



デベロッパーガイド

# Amazon Elastic Transcoder



# Amazon Elastic Transcoder: デベロッパーガイド

# Table of Contents

Amazon Elastic Transcoder とは .....	1
Elastic Transcoder へのアクセス .....	2
のリージョンとエンドポイント .....	3
Elastic Transcoder リソースのリージョンの選択 .....	3
パイプライン、ジョブ、プリセットの制限 .....	3
概要 .....	5
バケットの作成 .....	5
パイプラインの作成 .....	6
プリセットを作成する .....	8
ジョブを作成する .....	8
進捗状況をモニタリングする .....	9
パイプラインの使用 .....	10
パイプラインの作成 .....	10
パイプラインを一時停止し、再アクティブ化する .....	11
パイプラインの通知を更新する .....	12
パイプラインの一覧表示と確認を行う .....	12
パイプラインを削除する .....	13
パイプラインの設定 .....	14
リージョン .....	14
全般設定 .....	14
トランスコード済みのファイルやプレイリスト用に Amazon S3 Amazon S3 バケットを設定する .....	15
サムネイルに関する Amazon S3 バケットの設定 .....	17
通知 .....	20
ジョブの使用 .....	22
ジョブの作成 .....	22
リストの一覧表示と確認 .....	23
ジョブのキャンセル .....	25
ジョブの設定 .....	26
リージョン .....	27
全般設定 .....	27
入力詳細: パート 1 .....	28
入力詳細: パート 2、クリップとキャプションの設定 .....	30
出力詳細: パート 1 .....	34

出力詳細: パート 2 .....	36
出力詳細: パート 3、キャプションの設定 .....	38
(オプション) Output Encryption .....	40
( 動画/サムネイルのみ ) Watermarks .....	43
(FLAC/MP3/MP4 のみ) アルバムアート .....	45
(オプション) ユーザーメタデータ .....	50
(フラグメント化された MP4/MPEG-TS 出力のみ) Playlist .....	51
(フラグメント化された MP4/MPEG-TS 出力のみ、オプション) HLS Content Protection .....	53
( HLSv3 およびスムーズプレイリストのみ、オプション ) デジタル著作権管理 .....	56
プリセットの使用 .....	60
プリセットを作成する .....	60
プリセットを一覧表示する/プリセットの設定を表示する .....	61
プリセットの変更 .....	62
プリセットを削除する .....	62
プリセットの設定 .....	63
全般設定 .....	63
ビデオ設定 .....	65
ウォーターマークの設定 .....	79
オーディオ設定 .....	85
サムネイルの設定 .....	91
Amazon S3 バケットの整理 .....	95
コンテンツの保護 .....	97
アクセス権限の制御 .....	97
Elastic Transcoder に対するアクセス権限の制御 .....	98
パイプラインのサービスロール .....	105
データ暗号化のオプション .....	108
暗号化オプション .....	108
KMS の使用 .....	110
HLS のコンテンツ保護 .....	112
DRM .....	116
通知 .....	121
クリップの結合 .....	124
字幕 .....	125
ウォーターマーク .....	127
デジタル音声 .....	128
システムプリセット .....	130

サイズ設定のポリシーとアスペクト比 .....	136
サムネイルのアスペクト比 .....	136
Fit .....	137
Fill .....	138
[Stretch] (ストレッチ) .....	140
Keep .....	142
フィットするまで縮小する .....	144
圧縮して充填する .....	146
モニタリング .....	149
モニタリングツール .....	150
自動化ツール .....	150
手動ツール .....	150
Amazon CloudWatch でのモニタリング .....	151
メトリクスとディメンション .....	151
メトリクスの使用 .....	154
アラームの作成 .....	155
を使用した Elastic Transcoder API コールのログ記録 AWS CloudTrail .....	158
CloudTrail の Elastic Transcoder 情報 .....	158
Elastic Transcoder ログファイルエントリの理解 .....	159
AWS SDK サンプルコード .....	164
AWS SDK サンプルコードの背景情報 .....	165
HTTP Live Streaming ( HLS ) .....	165
通知 .....	165
環境をセットアップする .....	166
Java .....	166
Ruby .....	171
PHP .....	175
Python .....	179
Java サンプル .....	183
HLS サンプル .....	183
通知サンプル .....	184
Ruby サンプル .....	186
HLS サンプル .....	186
通知サンプル .....	187
PHP サンプル .....	189
HLS サンプル .....	189

