputs:Duration
Duration of the output file, in seconds.

Outputs:Width
Width of the output file, in pixels.

Outputs:Height
Height of the output file, in pixels.

Outputs:Status
The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
- When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.
- `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
- When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the terminal status for `Job:Status` is also `Error`.

The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

Outputs:StatusDetail
Information that further explains `Outputs:Status`.

(Video Only) Outputs:Captions
You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file.

  Valid input values include: `CEA-608 (EIA-608`, first non-empty channel only), `CEA-708 (EIA-708`, first non-empty channel only), and `mov-text`

  Valid outputs include: `mov-text`

  Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file.

  Valid input values include: `dfxp` (first div element only), `ebu-tt`, `scc`, `smpt`, `srt`, `ttml` (first div element only), and `webvtt`

  Valid outputs include: `dfxp` (first div element only), `scc`, `srt`, and `webvtt`.

If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to null. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a null `CaptionSources` array.

For more information about embedded files, see the Subtitle (caption) Wikipedia page.

For more information about sidecar files, see the Metadata Platform and Sidecar file Wikipedia pages.

(Video Only) Outputs:Captions:MergePolicy
A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If **CaptionSources** is empty, Elastic Transcoder omits all sidecar captions from the output files.
- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`.

`MergePolicy` cannot be null.

(Video/Sidecar Only, Optional) Outputs:Captions:CaptionSources
Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

(Video Only) Outputs:Captions:CaptionSources:Key
The name of the sidecar caption file that you want Elastic Transcoder to include with the outputs.

(Video Only) Outputs:Captions:CaptionSources:Language
A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information about ISO language codes, see List of ISO 639-1 codes.

(Video Only, Optional) Outputs:Captions:CaptionSources:TimeOffset
For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the TimeOffset in the form [+-]SS.sss or [+-]HH:mm:SS.ss.

(Video Only, Optional) Outputs:Captions:CaptionSources:Label
The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

(Video Only) Outputs:Captions:CaptionFormats
The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

(Video Only) Outputs:Captions:CaptionFormats:Format
The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption for this output.

- **Embedded Caption Formats:**

Container Type	Supported Embedded types
MP4	mov-text
Non-MP4 containers	Embedded captions are not supported

Elastic Transcoder supports a maximum of one embedded format per output.

- **Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

Container Type	Supported Sidecar types
fmp4	dfxp
Non-fmp4 containers	All sidecar types

Note
`fmp4` captions have an extension of `.ismt`

(Video/Sidecar Only) Outputs:Captions:CaptionFormats:Pattern
The prefix for caption filenames, in the form *`description`*`-{language}`, where:

- *`description`* is a description of the video.
- `{language}` is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include `{language}` in the file name pattern, Elastic Transcoder automatically appends "`{language}`" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be `Sydney-en-sunrise00000.srt`.

(Automatic) Outputs:AppliedColorSpaceConversion
If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

For more information about `ColorSpaceConversionMode`, see ColorSpaceConversion.

(Optional) Outputs:UserMetadata
User-defined metadata that you want to associate with an Elastic Transcoder job. You specify metadata in `key/value` pairs. You can use the `key/value` pairs to track details about a file, for example, `Season 1: Episode 3`.

You can add up to 10 key/value pairs to each job. Elastic Transcoder does not guarantee that `key/value` pairs are returned in the same order in which you specify them.

Outputs:UserMetadata:Key
The key of the metadata `key/value` pair that you want returned with the output file. Each key must be a unique string between `1-128` characters, and must use only characters from the following list:

- `0-9`
- `A-Z` and `a-z`
- `Space`
- The following symbols: `_.:/=+-%@`

You can use keys as a numbering system for organizing your metadata, for storing an extra 128 characters of metadata, or for labeling the metadata stored in the **value**. If you want to use only value metadata, you can put throw-away strings in your keys such as `key1`, and ignore the keys when you retrieve your metadata from Elastic Transcoder.

Important
You must specify unique strings for all of the keys in a job. If the same string is used for more than one key in a job, Elastic Transcoder returns only one of the key/value pairs that use that key. There is no way to guarantee which value is returned.

Outputs:UserMetadata:Value
The value of the metadata `key/value` pair that you want returned with your job. Each value must be a string between `0-256` characters, and must use only characters from the following list:

- `0-9`
- `A-Z` and `a-z`
- `Space`
- The following symbols: `_.:/=+-%@`

(Fragmented MP4/MPEG-TS Outputs Only) Playlists
If you specify a preset in `PresetId` for which the value of `Container` is either **ts** (MPEG-TS) or **fmp4** (Fragmented MP4), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create.

We recommend that you create at most one master playlist per playlist format. The maximum number of master playlists in a job is 30.

Playlists:Format
The format of the output playlist. Valid formats include `HLSv3`, `HLSv4`, and `Smooth`.

Playlists:Name
The name that you want Elastic Transcoder to assign to a master playlist, for example, `nyc-vacation.m3u8`. If the name includes a `/` character, the section of the name before the `/` must be identical for all `Name` objects. If you create more than one master playlist, the values of all `Name` objects must be unique.

Note
Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for `HLSv3` playlists, `_v4.m3u8` and `_iframe.m3u8` for `HLSv4` playlists, and `.ism` and `.ismc` for `Smooth` playlists). If you include a file extension in `Name`, the file name will have two extensions.

Any segment duration settings, clip settings, or caption settings must be the same for all outputs in the playlist. For `Smooth` playlists, the `Audio:Profile`, `Video:Profile`, and `Video:FrameRate` to `Video:KeyframesMaxDist` ratio must be the same for all outputs. For more information, see KeyframesMaxDist (p. 282).

Playlists:OutputKeys
For each output in this job that you want to include in a master playlist, the value of the `Outputs:Key` object. If you include more than one output in a playlist, the value of `SegmentDuration` for all of the outputs must be the same.

For `HLSv4` master playlists, Elastic Transcoder chooses which combinations of audio and video inputs will be linked in the output playlists. The first audio and video inputs will be linked and rendered as the default playback experience, allowing you to choose your preferred playback default. For other individual playlists in the master playlist, Elastic Transcoder chooses which audio and video bit rate combinations will provide the best playback.

(Optional) Outputs:Playlists:HlsContentProtection
The HLS content protection settings, if any, that you want Elastic Transcoder to apply to your output files. If you want to use HLS content protection do not specify encryption settings for the output file or captions. HLS content protection encrypts each segment of a file so that they can be streamed encrypted and only decrypted on playback, while the output file and caption encryptions encrypt the file all at once. Elastic Transcoder does not support files that are encrypted both ways.

Outputs:Playlists:HlsContentProtection:Method
The content protection method for your output. The only valid value is:

`aes-128`.

This value will be written into the `method` attribute of the `EXT-X-KEY` metadata tag in the output playlist.

(Optional) Outputs:Playlists:HlsContentProtection:Key
If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

For more information about encrypting your key with AWS KMS, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Outputs:Playlists:HlsContentProtection:KeyMd5
The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

(Optional) Outputs:Playlists:HlsContentProtection:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

For more information, go to Initialization Vector.

Outputs:Playlists:HlsContentProtection:LicenseAcquisitionUrl
The location of the license key required to decrypt your HLS playlist. The URL must be an absolute path, and is referenced in the URI attribute of the EXT-X-KEY metadata tag in the playlist file. For example:

```
https://www.example.com/exampleKey/
```

Outputs:Playlists:HlsContentProtection:KeyStoragePolicy
Specify whether you want Elastic Transcoder to write your HLS license key to an Amazon S3 bucket. If you choose `WithVariantPlaylists`, Elastic Transcoder will write your encrypted key into the same Amazon S3 bucket as the associated playlist.

> **Important**
> If you chose `NoStore`, Elastic Transcoder will not store your key. You are responsible for storing it and providing it to your users by giving them the **License Acquisition URL** where you are storing the key.

PipelineId
The value of the `Id` object for the pipeline that you want Elastic Transcoder to use for transcoding. The pipeline determines several settings, including the Amazon S3 bucket from which Elastic Transcoder gets the files to transcode and the bucket into which Elastic Transcoder puts the transcoded files.

Status
If you specified more than one output for the job, the status of the entire job. When Elastic Transcoder starts processing a job, the value of `Job:Status` changes to `Progressing` and doesn't change until Elastic Transcoder has finished processing all outputs. When processing is complete, `Job:Status` changes either to `Complete` or, if any of the outputs failed, to `Error`.

If you specified only one output for the job, `Job:Status` is the same as `Outputs:Status`.

The value of `Job:Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

NextPageToken

A value that you use to access the second and subsequent pages of results, if any. When the jobs in the specified pipeline fit on one page or when you've reached the last page of results, the value of `NextPageToken` is `null`.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

The following example request creates a job.

Sample Request

The following example request gets a list of all of the jobs that you have ever created that have a status of `Complete`.

```
GET /2012-09-25/jobsByStatus/Complete?Ascending=true HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Sample Response

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Jobs":[
      {
         "Id":"3333333333333-abcde3",
         "Input":{
            "Key":"cooking/lasagna.mp4",

            "FrameRate":"auto",
            "Resolution":"auto",
            "AspectRatio":"auto",
            "Interlaced":"auto",
            "Container":"mp4"
         },
         "OutputKeyPrefix":"",
```

```
      "Outputs":[
         {
            "Id":"1",
            "Key":"mp4/lasagna-kindlefirehd.mp4",

            "ThumbnailPattern":"mp4/thumbnails/lasagna-{count}",
            "Rotate":"0",
            "PresetId":"1351620000000-100080",
            "Watermarks":[
               {
                  "InputKey":"logo/128x64.png",
                  "PresetWatermarkId":"company logo 128x64",

               }
            ],
            "Duration":"1003",
            "Width":"1280",
            "Height":"720",
            "Status":"Complete",
            "StatusDetail":"",
            "Captions":{
               "MergePolicy":"MergeOverride",
               "CaptionSources":[
                  {
                     "Key":"scc/lasagna-kindlefirehd.scc",

                     "Language":"en",
                     "Label":"English"
                  },
                  {
                     "Key":"srt/lasagna-kindlefirehd.srt",

                     "Language":"fr",
                     "TimeOffset":"1:00:00",
                     "Label":"French"
                  }
               ],
               "CaptionFormats":[
                  {
                     "Format":"scc",
                     "Pattern":"scc/lasagna-{language}"
                  },
                  {
                     "Format":"srt",
                     "Pattern":"srt/lasagna-{language}"
                  },
                  {
                     "Format":"mov-text"
                  }
               ]
            },
            "AppliedColorSpaceConversion":"None"
         },
         {
            "Id":"2",
            "Key":"iphone/lasagna-1024k",
            "ThumbnailPattern":"iphone/th1024k/lasagna-{count}",
            "Rotate":"0",
```

```
            "PresetId":"1351620000000-987654",
            "SegmentDuration":"5",
            "Duration":"1003",
            "Width":"1136",
            "Height":"640",
            "Status":"Complete",
            "StatusDetail":""
         },
      ],
      "PipelineId":"1111111111111-abcde1",
      "Playlists":[
         {
            "Format":"HLSv3",
            "Name":"playlist-iPhone-lasagna.m3u8",
            "OutputKeys":[
               "iphone/lasagna-1024k",
               "iphone/lasagna-512k"
            ]
         }
      ],
      "Status":"Complete"
   },
   {
      "Id":"4444444444444-abcde4",
      "Input":{
         "Key":"cooking/spaghetti.mp4",
         "FrameRate":"auto",
         "Resolution":"auto",
         "AspectRatio":"auto",
         "Interlaced":"auto",
         "Container":"mp4"
      },
      "Outputs":[
         {
            "Id":"3",
            "Key":"iphone/spaghetti-512k",
            "ThumbnailPattern":"iphone/th512k/spaghetti-{count}",
            "Rotate":"0",
            "PresetId":"1351620000000-456789",
            "SegmentDuration":"5",
            "Watermarks":[
               {
                  "InputKey":"logo/128x64.png",
                  "PresetWatermarkId":"company logo 128x64"
               }
            ],
            "Duration":"1003",
            "Width":"1136",
            "Height":"640",
            "Status":"Complete",
            "StatusDetail":""
         }
      ],
      "Playlists":[
         {
            "Format":"HLSv3",
            "Name":"playlist-iPhone-spaghetti.m3u8",
            "OutputKeys":[
```

```
                  "iphone/spaghetti-512k"
               ]
            }
         ],
         "UserMetadata":
            {
               "Food type":"Italian",
               "Cook book":"recipe notebook"
            },
         "Status":"Complete"
      }
   ],
   "NextPageToken":null
}
```

Read Job

Topics

- Description (p. 251)
- Requests (p. 251)
- Responses (p. 251)
- Errors (p. 267)
- Examples (p. 267)

Description

To get detailed information about a job, send a GET request to the `/2012-09-25/jobs/`*`jobId`* resource.