通知サンプル .....	190
Python サンプル .....	192
HLS サンプル .....	192
通知サンプル .....	193
API リファレンス .....	196
AWS SDK の使用 .....	196
HTTP リクエストの実行 .....	196
HTTP ヘッダーの内容 .....	197
HTTP リクエストボディ .....	198
HTTP レスポンス .....	199
リクエストへの署名 .....	200
エラー処理 .....	202
API のエラーコード ( クライアントエラーとサーバーエラー ) .....	203
ジョブ処理中のエラー .....	209
エラーの捕捉 .....	216
エラーの再試行とエクスポネンシャルバックオフ .....	217
パイプラインのオペレーション .....	218
パイプラインを作成する .....	219
パイプラインを一覧表示する .....	228
パイプラインを読み取る .....	235
パイプラインを更新する .....	242
パイプラインのステータスを更新する .....	252
パイプラインの通知を更新する .....	256
パイプラインを削除する .....	260
ロールのテスト .....	263
ジョブのオペレーション .....	267
ジョブを作成する .....	267
パイプラインに基づくジョブのリスト .....	283
ステータスに基づくジョブのリスト .....	297
ジョブの読み取り .....	311
ジョブをキャンセルする .....	323
プリセットのオプション .....	326
プリセットを作成する .....	327
プリセットのリスト .....	336
プリセットを読み取る .....	342
プリセットを削除する .....	349

---

ドキュメント履歴 .....	352
AWS の用語集 .....	357
.....	ccclviii

# Amazon Elastic Transcoder とは

## トピック

- [Elastic Transcoder へのアクセス](#)
- [Elastic Transcoder リソースのリージョンの選択](#)
- [Elastic Transcoder パイプライン、ジョブ、プリセットの数の制限](#)

Amazon Elastic Transcoder では、Amazon Simple Storage Service (Amazon S3) に保存したメディアファイルをユーザーの再生デバイスに対応した形式のメディアファイルに変換できます。たとえば、サイズが大きい高品質のデジタルメディアファイルを、ユーザーがモバイルデバイス、タブレット、ウェブブラウザ、および接続したテレビで再生できる形式に変換できます。

Elastic Transcoder には以下の 4 つの構成要素があります。

- ジョブは、トランスコーディングの処理を実行します。各ジョブが 1 つのファイルを最大 30 の形式に変換できます。たとえば、1 つのメディアファイルを 6 つの異なる形式に変換する場合は、1 つのジョブを作成することで、6 つのすべての形式のファイルを作成できます。

ジョブの作成時に、トランスコードするファイルの名前や、Elastic Transcoder によってトランスコード済みファイルに付けられる名前など、いくつかの設定を指定します。トランスコードする形式ごとに、プリセットと呼ばれるテンプレート (以下を参照) を指定します。テンプレートには、トランスコードしたファイルに使用するオーディオと動画の設定が含まれます。

- パイプラインは、変換ジョブを管理するキューです。ジョブを作成するとき、ジョブを追加するパイプラインを指定します。Elastic Transcoder によってパイプライン内のジョブの処理がスタートされます。処理は、キューにジョブを追加した順序で行われます。複数の形式にトランスコードするようにジョブを設定した場合は、ジョブで形式を指定した順に Elastic Transcoder によって各形式のファイルが作成されます。

通常の設定では、標準的な優先度のジョブ用と、高い優先度のジョブ用に 2 つのパイプラインを作成します。ほとんどのジョブは優先度が標準であるパイプラインに入ります。優先度が高いパイプラインは、ファイルを直ちに変換する必要がある場合にのみ使用します。

新しいジョブの作成時にパイプラインにジョブがすでにある場合、Elastic Transcoder によって最新のジョブがキューに格納され、リソースがそのパイプラインに使用可能になるとすぐにそのジョブが開始されます。パイプラインでそのリソースがすべて使用中である場合は、現在処理中のい

れかのジョブズの処理が完了すると、Elastic Transcoder によってパイプライン内の次のジョブの処理が開始されます。

パイプラインでは同時に複数のジョブを処理できます。ただしジョブの実行に要する時間は、変換するファイルのサイズやジョブの仕様によって大きく異なります。したがってジョブは必ずしも、作成した順に完了するわけではありません。

パイプラインを一時停止することで、ジョブの処理を一時停止できます。

- プリセットは、メディアファイルに対してある形式から別の形式へのトランスコードを行うための設定の大半が含まれているテンプレートです。Elastic Transcoder には、一般的な形式 (iPod と iPhone の複数のバージョンなど) 用にデフォルトのプリセットがいくつか用意されています。デフォルトのプリセットに含まれていない形式用に独自のプリセットを作成することもできます。使用するプリセットはジョブの作成時に指定します。
- 通知は、ジョブのステータスを知らせる機能で、必要に応じて Elastic Transcoder と Amazon Simple Notification Service で設定できます。例えば、Elastic Transcoder によるジョブ処理の開始時や、Elastic Transcoder によるジョブの終了時を知らせるように設定したり、ジョブの処理中に Elastic Transcoder で警告またはエラーが発生したときに知らせるように設定することができます。通知により、ジョブが終了したかどうかを調べるためのポーリングが不要になります。通知はパイプラインの作成時に設定します。

## Elastic Transcoder へのアクセス

Elastic Transcoder は、REST ベースのウェブサービスであり、HTTPS をプロトコルとして、JavaScript Object Notation (JSON) をメッセージ形式として使用します。アプリケーションコードから直接、Elastic Transcoder API にリクエストを行うことができます。この REST API を直接使用するときは、リクエストの署名と認証のためのコードを書く必要があります。API や、リクエストへの署名の詳細については、「[API リファレンス](#)」を参照してください。

Elastic Transcoder にはマネジメントコンソールもあります。コンソールでは、Elastic Transcoder API を使用して実行できるオペレーションと同じオペレーションをすべて実行できます。コンソールによりパイプライン、プリセット、ジョブを作成および管理する方法については、以下の該当するトピックを参照してください。

- [ジョブの使用](#)
- [パイプラインの使用](#)
- [プリセットの使用](#)

## のリージョンとエンドポイント

特定の AWS リージョンでパイプラインを作成します。Elastic Transcoder のリクエストを常にリージョン固有のエンドポイントに送信します。AWS でサポートされているリージョンのリストについては、Amazon Web Services 全般のリファレンスの「[リージョンとエンドポイント](#)」を参照してください。

## Elastic Transcoder リソースのリージョンの選択

パイプラインやジョブは、特定のリージョンと密接に関連付けられています。パイプラインやジョブを作成する場合、現在のリージョンで作成されます。新しいジョブの作成する場合は、現在のリージョンのパイプラインを指定する必要があります。

Elastic Transcoder リソースとは異なるリージョンの Amazon S3 バケットを指定することもできますが、AWS リージョン間でのファイル転送には追加料金が発生するためお勧めしません。

リージョン間の料金の詳細については、「[Amazon S3 料金表](#)」の「データ転送料金表」を参照してください。

## Elastic Transcoder パイプライン、ジョブ、プリセットの数の制限

Elastic Transcoder のパイプライン、ジョブ、プリセットには以下のような制限があります。

- パイプライン: リージョンごとに AWS アカウントあたり 4 パイプライン
- キュー内のジョブの最大数: パイプラインあたり 1,000,000 ジョブ
- 出力の最大数: ジョブあたり 30 出力
- パイプラインごとに同時に処理されるジョブの最大数: パイプラインあたり 100 ジョブ
- プリセット: AWS アカウントあたり 50 ユーザー定義プリセット (Elastic Transcoder の定義済みプリセットはこの制限数に含まれません)
- ジョブリクエストを送信できる最大レート:
  - Create Job: 継続的なレートでは、AWS アカウントごとに 1 秒あたり 2 個の Create Job リクエストを送信できます。瞬間的なバーストでは、1 秒あたり 100 個のリクエストに対応できません。
  - ジョブの読み取り: 継続的なレートでは、AWS アカウントごとに 1 秒あたり 4 個の Read Job リクエストを送信できます。瞬間的なバーストでは、1 秒あたり 50 個のリクエストに対応できません。