Requests

Syntax

```
GET /2012-09-25/jobs/jobId HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Request Parameter

This operation takes the following request parameter.

jobId
The identifier of the job for which you want to get detailed information.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

This operation does not have a request body.

Responses

Syntax

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
```

```
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Job":{
      "Id":"Id that Elastic Transcoder assigned to the job",
      "Input":{
         "Key":"name of the file to transcode",
         "Encryption":{
            "Mode":"AES-CBC-PKCS7|AES-CTR|AES-GCM",
            "Key":"encrypted and base64-encoded decryption key",
            "KeyMd5":"base64-encoded key digest",
            "InitializationVector":"base64-encoded initialization
               vector"
         },
         "FrameRate":"auto|10|15|23.97|24|25|29.97|30|50|60",
         "Resolution":"auto",
         "AspectRatio":"auto|1:1|4:3|3:2|16:9",
         "Interlaced":"auto|true|false",
         "Container":"auto|3gp|aac|asf|avi|divx|flv|m4a|mkv|mov|mp3|
            mp4|mpeg|mpeg-ps|mpeg-ts|mxf|ogg|vob|wav|webm"
      },
      "OutputKeyPrefix",
      "Outputs":[
         {
            "Id":"sequential counter",
            "Key":"name of the transcoded file",
            "Encryption":{
               "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                  AES-CTR|AES-GCM",
               "Key":"encrypted and base64-encoded encryption key",
               "KeyMd5":"base64-encoded key digest",
               "InitializationVector":"base64-encoded initialization
                  vector"
            },
            "ThumbnailPattern":""|"pattern",
            "Rotate":"auto|0|90|180|270",
            "PresetId":"PresetId for the job",
            "SegmentDuration":"[1,60]",
            "Watermarks":[
               {
                  "InputKey":"name of the .png or .jpg file",
                  "PresetWatermarkId":"value of Video:Watermarks:Id in
                     preset"
                  "Encryption":{
                     "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                        AES-CTR|AES-GCM",
                     "Key":"encrypted and base64-encoded encryption key",
                     "KeyMd5":"base64-encoded key digest",
                     "InitializationVector":"base64-encoded
                        initialization vector"
                  }
               },
               {...}
            ],
            "AlbumArt":[
               {
                  "AlbumArtMerge":"Replace|Prepend|Append|Fallback",
```

```
            "AlbumArtArtwork":"can be empty, but not null":[
               {
                  "AlbumArtInputKey":"name of the file to use as album
                     art",
                  "Encryption":{
                     "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                        AES-CTR|AES-GCM",
                     "Key":"encrypted and base64-encoded encryption
                        key",
                     "KeyMd5":"base64-encoded key digest",
                     "InitializationVector":"base64-encoded
                        initialization vector"
                  },
                  "AlbumArtMaxWidth":"maximum width of output album art
                     in pixels",
                  "AlbumArtMaxHeight":"maximum height of output album
                     art in pixels",
                  "AlbumArtSizingPolicy":"Fit|Fill|Stretch|Keep|
                     ShrinkToFit|ShrinkToFill",
                  "AlbumArtPaddingPolicy":"Pad|NoPad",
                  "AlbumArtFormat":"jpg|png"
               },
               {...}
            ]
         },
         {...}
      ],
      "Composition":[
         {
            "TimeSpan":
               {
                  "StartTime":"starting place of the clip, in
                     HH:mm:ss.SSS or sssss.SSS",
                  "Duration":"duration of the clip, in HH:mm:ss.SSS
                     or sssss.SSS"
               }
         }
      ],
      "Duration":"duration in seconds",
      "Width":"width in pixels",
      "Height":"height in pixels",
      "Status":"Submitted|In Progress|Complete|Canceled|Error",
      "StatusDetail":"additional information about job status",
      "Captions":{
         "MergePolicy":"MergeOverride|MergeRetain|Override",
         "CaptionSources":[
            {
               "Key":"name of the input caption file",
               "Encryption":{
                  "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                     AES-CTR|AES-GCM",
                  "Key":"encrypted and base64-encoded encryption key",
                  "KeyMd5":"base64-encoded key digest",
                  "InitializationVector":"base64-encoded
                     initialization vector"
               },
               "Language":"language of the input caption file",
               "TimeOffset":"starting place of the captions, in
```

```
                        either [-+]SS.sss or [-+]HH:mm:SS.ss",
                     "Label":"label for the caption"
                  },
                  {...}
               ],
               "CaptionFormats":[
                  {
                     "Format":"dfxp|mov-text|scc|srt|webvtt",
                     "Pattern":"myCaption/file-language",
                     "Encryption":{
                        "Mode":"S3|S3-AWS-KMS|AES-CBC-PKCS7|
                           AES-CTR|AES-GCM",
                        "Key":"encrypted and base64-encoded encryption key",
                        "KeyMd5":"base64-encoded key digest",
                        "InitializationVector":"base64-encoded
                           initialization vector"
                     }
                  },
                  {...}
               ]
            },
              "AppliedColorSpaceConversion":"None|Bt601ToBt709|
                 Bt709ToBt601"
         },
         {...}
      ],
      "Playlists":[
         {
            "Format":"HLSv3|HLSv4|Smooth",
            "Name":"name",
            "OutputKeys":[
               "Outputs:Key to include in this playlist",
               ...
            ],
            "HlsContentProtection":{
               "Method":"aes-128",
               "Key":"encrypted and base64-encoded protection key",
               "KeyMd5":"base64-encoded key digest",
               "InitializationVector":"base64-encoded
                        initialization vector",
               "LicenseAcquisitionUrl":"license aquisition url",
               "KeyStoragePolicy":"NoStore|WithVariantPlaylists"
            }
         },
         {...}
      ],
      "UserMetadata":
         {
             "Key":"Value",
             "Second user metadata key":"Second user metadata value"
         },
      "PipelineId":"Id of the pipeline that transcoded the job",
      "Status":"Submitted|Progressing|Complete|Canceled|Error"
   }
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

For each job that satisfies the search criteria, the response body contains the values that you specified when you created the job. For more information, see Read Job (p. 251).

Id
The identifier that Elastic Transcoder assigned to the job. You use this value to get settings for the job or to delete the job.

Input
Information about the file that Elastic Transcoder transcoded. These are values that you specified when you created the job.

Input:Key
The name of the file that you want to transcode. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/lasagna.mpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

(Optional) Input:Encryption
The encryption settings, if any, that are used for decrypting your input files. If your input file is encrypted, you must specify the mode that Elastic Transcoder will use to decrypt your file.

(Required for Encryption) Input:Encryption:Mode
The specific encryption mode that you want Elastic Transcoder to use when decrypting your files.

Elastic Transcoder supports the following options:

- **Amazon S3 Server-Side Encryption:** Amazon S3 handles the encryption and decryption of your files. As long as Elastic Transcoder has access permissions to your Amazon S3 bucket, you don't need to take any action.

  For more information, see Protecting Data Using Server-Side Encryption in the *Amazon Simple Storage Service Developer Guide*.
- **Server-Side Encryption Using Customer-Provided Keys:** Elastic Transcoder supports three types of encryption using customer-provided keys:
  - **AES-CBC-PKCS7:** A padded cipher-block mode of operation.
  - **AES-CTR:** AES Counter Mode.
  - **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

  If you chose one of the AES-encryption modes, you must also specify the following three values (all three must be base64-encoded):
  - **Encryption Key**
  - **Encryption Key MD5**
  - **Encryption Initialization Vector**

(Optional) Input:Encryption:Key
The data encryption key used to encrypt your file. The key must be base64-encoded and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

The key must also be encrypted by using AWS KMS. For more information, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Input:Encryption:KeyMd5
The MD5 digest of the key used to encrypt your input file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

(Optional) Input:Encryption:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you used to encrypt your input files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

For more information, go to Initialization Vector.

(Optional) Input:FrameRate
The frame rate of the input file. If you want Elastic Transcoder to automatically detect the frame rate of the input file, specify `auto`. If you want to specify the frame rate for the input file, enter one of the following values:

`10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `50`, `60`

The default value is `auto`.

(Optional) Input:Resolution
The resolution, in pixels, of the input file. This value must be `auto`, which causes Elastic Transcoder to automatically detect the resolution of the input file.

(Optional) Input:AspectRatio
The aspect ratio of the input file. If you want Elastic Transcoder to automatically detect the aspect ratio of the input file, specify `auto`. If you want to specify the aspect ratio for the output file, enter one of the following values:

`1:1`, `4:3`, `3:2`, `16:9`

The default value is `auto`.

(Optional) Input:Interlaced
Whether the input file is interlaced. If you want Elastic Transcoder to automatically detect whether the input file is interlaced, specify `auto`. If you want to specify whether the input file is interlaced, enter one of the following values:

`true`, `false`

The default value is `auto`.

(Optional) Input:Container
The container type for the input file. If you want Elastic Transcoder to automatically detect the container type of the input file, specify `auto`. If you want to specify the container type for the input file, enter one of the following values:

`3gp`, `aac`, `asf`, `avi`, `divx`, `flv`, `m4a`, `mkv`, `mov`, `mp3`, `mp4`, `mpeg`, `mpeg-ps`, `mpeg-ts`, `mxf`, `ogg`, `vob`, `wav`, `webm`

OutputKeyPrefix
The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, and playlists. If you specify a value, it must contain a `/` somewhere after the first character, which simplifies Amazon S3 file management.

Outputs
Information about the output files. You can create a maximum of 30 outputs per job. If you specify more than one output for a job, Elastic Transcoder creates the files for each output in the order in which you specify them in the job. The `Outputs:Id` object identifies the position of an output in the sequence.

Note
In early versions of Elastic Transcoder, you could create just one output per job, so the object name was `Output`. The `Output` syntax still works, but we recommend that you use the `Outputs` syntax for all jobs, even when you want Elastic Transcoder to transcode a file into only one format. Do not use both the `Outputs` and `Output` syntaxes in the same request.

Outputs:Id
A sequential counter, starting with 1, that identifies an output among the outputs from the current job. In the `Output` syntax, this value is always `1`.

Outputs:Key
The name that you want Elastic Transcoder to assign to the transcoded file and playlist. Elastic Transcoder saves the file or files in the Amazon S3 bucket specified by the `OutputBucket` object in the pipeline that you specify in `PipelineId`.

If the bucket already contains a file that has the specified name, the output fails. In the `Create Job` response, the value of `Outputs:Status` for that output will be `Error`, as will the final value of Status for the job. However, other outputs in the same job may succeed.

The format for file names depends the container type and whether the segment duration is set. If the container type is not `ts` or the segment duration is not provided, the name of the output file is a concatenation of `OutputKeyPrefix` and `Key`.

If the container type is `ts` and segment duration is provided, Elastic Transcoder uses the value of `Key` to name both the playlist for the output and the `.ts` files:

- **Playlist:**
  - **HLSv3:** The file name is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `.m3u8`:

    OutputKeyPrefix`Key`*.m3u8*
  - **HLSv4:** The file name is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `_v4.m3u8`. Video outputs create a second file with a file name that is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `_iframe.m3u8`:

    OutputKeyPrefix`Key`*_v4.m3u8*

    OutputKeyPrefix`Key`*_iframe.m3u8*
- **Segment (.ts) files:**
  - **HLSv3:** The file name is a concatenation of `OutputKeyPrefix` and `Key`, plus a five-digit sequential counter beginning with `00000`, and the file name extension `.ts`:

    OutputKeyPrefix`Key`*00000.ts*
  - **HLSv4:** The file name is a concatenation of `OutputKeyPrefix` and `Key` plus the file name extension `.ts`:

    OutputKeyPrefix`Key`*.ts*

If a segmented ts output is not included in a master playlist, Elastic Transcoder treats the output as `HLSv3`.

Note
Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for `HLSv3` playlists, `_v4.m3u8` and `_iframe.m3u8` for `HLSv4` playlists). If you include a file extension in `Output Key`, the file name will have two extensions.

`OutputKeyPrefix` groups all of the files for a job together in your Amazon S3 bucket. If you want to group the files for each output within a job, you can include a prefix in the value of `Key`, for example:

*OutputKeyPrefix*iPhone/Key*00000.ts*

*OutputKeyPrefix*KindleFireHD/Key*00000.ts*

(Optional) Outputs:Encryption
The encryption settings, if any, that you want Elastic Transcoder to apply to your output files. If you choose to use encryption, you must specify a mode to use. If you choose not to use encryption, Elastic Transcoder will write an unencrypted file to your Amazon S3 bucket.

(Required for Encryption) Outputs:Encryption:Mode
The specific server-side encryption mode that you want Elastic Transcoder to use when encrypting your output files individually. Elastic Transcoder supports the following **Encryption Mode** options:

- **S3:** Amazon S3 creates and manages the keys used for encrypting your files.

  For more information, see Protecting Data Using Server-Side Encryption in the *Amazon Simple Storage Service Developer Guide*.
- **S3-AWS-KMS:** Amazon S3 calls AWS KMS, which creates and manages the keys that are used for encrypting your files. If you specify **S3-AWS-KMS** and you don't want to use the default key, you must add the AWS-KMS key that you want to use to your pipeline.

  For more information, see Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys in the *Amazon Simple Storage Service Developer Guide*.
- **AES-CBC-PKCS7:** A padded cipher-block mode of operation.
- **AES-CTR:** AES Counter Mode.
- **AES-GCM:** AES Galois Counter Mode, a mode of operation that is an authenticated encryption format, meaning that a file, key, or initialization vector that has been tampered with will fail the decryption process.

If you chose one of the AES-encryption modes, you must also specify the following three values (all three must be base64-encoded):

- **Encryption Key**
- **Encryption Key MD5**
- **Encryption Initialization Vector**

If you chose one of the AES-encryption modes, and you want Elastic Transcoder to generate a 128-bit bit AES encryption key for you, do not specify values for the **Encryption Key**, **Encryption Key MD5**, or **Encryption Initialization Vector**.

> **Important**
> For the AES modes, your media-specific private encryption keys and your unencrypted data are never stored by AWS; therefore, it is important that you safely manage your encryption keys. If you lose them, you won't be able to decrypt your data.

(Optional) Outputs:Encryption:Key
If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

128, 192, or 256.

For more information about encrypting your key with AWS KMS, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Outputs:Encryption:KeyMd5
The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

(Optional) Outputs:Encryption:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

For more information, go to Initialization Vector.

(Optional) Outputs:ThumbnailPattern
Whether you want Elastic Transcoder to create thumbnails for your videos and, if so, how you want Elastic Transcoder to name the files.

If you don't want Elastic Transcoder to create thumbnails, specify `""`.

If you do want Elastic Transcoder to create thumbnails, specify the information that you want to include in the file name for each thumbnail. You can specify the following values in any sequence:

- **`{count}` (Required):** If you want to create thumbnails, you must include `{count}` in the `ThumbnailPattern` object. Wherever you specify `{count}`, Elastic Transcoder adds a five-digit sequence number (beginning with **00001**) to thumbnail file names. The number indicates where a given thumbnail appears in the sequence of thumbnails for a transcoded file.

  Important
  If you specify a literal value and/or `{resolution}` but you omit `{count}`, Elastic Transcoder returns a validation error and does not create the job.

- **(Optional) Literal values:** You can specify literal values anywhere in the `ThumbnailPattern` object, for example, as a file name prefix or as a delimiter between `{resolution}` and `{count}`.
- **(Optional) `{resolution}`:** If you want Elastic Transcoder to include the resolution in the file name, include `{resolution}` in the `ThumbnailPattern` object.

When creating thumbnails, Elastic Transcoder automatically saves the files in the format (`.jpg` or `.png`) that appears in the preset that you specified in `PresetId`. Elastic Transcoder also appends the applicable file name extension.

As with `Outputs:Key`, you can include a prefix in `ThumbnailPattern` that groups the applicable files together, for example, all of the thumbnails for one video in one format, or all of the thumbnails with the corresponding output file.

(Optional) Outputs:Rotate
The number of degrees clockwise by which you want Elastic Transcoder to rotate the output relative to the input. Valid values include:

`auto`, `0`, `90`, `180`, `270`

The value `auto` generally works only if the file that you're transcoding contains rotation metadata.

Outputs:PresetId
The value of the `Id` object for the preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding. To use a preset that you created, specify the preset ID that Elastic Transcoder returned in the response when you created the preset.

Note
If you created any presets before AAC profiles were added, Elastic Transcoder uses the AAC-LC profile for those presets.

For a list of system presets, see System Presets (p. 70) (You can also get these IDs using List Presets (p. 296).)

(Fragmented MP4/MPEG-TS Outputs Only) Outputs:SegmentDuration
If you specify a preset for the current output for which the value of `Container` is either **ts** (MPEG-TS) or **fmp4** (Fragmented MP4), `SegmentDuration` is the target maximum duration of each segment

in seconds. For `HLSv3` format playlists, each media segment is stored in a separate `.ts` file. For `HLSv4` and `Smooth` playlists, all media segments for an output are stored in a single file. Each segment is approximately the length of the `SegmentDuration`, though individual segments might be shorter or longer.

The range of valid values is 1 to 60 seconds. If the duration of the video is not evenly divisible by `SegmentDuration`, the duration of the last segment is the remainder of:

```
total length/SegmentDuration
```

Elastic Transcoder creates an output-specific playlist for each `HLS` output that you specify in `OutputKeys`. To add an output to a master playlist for this job, include it in Outputs in Master Playlist.

Elastic Transcoder applies this segmenting to any captions associated with the output video.

For more information, see HTTP Live Streaming.

(Video Only) Outputs:Watermarks

Information about the watermarks that you want Elastic Transcoder to add to the video or artwork during transcoding. You can specify up to four watermarks for each output. Settings for each watermark must be defined in the preset that you specify in `Outputs:PresetId` for the current output.

Watermarks are added to the output file in the sequence in which you list them in the job output—the first watermark in the list is added to the output file first, the second watermark in the list is added next, and so on. As a result, if the settings in a preset cause Elastic Transcoder to place all watermarks in the same location, the second watermark that you list in `Outputs:Watermarks` will cover the first one, the third one will cover the second, and the fourth one will cover the third.

For more information about watermarks, see Watermarks (p. 69).

(Video Only) Outputs:Watermarks:InputKey

The name of the `.png` or `.jpg` file that you want to use for the watermark. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `logos/128x64.png`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

(Video Only) Outputs:Watermarks:PresetWatermarkId

The ID of the watermark settings that Elastic Transcoder uses to add watermarks to the file during transcoding. The settings are in the preset specified by `Outputs:PresetId` for the current output. In that preset, the value of `Watermarks:Id` tells Elastic Transcoder which settings to use.

(MP3/MP4 Only) Outputs:AlbumArt

The album art to be associated with the output file, if any.

To remove artwork or leave the artwork empty, you can either set `Artwork` to null, or set the `MergePolicy` to `Replace` and use an empty `Artwork` array.

To pass through existing artwork unchanged, set the `MergePolicy` to `Prepend`, `Append`, or `Fallback`, and use an empty `Artwork` array.

> **Note**
> Album Art is available only for containers of type `mp3` or `mp4`.

(MP3/MP4 Only) Outputs:AlbumArt:MergePolicy

A policy that determines how Elastic Transcoder handles the existence of multiple album artwork files.

- **Replace:** The specified album art replaces any existing album art.
- **Prepend:** The specified album art is placed in front of any existing album art.
- **Append:** The specified album art is placed after any existing album art.

- **Fallback:** If the input file contains artwork, Elastic Transcoder uses that artwork for the output. If the input does not contain artwork, Elastic Transcoder uses the specified album art file.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork
The file to be used as album art. There can be multiple artworks associated with an audio file, to a maximum of 20.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:InputKey
The name of the file to be used as album art. To determine which Amazon S3 bucket contains the specified file, Elastic Transcoder checks the pipeline specified by `PipelineId`; the `InputBucket` object in that pipeline identifies the bucket.

If the file name includes a prefix, for example, `cooking/pie.jpg`, include the prefix in the key. If the file isn't in the specified bucket, Elastic Transcoder returns an error.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:MaxWidth
The maximum width of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:MaxHeight
The maximum height of the output album art in pixels. If you specify `auto`, Elastic Transcoder uses 600 as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:SizingPolicy
A value that controls scaling of the output album art:

- **Fit:** Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill:** Elastic Transcoder scales the output art so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output art and then crops it to the dimension (if any) that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches the output art to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input art and the output art are different, the output art will be distorted.
- **Keep:** Elastic Transcoder does not scale the output art. If either dimension of the input art exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output art.
- **ShrinkToFit:** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the art up.
- **ShrinkToFill:** Elastic Transcoder scales the output art down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the art up.

The following table shows possible effects of `SizingPolicy` settings on the output album art:

SizingPolicy	Output Album Art May Be Scaled Up	Output Album Art May Be Padded When Padding Policy Is "Pad"	Output Album Art May Have a Different Pixel Aspect Ratio than Input Album Art	Output Album Art May Be Cropped
Fit	Yes	Yes		
Fill	Yes			Yes
Stretch	Yes		Yes	

SizingPolicy	Output Album Art May Be Scaled Up	Output Album Art May Be Padded When Padding Policy Is "Pad"	Output Album Art May Have a Different Pixel Aspect Ratio than Input Album Art	Output Album Art May Be Cropped
Keep	Yes	Yes		Yes
ShrinkToFit		Yes		
ShrinkToFill		Yes		Yes

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:PaddingPolicy
When you set `PaddingPolicy` to `Pad`, Elastic Transcoder might add white bars to the top and bottom and/or left and right sides of the output album art to make the total size of the output art match the values that you specified for `MaxWidth` and `MaxHeight`. For more information, see the table at `AlbumArt:Art:SizingPolicy`.

(MP3/MP4 Only) Outputs:AlbumArt:Artwork:AlbumArtFormat
The format of album art, if any. Valid formats are `jpg` and `png`.

Outputs:Composition
Using a composition, you can create an output file that contains an excerpt from the input file. This excerpt, called a clip, can come from the beginning, middle, or end of the file. The `Composition` object contains settings for the clips that make up an output file.

Currently, you can specify settings for only a single clip per output file. The `Composition` object cannot be null. All jobs in a playlist must have the same clip settings.

Outputs:Composition:TimeSpan
Settings that determine when a clip begins and how long it lasts.

Outputs:Composition:TimeSpan:StartTime
The place in the input file where you want a clip to start. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder starts at the beginning of the input file.

Outputs:Composition:TimeSpan:Duration
The duration of the clip. The format can be either HH:mm:ss.SSS (maximum value: 23:59:59.999; SSS is thousandths of a second) or sssss.SSS (maximum value: 86399.999). If you don't specify a value, Elastic Transcoder creates an output file from StartTime to the end of the file.

If you specify a value longer than the duration of the input file , Elastic Transcoder transcodes the file and returns a warning message.

Outputs:Duration
Duration of the output file, in seconds.

Outputs:Width
Width of the output file, in pixels.

Outputs:Height
Height of the output file, in pixels.

Outputs:Status
The status of one output in a job. If you specified only one output for the job, `Outputs:Status` is always the same as `Job:Status`. If you specified more than one output:

- `Job:Status` and `Outputs:Status` for all of the outputs is `Submitted` until Elastic Transcoder starts to process the first output.
- When Elastic Transcoder starts to process the first output, `Outputs:Status` for that output and `Job:Status` both change to `Progressing`. For each output, the value of `Outputs:Status` remains `Submitted` until Elastic Transcoder starts to process the output.

- `Job:Status` remains `Progressing` until all of the outputs reach a terminal status, either `Complete` or `Error`.
- When all of the outputs reach a terminal status, `Job:Status` changes to `Complete` only if `Outputs:Status` for all of the outputs is `Complete`. If `Outputs:Status` for one or more outputs is `Error`, the terminal status for `Job:Status` is also `Error`.

The value of `Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

Outputs:StatusDetail
Information that further explains `Outputs:Status`.

(Video Only) Outputs:Captions
You can configure Elastic Transcoder to transcode captions, or subtitles, from one format to another. All captions must be in UTF-8. Elastic Transcoder supports two types of captions:

- **Embedded:** Embedded captions are included in the same file as the audio and video. Elastic Transcoder supports only one embedded caption per language, to a maximum of 300 embedded captions per file.

  Valid input values include: `CEA-608 (EIA-608`, first non-empty channel only), `CEA-708 (EIA-708`, first non-empty channel only), and `mov-text`

  Valid outputs include: `mov-text`

  Elastic Transcoder supports a maximum of one embedded format per output.
- **Sidecar:** Sidecar captions are kept in a separate metadata file from the audio and video data. Sidecar captions require a player that is capable of understanding the relationship between the video file and the sidecar file. Elastic Transcoder supports only one sidecar caption per language, to a maximum of 20 sidecar captions per file.

  Valid input values include: `dfxp` (first div element only), `ebu-tt`, `scc`, `smpt`, `srt`, `ttml` (first div element only), and `webvtt`

  Valid outputs include: `dfxp` (first div element only), `scc`, `srt`, and `webvtt`.

If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

Elastic Transcoder does not support OCR (Optical Character Recognition), does not accept pictures as a valid input for captions, and is not available for audio-only transcoding. Elastic Transcoder does not preserve text formatting (for example, italics) during the transcoding process.

To remove captions or leave the captions empty, set `Captions` to null. To pass through existing captions unchanged, set the `MergePolicy` to `MergeRetain`, and pass in a null `CaptionSources` array.

For more information about embedded files, see the Subtitle (caption) Wikipedia page.

For more information about sidecar files, see the Metadata Platform and Sidecar file Wikipedia pages.

(Video Only) Outputs:Captions:MergePolicy
A policy that determines how Elastic Transcoder handles the existence of multiple captions.

- **MergeOverride:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the sidecar captions and ignores the embedded captions for that language.
- **MergeRetain:** Elastic Transcoder transcodes both embedded and sidecar captions into outputs. If captions for a language are embedded in the input file and also appear in a sidecar file, Elastic Transcoder uses the embedded captions and ignores the sidecar captions for that language. If **CaptionSources** is empty, Elastic Transcoder omits all sidecar captions from the output files.
- **Override:** Elastic Transcoder transcodes only the sidecar captions that you specify in `CaptionSources`.

`MergePolicy` cannot be null.

(Video/Sidecar Only, Optional) Outputs:Captions:CaptionSources
Source files for the input sidecar captions used during the transcoding process. To omit all sidecar captions, leave `CaptionSources` blank.

(Video Only) Outputs:Captions:CaptionSources:Key
The name of the sidecar caption file that you want Elastic Transcoder to include with the outputs.

(Video Only) Outputs:Captions:CaptionSources:Language
A string that specifies the language of the caption. Specify this as one of:

- 2-character ISO 639-1 code
- 3-character ISO 639-2 code

For more information about ISO language codes, see List of ISO 639-1 codes.

(Video Only, Optional) Outputs:Captions:CaptionSources:TimeOffset
For clip generation or captions that do not start at the same time as the associated video file, the `TimeOffset` tells Elastic Transcoder how much of the video to encode before including captions.

Specify the TimeOffset in the form [+-]SS.sss or [+-]HH:mm:SS.ss.

(Video Only, Optional) Outputs:Captions:CaptionSources:Label
The label of the caption shown in the player when choosing a language. We recommend that you put the caption language name here, in the language of the captions.

(Video Only) Outputs:Captions:CaptionFormats
The file format of the output captions. If you leave this value blank, Elastic Transcoder returns an error.

(Video Only) Outputs:Captions:CaptionFormats:Format
The format you specify determines whether Elastic Transcoder generates an embedded or sidecar caption for this output.

- **Embedded Caption Formats:**

Container Type	Supported Embedded types
MP4	mov-text
Non-MP4 containers	Embedded captions are not supported

Elastic Transcoder supports a maximum of one embedded format per output.

- **Sidecar Caption Formats:** Elastic Transcoder supports dfxp (first div element only), scc, srt, and webvtt. If you want ttml or smpte-tt compatible captions, specify dfxp as your output format.

Container Type	Supported Sidecar types
fmp4	dfxp
Non-fmp4 containers	All sidecar types

Note
`fmp4` captions have an extension of `.ismt`

(Video/Sidecar Only) Outputs:Captions:CaptionFormats:Pattern
The prefix for caption filenames, in the form *`description`*-`{language}`, where:

- *`description`* is a description of the video.
- `{language}` is a literal value that Elastic Transcoder replaces with the two- or three-letter code for the language of the caption in the output file names.

If you don't include `{language}` in the file name pattern, Elastic Transcoder automatically appends "`{language}`" to the value that you specify for the description. In addition, Elastic Transcoder automatically appends the count to the end of the segment files.

For example, suppose you're transcoding into srt format. When you enter "Sydney-{language}-sunrise", and the language of the captions is English (en), the name of the first caption file will be `Sydney-en-sunrise00000.srt`.

(Automatic) Outputs:AppliedColorSpaceConversion

If Elastic Transcoder used a preset with a `ColorSpaceConversionMode` to transcode the output file, the `AppliedColorSpaceConversion` parameter shows the conversion used. If no `ColorSpaceConversionMode` was defined in the preset, this parameter will not be included in the job response.

For more information about `ColorSpaceConversionMode`, see ColorSpaceConversion.

(Optional) Outputs:UserMetadata

User-defined metadata that you want to associate with an Elastic Transcoder job. You specify metadata in `key/value` pairs. You can use the `key/value` pairs to track details about a file, for example, `Season 1: Episode 3`.

You can add up to 10 key/value pairs to each job. Elastic Transcoder does not guarantee that `key/value` pairs are returned in the same order in which you specify them.

Outputs:UserMetadata:Key

The key of the metadata `key/value` pair that you want returned with the output file. Each key must be a unique string between `1-128` characters, and must use only characters from the following list:

- `0-9`
- `A-Z` and `a-z`
- `Space`
- The following symbols: `_.:/=+-%@`

You can use keys as a numbering system for organizing your metadata, for storing an extra 128 characters of metadata, or for labeling the metadata stored in the **value**. If you want to use only value metadata, you can put throw-away strings in your keys such as `key1`, and ignore the keys when you retrieve your metadata from Elastic Transcoder.

> **Important**
> You must specify unique strings for all of the keys in a job. If the same string is used for more than one key in a job, Elastic Transcoder returns only one of the key/value pairs that use that key. There is no way to guarantee which value is returned.

Outputs:UserMetadata:Value

The value of the metadata `key/value` pair that you want returned with your job. Each value must be a string between `0-256` characters, and must use only characters from the following list:

- `0-9`
- `A-Z` and `a-z`
- `Space`
- The following symbols: `_.:/=+-%@`

(Fragmented MP4/MPEG-TS Outputs Only) Playlists

If you specify a preset in `PresetId` for which the value of `Container` is either **ts** (MPEG-TS) or **fmp4** (Fragmented MP4), `Playlists` contains information about the master playlists that you want Elastic Transcoder to create.

We recommend that you create at most one master playlist per playlist format. The maximum number of master playlists in a job is 30.

Playlists:Format

The format of the output playlist. Valid formats include `HLSv3`, `HLSv4`, and `Smooth`.

Playlists:Name

The name that you want Elastic Transcoder to assign to a master playlist, for example, `nyc-vacation.m3u8`. If the name includes a `/` character, the section of the name before the `/` must be identical for all `Name` objects. If you create more than one master playlist, the values of all `Name` objects must be unique.

> **Note**
> Elastic Transcoder automatically appends the relevant file extension to the file name (`.m3u8` for `HLSv3` playlists, `_v4.m3u8` and `_iframe.m3u8` for `HLSv4` playlists, and `.ism` and `.ismc` for `Smooth` playlists). If you include a file extension in `Name`, the file name will have two extensions.

Any segment duration settings, clip settings, or caption settings must be the same for all outputs in the playlist. For `Smooth` playlists, the `Audio:Profile`, `Video:Profile`, and `Video:FrameRate` to `Video:KeyframesMaxDist` ratio must be the same for all outputs. For more information, see KeyframesMaxDist (p. 282).

Playlists:OutputKeys

For each output in this job that you want to include in a master playlist, the value of the `Outputs:Key` object. If you include more than one output in a playlist, the value of `SegmentDuration` for all of the outputs must be the same.

For `HLSv4` master playlists, Elastic Transcoder chooses which combinations of audio and video inputs will be linked in the output playlists. The first audio and video inputs will be linked and rendered as the default playback experience, allowing you to choose your preferred playback default. For other individual playlists in the master playlist, Elastic Transcoder chooses which audio and video bit rate combinations will provide the best playback.

(Optional) Outputs:Playlists:HlsContentProtection

The HLS content protection settings, if any, that you want Elastic Transcoder to apply to your output files. If you want to use HLS content protection do not specify encryption settings for the output file or captions. HLS content protection encrypts each segment of a file so that they can be streamed encrypted and only decrypted on playback, while the output file and caption encryptions encrypt the file all at once. Elastic Transcoder does not support files that are encrypted both ways.

Outputs:Playlists:HlsContentProtection:Method

The content protection method for your output. The only valid value is:

`aes-128`.

This value will be written into the `method` attribute of the `EXT-X-KEY` metadata tag in the output playlist.

(Optional) Outputs:Playlists:HlsContentProtection:Key

If you want Elastic Transcoder to generate a key for you, leave this field blank.

If you choose to supply your own key, you must encrypt the key by using AWS KMS. The key must be base64-encoded, and it must be one of the following bit lengths before being base64-encoded:

`128`, `192`, or `256`.

For more information about encrypting your key with AWS KMS, see Encrypting and Decrypting Data in the *AWS Key Management Service Developer Guide*.

(Optional) Outputs:Playlists:HlsContentProtection:KeyMd5

The MD5 digest of the key that you want Elastic Transcoder to use to encrypt your output file, and that you want Elastic Transcoder to use as a checksum to make sure your key was not corrupted in transit. The key MD5 must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

(Optional) Outputs:Playlists:HlsContentProtection:InitializationVector
The series of random bits created by a random bit generator, unique for every encryption operation, that you want Elastic Transcoder to use to encrypt your output files. The initialization vector must be base64-encoded, and it must be exactly 16 bytes before being base64-encoded.

If Elastic Transcoder is generating your key for you, you must leave this field blank.

For more information, go to Initialization Vector.

Outputs:Playlists:HlsContentProtection:LicenseAcquisitionUrl
The location of the license key required to decrypt your HLS playlist. The URL must be an absolute path, and is referenced in the URI attribute of the EXT-X-KEY metadata tag in the playlist file. For example:

```
https://www.example.com/exampleKey/
```

Outputs:Playlists:HlsContentProtection:KeyStoragePolicy
Specify whether you want Elastic Transcoder to write your HLS license key to an Amazon S3 bucket. If you choose `WithVariantPlaylists`, Elastic Transcoder will write your encrypted key into the same Amazon S3 bucket as the associated playlist.

> **Important**
> If you chose `NoStore`, Elastic Transcoder will not store your key. You are responsible for storing it and providing it to your users by giving them the **License Acquisition URL** where you are storing the key.

PipelineId
The value of the `Id` object for the pipeline that you want Elastic Transcoder to use for transcoding. The pipeline determines several settings, including the Amazon S3 bucket from which Elastic Transcoder gets the files to transcode and the bucket into which Elastic Transcoder puts the transcoded files.

Status
If you specified more than one output for the job, the status of the entire job. When Elastic Transcoder starts processing a job, the value of `Job:Status` changes to `Progressing` and doesn't change until Elastic Transcoder has finished processing all outputs. When processing is complete, `Job:Status` changes either to `Complete` or, if any of the outputs failed, to `Error`.

If you specified only one output for the job, `Job:Status` is the same as `Outputs:Status`.

The value of `Job:Status` is one of the following: `Submitted`, `Progressing`, `Complete`, `Canceled`, or `Error`.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

Sample Request

The following example request gets the job that has the job ID 3333333333333-abcde3.

```
GET /2012-09-25/jobs/3333333333333-abcde3 HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
```

```
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Sample Response

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Job":{
      "Id":"3333333333333-abcde3",
      "Input":{
         "Key":"cooking/lasagna.mp4",

         "FrameRate":"auto",
         "Resolution":"auto",
         "AspectRatio":"auto",
         "Interlaced":"auto",
         "Container":"mp4"
      },
      "OutputKeyPrefix":"",
      "Outputs":[
         {
            "Id":"1",
            "Key":"mp4/lasagna-kindlefirehd.mp4",

            "ThumbnailPattern":"mp4/thumbnails/lasagna-{count}",
            "Rotate":"0",
            "PresetId":"1351620000000-100080",
            "Watermarks":[
               {
                  "InputKey":"logo/128x64.png",
                  "PresetWatermarkId":"company logo 128x64",

               }
            ],
            "Duration":"1003",
            "Width":"1280",
            "Height":"720",
            "Status":"Progressing",
            "StatusDetail":"",
            "Captions":{
               "MergePolicy":"MergeOverride",
               "CaptionSources":[
                  {
                     "Key":"scc/lasagna-kindlefirehd.scc",

                     "Language":"en",
                     "Label":"English"
```

```
            },
            {
               "Key":"srt/lasagna-kindlefirehd.srt",

               "Language":"fr",
               "TimeOffset":"1:00:00",
               "Label":"French"
            }
         ],
         "CaptionFormats":[
            {
               "Format":"scc",
               "Pattern":"scc/lasagna-{language}",

            },
            {
               "Format":"srt",
               "Pattern":"srt/lasagna-{language}",

            },
            {
               "Format":"mov-text"
            }
         ]
      }
   },
   {
      "Id":"2",
      "Key":"iphone/lasagna-1024k",
      "ThumbnailPattern":"iphone/th1024k/lasagna-{count}",
      "Rotate":"0",
      "PresetId":"1351620000000-987654",
      "SegmentDuration":"5",
      "Duration":"1003",
      "Width":"1136",
      "Height":"640",
      "Status":"Progressing",
      "StatusDetail":"",
      "AppliedColorSpaceConversion":"None"
   },
   {
      "Id":"3",
      "Key":"iphone/lasagna-512k",
      "ThumbnailPattern":"iphone/th512k/lasagna-{count}",
      "Rotate":"0",
      "PresetId":"1351620000000-456789",
      "SegmentDuration":"5",
      "Duration":"1003",
      "Width":"1136",
      "Height":"640",
      "Status":"Complete",
      "StatusDetail":""
   }
],
"Playlists":[
   {
      "Format":"HLSv3",
      "Name":"playlist-iPhone-lasagna.m3u8",
```

```
            "OutputKeys":[
               "iphone/lasagna-1024k",
               "iphone/lasagna-512k"
            ]
         }
      ],
      "UserMetadata":
         {
            "Food type":"Italian",
            "Cook book":"recipe notebook"
         },
      "PipelineId":"1111111111111-abcde1",
      "Status":"Progressing"
   }
}
```

Cancel Job

Topics

- Description (p. 271)
- Requests (p. 271)
- Responses (p. 272)
- Errors (p. 272)
- Examples (p. 272)

Description

To cancel a job that Elastic Transcoder has not begun to process, send a DELETE request to the `/2012-09-25/jobs/`*`jobId`* resource. Elastic Transcoder might still return the job if you submit a `Read Job`, `List Job by Pipeline`, or `List Job by Status` request, but Elastic Transcoder won't transcode the input file, and you won't be charged for the job.

> **Note**
> You can only cancel a job that has a status of **Submitted**. To prevent a pipeline from starting to process a job while you're getting the job identifier, use Update Pipeline Status (p. 167) to temporarily pause the pipeline.

Requests

Syntax

```
DELETE /2012-09-25/jobs/jobId HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Request Parameter

This operation takes the following request parameter.

jobId
The identifier of the job that you want to cancel.

To get a list of the jobs (including their `jobId`) that have a status of `Submitted`, use the List Jobs by Status (p. 230) API action.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

This operation does not have a request body.

Responses

Syntax

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Success":"true"
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

The response body contains the following JSON object.

Success
If the job is successfully canceled, the value of `Success` is `true`.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

The following example request cancels the job that has the ID `3333333333333-abcde3`.

Sample Request

```
DELETE /2012-09-25/jobs/3333333333333-abcde3 HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Sample Response

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT
{
   "Success":"true"
}
```

Preset Operations

Topics

- Create Preset (p. 275)
- List Presets (p. 296)
- Read Preset (p. 315)
- Delete Preset (p. 334)

Presets are templates that contain most of the settings for transcoding media files from one format to another. Elastic Transcoder includes some default presets for common formats, for example, several iPod and iPhone versions. You can also create your own presets for formats that aren't included among the default presets. You specify which preset you want to use when you create a job.

This section describes operations that you can perform on presets using the Elastic Transcoder API. For more information about presets, including how to perform the same operations using the Elastic Transcoder console, see Working with Presets (p. 39).

Create Preset

Topics

- Description (p. 275)
- Requests (p. 275)
- Responses (p. 291)
- Errors (p. 293)
- Examples (p. 293)

Description

To create a preset, send a POST request to the `/2012-09-25/presets` resource.

> **Important**
> Elastic Transcoder checks the settings that you specify to ensure that they meet Elastic Transcoder requirements and to determine whether they comply with the applicable standard. If your settings are not valid for Elastic Transcoder, Elastic Transcoder returns an HTTP 400 response (`ValidationException`) and does not create the preset. If the settings are valid for Elastic Transcoder but aren't strictly compliant with the standard, Elastic Transcoder creates the preset and returns a warning message in the response. This helps you determine whether your settings comply with the standard while giving you greater flexibility with respect to the file that Elastic Transcoder produces.

> **Note**
> Elastic Transcoder includes several system presets for common output formats. For more information, see PresetId in the topic Create Job (p. 181).

Requests

Syntax

```
POST /2012-09-25/presets HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
   "Name":"preset name",
   "Description":"preset description",
   "Container":"flv|fmp4|gif|mp3|mp4|mpg|ogg|ts|webm",
   "Audio":{
      "Codec":"AAC|mp2|mp3|vorbis",
      "CodecOptions":{
          "Profile":"auto|AAC-LC|HE-AAC|HE-AACv2"
      },
      "SampleRate":"auto|22050|32000|44100|48000|96000",
      "BitRate":"audio bit rate of output file in kilobits/second",
      "Channels":"auto|0|1|2",
```

```
   },
   "Video":{
      "Codec":"gif|H.264|mpeg2|vp8",
      "CodecOptions":{
         "Profile":"baseline|main|high|0|1|2|3",
         "Level":"1|1b|1.1|1.2|1.3|2|2.1|2.2|3|3.1|3.2|4|4.1",
         "MaxReferenceFrames":"maximum number of reference frames",
         "MaxBitRate":"maximum bit rate",
         "BufferSize":"maximum buffer size",
         "InterlacedMode":"Progressive|TopFirst|BottomFirst|Auto",
         "ColorSpaceConversion":"None|Bt709ToBt601|Bt601ToBt709|Auto",
         "ChromaSubsampling":"yuv420p|yuv422p",
         "LoopCount":"Infinite|[0,100]"
      },
      "KeyframesMaxDist":"maximum frames between key frames",
      "FixedGOP":"true|false",
      "BitRate":"auto|video bit rate of output file in kilobits/second",
      "FrameRate":"auto|10|15|23.97|24|25|29.97|30|50|60",
      "MaxFrameRate":"10|15|23.97|24|25|29.97|30|50|60",
      "MaxWidth":"auto|[128,4096]",
      "MaxHeight":"auto|[96,3072]",
      "SizingPolicy":"Fit|Fill|Stretch|Keep|ShrinkToFit|ShrinkToFill",
      "PaddingPolicy":"Pad|NoPad",
      "DisplayAspectRatio":"auto|1:1|4:3|3:2|16:9",
      "Resolution":"auto|width in pixelsxheight in pixels" <not recommended>,
      "AspectRatio":"auto|1:1|4:3|3:2|16:9" <not recommended>,
      "Watermarks":[
         {
            "Id":"unique identifier up to 40 characters",
            "MaxWidth":"[16,Video:MaxWidth]px|[0,100]%",
            "MaxHeight":"[16,Video:MaxHeight]px|[0,100]%",
            "SizingPolicy":"Fit|Stretch|ShrinkToFit",
            "HorizontalAlign":"Left|Right|Center",
            "HorizontalOffset":"[0,100]%|[0,Video:MaxWidth]px",
            "VerticalAlign":"Top|Bottom|Center",
            "VerticalOffset":"[0,100]%|[0,Video:MaxHeight]px",
            "Opacity":"[0,100]",
            "Target":"Content|Frame"
         },
         {...}
      ]
   },
   "Thumbnails":{
      "Format":"jpg|png",
      "Interval":"number of seconds between thumbnails",
      "MaxWidth":"auto|[32,4096]",
      "MaxHeight":"auto|[32,3072]",
      "SizingPolicy":"Fit|Fill|Stretch|Keep|ShrinkToFit|ShrinkToFill",
      "PaddingPolicy":"Pad|NoPad",
      "Resolution":"width in pixelsxheight in pixels" <not recommended>,
      "AspectRatio":"auto|1:1|4:3|3:2|16:9" <not recommended>
   }
}
```

Request Parameters

This operation does not use request parameters.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

The JSON string in the request body contains the following objects.

Name
The name of the preset. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

Description
A description of the preset.

Constraints: Maximum 255 characters

Container
The container type for the output file. Valid values include `flv`, `fmp4`, `gif`, `mp3`, `mp4`, `mpg`, `ogg`, `ts`, and `webm`. The following restrictions apply:

Container	Audio Codecs	Video Codecs
FLV	AAC, mp3	H.264
FMP4	AAC	H.264
GIF	Not Allowed	gif
MP3	mp3	Not Allowed
MP4	AAC, mp3	H.264
MPG	mp2	mpeg2
OGG	vorbis	Not Allowed
TS	AAC, mp3	H.264
WEBM	vorbis	vp8

Audio:Codec
The audio codec for the output file. Valid values include `AAC`, `mp2`, `mp3`, and `vorbis`. The following chart marks the allowed combinations:

Container & Audio Codecs	AAC	MP2	MP3	Vorbis
FLV	x		x	
GIF				
FMP4	x			
MP3			x	
MP4	x		x	

Container & Audio Codecs	AAC	MP2	MP3	Vorbis
MPG		x		
OGG				x
TS	x		x	
WEBM				x

(AAC Only) Audio:CodecOptions:Profile

If you specified AAC for `Audio:Codec`, choose the AAC profile for the output file. Elastic Transcoder supports the following profiles:

- `auto`: If you specify `auto`, Elastic Transcoder selects the profile based on the bit rate selected for the output file.
- `AAC-LC`: The most common AAC profile. Use for bit rates larger than 64 kbps. For more information, see Advanced Audio Coding.
- `HE-AAC`: Not supported on some older players and devices. Use for bit rates between 40 and 80 kbps. For more information, see High-Efficiency Advanced Audio Coding.
- `HE-AACv2`: Not supported on some players and devices. Use for bit rates less than 48 kbps. For more information, see High-Efficiency Advanced Audio Coding..

All outputs in a `Smooth` playlist must have the same value for `Profile`.

> **Note**
> If you created any presets before AAC profiles were added, Elastic Transcoder automatically updated your presets to use AAC-LC.

For more information about AAC, go to Audio Profiles in the Wikipedia entry "MPEG-4 Part 3."

Audio:SampleRate

The sample rate of the audio stream in the output file, in hertz. Valid values include:

`auto`, `22050`, `32000`, `44100`, `48000`, `96000`

If you specify `auto`, Elastic Transcoder automatically detects the sample rate.

Audio:BitRate

The bit rate of the audio stream in the output file, in kilobits/second. Enter an integer between 64 and 320, inclusive.

Audio:Channels

The number of audio channels in the output file. Valid values include:

`auto`, `0`, `1`, `2`

If you want Elastic Transcoder to automatically detect the number of channels in the input file and use that value for the output file, specify `auto`. If you want Elastic Transcoder to omit audio from the output file, specify `0`.

Video:Codec

The video codec for the output file. Valid values include `gif`, `H.264`, `mpeg2`, and `vp8`. The following chart marks the allowed combinations:

Container & Video Codecs	GIF	H.264	MPEG2	VP8
FLV		x		

Container & Video Codecs	GIF	H.264	MPEG2	VP8
GIF	x			
FMP4		x		
MP3				
MP4		x		
MPG			x	
OGG				
TS		x		
WEBM				x

For more information about the H.264 video-compression format, go to the wikipedia page on H.264/MPEG-4 AVC.

For more information about VP8, see the WebM project website.

(H.264/VP8 Only) Video:CodecOptions:Profile

If you specified `H.264` for `Video:Codec`, the H.264 profile that you want to use for the output file. Elastic Transcoder supports the following profiles:

- `baseline`: The profile most commonly used for video conferencing and for mobile applications.
- `main`: The profile used for standard-definition digital TV broadcasts.
- `high`: The profile used for high-definition digital TV broadcasts and for Blu-ray discs.

If you specified `vp8` for the video codec, the vp8 profile that you want to use for the output file. Elastic Transcoder supports the following profiles: `0`, `1`, `2`, `3`. You can specify `0`, `1`, `2`, or `3` only when the container type is `webm`.

For more information about profiles, see Profiles in the Wikipedia entry "H.264/MPEG-4 AVC."

(H.264 Only) Video:CodecOptions:Level

Applicable only when the value of `Video:Codec` is `H.264`. The H.264 level that you want to use for the output file. Elastic Transcoder supports the following levels:

`1`, `1b`, `1.1`, `1.2`, `1.3`, `2`, `2.1`, `2.2`, `3`, `3.1`, `3.2`, `4`, `4.1`

For more information about levels, see Levels in the Wikipedia entry "H.264/MPEG-4 AVC."

(H.264 Only) Video:CodecOptions:MaxReferenceFrames

Applicable only when the value of `Video:Codec` is `H.264`. The maximum number of previously decoded frames to use as a reference for decoding future frames. Valid values are integers 0 through 16, but we recommend that you not use a value greater than:

```
Min(Floor(Maximum decoded picture buffer in macroblocks * 256 / (Width in
pixels * Height in pixels)), 16)
```

where:

- `Width in pixels` and `Height in pixels` represent either `Video:MaxWidth` and `Video:MaxHeight`, or `Video:Resolution`.
- `Maximum decoded picture buffer in macroblocks` depends on the value of the `Video:CodecOptions:Level` object. (A macroblock is a block of pixels measuring 16x16.) See the table below.

For more information about encoding based on previously encoded pictures, see Decoded picture buffering in the Wikipedia entry "H.264/MPEG-4 AVC." Note that the Wikipedia calculation for maximum decoded picture buffer, which is similar to the calculation for maximum reference frames, uses macroblocks instead of pixels for the width and height of the video.

`Video:CodecOptions:Level`	Maximum Decoded Picture Buffer in Macroblocks
1	396
1b	396
1.1	900
1.2	2376
1.3	2376
2	2376
2.1	4752
2.2	8100
3	8100
3.1	18000
3.2	20480
4	32768
4.1	32768

(Optional, H.264/MPEG2/VP8 only) Video:CodecOptions:MaxBitRate

The maximum number of kilobits per second in the output video. Specify a value between 16 and 62,500, inclusive.

If you specify `auto` for `BitRate`, Elastic Transcoder uses the bit rate of the input video as the average bit rate of the output video. `MaxBitRate` allows you to cap the bit rate of the output video, which is useful when the maximum bit rate supported by a target device is lower than the bit rate of the input video. Reducing the maximum bit rate might reduce the quality of the video.

(Optional, H.264/MPEG2/VP8 only) Video:CodecOptions:BufferSize

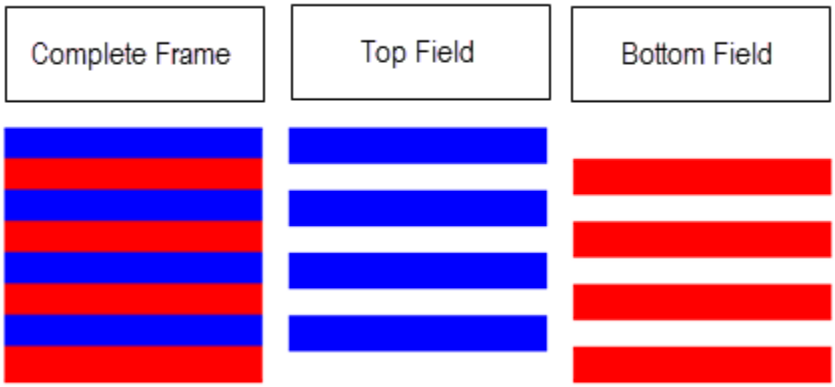
The maximum number of kilobits in any *x* seconds of the output video. This window is commonly 10 seconds, the standard segment duration when you're using ts for the container type of the output video. Specify an integer greater than 0. If you specify `MaxBitRate` and omit `BufferSize`, Elastic Transcoder sets `BufferSize` to 10 times the value of `MaxBitRate`.

(Optional, H.264/MPEG2 Only) Video:CodecOptions:InterlacedMode

The interlace mode for the output video.

Interlaced video is used to double the perceived frame rate for a video by interlacing two fields (one field on every other line, the other field on the other lines) so that the human eye registers multiple pictures per frame. Interlacing reduces the bandwidth required for transmitting a video, but can result in blurred images and flickering.

The two sets of lines are known as fields, and an interlaced frame splits two images across the fields:



Valid values include `Progressive` (no interlacing, top to bottom), `TopFirst` (top field first), `BottomFirst` (bottom field first), and `Auto`.

If `InterlaceMode` is not specified, Elastic Transcoder uses `Progressive` for the output. If `Auto` is specified, Elastic Transcoder interlaces the output.

For more information, go to the Wikipedia page Interlaced video.

(Optional, H.264/MPEG2 Only) Video:CodecOptions:ColorSpaceConversion

The color space conversion Elastic Transcoder applies to the output video. Color spaces are the algorithms used by the computer to store information about how to render color. `Bt.601` is the standard for standard definition video, while `Bt.709` is the standard for high definition video.

Valid values include `None`, `Bt709toBt601`, `Bt601toBt709`, and `Auto`.

If you chose `Auto` for `ColorSpaceConversionMode` and your output is interlaced, your frame rate is one of `23.97`, `24`, `25`, `29.97`, `50`, or `60`, your `SegmentDuration` is null, and you are using one of the resolution changes from the graph below, Elastic Transcoder applies the following color space conversions:

Input File	Input Resolution	Output Resolution	Applied Color Space Conversion
Standard to HD	720x480	1920x1080	Bt601ToBt709
Standard to HD	720x576	1920x1080	Bt601ToBt709
HD to Standard	1920x1080	720x576	Bt709ToBt601
HD to Standard	1920x1080	720x480	Bt709ToBt601

Note

Elastic Transcoder may change the behavior of the `ColorspaceConversionMode Auto` mode in the future. All outputs in a playlist must use the same `ColorSpaceConversionMode`.

If you do not specify a `ColorSpaceConversionMode`, Elastic Transcoder does not change the color space of a file.

If you are unsure what `ColorSpaceConversionMode` was applied to your output file, you can check the `AppliedColorSpaceConversion` parameter included in your job response. If your job does

not have an `AppliedColorSpaceConversion` in its response, no `ColorSpaceConversionMode` was applied.

For more information about color space, go to the Wikipedia page Color space. For more information about `Bt.601` and `Bt.709`, go to the Wikipedia pages Rec. 601 and Rec. 709.

Video:CodecOptions:ChromaSubsampling
The sampling pattern for the chroma (color) channels of the output video. Valid values include `yuv420p` and `yuv422p`.

`yuv420p` samples the chroma information of every other horizontal and every other vertical line, `yuv422p` samples the color information of every horizontal line and every other vertical line.

To learn more about chroma subsampling, go to the wikidepia page Chroma subsampling.

(Gif Only) Video:CodecOptions:LoopCount
The number of times you want the output gif to loop.

Valid values include `Infinite` and integers between `0` and `100`, inclusive.

(H.264/MPEG2/VP8 Only) Video:KeyframesMaxDist
The maximum number of frames between key frames. Not applicable for containers of type `gif`. Key frames are fully encoded frames; the frames between key frames are encoded based, in part, on the content of the key frames. The value is an integer formatted as a string; valid values are between 1 (every frame is a key frame) and 100000, inclusive. A higher value results in higher compression but might also discernibly decrease video quality.

For `Smooth` outputs, the `FrameRate` must have a constant ratio to the `KeyframesMaxDist`. This allows `Smooth` playlists to switch between different quality levels while the file is being played.

For example, an input file can have a `FrameRate` of 30 with a `KeyframesMaxDist` of 90. The output file then needs to have a ratio of 1:3. Valid outputs would have `FrameRate` of 30, 25, and 10, and `KeyframesMaxDist` of 90, 75, and 30, respectively.

Alternately, this can be achieved by setting `FrameRate` to auto and having the same values for `MaxFrameRate` and `KeyframesMaxDist`.

For more information about key frames, see the Wikipedia entry Video compression picture types.

(H.264/MPEG2/VP8 Only) Video:FixedGOP
Whether to use a fixed value for `Video:FixedGOP`. Not applicable for containers of type `gif`. Valid values are `true` and `false`:

- `true`: Elastic Transcoder uses the value of `Video:KeyframesMaxDist` for the distance between key frames (the number of frames in a group of pictures, or GOP).
- `false`: The distance between key frames can vary.

> **Note**
> `FixedGOP` must be set to `true` for `fmp4` containers.

Video:BitRate
The bit rate of the video stream in the output file, in kilobits/second. You can configure variable bit rate or constant bit rate encoding:

- **Variable bit rate encoding:** Specify **auto**. Elastic Transcoder optimizes the bit rate and maintains a consistent quality for each frame of the output.
- **Constant bit rate encoding:** Specify the bit rate.

If you specified `H.264` for `Video:Codec`: Valid values depend on the values of the `Video:CodecOptions:Level` and `Video:CodecOptions:Profile` objects.

> **Note**
> If you specified `vp8` for `Video:Codec`, do not use the following table; `Level` applies only when the video codec is H.264.

If you specify a value other than `auto`, we recommend that you specify a value less than or equal to the maximum H.264-compliant value listed in the following table for your level and profile:

`Level`	Maximum Video Bit Rate in kilobits/second: Baseline Profile and Main Profile	Maximum Video Bit Rate in kilobits/second: High Profile
1	64	80
1b	128	160
1.1	192	240
1.2	384	480
1.3	768	960
2	2000	2500
2.1	4000	5000
2.2	4000	5000
3	10000	12500
3.1	14000	17500
3.2	20000	25000
4	20000	25000
4.1	50000	62500

Video:FrameRate

The frames per second for the video stream in the output file. Valid values include:

`auto`, `10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `50`, `60`

If you want to preserve the frame rate of the input file and use it for the output file, specify `auto`.

If you specified `H.264` for Video:Codec: If you specify a frame rate, we recommend that you perform the following calculation:

```
Frame rate = maximum recommended decoding speed in luma samples/second /
(width in pixels * height in pixels)
```

where:

- `width in pixels` and `height in pixels` represent the `Video:Resolution` of the output video.
- `maximum recommended decoding speed in Luma samples/second` is less than or equal to the maximum value listed in the following table, based on the value that you specified for `Video:CodecOptions:Level`.

  Note

  If you specified `vp8` for `Video:Codec`, do not use the previous equation or the following table; `Level` applies only when the video codec is H.264.

Level	Maximum Recommended Decoding Speed in Luma Samples/Second
1	380160
1b	380160
1.1	768000
1.2	1536000
1.3	3041280
2	3041280
2.1	5068800
2.2	5184000
3	10368000
3.1	27648000
3.2	55296000
4	62914560
4.1	62914560

Video:MaxFrameRate

If you specify `auto` for `FrameRate`, Elastic Transcoder uses the frame rate of the input video for the frame rate of the output video, up to the maximum frame rate. If you do not specify a `MaxFrameRate`, Elastic Transcoder will use a default of `30`.

Specify the maximum frame rate that you want Elastic Transcoder to use when the frame rate of the input video is greater than either the desired maximum frame rate of the output video or the default maximum frame rate. Valid values include:

`10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `50`, `60`

Elastic Transcoder uses the highest supported frame rate that meets both of the following criteria:

- The frame rate is less than or equal to the maximum frame rate.
- The frame rate divides into the input frame rate evenly, with no remainder.

For example, if you have an input file with a frame rate of 50 and specify a value of 30 for `MaxFrameRate`, Elastic Transcoder produces an output video for which the frame rate is 25 frames per second, because 25 is less than 30, and 50 divided by 25 is 2.

Video:MaxWidth (Option 1)

The maximum width of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 128 and 4096, inclusive.

For more information, see `Video:MaxHeight`.

Video:MaxHeight (Option 1)

The maximum height of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 96 and 3072, inclusive.

If you specified `H.264` for `Video:Codec`: We recommend that you specify values for `MaxWidth` and `MaxHeight` so the product of the two values is less than or equal to the applicable value in the following table.

Note
If you specified `vp8` for `Video:Codec`, do not use the following table; `Level` applies only when the video codec is H.264.

Value of Level	Maximum Value of `MaxWidth` x `MaxHeight`
1	25344
1b	25344
1.1	101376
1.2	101376
1.3	101376
2	101376
2.1	202752
2.2	404720
3	404720
3.1	921600
3.2	1310720
4	2097152
4.1	2097152

Video:SizingPolicy (Option 1)

A value that controls scaling of the output video:

- **Fit:** Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Fill:** Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output video and then crops it to the dimension (if any) that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches the output video to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input video and the output video are different, the output video will be distorted.
- **Keep:** Elastic Transcoder does not scale the output video. If either dimension of the input video exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output video.
- **ShrinkToFit:** Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the video up.
- **ShrinkToFill:** Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the video up.

The following table shows possible effects of `SizingPolicy` settings on the output video:

SizingPolicy	Output Video May Be Scaled Up	Output Video May Be Padded When Padding Policy Is "Pad"	Output Video May Have a Different Pixel Aspect Ratio than Input Video	Output Video May Be Cropped
Fit	Yes	Yes		
Fill	Yes			Yes
Stretch	Yes		Yes	
Keep	Yes	Yes		Yes
ShrinkToFit		Yes		
ShrinkToFill		Yes		Yes

Video:PaddingPolicy (Option 1)
When you set `PaddingPolicy` to `Pad`, Elastic Transcoder might add black bars to the top and bottom and/or left and right sides of the output video to make the total size of the output video match the values that you specified for `MaxWidth` and `MaxHeight`. For more information, see the table at `Video:SizingPolicy`.

Video:DisplayAspectRatio (Option 1)
The value that Elastic Transcoder adds to the metadata in the output file. If you set `DisplayAspectRatio` to `auto`, Elastic Transcoder chooses an aspect ratio that ensures square pixels. If you specify another option, Elastic Transcoder sets that value in the output file.

Video:Resolution (Option 2)

> **Note**
> To better control resolution and aspect ratio of output videos, we recommend that you use the Video—Option 1 settings, `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of the two Video—Option 2 settings, `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of the video in the output file, in pixels. Valid values are `auto` and *`width`*x*`height`*:

- `auto`: Elastic Transcoder attempts to preserve the width and height of the input file, subject to the following rules.
- *`width`*x*`height`*: The width and height of the output video in pixels.

Note the following about specifying the width and height:

- The width must be an even integer between 128 and 4096, inclusive.
- The height must be an even integer between 96 and 3072, inclusive.
- If you specify a resolution that is less than the resolution of the input file, Elastic Transcoder rescales the output file to the lower resolution.
- If you specify a resolution that is greater than the resolution of the input file, Elastic Transcoder rescales the output to the higher resolution.
- We recommend that you specify a resolution for which the product of width and height is less than or equal to the applicable value in the following table:

Level	Maximum Value of `width` x `height`
1	25344
1b	25344

Level	Maximum Value of `width` x `height`
1.1	101376
1.2	101376
1.3	101376
2	101376
2.1	202752
2.2	404720
3	404720
3.1	921600
3.2	1310720
4	2097152
4.1	2097152

Video:AspectRatio (Option 2)

> **Note**
> To better control resolution and aspect ratio of output videos, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of `Resolution` and `AspectRatio`.

The display aspect ratio of the video in the output file. Valid values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the input file.

If you specify an aspect ratio for the output file that differs from aspect ratio of the input file, Elastic Transcoder adds pillarboxing (black bars on the sides) or letterboxing (black bars on the top and bottom) to maintain the aspect ratio of the active region of the video.

Video:Watermarks

Settings for the size, position, scale, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear for the duration of the transcoded video.

Watermarks can be in `.png` or `.jpg` format. If you want to display a watermark that is not rectangular, use the `.png` format, which supports transparency.

When you create a job that uses this preset, you specify the `.png` or `.jpg` graphics that you want Elastic Transcoder to include in the transcoded videos. Elastic Transcoder does not require you to specify as many watermarks in each job output as you specified in the corresponding preset. For example, you might specify settings for four watermarks in a preset and specify only one watermark in a job output.

To configure watermark settings so your graphic is not distorted, set the value of `SizingPolicy` to `ShrinkToFit`, and set the values of `MaxWidth` and `MaxHeight` to the same percentage. If you want the graphic to appear in the same size as the original, set `MaxWidth` and `MaxHeight` to 100%.

For more information, see Watermarks (p. 69).

Video:Watermarks:Id

A unique identifier for the settings for one watermark. The value of `Id` can be up to 40 characters long. You can specify settings for up to four watermarks.

Video:Watermarks:MaxWidth

The maximum width of the watermark in one of the following formats:

- *`number of pixels`*`px`: The minimum value is 16 pixels, and the maximum value is the value of `Video:MaxWidth`.
- *`integer percentage`*`%`: The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

Video:Watermarks:MaxHeight

The maximum height of the watermark in one of the following formats:

- *`number of pixels`*`px`: The minimum value is 16 pixels, and the maximum value is the value of `Video:MaxHeight`.
- *`integer percentage`*`%`: The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

Video:Watermarks:SizingPolicy

A value that controls scaling of the watermark:

- **Fit:** Elastic Transcoder scales the watermark so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Stretch:** Elastic Transcoder stretches the watermark to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the watermark and the values of `MaxWidth` and `MaxHeight` are different, the watermark will be distorted.
- **ShrinkToFit:** Elastic Transcoder scales the watermark down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the watermark up.

Video:Watermarks:HorizontalAlign

The horizontal position of the watermark unless you specify a nonzero value for `HorizontalOffset`:

- **Left:** The left edge of the watermark is aligned with the left border of the video.
- **Right:** The right edge of the watermark is aligned with the right border of the video.
- **Center:** The watermark is centered between the left and right borders.

Video:Watermarks:HorizontalOffset

The amount by which you want the horizontal position of the watermark to be offset from the position specified by `HorizontalAlign`:

- *`number of pixels`*`px`: The minimum value is 0 pixels, and the maximum value is the value of `Video:MaxWidth`.
- *`integer percentage`*`%`: The range of valid values is 0 to 100.

For example, if you specify `Left` for `HorizontalAlign` and `5px` for `HorizontalOffset`, the left side of the watermark appears 5 pixels from the left border of the output video.

`HorizontalOffset` is valid only when the value of `HorizontalAlign` is `Left` or `Right`.

If you specify an offset that causes the watermark to extend beyond the left or right border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

Video:Watermarks:VerticalAlign

The vertical position of the watermark unless you specify a nonzero value for `VerticalOffset`:

- **Top:** The top edge of the watermark is aligned with the top border of the video.
- **Bottom:** The bottom edge of the watermark is aligned with the bottom border of the video.
- **Center:** The watermark is centered between the top and bottom borders.

Video:Watermarks:VerticalOffset
The amount by which you want the vertical position of the watermark to be offset from the position specified by `VerticalAlign`:

- *`number of pixels`*`px`: The minimum value is 0 pixels, and the maximum value is the value of `Video:MaxHeight`.
- *`integer percentage`*`%`: The range of valid values is 0 to 100.

For example, if you specify `Top` for `VerticalAlign` and `5px` for `VerticalOffset`, the top of the watermark appears 5 pixels from the top border of the output video.

`VerticalOffset` is valid only when the value of `VerticalAlign` is `Top` or `Bottom`.

If you specify an offset that causes the watermark to extend beyond the top or bottom border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

Video:Watermarks:Opacity
A percentage that indicates how much you want a watermark to obscure the video in the location where it appears. Valid values are 0 (the watermark is invisible) to 100 (the watermark completely obscures the video in the specified location). The data type of `Opacity` is `float`.

Elastic Transcoder supports transparent `.png` graphics. If you use a transparent `.png`, the transparent portion of the video appears as if you had specified a value of 0 for `Opacity`. The `.jpg` file format doesn't support transparency.

Video:Watermarks:Target
A value that determines how Elastic Transcoder interprets values that you specified for `Video:Watermarks:HorizontalOffset`, `Video:Watermarks:VerticalOffset`, `Video:Watermarks:MaxWidth`, and `Video:Watermarks:MaxHeight`:

- **Content:** `HorizontalOffset` and `VerticalOffset` values are calculated based on the borders of the video **excluding** black bars added by Elastic Transcoder, if any.

  In addition, `MaxWidth` and `MaxHeight`, if specified as a percentage, are calculated based on the borders of the video **excluding** black bars added by Elastic Transcoder, if any.
- **Frame:** `HorizontalOffset` and `VerticalOffset` values are calculated based on the borders of the video **including** black bars added by Elastic Transcoder, if any.

  In addition, `MaxWidth` and `MaxHeight`, if specified as a percentage, are calculated based on the borders of the video **including** black bars added by Elastic Transcoder, if any.

(Video Only) Thumbnails:Format
The format of thumbnails, if any. Valid formats are `jpg` and `png`.

You specify whether you want Elastic Transcoder to create thumbnails when you create a job. For more information, see ThumbnailPattern (p. 188).

(Video Only) Thumbnails:Interval
The approximate number of seconds between thumbnails. The value must be an integer. The actual interval can vary by several seconds from one thumbnail to the next.

(Option 1, Video Only) Thumbnails:MaxWidth
The maximum width of thumbnails, in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

(Option 1, Video Only) Thumbnails:MaxHeight
The maximum height of thumbnails, in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

(Option 1, Video Only) Thumbnails:SizingPolicy
A value that controls scaling of thumbnails:

- **Fit:** Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings without exceeding the other value.
- **Fill:** Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings and matches or exceeds the other value. Elastic Transcoder centers the image in thumbnails and then crops to the dimension, if any, that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches thumbnails to match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings. If the relative proportions of the input video and thumbnails are different, the thumbnails will be distorted.
- **Keep:** Elastic Transcoder does not scale thumbnails. If either dimension of the input video exceeds the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings, Elastic Transcoder crops the thumbnails.
- **ShrinkToFit:** Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of thumbnail `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.
- **ShrinkToFill:** Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.

The following table shows possible effects of `SizingPolicy` settings on thumbnails:

SizingPolicy	Thumbnails May Be Scaled Up	Thumbnails May Be Padded When Padding Policy Is "Pad"	Thumbnails May Have a Different Pixel Aspect Ratio than Input Video	Thumbnails May Be Cropped
Fit	Yes	Yes		
Fill	Yes			Yes
Stretch	Yes		Yes	
Keep	Yes	Yes		Yes
ShrinkToFit		Yes		
ShrinkToFill		Yes		Yes

(Option 1, Video Only) Thumbnails:PaddingPolicy
When you set `PaddingPolicy` to `Pad`, Elastic Transcoder might add black bars to the top and bottom and/or left and right sides of thumbnails to make the total size of the thumbnails match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings. For more information, see the table at `Thumbnails:SizingPolicy`.