<https://console.aws.amazon.com/support/home#/case/create?issueType=service-limit-increase&limitType=service-code-elastic-transcoders> で制限数の引き上げをリクエストできます。

# Elastic Transcoder の使用開始

このトピックの例では、Amazon Elastic Transcoder を使用してメディアファイルに対してあるメディア形式から別の形式へのトランスコードを行う方法の概要を説明します。いくつかの基本的なステップを実行するだけで、Elastic Transcoder を使用してメディアファイルのトランスコードを開始することができます。最初のステップでは、AWS にサインアップします。その後で、Amazon S3 バケットを作成し、トランスコードを行うメディアファイルをアップロードします。次に、ジョブを処理するためのパイプラインを作成し、特定のファイルを特定の形式に変換するジョブを作成します。プリセット (テンプレート) が提供されていない形式に変換する場合は、ジョブを作成する前に、カスタムプリセットを作成できます。

## Note

ジョブ、パイプライン、プリセット (Elastic Transcoder の基本概念) の詳細が分からない場合は、簡単な概要を示した「[Amazon Elastic Transcoder とは](#)」を見直しておきましょう。

## トピック

- [Amazon S3 バケットを 1 つまたは 2 つ作成し、メディアファイルをアップロードする](#)
- [パイプラインの作成](#)
- [\(オプション\) プリセットを作成する](#)
- [ジョブを作成する](#)
- [ジョブの進捗状況をモニタリングする](#)

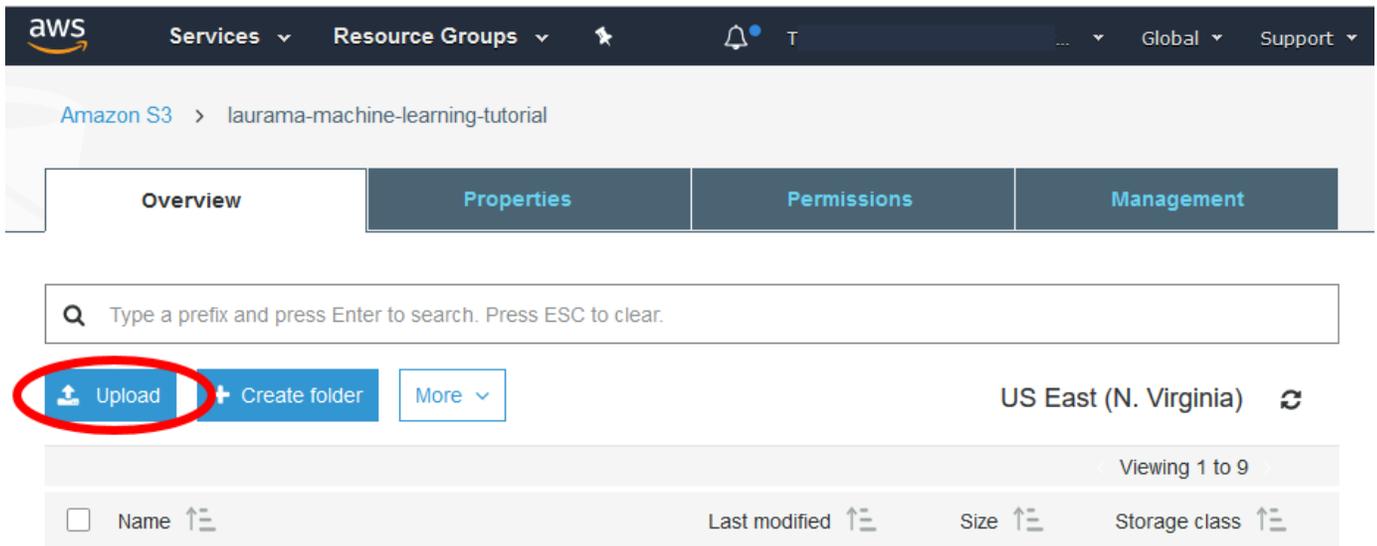
## Amazon S3 バケットを 1 つまたは 2 つ作成し、メディアファイルをアップロードする

トランスコード対象ファイル用の Amazon S3 バケット (入力バケット) と、トランスコード済みファイル用の別のバケット (出力バケット) を作成します。入力バケットと出力バケット用に、同じバケットを使用することもできます。

### Amazon S3 バケットを作成し、メディアファイルをアップロードする方法

1. にサインイン AWS マネジメントコンソールし、<https://console.aws.amazon.com/s3/> で Amazon S3 コンソールを開きます。

2. Amazon S3 コンソールで、[Create Bucket] (バケットの作成) をクリックします。
3. [Create Bucket] (バケットの作成) ダイアログボックスに、バケット名を入力します。入力バケットと出力バケットを異なるバケットとして作成する場合は、それぞれのバケットに適切な名前を指定してください。
4. バケットのリージョンを選択します。デフォルトでは、Amazon S3 により米国スタンダードリージョンにバケットが作成されます。レイテンシーを最適化するため、コストを最小化するため、また規制要件に対応するために、最寄りのリージョンを選択してください。このリージョンは、Elastic Transcoder でトランスコードを実行するリージョンでもあります。
5. [Create (作成)] をクリックします。
6. トランスコード対象ファイルと Elastic Transcoder でトランスコード処理されたファイルに対して個別のバケットを作成する場合は、ステップ 2~5 を繰り返します。
7. [Buckets (バケット)] ペインで、入力バケットの名前をクリックします。
8. [Actions (アクション)] をクリックし、[Upload (アップロード)] をクリックします。
9. [Upload - Select Files (アップロード - ファイルの選択)] ページで [[Add Files (ファイルの追加)] をクリックし、変換するメディアファイルをアップロードします。



10. [Start Upload (アップロードを開始)] をクリックします。

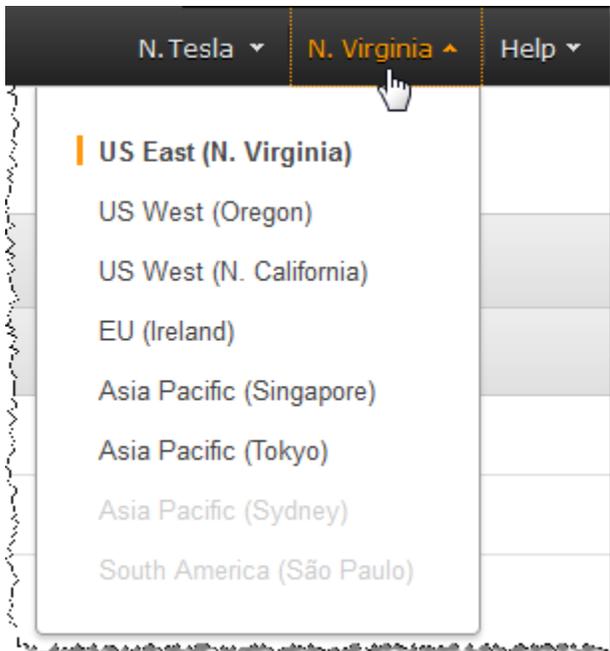
## パイプラインの作成

パイプラインは、ファイルの変換ジョブを管理します。通常、2 つ以上のパイプラインを作成します。たとえば、優先度が標準であるジョブを管理するためのパイプラインと、優先度が高いジョブを

管理するためのパイプラインです。この例では、1つのパイプラインを作成します。パイプラインの詳細については、「[パイプラインの使用](#)」を参照してください。

## Elastic Transcoder コンソールを使用してパイプラインを作成する方法

1. <https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. Elastic Transcoder コンソールのナビゲーションバーで、パイプラインを作成するリージョンを選択します。



3. コンソールの左側のペインで、[Pipelines (パイプライン)] をクリックします。
4. [Pipelines (パイプライン)] ページで、[Create New Pipeline (新しいパイプラインの作成)] をクリックします。
5. 適切な値を入力します。各フィールドの詳細については「[Elastic Transcoder パイプラインの作成時に指定する設定](#)」を参照してください。

### Note

ジョブの進捗状況をモニタリングする場合は、Elastic Transcoder でジョブの処理が開始されたとき、および Elastic Transcoder でジョブが完了したときに通知する Amazon Simple Notification Service (Amazon SNS) トピックを指定します。