(Option 2, Video Only) Thumbnails:Resolution

> **Note**
> To better control resolution and aspect ratio of thumbnails, we recommend that you use the thumbnail values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead

of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of thumbnail files in pixels, in the format *`Width`*x*`Height`*, where both values are even integers. The values cannot exceed the width and height that you specified in the `Video:Resolution` object.

(Option 2, Video Only) Thumbnails:AspectRatio

Note
To better control resolution and aspect ratio of thumbnails, we recommend that you use the thumbnail values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead of `Resolution` and `AspectRatio`.

The aspect ratio of thumbnails. Valid values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the video in the output file.

Responses

Syntax

```
Status: 201 Created
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT
{
   "Preset":{
      "Id":"Id for the new preset",
      "Type":"Custom|System",
      "Name":"preset name",
      "Description":"preset description",
      "Container":"flv|fmp4|gif|mp3|mp4|mpg|ogg|ts|webm",
      "Audio":{
         "Codec":"AAC|mp2|mp3|vorbis",
         "CodecOptions":{
            "Profile":"auto|AAC-LC|HE-AAC|HE-AACv2"
         },
         "SampleRate":"auto|22050|32000|44100|48000|96000",
         "BitRate":"audio bit rate of output file in kilobits/second",
         "Channels":"auto|0|1|2"
      },
      "Video":{
         "Codec":"gif|H.264|mpeg2|vp8",
         "CodecOptions":{
            "Profile":"baseline|main|high|0|1|2|3",
            "Level":"1|1b|1.1|1.2|1.3|2|2.1|2.2|3|3.1|3.2|4|4.1",
            "MaxReferenceFrames":maximum number of reference frames,
            "MaxBitRate":"maximum bit rate",
            "BufferSize":"maximum buffer size",
            "InterlacedMode":"Progressive|TopFirst|BottomFirst|Auto",
            "ColorSpaceConversionMode":"None|Bt709ToBt601|Bt601ToBt709|Auto",
            "ChromaSubsampling":"yuv420p|yuv422p",
            "LoopCount":"Infinite|[0,100]"
```

```
        },
        "KeyframesMaxDist":maximum frames between key frames,
        "FixedGOP":"true|false",
        "BitRate":"auto|video bit rate of output file in kilobits/second",
        "FrameRate":"auto|10|15|23.97|24|25|29.97|30|50|60",
        "MaxFrameRate":"10|15|23.97|24|25|29.97|30|50|60",
        "MaxWidth":"auto|[128,4096]",
        "MaxHeight":"auto|[96,3072]",
        "SizingPolicy":"Fit|Fill|Stretch|Keep|ShrinkToFit|ShrinkToFill",
        "PaddingPolicy":"Pad|NoPad",
        "DisplayAspectRatio":"auto|1:1|4:3|3:2|16:9",
        "Resolution":"width in pixelsxheight in pixels" <not recommended>,
        "AspectRatio":"auto|1:1|4:3|3:2|16:9" <not recommended>
        "Watermarks":[
           {
              "Id":"unique identifier up to 40 characters",
              "MaxWidth":"[16,Video:MaxWidth]px|[0,100]%",
              "MaxHeight":"[16,Video:MaxHeight]px|[0,100]%",
              "SizingPolicy":"Fit|Stretch|ShrinkToFit",
              "HorizontalAlign":"Left|Right|Center",
              "HorizontalOffset":"[0,100]%|[0,Video:MaxWidth]px",
              "VerticalAlign":"Top|Bottom|Center",
              "VerticalOffset":"[0,100]%|[0,Video:MaxHeight]px",
              "Opacity":"[0,100]",
              "Target":"Content|Frame"
           }
        ]
     },
     "Thumbnails":{
        "Format":"jpg|png",
        "Interval":"number of seconds between thumbnails",
        "MaxWidth":"auto|[32,4096]",
        "MaxHeight":"auto|[32,3072]",
        "SizingPolicy":"Fit|Fill|Stretch|Keep|ShrinkToFit|ShrinkToFill",
        "PaddingPolicy":"Pad|NoPad",
        "Resolution":"width in pixelsxheight in pixels",
        "AspectRatio":"auto|1:1|4:3|3:2|16:9"
     }
  },
  "Warning":"message about codec compatibility"
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

When you create a job, Elastic Transcoder returns the values that you specified in the request. For more information, see Request Body (p. 277).

In addition, Elastic Transcoder returns the following values.

Id

Identifier for the new preset. You use this value to get settings for the preset or to delete it.

Type
Whether the preset is a default preset provided by Elastic Transcoder (`System`) or a preset that you have defined (`Custom`).

Warning
When the settings for a preset are not compliant with the standards for the Codec but the settings might produce acceptable output, Elastic Transcoder creates the preset and includes a warning explaining how the preset settings are not compliant.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

The following example request creates a preset named DefaultPreset.

Sample Request

```
POST /2012-09-25/presets HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
   "Name":"DefaultPreset",
   "Description":"Use for published videos",
   "Container":"mp4",
   "Audio":{
      "Codec":"AAC",
      "CodecOptions":{
         "Profile":"AAC-LC"
      },
      "SampleRate":"44100",
      "BitRate":"96",
      "Channels":"2"
   },
   "Video":{
      "Codec":"H.264",
      "CodecOptions":{
         "Profile":"main",
         "Level":"2.2",
         "MaxReferenceFrames":"3",
         "MaxBitRate":"",
         "BufferSize":"",
         "InterlacedMode":"Progressive",
         "ColorSpaceConversionMode":"None"
      },
      "KeyframesMaxDist":"240",
      "FixedGOP":"false",
```

```
      "BitRate":"1600",
      "FrameRate":"auto",
      "MaxFrameRate":"30",
      "MaxWidth":"auto",
      "MaxHeight":"auto",
      "SizingPolicy":"Fit",
      "PaddingPolicy":"Pad",
      "DisplayAspectRatio":"auto",
      "Watermarks":[
         {
            "Id":"company logo",
            "MaxWidth":"20%",
            "MaxHeight":"20%",
            "SizingPolicy":"ShrinkToFit",
            "HorizontalAlign":"Right",
            "HorizontalOffset":"10px",
            "VerticalAlign":"Bottom",
            "VerticalOffset":"10px",
            "Opacity":"55.5",
            "Target":"Content"
         }
      ]
   },
   "Thumbnails":{
      "Format":"png",
      "Interval":"120",
      "MaxWidth":"auto",
      "MaxHeight":"auto",
      "SizingPolicy":"Fit",
      "PaddingPolicy":"Pad"
   }
}
```

Sample Response

```
Status: 201 Created
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Preset":{
      "Id":"5555555555555-abcde5",
      "Type":"Custom",
      "Name":"DefaultPreset",
      "Description":"Use for published videos",
      "Container":"mp4",
      "Audio":{
         "Codec":"AAC",
         "CodecOptions":{
            "Profile":"AAC-LC"
         },
         "SampleRate":"44100",
         "BitRate":"96",
         "Channels":"2"
```

```
      },
      "Video":{
         "Codec":"H.264",
         "CodecOptions":{
            "Profile":"main",
            "Level":"2.2",
            "MaxReferenceFrames":"3",
            "MaxBitRate":"",
            "BufferSize":"",
            "InterlacedMode":"Progressive",
            "ColorSpaceConversionMode":"None|Bt709ToBt601|Bt601ToBt709|Auto"
         },
         "KeyframesMaxDist":"240",
         "FixedGOP":"false",
         "BitRate":"1600",
         "FrameRate":"auto",
         "MaxFrameRate":"30",
         "MaxWidth":"auto",
         "MaxHeight":"auto",
         "SizingPolicy":"Fit",
         "PaddingPolicy":"Pad",
         "DisplayAspectRatio":"auto",
         "Watermarks":[
            {
               "Id":"company logo",
               "MaxWidth":"20%",
               "MaxHeight":"20%",
               "SizingPolicy":"ShrinkToFit",
               "HorizontalAlign":"Right",
               "HorizontalOffset":"10px",
               "VerticalAlign":"Bottom",
               "VerticalOffset":"10px",
               "Opacity":"55.5",
               "Target":"Content"
            }
         ]
      },
      "Thumbnails":{
         "Format":"png",
         "Interval":"120",
         "MaxWidth":"auto",
         "MaxHeight":"auto",
         "SizingPolicy":"Fit",
         "PaddingPolicy":"Pad"
      }
   },
   "Warning":""
}
```

List Presets

Topics

- Description (p. 296)
- Requests (p. 296)
- Responses (p. 297)
- Errors (p. 312)
- Examples (p. 312)

Description

To get a list of all presets associated with the current AWS account, send a GET request to the `/2012-09-25/presets` resource.

Requests

Syntax

```
GET /2012-09-25/presets/Ascending=true|false&
                PageToken=value for accessing the next page of
                results HTTP/1.1 Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Request Parameters

This operation takes the following request parameters. Elastic Transcoder returns all of the presets available.

Ascending
To list presets in chronological order by the date and time that they were submitted, enter `true`. To list presets in reverse chronological order, enter `false`.

PageToken
When Elastic Transcoder returns more than one page of results, use `PageToken` in subsequent `GET` requests to get each successive page of results.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

The JSON string in the request body contains the following objects.

Responses

Syntax

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Presets":[
      {
         "Id":"preset ID",
         "Type":"Custom|System",
         "Name":"preset name",
         "Description":"preset description",
         "Container":"flv|fmp4|gif|mp3|mp4|mpg|ogg|ts|webm",
         "Audio":{
            "Codec":"AAC|mp2|mp3|vorbis",
            "CodecOptions":{
               "Profile":"auto|AAC-LC|HE-AAC|HE-AACv2"
            },
            "SampleRate":"auto|22050|32000|44100|48000|96000",
            "BitRate":"audio bit rate of output file in kilobits/second",
            "Channels":"auto|0|1|2"
         },
         "Video":{
            "Codec":"gif|H.264|mpeg2|vp8",
            "CodecOptions":{
               "Profile":"baseline|main|high|0|1|2|3",
               "Level":"1|1b|1.1|1.2|1.3|2|2.1|2.2|
                  3|3.1|3.2|4|4.1",
               "MaxReferenceFrames":"maximum number of reference frames",
               "MaxBitRate":"maximum bit rate",
               "BufferSize":"maximum buffer size",
               "InterlacedMode":"Progressive|TopFirst|BottomFirst|Auto",
               "ColorSpaceConversion":"None|Bt709ToBt601|Bt601ToBt709|Auto",
               "ChromaSubsampling":"yuv420p|yuv422p",
               "LoopCount":"Infinite|[0,100]"
            },
            "KeyframesMaxDist":"maximum frames between key frames",
            "FixedGOP":"true|false",
            "BitRate":"auto|video bit rate of output file in
               kilobits/second",
            "FrameRate":"auto|10|15|23.97|24|25|29.97|30|50|60",
            "MaxFrameRate":"10|15|23.97|24|25|29.97|30|50|60",
            "MaxWidth":"auto|[128,4096]",
            "MaxHeight":"auto|[96,3072]",
            "SizingPolicy":"Fit|Fill|Stretch|Keep|ShrinkToFit|ShrinkToFill",
            "PaddingPolicy":"Pad|NoPad",
            "DisplayAspectRatio":"auto|1:1|4:3|3:2|16:9",
            "Resolution":"width in pixelsxheight in pixels",
            "AspectRatio":"auto|1:1|4:3|3:2|16:9",
            "Watermarks":[
               {
                  "Id":"unique identifier up to 40 characters",
```

```
                "MaxWidth":"[16,Video:MaxWidth]px|[0,100]%",
                "MaxHeight":"[16,Video:MaxHeight]px|[0,100]%",
                "SizingPolicy":"Fit|Stretch|ShrinkToFit",
                "HorizontalAlign":"Left|Right|Center",
                "HorizontalOffset":"[0,100]%|[0,Video:MaxWidth]px",
                "VerticalAlign":"Top|Bottom|Center",
                "VerticalOffset":"[0,100]%|[0,Video:MaxHeight]px",
                "Opacity":"[0,100]",
                "Target":"Content|Frame"
             },
             {...}
          ]
       },
       "Thumbnails":{
          "Format":"jpg|png",
          "Interval":"number of seconds between thumbnails",
          "MaxWidth":"auto|[32,4096]",
          "MaxHeight":"auto|[32,3072]",
          "SizingPolicy":"Fit|Fill|Stretch|Keep|ShrinkToFit|ShrinkToFill",
          "PaddingPolicy":"Pad|NoPad",
          "Resolution":"width in pixelsxheight in pixels",
          "AspectRatio":"auto|1:1|4:3|3:2|16:9"
       },
    },
    {...},
 ],
 "NextPageToken":value for accessing the next page of results|null
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

The JSON string in the request body contains the following objects. For more detail about the individual objects, see Request Body (p. 277) in the topic Create Preset (p. 275).

Id
Identifier for the new preset. You use this value to get settings for the preset or to delete it.

Type
Whether the preset is a default preset provided by Elastic Transcoder (`System`) or a preset that you have defined (`Custom`).

Name
The name of the preset. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

Description
A description of the preset.

Constraints: Maximum 255 characters

Container
The container type for the output file. Valid values include `flv`, `fmp4`, `gif`, `mp3`, `mp4`, `mpg`, `ogg`, `ts`, and `webm`. The following restrictions apply:

Container	Audio Codecs	Video Codecs
FLV	AAC, mp3	H.264
FMP4	AAC	H.264
GIF	Not Allowed	gif
MP3	mp3	Not Allowed
MP4	AAC, mp3	H.264
MPG	mp2	mpeg2
OGG	vorbis	Not Allowed
TS	AAC, mp3	H.264
WEBM	vorbis	vp8

Audio:Codec

The audio codec for the output file. Valid values include `AAC`, `mp2`, `mp3`, and `vorbis`. The following chart marks the allowed combinations:

Container & Audio Codecs	AAC	MP2	MP3	Vorbis
FLV	x		x	
GIF				
FMP4	x			
MP3			x	
MP4	x		x	
MPG		x		
OGG				x
TS	x		x	
WEBM				x

(AAC Only) Audio:CodecOptions:Profile

If you specified `AAC` for `Audio:Codec`, choose the AAC profile for the output file. Elastic Transcoder supports the following profiles:

- `auto`: If you specify `auto`, Elastic Transcoder selects the profile based on the bit rate selected for the output file.
- `AAC-LC`: The most common AAC profile. Use for bit rates larger than 64 kbps. For more information, see Advanced Audio Coding.
- `HE-AAC`: Not supported on some older players and devices. Use for bit rates between 40 and 80 kbps. For more information, see High-Efficiency Advanced Audio Coding.
- `HE-AACv2`: Not supported on some players and devices. Use for bit rates less than 48 kbps. For more information, see High-Efficiency Advanced Audio Coding..

All outputs in a `Smooth` playlist must have the same value for `Profile`.

Note
If you created any presets before AAC profiles were added, Elastic Transcoder automatically updated your presets to use AAC-LC.

For more information about AAC, go to Audio Profiles in the Wikipedia entry "MPEG-4 Part 3."

Audio:SampleRate
The sample rate of the audio stream in the output file, in hertz. Valid values include:

`auto`, `22050`, `32000`, `44100`, `48000`, `96000`

If you specify `auto`, Elastic Transcoder automatically detects the sample rate.

Audio:BitRate
The bit rate of the audio stream in the output file, in kilobits/second. Enter an integer between 64 and 320, inclusive.

Audio:Channels
The number of audio channels in the output file. Valid values include:

`auto`, `0`, `1`, `2`

If you want Elastic Transcoder to automatically detect the number of channels in the input file and use that value for the output file, specify `auto`. If you want Elastic Transcoder to omit audio from the output file, specify `0`.

Video:Codec
The video codec for the output file. Valid values include `gif`, `H.264`, `mpeg2`, and `vp8`. The following chart marks the allowed combinations:

Container & Video Codecs	GIF	H.264	MPEG2	VP8
FLV		x		
GIF	x			
FMP4		x		
MP3				
MP4		x		
MPG			x	
OGG				
TS		x		
WEBM				x

For more information about the H.264 video-compression format, go to the wikipedia page on H.264/MPEG-4 AVC.

For more information about VP8, see the WebM project website.

(H.264/VP8 Only) Video:CodecOptions:Profile
If you specified `H.264` for `Video:Codec`, the H.264 profile that you want to use for the output file. Elastic Transcoder supports the following profiles:

- `baseline`: The profile most commonly used for video conferencing and for mobile applications.
- `main`: The profile used for standard-definition digital TV broadcasts.
- `high`: The profile used for high-definition digital TV broadcasts and for Blu-ray discs.

If you specified `vp8` for the video codec, the vp8 profile that you want to use for the output file. Elastic Transcoder supports the following profiles: `0`, `1`, `2`, `3`. You can specify `0`, `1`, `2`, or `3` only when the container type is `webm`.

For more information about profiles, see Profiles in the Wikipedia entry "H.264/MPEG-4 AVC."

(H.264 Only) Video:CodecOptions:Level

Applicable only when the value of `Video:Codec` is `H.264`. The H.264 level that you want to use for the output file. Elastic Transcoder supports the following levels:

`1`, `1b`, `1.1`, `1.2`, `1.3`, `2`, `2.1`, `2.2`, `3`, `3.1`, `3.2`, `4`, `4.1`

For more information about levels, see Levels in the Wikipedia entry "H.264/MPEG-4 AVC."

(H.264 Only) Video:CodecOptions:MaxReferenceFrames

Applicable only when the value of `Video:Codec` is `H.264`. The maximum number of previously decoded frames to use as a reference for decoding future frames. Valid values are integers 0 through 16, but we recommend that you not use a value greater than:

```
Min(Floor(Maximum decoded picture buffer in macroblocks * 256 / (Width in pixels * Height in pixels)), 16)
```

where:

- `Width in pixels` and `Height in pixels` represent either `Video:MaxWidth` and `Video:MaxHeight`, or `Video:Resolution`.
- `Maximum decoded picture buffer in macroblocks` depends on the value of the `Video:CodecOptions:Level` object. (A macroblock is a block of pixels measuring 16x16.) See the table below.

For more information about encoding based on previously encoded pictures, see Decoded picture buffering in the Wikipedia entry "H.264/MPEG-4 AVC." Note that the Wikipedia calculation for maximum decoded picture buffer, which is similar to the calculation for maximum reference frames, uses macroblocks instead of pixels for the width and height of the video.

`Video:CodecOptions:Level`	Maximum Decoded Picture Buffer in Macroblocks
1	396
1b	396
1.1	900
1.2	2376
1.3	2376
2	2376
2.1	4752
2.2	8100
3	8100
3.1	18000
3.2	20480
4	32768
4.1	32768

Video:CodecOptions:MaxBitRate

The maximum number of kilobits per second in the output video. Specify a value between 16 and 62,500, inclusive.

If you specify `auto` for `BitRate`, Elastic Transcoder uses the bit rate of the input video as the average bit rate of the output video. `MaxBitRate` allows you to cap the bit rate of the output video, which is useful when the maximum bit rate supported by a target device is lower than the bit rate of the input video. Reducing the maximum bit rate might reduce the quality of the video.

(Optional, H.264/MPEG2/VP8 only) Video:CodecOptions:BufferSize

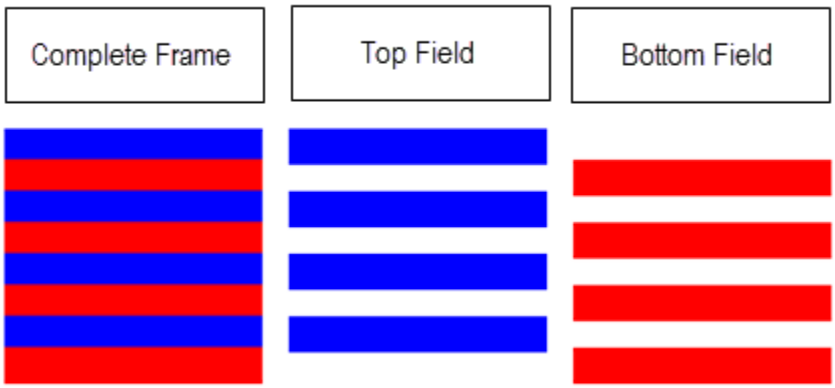
The maximum number of kilobits in any *x* seconds of the output video. This window is commonly 10 seconds, the standard segment duration when you're using ts for the container type of the output video. Specify an integer greater than 0. If you specify `MaxBitRate` and omit `BufferSize`, Elastic Transcoder sets `BufferSize` to 10 times the value of `MaxBitRate`.

(Optional, H.264/MPEG2 Only) Video:CodecOptions:InterlacedMode

The interlace mode for the output video.