6. [Create Pipeline (パイプラインの作成)] をクリックします。

## (オプション) プリセットを作成する

プリセットは、Elastic Transcoder のトランスコード処理で適用する設定 (トランスコード済みファイルのコーデックや解像度など) が含まれているテンプレートです。ジョブの作成時に、使用するプリセットを指定します。多くの一般的なデバイスで再生可能なメディアファイルを作成するシステムプリセットと、特定のデバイスを対象としたシステムプリセットが用意されています。互換性を重視する場合は、様々なデバイスで再生可能な出力を作成する「preset」 (汎用プリセット) を選択します。品質とファイルサイズを最適化する場合は、特定のデバイスまたはデバイスのクラスに応じた出力を作成するプリセットを選択します。システムプリセットの最新のリストについては、Elastic Transcoder の詳細ページにあるプリセットのリストを参照してください。

必要な設定が含まれているシステムプリセットがない場合は、独自のプリセットを作成できます。詳細については、「[Elastic Transcoder でプリセットを作成する](#)」を参照してください。

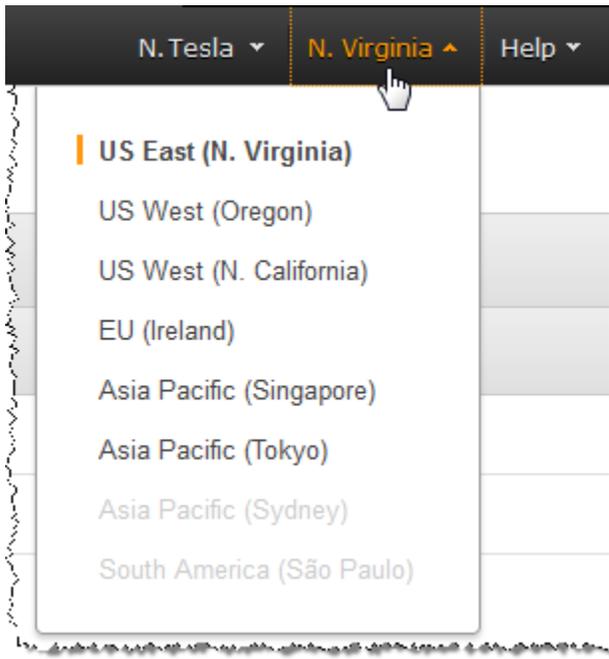
## ジョブを作成する

ジョブは変換の実行を実行します。トランスコード対象ファイル (入力ファイル) の名前、Elastic Transcoder によるトランスコード済みファイルに付ける名前、Elastic Transcoder で使用するプリセット、およびその他の設定は、ユーザーが指定します。Elastic Transcoder では、ユーザーがパイプラインで指定した Amazon S3 入力バケットから入力ファイルが取得され、そのファイルのトランスコードが行われて、ユーザーがパイプラインで指定した Amazon S3 出力バケットにトランスコード済みファイルが保存されます。

ジョブの詳細については、「[ジョブの使用](#)」を参照してください。

Elastic Transcoder コンソールを使用してジョブを作成する方法

1. <https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. Elastic Transcoder コンソールのナビゲーションバーで、ジョブを作成するリージョンを選択します。



3. コンソールの左側のペインで、[Pipelines] (パイプライン) をクリックします。(ファイルのトランスコードに使用するジョブをパイプライン (キュー) に作成します。)
4. [Pipelines] (パイプライン) ページで、[Create New Job] (新しいジョブの作成) をクリックします。
5. 適切な値を入力します。各フィールドの詳細については「[Elastic Transcoder ジョブの作成時に指定する設定](#)」を参照してください。
6. [Create Job](新しいジョブの作成) をクリックします。

## ジョブの進捗状況をモニタリングする

[パイプラインの作成](#) でパイプラインを作成した場合は通知を設定するオプションがあり、そのため、Elastic Transcoder によるジョブの処理の開始時と終了時に、Elastic Transcoder から Amazon Simple Notification Service(Amazon SNS) トピックにメッセージが送信されます。通知を設定した場合と、該当する Amazon SNS トピックにサブスクライブした場合には、ジョブの進捗状況をモニタリングできます。