Interlaced video is used to double the perceived frame rate for a video by interlacing two fields (one field on every other line, the other field on the other lines) so that the human eye registers multiple pictures per frame. Interlacing reduces the bandwidth required for transmitting a video, but can result in blurred images and flickering.

The two sets of lines are known as fields, and an interlaced frame splits two images across the fields:



Valid values include `Progressive` (no interlacing, top to bottom), `TopFirst` (top field first), `BottomFirst` (bottom field first), and `Auto`.

If `InterlaceMode` is not specified, Elastic Transcoder uses `Progressive` for the output. If `Auto` is specified, Elastic Transcoder interlaces the output.

For more information, go to the Wikipedia page Interlaced video.

(Optional, H.264/MPEG2 Only) Video:CodecOptions:ColorSpaceConversion

The color space conversion Elastic Transcoder applies to the output video. Color spaces are the algorithms used by the computer to store information about how to render color. `Bt.601` is the standard for standard definition video, while `Bt.709` is the standard for high definition video.

Valid values include `None`, `Bt709toBt601`, `Bt601toBt709`, and `Auto`.

If you chose `Auto` for `ColorSpaceConversionMode` and your output is interlaced, your frame rate is one of `23.97`, `24`, `25`, `29.97`, `50`, or `60`, your `SegmentDuration` is null, and you are using one of the resolution changes from the graph below, Elastic Transcoder applies the following color space conversions:

Input File	Input Resolution	Output Resolution	Applied Color Space Conversion
Standard to HD	720x480	1920x1080	Bt601ToBt709
Standard to HD	720x576	1920x1080	Bt601ToBt709
HD to Standard	1920x1080	720x576	Bt709ToBt601
HD to Standard	1920x1080	720x480	Bt709ToBt601

Note
Elastic Transcoder may change the behavior of the `ColorspaceConversionMode Auto` mode in the future. All outputs in a playlist must use the same `ColorSpaceConversionMode`.

If you do not specify a `ColorSpaceConversionMode`, Elastic Transcoder does not change the color space of a file.

If you are unsure what `ColorSpaceConversionMode` was applied to your output file, you can check the `AppliedColorSpaceConversion` parameter included in your job response. If your job does not have an `AppliedColorSpaceConversion` in its response, no `ColorSpaceConversionMode` was applied.

For more information about color space, go to the Wikipedia page Color space. For more information about `Bt.601` and `Bt.709`, go to the Wikipedia pages Rec. 601 and Rec. 709.

Video:CodecOptions:ChromaSubsampling
The sampling pattern for the chroma (color) channels of the output video. Valid values include `yuv420p` and `yuv422p`.

`yuv420p` samples the chroma information of every other horizontal and every other vertical line, `yuv422p` samples the color information of every horizontal line and every other vertical line.

To learn more about chroma subsampling, go to the wikidepia page Chroma subsampling.

(Gif Only) Video:CodecOptions:LoopCount
The number of times you want the output gif to loop.

Valid values include `Infinite` and integers between `0` and `100`, inclusive.

(H.264/MPEG2/VP8 Only) Video:KeyframesMaxDist
The maximum number of frames between key frames. Not applicable for containers of type `gif`. Key frames are fully encoded frames; the frames between key frames are encoded based, in part, on the content of the key frames. The value is an integer formatted as a string; valid values are between 1 (every frame is a key frame) and 100000, inclusive. A higher value results in higher compression but might also discernibly decrease video quality.

For `Smooth` outputs, the `FrameRate` must have a constant ratio to the `KeyframesMaxDist`. This allows `Smooth` playlists to switch between different quality levels while the file is being played.

For example, an input file can have a `FrameRate` of 30 with a `KeyframesMaxDist` of 90. The output file then needs to have a ratio of 1:3. Valid outputs would have `FrameRate` of 30, 25, and 10, and `KeyframesMaxDist` of 90, 75, and 30, respectively.

Alternately, this can be achieved by setting `FrameRate` to auto and having the same values for `MaxFrameRate` and `KeyframesMaxDist`.

For more information about key frames, see the Wikipedia entry Video compression picture types.

(H.264/MPEG2/VP8 Only) Video:FixedGOP

Whether to use a fixed value for `Video:FixedGOP`. Not applicable for containers of type `gif`. Valid values are `true` and `false`:

- `true`: Elastic Transcoder uses the value of `Video:KeyframesMaxDist` for the distance between key frames (the number of frames in a group of pictures, or GOP).
- `false`: The distance between key frames can vary.

Note

`FixedGOP` must be set to `true` for `fmp4` containers.

Video:BitRate

The bit rate of the video stream in the output file, in kilobits/second. You can configure variable bit rate or constant bit rate encoding:

- **Variable bit rate encoding:** Specify **auto**. Elastic Transcoder optimizes the bit rate and maintains a consistent quality for each frame of the output.
- **Constant bit rate encoding:** Specify the bit rate.

If you specified `H.264` for `Video:Codec`: Valid values depend on the values of the `Video:CodecOptions:Level` and `Video:CodecOptions:Profile` objects.

Note

If you specified `vp8` for `Video:Codec`, do not use the following table; `Level` applies only when the video codec is H.264.

If you specify a value other than `auto`, we recommend that you specify a value less than or equal to the maximum H.264-compliant value listed in the following table for your level and profile:

`Level`	Maximum Video Bit Rate in kilobits/second: Baseline Profile and Main Profile	Maximum Video Bit Rate in kilobits/second: High Profile
1	64	80
1b	128	160
1.1	192	240
1.2	384	480
1.3	768	960
2	2000	2500
2.1	4000	5000
2.2	4000	5000
3	10000	12500
3.1	14000	17500
3.2	20000	25000
4	20000	25000
4.1	50000	62500

Video:FrameRate

The frames per second for the video stream in the output file. Valid values include:

`auto`, `10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `50`, `60`

If you want to preserve the frame rate of the input file and use it for the output file, specify `auto`.

If you specified `H.264` for Video:Codec: If you specify a frame rate, we recommend that you perform the following calculation:

```
Frame rate = maximum recommended decoding speed in luma samples/second / (width in pixels * height in pixels)
```

where:

- `width in pixels` and `height in pixels` represent the `Video:Resolution` of the output video.
- `maximum recommended decoding speed in Luma samples/second` is less than or equal to the maximum value listed in the following table, based on the value that you specified for `Video:CodecOptions:Level`.

  Note
  If you specified `vp8` for `Video:Codec`, do not use the previous equation or the following table; `Level` applies only when the video codec is H.264.

Level	Maximum Recommended Decoding Speed in Luma Samples/Second
1	380160
1b	380160
1.1	768000
1.2	1536000
1.3	3041280
2	3041280
2.1	5068800
2.2	5184000
3	10368000
3.1	27648000
3.2	55296000
4	62914560
4.1	62914560

Video:MaxFrameRate

If you specify `auto` for `FrameRate`, Elastic Transcoder uses the frame rate of the input video for the frame rate of the output video, up to the maximum frame rate. If you do not specify a `MaxFrameRate`, Elastic Transcoder will use a default of `30`.

Specify the maximum frame rate that you want Elastic Transcoder to use when the frame rate of the input video is greater than either the desired maximum frame rate of the output video or the default maximum frame rate. Valid values include:

`10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `50`, `60`

Elastic Transcoder uses the highest supported frame rate that meets both of the following criteria:

- The frame rate is less than or equal to the maximum frame rate.
- The frame rate divides into the input frame rate evenly, with no remainder.

For example, if you have an input file with a frame rate of 50 and specify a value of 30 for `MaxFrameRate`, Elastic Transcoder produces an output video for which the frame rate is 25 frames per second, because 25 is less than 30, and 50 divided by 25 is 2.

Video:MaxWidth

The maximum width of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 128 and 4096, inclusive.

For more information, see `Video:MaxHeight`.

Video:MaxHeight

The maximum height of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 96 and 3072, inclusive.

If you specified `H.264` for `Video:Codec`: We recommend that you specify values for `MaxWidth` and `MaxHeight` so the product of the two values is less than or equal to the applicable value in the following table.

Note
If you specified `vp8` for `Video:Codec`, do not use the following table; `Level` applies only when the video codec is H.264.

Value of Level	Maximum Value of `MaxWidth` x `MaxHeight`
1	25344
1b	25344
1.1	101376
1.2	101376
1.3	101376
2	101376
2.1	202752
2.2	404720
3	404720
3.1	921600
3.2	1310720
4	2097152
4.1	2097152

Video:SizingPolicy

A value that controls scaling of the output video:

- **Fit:** Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.

- **Fill:** Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output video and then crops it to the dimension (if any) that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches the output video to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input video and the output video are different, the output video will be distorted.
- **Keep:** Elastic Transcoder does not scale the output video. If either dimension of the input video exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output video.
- **ShrinkToFit:** Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the video up.
- **ShrinkToFill:** Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the video up.

The following table shows possible effects of `SizingPolicy` settings on the output video:

SizingPolicy	Output Video May Be Scaled Up	Output Video May Be Padded When Padding Policy Is "Pad"	Output Video May Have a Different Pixel Aspect Ratio than Input Video	Output Video May Be Cropped
Fit	Yes	Yes		
Fill	Yes			Yes
Stretch	Yes		Yes	
Keep	Yes	Yes		Yes
ShrinkToFit		Yes		
ShrinkToFill		Yes		Yes

Video:PaddingPolicy

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder might add black bars to the top and bottom and/or left and right sides of the output video to make the total size of the output video match the values that you specified for `MaxWidth` and `MaxHeight`. For more information, see the table at `Video:SizingPolicy`.

Video:DisplayAspectRatio

The value that Elastic Transcoder adds to the metadata in the output file. If you set `DisplayAspectRatio` to `auto`, Elastic Transcoder chooses an aspect ratio that ensures square pixels. If you specify another option, Elastic Transcoder sets that value in the output file.

Video:Resolution

> **Note**
>
> To better control resolution and aspect ratio of output videos, we recommend that you use the Video—Option 1 settings, `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of the two Video—Option 2 settings, `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of the video in the output file, in pixels. Valid values are `auto` and *`width`*x*`height`*:

- `auto`: Elastic Transcoder attempts to preserve the width and height of the input file, subject to the following rules.

- *widthxheight*: The width and height of the output video in pixels.

Note the following about specifying the width and height:

- The width must be an even integer between 128 and 4096, inclusive.
- The height must be an even integer between 96 and 3072, inclusive.
- If you specify a resolution that is less than the resolution of the input file, Elastic Transcoder rescales the output file to the lower resolution.
- If you specify a resolution that is greater than the resolution of the input file, Elastic Transcoder rescales the output to the higher resolution.
- We recommend that you specify a resolution for which the product of width and height is less than or equal to the applicable value in the following table:

Level	Maximum Value of `width` x `height`
1	25344
1b	25344
1.1	101376
1.2	101376
1.3	101376
2	101376
2.1	202752
2.2	404720
3	404720
3.1	921600
3.2	1310720
4	2097152
4.1	2097152

Video:AspectRatio

> **Note**
> To better control resolution and aspect ratio of output videos, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of `Resolution` and `AspectRatio`.

The display aspect ratio of the video in the output file. Valid values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the input file.

If you specify an aspect ratio for the output file that differs from aspect ratio of the input file, Elastic Transcoder adds pillarboxing (black bars on the sides) or letterboxing (black bars on the top and bottom) to maintain the aspect ratio of the active region of the video.

Video:Watermarks

Settings for the size, position, scale, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear for the duration of the transcoded video.

Watermarks can be in `.png` or `.jpg` format. If you want to display a watermark that is not rectangular, use the `.png` format, which supports transparency.

When you create a job that uses this preset, you specify the `.png` or `.jpg` graphics that you want Elastic Transcoder to include in the transcoded videos. Elastic Transcoder does not require you to specify as many watermarks in each job output as you specified in the corresponding preset. For example, you might specify settings for four watermarks in a preset and specify only one watermark in a job output.

To configure watermark settings so your graphic is not distorted, set the value of `SizingPolicy` to `ShrinkToFit`, and set the values of `MaxWidth` and `MaxHeight` to the same percentage. If you want the graphic to appear in the same size as the original, set `MaxWidth` and `MaxHeight` to 100%.

For more information, see Watermarks (p. 69).

Video:Watermarks:Id

A unique identifier for the settings for one watermark. The value of `Id` can be up to 40 characters long. You can specify settings for up to four watermarks.

Video:Watermarks:MaxWidth

The maximum width of the watermark in one of the following formats:

- *`number of pixels`*`px`: The minimum value is 16 pixels, and the maximum value is the value of `Video:MaxWidth`.
- *`integer percentage`*`%`: The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

Video:Watermarks:MaxHeight

The maximum height of the watermark in one of the following formats:

- *`number of pixels`*`px`: The minimum value is 16 pixels, and the maximum value is the value of `Video:MaxHeight`.
- *`integer percentage`*`%`: The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

Video:Watermarks:SizingPolicy

A value that controls scaling of the watermark:

- **Fit:** Elastic Transcoder scales the watermark so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Stretch:** Elastic Transcoder stretches the watermark to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the watermark and the values of `MaxWidth` and `MaxHeight` are different, the watermark will be distorted.
- **ShrinkToFit:** Elastic Transcoder scales the watermark down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the watermark up.

Video:Watermarks:HorizontalAlign

The horizontal position of the watermark unless you specify a nonzero value for `HorizontalOffset`:

- **Left:** The left edge of the watermark is aligned with the left border of the video.
- **Right:** The right edge of the watermark is aligned with the right border of the video.
- **Center:** The watermark is centered between the left and right borders.

Video:Watermarks:HorizontalOffset

The amount by which you want the horizontal position of the watermark to be offset from the position specified by `HorizontalAlign`:

- *number of pixels*px: The minimum value is 0 pixels, and the maximum value is the value of Video:MaxWidth.
- *integer percentage*%: The range of valid values is 0 to 100.

For example, if you specify Left for HorizontalAlign and 5px for HorizontalOffset, the left side of the watermark appears 5 pixels from the left border of the output video.

HorizontalOffset is valid only when the value of HorizontalAlign is Left or Right.

If you specify an offset that causes the watermark to extend beyond the left or right border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of Target to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

Video:Watermarks:VerticalAlign

The vertical position of the watermark unless you specify a nonzero value for VerticalOffset:

- **Top:** The top edge of the watermark is aligned with the top border of the video.
- **Bottom:** The bottom edge of the watermark is aligned with the bottom border of the video.
- **Center:** The watermark is centered between the top and bottom borders.

Video:Watermarks:VerticalOffset

The amount by which you want the vertical position of the watermark to be offset from the position specified by VerticalAlign:

- *number of pixels*px: The minimum value is 0 pixels, and the maximum value is the value of Video:MaxHeight.
- *integer percentage*%: The range of valid values is 0 to 100.

For example, if you specify Top for VerticalAlign and 5px for VerticalOffset, the top of the watermark appears 5 pixels from the top border of the output video.

VerticalOffset is valid only when the value of VerticalAlign is Top or Bottom.

If you specify an offset that causes the watermark to extend beyond the top or bottom border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of Target to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

Video:Watermarks:Opacity

A percentage that indicates how much you want a watermark to obscure the video in the location where it appears. Valid values are 0 (the watermark is invisible) to 100 (the watermark completely obscures the video in the specified location). The data type of Opacity is float.

Elastic Transcoder supports transparent .png graphics. If you use a transparent .png, the transparent portion of the video appears as if you had specified a value of 0 for Opacity. The .jpg file format doesn't support transparency.

Video:Watermarks:Target

A value that determines how Elastic Transcoder interprets values that you specified for Video:Watermarks:HorizontalOffset, Video:Watermarks:VerticalOffset, Video:Watermarks:MaxWidth, and Video:Watermarks:MaxHeight:

- **Content:** HorizontalOffset and VerticalOffset values are calculated based on the borders of the video **excluding** black bars added by Elastic Transcoder, if any.

In addition, `MaxWidth` and `MaxHeight`, if specified as a percentage, are calculated based on the borders of the video **excluding** black bars added by Elastic Transcoder, if any.

- **Frame:** `HorizontalOffset` and `VerticalOffset` values are calculated based on the borders of the video **including** black bars added by Elastic Transcoder, if any.

  In addition, `MaxWidth` and `MaxHeight`, if specified as a percentage, are calculated based on the borders of the video **including** black bars added by Elastic Transcoder, if any.

(Video Only) Thumbnails:Format
The format of thumbnails, if any. Valid formats are `jpg` and `png`.

You specify whether you want Elastic Transcoder to create thumbnails when you create a job. For more information, see ThumbnailPattern (p. 188).

(Video Only) Thumbnails:Interval
The approximate number of seconds between thumbnails. The value must be an integer. The actual interval can vary by several seconds from one thumbnail to the next.

(Video Only) Thumbnails:MaxWidth
The maximum width of thumbnails, in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

(Video Only) Thumbnails:MaxHeight
The maximum height of thumbnails, in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

(Video Only) Thumbnails:SizingPolicy
A value that controls scaling of thumbnails:

- **Fit:** Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings without exceeding the other value.
- **Fill:** Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings and matches or exceeds the other value. Elastic Transcoder centers the image in thumbnails and then crops to the dimension, if any, that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches thumbnails to match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings. If the relative proportions of the input video and thumbnails are different, the thumbnails will be distorted.
- **Keep:** Elastic Transcoder does not scale thumbnails. If either dimension of the input video exceeds the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings, Elastic Transcoder crops the thumbnails.
- **ShrinkToFit:** Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of thumbnail `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.
- **ShrinkToFill:** Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.

The following table shows possible effects of `SizingPolicy` settings on thumbnails:

SizingPolicy	Thumbnails May Be Scaled Up	Thumbnails May Be Padded When Padding Policy Is "Pad"	Thumbnails May Have a Different Pixel Aspect Ratio than Input Video	Thumbnails May Be Cropped
Fit	Yes	Yes		
Fill	Yes			Yes

SizingPolicy	Thumbnails May Be Scaled Up	Thumbnails May Be Padded When Padding Policy Is "Pad"	Thumbnails May Have a Different Pixel Aspect Ratio than Input Video	Thumbnails May Be Cropped
Stretch	Yes		Yes	
Keep	Yes	Yes		Yes
ShrinkToFit		Yes		
ShrinkToFill		Yes		Yes

(Video Only) Thumbnails:PaddingPolicy

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder might add black bars to the top and bottom and/or left and right sides of thumbnails to make the total size of the thumbnails match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings. For more information, see the table at `Thumbnails:SizingPolicy`.

(Video Only) Thumbnails:Resolution

> **Note**
> To better control resolution and aspect ratio of thumbnails, we recommend that you use the thumbnail values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of thumbnail files in pixels, in the format *`Width`*x*`Height`*, where both values are even integers. The values cannot exceed the width and height that you specified in the `Video:Resolution` object.

(Video Only) Thumbnails:AspectRatio

> **Note**
> To better control resolution and aspect ratio of thumbnails, we recommend that you use the thumbnail values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead of `Resolution` and `AspectRatio`.

The aspect ratio of thumbnails. Valid values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the video in the output file.

NextPageToken

A value that you use to access the second and subsequent pages of results, if any. When the presets fit on one page or when you've reached the last page of results, the value of `NextPageToken` is `null`.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

The following example request creates a preset named `DefaultPreset`.

Sample Request

```
GET /2012-09-25/presets HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Sample Response

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Presets":[
      {
         "Id":"5555555555555-abcde5",
         "Type":"Custom",
         "Name":"DefaultPreset",
         "Description":"Use for published videos",
         "Container":"mp4",
         "Audio":{
            "BitRate":"96",
            "Channels":"2",
            "Codec":"AAC",
            "CodecOptions":{
               "Profile":"AAC-LC"
            },
            "SampleRate":"44100"
         },
         "Video":{
            "Codec":"H.264",
            "CodecOptions":{
               "Profile":"main",
               "Level":"2.2",
               "MaxReferenceFrames":"3",
               "MaxBitRate":"",
               "BufferSize":"",
               "InterlacedMode":"Progressive",
               "ColorSpaceConversionMode":"None"
            },
            "KeyframesMaxDist":"240",
            "FixedGOP":"false",
            "BitRate":"1600",
            "FrameRate":"auto",
            "MaxFrameRate":"30",
            "MaxWidth":"auto",
            "MaxHeight":"auto",
```

```
            "SizingPolicy":"Fit",
            "PaddingPolicy":"Pad",
            "DisplayAspectRatio":"auto",
            "Watermarks":[
               {
                  "Id":"company logo",
                  "MaxWidth":"20%",
                  "MaxHeight":"20%",
                  "SizingPolicy":"ShrinkToFit",
                  "HorizontalAlign":"Right",
                  "HorizontalOffset":"10px",
                  "VerticalAlign":"Bottom",
                  "VerticalOffset":"10px",
                  "Opacity":"55.5",
                  "Target":"Content"
               }
            ]
         }
         "Thumbnails":{
            "Format":"png",
            "Interval":"120",
            "MaxWidth":"auto",
            "MaxHeight":"auto",
            "SizingPolicy":"Fit",
            "PaddingPolicy":"Pad"
         },
      },
      {...}
   ]
}
```

Read Preset

Topics

- Description (p. 315)
- Requests (p. 315)
- Responses (p. 316)
- Errors (p. 331)
- Examples (p. 331)

Description

To get detailed information about a preset, send a GET request to the `/2012-09-25/presets/`*`presetId`* resource.

Requests

Syntax

```
GET /2012-09-25/presets/presetId HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
```

Request Parameter

This operation takes the following request parameter.

presetId
The identifier of the preset for which you want to get detailed information.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

This operation does not have a request body.

Responses

Syntax

```
Status: 200 OK
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature

{
   "Preset":{
      "Id":"preset identifier",
      "Type":"Custom|System",
      "Name":"preset name",
      "Description":"preset description",
      "Container":"flv|fmp4|gif|mp3|mp4|mpg|ogg|ts|webm",
      "Audio":{
         "Codec":"AAC|mp2|mp3|vorbis",
         "CodecOptions":{
            "Profile":"auto|AAC-LC|HE-AAC|HE-AACv2"
         },
         "SampleRate":"auto|22050|32000|44100|48000|96000",
         "BitRate":"audio bit rate of output file in kilobits/second",
         "Channels":"auto|0|1|2"
      },
      "Video":{
         "Codec":"gif|H.264|mpeg2|vp8",
         "CodecOptions":{
            "Profile":"baseline|main|high|0|1|2|3",
            "Level":"1|1b|1.1|1.2|1.3|2|2.1|2.2|3|3.1|3.2|4|4.1",
            "MaxReferenceFrames":"maximum number of reference frames",
            "MaxBitRate":"maximum bit rate",
            "BufferSize":"maximum buffer size",
            "InterlacedMode":"Progressive|TopFirst|BottomFirst|Auto",
            "ColorSpaceConversion":"None|Bt709ToBt601|Bt601ToBt709|Auto",
            "ChromaSubsampling":"yuv420p|yuv422p",
            "LoopCount":"Infinite|[0,100]"
         },
         "KeyframesMaxDist":"maximum frames between key frames",
         "FixedGOP":"true|false",
         "BitRate":"auto|video bit rate of output file in kilobits/second",
         "FrameRate":"auto|10|15|23.97|24|25|29.97|30|50|60",
         "MaxFrameRate":"10|15|23.97|24|25|29.97|30|50|60",
         "MaxWidth":"auto|[128,4096]",
         "MaxHeight":"auto|[96,3072]",
         "SizingPolicy":"Fit|Fill|Stretch|Keep|ShrinkToFit|ShrinkToFill",
         "PaddingPolicy":"Pad|NoPad",
         "DisplayAspectRatio":"auto|1:1|4:3|3:2|16:9",
         "Resolution":"auto|width in pixelsxheight in pixels",
         "AspectRatio":"auto|1:1|4:3|3:2|16:9"
         "Watermarks":[
```

```
            {
               "Id":"unique identifier up to 40 characters",
               "MaxWidth":"[16,Video:MaxWidth]px|[0,100]%",
               "MaxHeight":"[16,Video:MaxHeight]px|[0,100]%",
               "SizingPolicy":"Fit|Stretch|ShrinkToFit",
               "HorizontalAlign":"Left|Right|Center",
               "HorizontalOffset":"[0,100]%|[0,Video:MaxWidth]px",
               "VerticalAlign":"Top|Bottom|Center",
               "VerticalOffset":"[0,100]%|[0,Video:MaxHeight]px",
               "Opacity":"[0,100]",
               "Target":"Content|Frame"
            },
            {...}
         ]
      },
      "Thumbnails":{
         "Format":"jpg|png",
         "Interval":"number of seconds between thumbnails",
         "MaxWidth":"auto|[32,4096]",
         "MaxHeight":"auto|[32,3072]",
         "SizingPolicy":"Fit|Fill|Stretch|Keep|ShrinkToFit|ShrinkToFill",
         "PaddingPolicy":"Pad|NoPad",
         "Resolution":"width in pixelsxheight in pixels",
         "AspectRatio":"auto|1:1|4:3|3:2|16:9"
      }
   }
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

When you get a preset, Elastic Transcoder returns the values that you specified when you created the preset. For more information, see Request Body (p. 277).

In addition, Elastic Transcoder returns the following values.

Id
Identifier for the new preset. You use this value to get settings for the preset or to delete it.

Type
Whether the preset is a default preset provided by Elastic Transcoder (`System`) or a preset that you have defined (`Custom`).

Name
The name of the preset. We recommend that the name be unique within the AWS account, but uniqueness is not enforced.

Constraints: Maximum 40 characters

Description
A description of the preset.

Constraints: Maximum 255 characters

Container
The container type for the output file. Valid values include `flv`, `fmp4`, `gif`, `mp3`, `mp4`, `mpg`, `ogg`, `ts`, and `webm`. The following restrictions apply:

Container	Audio Codecs	Video Codecs
FLV	AAC, mp3	H.264
FMP4	AAC	H.264
GIF	Not Allowed	gif
MP3	mp3	Not Allowed
MP4	AAC, mp3	H.264
MPG	mp2	mpeg2
OGG	vorbis	Not Allowed
TS	AAC, mp3	H.264
WEBM	vorbis	vp8

Audio:Codec

The audio codec for the output file. Valid values include `AAC`, `mp2`, `mp3`, and `vorbis`. The following chart marks the allowed combinations:

Container & Audio Codecs	AAC	MP2	MP3	Vorbis
FLV	x		x	
GIF				
FMP4	x			
MP3			x	
MP4	x		x	
MPG		x		
OGG				x
TS	x		x	
WEBM				x

(AAC Only) Audio:CodecOptions:Profile

If you specified `AAC` for `Audio:Codec`, choose the AAC profile for the output file. Elastic Transcoder supports the following profiles:

- `auto`: If you specify `auto`, Elastic Transcoder selects the profile based on the bit rate selected for the output file.
- `AAC-LC`: The most common AAC profile. Use for bit rates larger than 64 kbps. For more information, see Advanced Audio Coding.
- `HE-AAC`: Not supported on some older players and devices. Use for bit rates between 40 and 80 kbps. For more information, see High-Efficiency Advanced Audio Coding.
- `HE-AACv2`: Not supported on some players and devices. Use for bit rates less than 48 kbps. For more information, see High-Efficiency Advanced Audio Coding..

All outputs in a `Smooth` playlist must have the same value for `Profile`.

Note
If you created any presets before AAC profiles were added, Elastic Transcoder automatically updated your presets to use AAC-LC.

For more information about AAC, go to Audio Profiles in the Wikipedia entry "MPEG-4 Part 3."

Audio:SampleRate
The sample rate of the audio stream in the output file, in hertz. Valid values include:

`auto`, `22050`, `32000`, `44100`, `48000`, `96000`

If you specify `auto`, Elastic Transcoder automatically detects the sample rate.

Audio:BitRate
The bit rate of the audio stream in the output file, in kilobits/second. Enter an integer between 64 and 320, inclusive.

Audio:Channels
The number of audio channels in the output file. Valid values include:

`auto`, `0`, `1`, `2`

If you want Elastic Transcoder to automatically detect the number of channels in the input file and use that value for the output file, specify `auto`. If you want Elastic Transcoder to omit audio from the output file, specify `0`.

Video:Codec
The video codec for the output file. Valid values include `gif`, `H.264`, `mpeg2`, and `vp8`. The following chart marks the allowed combinations:

Container & Video Codecs	GIF	H.264	MPEG2	VP8
FLV		x		
GIF	x			
FMP4		x		
MP3				
MP4		x		
MPG			x	
OGG				
TS		x		
WEBM				x

For more information about the H.264 video-compression format, go to the wikipedia page on H.264/MPEG-4 AVC.

For more information about VP8, see the WebM project website.

(H.264/VP8 Only) Video:CodecOptions:Profile
If you specified `H.264` for `Video:Codec`, the H.264 profile that you want to use for the output file. Elastic Transcoder supports the following profiles:

- `baseline`: The profile most commonly used for video conferencing and for mobile applications.
- `main`: The profile used for standard-definition digital TV broadcasts.
- `high`: The profile used for high-definition digital TV broadcasts and for Blu-ray discs.

If you specified `vp8` for the video codec, the vp8 profile that you want to use for the output file. Elastic Transcoder supports the following profiles: `0`, `1`, `2`, `3`. You can specify `0`, `1`, `2`, or `3` only when the container type is `webm`.

For more information about profiles, see Profiles in the Wikipedia entry "H.264/MPEG-4 AVC."

(H.264 Only) Video:CodecOptions:Level

Applicable only when the value of `Video:Codec` is `H.264`. The H.264 level that you want to use for the output file. Elastic Transcoder supports the following levels:

`1`, `1b`, `1.1`, `1.2`, `1.3`, `2`, `2.1`, `2.2`, `3`, `3.1`, `3.2`, `4`, `4.1`

For more information about levels, see Levels in the Wikipedia entry "H.264/MPEG-4 AVC."

(H.264 Only) Video:CodecOptions:MaxReferenceFrames

Applicable only when the value of `Video:Codec` is `H.264`. The maximum number of previously decoded frames to use as a reference for decoding future frames. Valid values are integers 0 through 16, but we recommend that you not use a value greater than:

```
Min(Floor(Maximum decoded picture buffer in macroblocks * 256 / (Width in pixels * Height in pixels)), 16)
```

where:

- `Width in pixels` and `Height in pixels` represent either `Video:MaxWidth` and `Video:MaxHeight`, or `Video:Resolution`.
- `Maximum decoded picture buffer in macroblocks` depends on the value of the `Video:CodecOptions:Level` object. (A macroblock is a block of pixels measuring 16x16.) See the table below.

For more information about encoding based on previously encoded pictures, see Decoded picture buffering in the Wikipedia entry "H.264/MPEG-4 AVC." Note that the Wikipedia calculation for maximum decoded picture buffer, which is similar to the calculation for maximum reference frames, uses macroblocks instead of pixels for the width and height of the video.

`Video:CodecOptions:Level`	Maximum Decoded Picture Buffer in Macroblocks
1	396
1b	396
1.1	900
1.2	2376
1.3	2376
2	2376
2.1	4752
2.2	8100
3	8100
3.1	18000
3.2	20480
4	32768
4.1	32768

Video:CodecOptions:MaxBitRate

The maximum number of kilobits per second in the output video. Specify a value between 16 and 62,500, inclusive.

If you specify `auto` for `BitRate`, Elastic Transcoder uses the bit rate of the input video as the average bit rate of the output video. `MaxBitRate` allows you to cap the bit rate of the output video, which is useful when the maximum bit rate supported by a target device is lower than the bit rate of the input video. Reducing the maximum bit rate might reduce the quality of the video.

(Optional, H.264/MPEG2/VP8 only) Video:CodecOptions:BufferSize

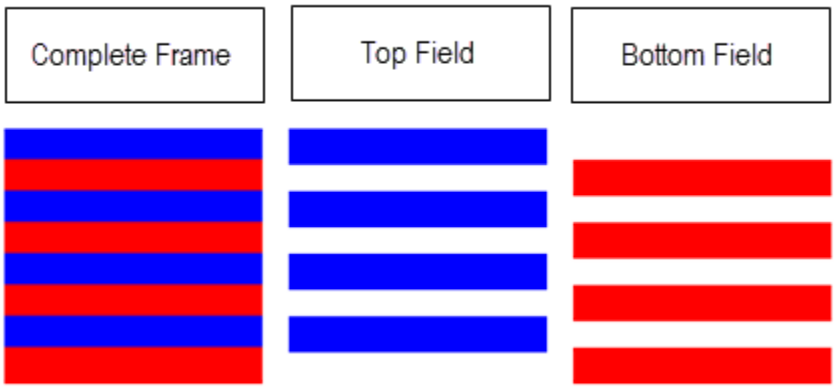
The maximum number of kilobits in any *x* seconds of the output video. This window is commonly 10 seconds, the standard segment duration when you're using ts for the container type of the output video. Specify an integer greater than 0. If you specify `MaxBitRate` and omit `BufferSize`, Elastic Transcoder sets `BufferSize` to 10 times the value of `MaxBitRate`.

(Optional, H.264/MPEG2 Only) Video:CodecOptions:InterlacedMode

The interlace mode for the output video.

Interlaced video is used to double the perceived frame rate for a video by interlacing two fields (one field on every other line, the other field on the other lines) so that the human eye registers multiple pictures per frame. Interlacing reduces the bandwidth required for transmitting a video, but can result in blurred images and flickering.

The two sets of lines are known as fields, and an interlaced frame splits two images across the fields:



Valid values include `Progressive` (no interlacing, top to bottom), `TopFirst` (top field first), `BottomFirst` (bottom field first), and `Auto`.

If `InterlaceMode` is not specified, Elastic Transcoder uses `Progressive` for the output. If `Auto` is specified, Elastic Transcoder interlaces the output.

For more information, go to the Wikipedia page Interlaced video.

(Optional, H.264/MPEG2 Only) Video:CodecOptions:ColorSpaceConversion

The color space conversion Elastic Transcoder applies to the output video. Color spaces are the algorithms used by the computer to store information about how to render color. `Bt.601` is the standard for standard definition video, while `Bt.709` is the standard for high definition video.

Valid values include `None`, `Bt709toBt601`, `Bt601toBt709`, and `Auto`.

If you chose `Auto` for `ColorSpaceConversionMode` and your output is interlaced, your frame rate is one of `23.97`, `24`, `25`, `29.97`, `50`, or `60`, your `SegmentDuration` is null, and you are using one of the resolution changes from the graph below, Elastic Transcoder applies the following color space conversions:

Input File	Input Resolution	Output Resolution	Applied Color Space Conversion
Standard to HD	720x480	1920x1080	Bt601ToBt709
Standard to HD	720x576	1920x1080	Bt601ToBt709
HD to Standard	1920x1080	720x576	Bt709ToBt601
HD to Standard	1920x1080	720x480	Bt709ToBt601

Note
Elastic Transcoder may change the behavior of the `ColorspaceConversionMode Auto` mode in the future. All outputs in a playlist must use the same `ColorSpaceConversionMode`.

If you do not specify a `ColorSpaceConversionMode`, Elastic Transcoder does not change the color space of a file.

If you are unsure what `ColorSpaceConversionMode` was applied to your output file, you can check the `AppliedColorSpaceConversion` parameter included in your job response. If your job does not have an `AppliedColorSpaceConversion` in its response, no `ColorSpaceConversionMode` was applied.

For more information about color space, go to the Wikipedia page Color space. For more information about `Bt.601` and `Bt.709`, go to the Wikipedia pages Rec. 601 and Rec. 709.

Video:CodecOptions:ChromaSubsampling
The sampling pattern for the chroma (color) channels of the output video. Valid values include `yuv420p` and `yuv422p`.

`yuv420p` samples the chroma information of every other horizontal and every other vertical line, `yuv422p` samples the color information of every horizontal line and every other vertical line.

To learn more about chroma subsampling, go to the wikidepia page Chroma subsampling.

(Gif Only) Video:CodecOptions:LoopCount
The number of times you want the output gif to loop.

Valid values include `Infinite` and integers between `0` and `100`, inclusive.

(H.264/MPEG2/VP8 Only) Video:KeyframesMaxDist
The maximum number of frames between key frames. Not applicable for containers of type `gif`. Key frames are fully encoded frames; the frames between key frames are encoded based, in part, on the content of the key frames. The value is an integer formatted as a string; valid values are between 1 (every frame is a key frame) and 100000, inclusive. A higher value results in higher compression but might also discernibly decrease video quality.

For `Smooth` outputs, the `FrameRate` must have a constant ratio to the `KeyframesMaxDist`. This allows `Smooth` playlists to switch between different quality levels while the file is being played.

For example, an input file can have a `FrameRate` of 30 with a `KeyframesMaxDist` of 90. The output file then needs to have a ratio of 1:3. Valid outputs would have `FrameRate` of 30, 25, and 10, and `KeyframesMaxDist` of 90, 75, and 30, respectively.

Alternately, this can be achieved by setting `FrameRate` to auto and having the same values for `MaxFrameRate` and `KeyframesMaxDist`.

For more information about key frames, see the Wikipedia entry Video compression picture types.

(H.264/MPEG2/VP8 Only) Video:FixedGOP

Whether to use a fixed value for `Video:FixedGOP`. Not applicable for containers of type `gif`. Valid values are `true` and `false`:

- `true`: Elastic Transcoder uses the value of `Video:KeyframesMaxDist` for the distance between key frames (the number of frames in a group of pictures, or GOP).
- `false`: The distance between key frames can vary.

> **Note**
> `FixedGOP` must be set to `true` for `fmp4` containers.

Video:BitRate

The bit rate of the video stream in the output file, in kilobits/second. You can configure variable bit rate or constant bit rate encoding:

- **Variable bit rate encoding:** Specify **auto**. Elastic Transcoder optimizes the bit rate and maintains a consistent quality for each frame of the output.
- **Constant bit rate encoding:** Specify the bit rate.

If you specified `H.264` for `Video:Codec`: Valid values depend on the values of the `Video:CodecOptions:Level` and `Video:CodecOptions:Profile` objects.

> **Note**
> If you specified `vp8` for `Video:Codec`, do not use the following table; `Level` applies only when the video codec is H.264.

If you specify a value other than `auto`, we recommend that you specify a value less than or equal to the maximum H.264-compliant value listed in the following table for your level and profile:

`Level`	Maximum Video Bit Rate in kilobits/second: Baseline Profile and Main Profile	Maximum Video Bit Rate in kilobits/second: High Profile
1	64	80
1b	128	160
1.1	192	240
1.2	384	480
1.3	768	960
2	2000	2500
2.1	4000	5000
2.2	4000	5000
3	10000	12500
3.1	14000	17500
3.2	20000	25000
4	20000	25000
4.1	50000	62500

Video:FrameRate

The frames per second for the video stream in the output file. Valid values include:

`auto`, `10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `50`, `60`

If you want to preserve the frame rate of the input file and use it for the output file, specify `auto`.

If you specified `H.264` for Video:Codec: If you specify a frame rate, we recommend that you perform the following calculation:

```
Frame rate = maximum recommended decoding speed in luma samples/second /
(width in pixels * height in pixels)
```

where:

- `width in pixels` and `height in pixels` represent the `Video:Resolution` of the output video.
- `maximum recommended decoding speed in Luma samples/second` is less than or equal to the maximum value listed in the following table, based on the value that you specified for `Video:CodecOptions:Level`.

  Note
  If you specified `vp8` for `Video:Codec`, do not use the previous equation or the following table; `Level` applies only when the video codec is H.264.

Level	Maximum Recommended Decoding Speed in Luma Samples/Second
1	380160
1b	380160
1.1	768000
1.2	1536000
1.3	3041280
2	3041280
2.1	5068800
2.2	5184000
3	10368000
3.1	27648000
3.2	55296000
4	62914560
4.1	62914560

Video:MaxFrameRate

If you specify `auto` for `FrameRate`, Elastic Transcoder uses the frame rate of the input video for the frame rate of the output video, up to the maximum frame rate. If you do not specify a `MaxFrameRate`, Elastic Transcoder will use a default of `30`.

Specify the maximum frame rate that you want Elastic Transcoder to use when the frame rate of the input video is greater than either the desired maximum frame rate of the output video or the default maximum frame rate. Valid values include:

`10`, `15`, `23.97`, `24`, `25`, `29.97`, `30`, `50`, `60`

Elastic Transcoder uses the highest supported frame rate that meets both of the following criteria:

- The frame rate is less than or equal to the maximum frame rate.
- The frame rate divides into the input frame rate evenly, with no remainder.

For example, if you have an input file with a frame rate of 50 and specify a value of 30 for `MaxFrameRate`, Elastic Transcoder produces an output video for which the frame rate is 25 frames per second, because 25 is less than 30, and 50 divided by 25 is 2.

Video:MaxWidth

The maximum width of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 128 and 4096, inclusive.

For more information, see `Video:MaxHeight`.

Video:MaxHeight

The maximum height of the output video in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 96 and 3072, inclusive.

If you specified `H.264` for `Video:Codec`: We recommend that you specify values for `MaxWidth` and `MaxHeight` so the product of the two values is less than or equal to the applicable value in the following table.

Note
If you specified `vp8` for `Video:Codec`, do not use the following table; `Level` applies only when the video codec is H.264.

Value of Level	Maximum Value of `MaxWidth` x `MaxHeight`
1	25344
1b	25344
1.1	101376
1.2	101376
1.3	101376
2	101376
2.1	202752
2.2	404720
3	404720
3.1	921600
3.2	1310720
4	2097152
4.1	2097152

Video:SizingPolicy

A value that controls scaling of the output video:

- **Fit:** Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.

- **Fill:** Elastic Transcoder scales the output video so it matches the value that you specified in either `MaxWidth` or `MaxHeight` and matches or exceeds the other value. Elastic Transcoder centers the output video and then crops it to the dimension (if any) that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches the output video to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the input video and the output video are different, the output video will be distorted.
- **Keep:** Elastic Transcoder does not scale the output video. If either dimension of the input video exceeds the values that you specified for `MaxWidth` and `MaxHeight`, Elastic Transcoder crops the output video.
- **ShrinkToFit:** Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the video up.
- **ShrinkToFill:** Elastic Transcoder scales the output video down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale the video up.

The following table shows possible effects of `SizingPolicy` settings on the output video:

SizingPolicy	Output Video May Be Scaled Up	Output Video May Be Padded When Padding Policy Is "Pad"	Output Video May Have a Different Pixel Aspect Ratio than Input Video	Output Video May Be Cropped
Fit	Yes	Yes		
Fill	Yes			Yes
Stretch	Yes		Yes	
Keep	Yes	Yes		Yes
ShrinkToFit		Yes		
ShrinkToFill		Yes		Yes

Video:PaddingPolicy

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder might add black bars to the top and bottom and/or left and right sides of the output video to make the total size of the output video match the values that you specified for `MaxWidth` and `MaxHeight`. For more information, see the table at `Video:SizingPolicy`.

Video:DisplayAspectRatio

The value that Elastic Transcoder adds to the metadata in the output file. If you set `DisplayAspectRatio` to `auto`, Elastic Transcoder chooses an aspect ratio that ensures square pixels. If you specify another option, Elastic Transcoder sets that value in the output file.

Video:Resolution

> **Note**
>
> To better control resolution and aspect ratio of output videos, we recommend that you use the Video—Option 1 settings, `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of the two Video—Option 2 settings, `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of the video in the output file, in pixels. Valid values are `auto` and *`width`*x*`height`*:

- `auto`: Elastic Transcoder attempts to preserve the width and height of the input file, subject to the following rules.

- *width*x*height*: The width and height of the output video in pixels.

Note the following about specifying the width and height:

- The width must be an even integer between 128 and 4096, inclusive.
- The height must be an even integer between 96 and 3072, inclusive.
- If you specify a resolution that is less than the resolution of the input file, Elastic Transcoder rescales the output file to the lower resolution.
- If you specify a resolution that is greater than the resolution of the input file, Elastic Transcoder rescales the output to the higher resolution.
- We recommend that you specify a resolution for which the product of width and height is less than or equal to the applicable value in the following table:

Level	Maximum Value of `width` x `height`
1	25344
1b	25344
1.1	101376
1.2	101376
1.3	101376
2	101376
2.1	202752
2.2	404720
3	404720
3.1	921600
3.2	1310720
4	2097152
4.1	2097152

Video:AspectRatio

> **Note**
> To better control resolution and aspect ratio of output videos, we recommend that you use the values `MaxWidth`, `MaxHeight`, `SizingPolicy`, `PaddingPolicy`, and `DisplayAspectRatio` instead of `Resolution` and `AspectRatio`.

The display aspect ratio of the video in the output file. Valid values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the input file.

If you specify an aspect ratio for the output file that differs from aspect ratio of the input file, Elastic Transcoder adds pillarboxing (black bars on the sides) or letterboxing (black bars on the top and bottom) to maintain the aspect ratio of the active region of the video.

Video:Watermarks

Settings for the size, position, scale, and opacity of graphics that you want Elastic Transcoder to overlay over videos that are transcoded using this preset. You can specify settings for up to four watermarks. Watermarks appear for the duration of the transcoded video.

Watermarks can be in `.png` or `.jpg` format. If you want to display a watermark that is not rectangular, use the `.png` format, which supports transparency.

When you create a job that uses this preset, you specify the `.png` or `.jpg` graphics that you want Elastic Transcoder to include in the transcoded videos. Elastic Transcoder does not require you to specify as many watermarks in each job output as you specified in the corresponding preset. For example, you might specify settings for four watermarks in a preset and specify only one watermark in a job output.

To configure watermark settings so your graphic is not distorted, set the value of `SizingPolicy` to `ShrinkToFit`, and set the values of `MaxWidth` and `MaxHeight` to the same percentage. If you want the graphic to appear in the same size as the original, set `MaxWidth` and `MaxHeight` to 100%.

For more information, see Watermarks (p. 69).

Video:Watermarks:Id

A unique identifier for the settings for one watermark. The value of `Id` can be up to 40 characters long. You can specify settings for up to four watermarks.

Video:Watermarks:MaxWidth

The maximum width of the watermark in one of the following formats:

- *`number of pixels`*`px`: The minimum value is 16 pixels, and the maximum value is the value of `Video:MaxWidth`.
- *`integer percentage`*`%`: The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

Video:Watermarks:MaxHeight

The maximum height of the watermark in one of the following formats:

- *`number of pixels`*`px`: The minimum value is 16 pixels, and the maximum value is the value of `Video:MaxHeight`.
- *`integer percentage`*`%`: The range of valid values is 0 to 100. Use the value of `Target` to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the calculation.

Video:Watermarks:SizingPolicy

A value that controls scaling of the watermark:

- **Fit:** Elastic Transcoder scales the watermark so it matches the value that you specified in either `MaxWidth` or `MaxHeight` without exceeding the other value.
- **Stretch:** Elastic Transcoder stretches the watermark to match the values that you specified for `MaxWidth` and `MaxHeight`. If the relative proportions of the watermark and the values of `MaxWidth` and `MaxHeight` are different, the watermark will be distorted.
- **ShrinkToFit:** Elastic Transcoder scales the watermark down so that its dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale the watermark up.

Video:Watermarks:HorizontalAlign

The horizontal position of the watermark unless you specify a nonzero value for `HorizontalOffset`:

- **Left:** The left edge of the watermark is aligned with the left border of the video.
- **Right:** The right edge of the watermark is aligned with the right border of the video.
- **Center:** The watermark is centered between the left and right borders.

Video:Watermarks:HorizontalOffset

The amount by which you want the horizontal position of the watermark to be offset from the position specified by `HorizontalAlign`:

- *number of pixels*px: The minimum value is 0 pixels, and the maximum value is the value of Video:MaxWidth.
- *integer percentage*%: The range of valid values is 0 to 100.

For example, if you specify Left for HorizontalAlign and 5px for HorizontalOffset, the left side of the watermark appears 5 pixels from the left border of the output video.

HorizontalOffset is valid only when the value of HorizontalAlign is Left or Right.

If you specify an offset that causes the watermark to extend beyond the left or right border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of Target to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

Video:Watermarks:VerticalAlign

The vertical position of the watermark unless you specify a nonzero value for VerticalOffset:

- **Top:** The top edge of the watermark is aligned with the top border of the video.
- **Bottom:** The bottom edge of the watermark is aligned with the bottom border of the video.
- **Center:** The watermark is centered between the top and bottom borders.

Video:Watermarks:VerticalOffset

The amount by which you want the vertical position of the watermark to be offset from the position specified by VerticalAlign:

- *number of pixels*px: The minimum value is 0 pixels, and the maximum value is the value of Video:MaxHeight.
- *integer percentage*%: The range of valid values is 0 to 100.

For example, if you specify Top for VerticalAlign and 5px for VerticalOffset, the top of the watermark appears 5 pixels from the top border of the output video.

VerticalOffset is valid only when the value of VerticalAlign is Top or Bottom.

If you specify an offset that causes the watermark to extend beyond the top or bottom border and Elastic Transcoder has not added black bars, the watermark is cropped. If Elastic Transcoder has added black bars, the watermark extends into the black bars. If the watermark extends beyond the black bars, it is cropped.

Use the value of Target to specify whether you want Elastic Transcoder to include the black bars that are added by Elastic Transcoder, if any, in the offset calculation.

Video:Watermarks:Opacity

A percentage that indicates how much you want a watermark to obscure the video in the location where it appears. Valid values are 0 (the watermark is invisible) to 100 (the watermark completely obscures the video in the specified location). The data type of Opacity is float.

Elastic Transcoder supports transparent .png graphics. If you use a transparent .png, the transparent portion of the video appears as if you had specified a value of 0 for Opacity. The .jpg file format doesn't support transparency.

Video:Watermarks:Target

A value that determines how Elastic Transcoder interprets values that you specified for Video:Watermarks:HorizontalOffset, Video:Watermarks:VerticalOffset, Video:Watermarks:MaxWidth, and Video:Watermarks:MaxHeight:

- **Content:** HorizontalOffset and VerticalOffset values are calculated based on the borders of the video **excluding** black bars added by Elastic Transcoder, if any.

In addition, `MaxWidth` and `MaxHeight`, if specified as a percentage, are calculated based on the borders of the video **excluding** black bars added by Elastic Transcoder, if any.

- **Frame:** `HorizontalOffset` and `VerticalOffset` values are calculated based on the borders of the video **including** black bars added by Elastic Transcoder, if any.

  In addition, `MaxWidth` and `MaxHeight`, if specified as a percentage, are calculated based on the borders of the video **including** black bars added by Elastic Transcoder, if any.

(Video Only) Thumbnails:Format
The format of thumbnails, if any. Valid formats are `jpg` and `png`.

You specify whether you want Elastic Transcoder to create thumbnails when you create a job. For more information, see ThumbnailPattern (p. 188).

(Video Only) Thumbnails:Interval
The approximate number of seconds between thumbnails. The value must be an integer. The actual interval can vary by several seconds from one thumbnail to the next.

(Video Only) Thumbnails:MaxWidth
The maximum width of thumbnails, in pixels. If you specify `auto`, Elastic Transcoder uses 1920 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 4096, inclusive.

(Video Only) Thumbnails:MaxHeight
The maximum height of thumbnails, in pixels. If you specify `auto`, Elastic Transcoder uses 1080 (Full HD) as the default value. If you specify a numeric value, enter an even integer between 32 and 3072, inclusive.

(Video Only) Thumbnails:SizingPolicy
A value that controls scaling of thumbnails:

- **Fit:** Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings without exceeding the other value.
- **Fill:** Elastic Transcoder scales thumbnails so they match the value that you specified in thumbnail `MaxWidth` or `MaxHeight` settings and matches or exceeds the other value. Elastic Transcoder centers the image in thumbnails and then crops to the dimension, if any, that exceeds the maximum value.
- **Stretch:** Elastic Transcoder stretches thumbnails to match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings. If the relative proportions of the input video and thumbnails are different, the thumbnails will be distorted.
- **Keep:** Elastic Transcoder does not scale thumbnails. If either dimension of the input video exceeds the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings, Elastic Transcoder crops the thumbnails.
- **ShrinkToFit:** Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of thumbnail `MaxWidth` and `MaxHeight` without exceeding either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.
- **ShrinkToFill:** Elastic Transcoder scales thumbnails down so that their dimensions match the values that you specified for at least one of `MaxWidth` and `MaxHeight` without dropping below either value. If you specify this option, Elastic Transcoder does not scale thumbnails up.

The following table shows possible effects of `SizingPolicy` settings on thumbnails:

SizingPolicy	Thumbnails May Be Scaled Up	Thumbnails May Be Padded When Padding Policy Is "Pad"	Thumbnails May Have a Different Pixel Aspect Ratio than Input Video	Thumbnails May Be Cropped
Fit	Yes	Yes		
Fill	Yes			Yes

SizingPolicy	Thumbnails May Be Scaled Up	Thumbnails May Be Padded When Padding Policy Is "Pad"	Thumbnails May Have a Different Pixel Aspect Ratio than Input Video	Thumbnails May Be Cropped
Stretch	Yes		Yes	
Keep	Yes	Yes		Yes
ShrinkToFit		Yes		
ShrinkToFill		Yes		Yes

(Video Only) Thumbnails:PaddingPolicy

When you set `PaddingPolicy` to `Pad`, Elastic Transcoder might add black bars to the top and bottom and/or left and right sides of thumbnails to make the total size of the thumbnails match the values that you specified for thumbnail `MaxWidth` and `MaxHeight` settings. For more information, see the table at `Thumbnails:SizingPolicy`.

(Video Only) Thumbnails:Resolution

> **Note**
> To better control resolution and aspect ratio of thumbnails, we recommend that you use the thumbnail values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead of `Resolution` and `AspectRatio`. The two groups of settings are mutually exclusive. Do not use them together.

The width and height of thumbnail files in pixels, in the format *`Width`*x*`Height`*, where both values are even integers. The values cannot exceed the width and height that you specified in the `Video:Resolution` object.

(Video Only) Thumbnails:AspectRatio

> **Note**
> To better control resolution and aspect ratio of thumbnails, we recommend that you use the thumbnail values `MaxWidth`, `MaxHeight`, `SizingPolicy`, and `PaddingPolicy` instead of `Resolution` and `AspectRatio`.

The aspect ratio of thumbnails. Valid values include:

`auto`, `1:1`, `4:3`, `3:2`, `16:9`

If you specify `auto`, Elastic Transcoder tries to preserve the aspect ratio of the video in the output file.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

The following example request gets the preset that has the preset ID `5555555555555-abcde5`.

Sample Request

```
GET /2012-09-25/presets/5555555555555-abcde5 HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
Content-Length: number of characters in the JSON string
```

Sample Response

```
Status: 200 OK
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature

{
   "Preset":{
      "Id":"5555555555555-abcde5",
      "Type":"Custom",
      "Name":"DefaultPreset",
      "Description":"Use for published videos",
      "Container":"mp4",
      "Audio":{
         "Codec":"AAC",
         "CodecOptions":{
            "Profile":"AAC-LC"
         },
         "SampleRate":"44100",
         "BitRate":"96",
         "Channels":"2"
      },
      "Video":{
         "Codec":"H.264",
         "CodecOptions":{
            "Profile":"main",
            "Level":"2.2",
            "MaxReferenceFrames":"3",
            "MaxBitRate":"",
            "BufferSize":"",
            "InterlacedMode":"Progressive",
            "ColorSpaceConversionMode":"None"
         },
         "KeyframesMaxDist":"240",
         "FixedGOP":"false",
```

```
         "BitRate":"1600",
         "FrameRate":"auto",
         "MaxFrameRate":"30",
         "MaxWidth":"auto",
         "MaxHeight":"auto",
         "SizingPolicy":"Fit",
         "PaddingPolicy":"Pad",
         "DisplayAspectRatio":"auto",
         "Watermarks":[
            {
               "Id":"company logo",
               "MaxWidth":"20%",
               "MaxHeight":"20%",
               "SizingPolicy":"ShrinkToFit",
               "HorizontalAlign":"Right",
               "HorizontalOffset":"10px",
               "VerticalAlign":"Bottom",
               "VerticalOffset":"10px",
               "Opacity":"55.5",
               "Target":"Content"
            }
         ]
      }
      "Thumbnails":{
         "Format":"png",
         "Interval":"120",
         "MaxHeight":"auto",
         "MaxWidth":"auto",
         "SizingPolicy":"Fit",
         "PaddingPolicy":"Pad"
      },
   },
   "Warning":""
}
```

Delete Preset

Topics

- Description (p. 334)
- Requests (p. 334)
- Responses (p. 335)
- Errors (p. 335)
- Examples (p. 335)

Description

To delete a preset, send a DELETE request to the `/2012-09-25/presets/`*`presetId`* resource.

> **Note**
> If the preset has been used, you cannot delete it.

Requests

Syntax

```
DELETE /2012-09-25/presets/presetId HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Request Parameters

This operation takes the following request parameter.

presetId
The identifier of the preset for which you want to get detailed information.

Request Headers

This operation uses only request headers that are common to all operations. For information about common request headers, see HTTP Header Contents (p. 114).

Request Body

This operation does not have a request body.

Responses

Syntax

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Success":"true"
}
```

Response Headers

This operation uses only response headers that are common to most responses. For information about common response headers, see HTTP Responses (p. 116).

Response Body

The response body contains the following JSON object.

Success
If the preset is successfully deleted, the value of `Success` is `true`.

Errors

For information about Elastic Transcoder exceptions and error messages, see Handling Errors in Elastic Transcoder (p. 118).

Examples

The following example request deletes the preset that has the ID `5555555555555-abcde5`.

Sample Request

```
DELETE /2012-09-25/pipelines/5555555555555-abcde5 HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder end
point/ets/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

Sample Response

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
```

```
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Success":"true"
}
```

Document History

The following table describes the important changes to the documentation in this release of Amazon Elastic Transcoder.

- **API Version:** 2012-09-25
- **Last documentation update:** January 13th, 2015

Change	Description	Release Date
New Features	With this release, Elastic Transcoder adds support for: • **Extended format support (PAL, NTSC, and more):** Elastic Transcoder now supports additional formats, interlaced signalling, chroma subsampling, and color conversion. For more information, see Preset Settings (p. 41).	March 17th, 2015
New Features	With this release, Elastic Transcoder adds support for: • **Cross-Regional Warnings:** Elastic Transcoder now warns you when you create, update, or retrieve a pipeline with cross-regional settings. For more information, see Warnings (p. 134).	February 19th, 2015
New Features	With this release, Elastic Transcoder adds support for: • **HLS Content Protection:** You can now set content protection settings for your streamed media. For more information, see (Fragmented MP4/MPEG-TS Outputs Only, Optional) HLS Content Protection (p. 36).	January 13th, 2015
New Features	With this release, Elastic Transcoder adds support for: • **User-Defined Metadata:** You can now create custom metadata for your files, which Elastic Transcoder will pass through the transcoding process unchanged. For more information, see (Optional) User Metadata (p. 34).	December 10th, 2014

Change	Description	Release Date
New Features	With this release, Elastic Transcoder adds support for: • **Encrypted Files:** You can now encrypt your output files and transcode files that are encrypted. For more information, see Data Encryption Options (p. 60).	November 24th, 2014
New Features	With this release, Elastic Transcoder adds support for: • **Logging with CloudTrail:** You can now set up logging of Elastic Transcoder apis calls through your account. For more information, see Logging Elastic Transcoder API Calls Using CloudTrail (p. 81).	October 27th, 2014
New Features	With this release, Elastic Transcoder adds support for: • **HTTP Live Streaming v4:** You can now generate HTTP Live Streaming v4 streaming files. For more information, see (Fragmented MP4/MPEG-TS Outputs Only) Playlist (p. 35).	October 13th, 2014
New Features	With this release, Elastic Transcoder adds support for: • **Smooth Streaming:** You can now generate Microsoft Smooth Streaming files. For more information, see (Fragmented MP4/MPEG-TS Outputs Only) Playlist (p. 35).	October 1st, 2014
New Features	With this release, Elastic Transcoder adds support for: • **Closed Captions:** You can now include closed captions in transcoded output. For more information, see Output Details, Part 2, Clip and Caption Settings (p. 25).	June 18th, 2014
New Features	With this release, Elastic Transcoder adds support for: • **Audio profiles:** You can now configure Elastic Transcoder to use different audio compression schemes, including AAC-LC, HE-AAC, and HE-AACv2. For more information, see Audio Codec Profiles at Request Body (p. 277). • **Preset and pipeline pagination:** You can now view more than the first 50 presets and pipelines with the list operation.	December 18th, 2013
New Features	With this release, Elastic Transcoder adds support for: • **Clip Generation:** You can use Elastic Transcoder to generate partial excerpts of content, or "clips," from your source media. This feature lets you transcode only a portion of your source media. For more information, see Composition at Create Job (p. 181).	November 8th, 2013

Change	Description	Release Date
New Features	With this release, Elastic Transcoder adds support for: • **Audio only transcoding:** You can now configure Elastic Transcoder to transcoding files that contain audio only content, strip out the video portion of a file that contains both audio and video, and create an audio only track for an HLS file set. For more information, see Container at General Settings (p. 41). • **Album Art:** When you configure Elastic Transcoder to use mp3 or mp4 containers, you can now specify a file in Amazon S3 to use for Album Art. For more information, see Album Art at Output Details, Part 1 (p. 23). • **Audio Metadata:** You can now pass through audio related metadata for audio files, such as artist, title, and genre.	October 9th, 2013
New Features	With this release, Elastic Transcoder adds support for: • **Maximum bit rate:** When you configure Elastic Transcoder to automatically detect the bit rate of the input video and use that value as the bit rate for the output video, you can now specify a maximum bit rate for the output video. For more information, see Maximum Bit Rate at Video Settings (p. 42). • **Maximum frame rate:** When you configure Elastic Transcoder to automatically detect the frame rate of the input video and use that value as the frame rate for the output video, you can now specify a maximum frame rate for the output video. For more information, see Maximum Frame Rate at Video Settings (p. 42). • **Watermarks:** You can now overlay up to four static images (watermarks) on every frame of an output video, and specify the size, position, scale, and opacity of each image. For more information, see Watermarks (p. 69).	July 8, 2013
New Features	With this release, Elastic Transcoder adds support for: • Apple HLS-compliant, pre-segmented files and playlists. • Automatic video bit rate optimization. • More control over the resolution and aspect ratio of output videos and thumbnails. • MPEG-TS as a container format for output. • Multiple outputs for the same job. • Specifying Amazon S3 permissions on the output videos and thumbnails, which simplifies using Amazon CloudFront to distribute your content. • Specifying separate Amazon S3 buckets for output videos and thumbnails. • Specifying the Amazon S3 storage class. • Vorbis as an audio codec for output. • VP8 as a video codec for output. • WebM as a container format for output.	May 16, 2013

Change	Description	Release Date
Initial Release	This is the first release of the *Amazon Elastic Transcoder Developer Guide*.	January 28, 2013

AWS Glossary

Blank

placeholder This page redirects to the AWS Glossary in the *AWS General Reference*.