# パイプラインの使用

## トピック

- [Elastic Transcoder でパイプラインを作成する](#)
- [Elastic Transcoder でパイプラインを一時停止し、再アクティブ化する](#)
- [Elastic Transcoder でパイプライン通知を更新する](#)
- [Elastic Transcoder でパイプラインの一覧表示と確認を行う](#)
- [Elastic Transcoder パイプラインを削除する](#)
- [Elastic Transcoder パイプラインの作成時に指定する設定](#)

パイプラインは、変換ジョブを管理するキューです。ジョブを作成するとき、ジョブを追加するパイプラインを指定します。Elastic Transcoder によってパイプライン内のジョブの処理がスタートされます。処理は、キューにジョブを追加した順序で行われます。

一般的な設定の 1 つは、2 つのパイプラインを作成することです。1 つはスタンダード優先度のジョブ用、もう 1 つは優先度の高いジョブ用です。ほとんどのジョブは優先度が標準であるパイプラインに入ります。優先度の高いパイプラインは、ファイルを直ちに変換する必要がある場合にのみ使用します。

ジョブの作成時に他のジョブがパイプライン内にある場合、Elastic Transcoder では、リソースが利用可能になると、新しいジョブの処理をスタートします。パイプラインでは同時に複数のジョブを処理できます。ただしジョブの実行に要する時間は、変換するファイルのサイズやジョブの仕様によって大きく異なります。このため、ジョブは、必ずしも作成された順序で実行されるわけではありません。

一時的にパイプラインを停止できます。これによりジョブの処理も停止されます。これは、1 つ以上のジョブをキャンセルする場合に役立ちます。この操作は、Elastic Transcoder でジョブの処理がスタートされるまで行うことができます。

## Elastic Transcoder でパイプラインを作成する

パイプラインは、AWS マネジメントコンソール または Elastic Transcoder Create Pipeline API アクションを使用して作成できます。次の手順では、コンソールを使用してパイプラインを作成する方法を説明します。API を使用してパイプラインを作成する方法については、「[パイプラインを作成する](#)」を参照してください。

## Elastic Transcoder コンソールを使用してパイプラインを作成する方法

1. にサインイン AWS マネジメントコンソールし、<https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. Elastic Transcoder コンソールのナビゲーションバーで、パイプラインを作成するリージョンを選択します。このリージョンは、ジョブが作成される場所です。
3. コンソールのナビゲーションペイン (左側) で、[Pipelines] (パイプライン) をクリックします。
4. [Pipelines] (パイプライン) ページで、[Create New Pipeline] (新しいパイプラインの作成) をクリックします。
5. 適切な値を入力します。各フィールドの詳細については「[Elastic Transcoder パイプラインの作成時に指定する設定](#)」を参照してください。
6. [Create Pipeline] (パイプラインの作成) をクリックします。

## Elastic Transcoder でパイプラインを一時停止し、再アクティブ化する

ジョブをキャンセルする場合は、Elastic Transcoder でジョブの処理が開始されないように、まず、対応するパイプラインを一時停止しておくことをお勧めします。ジョブのステータスが [Submitted] から [Progressing] に変わると、ジョブをキャンセルすることはできません。

次の手順では、コンソールを使用してパイプラインを一時停止し再アクティブ化する方法を説明します。API を使用してパイプラインを一時停止し再アクティブ化する方法については、「[パイプラインのステータスを更新する](#)」参照してください。

### Elastic Transcoder コンソールを使用してパイプライン一時停止し再アクティブ化する方法

1. にサインイン AWS マネジメントコンソールし、<https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. Elastic Transcoder コンソールのナビゲーションバーで、パイプラインを一時停止または再アクティブ化するリージョンを選択します。
3. 左側のナビゲーションペインで、[Pipelines] (パイプライン) をクリックします。
4. 一時停止または再アクティブ化するパイプラインの横にあるチェックボックスをオンにします。
5. 必要に応じて、[Pause] (一時停止) または [Activate] (アクティブにする) をクリックします。

## Elastic Transcoder でパイプライン通知を更新する

パイプラインを作成するとき、必要に応じて Elastic Transcoder を設定して、ジョブのステータスが変った場合に (Elastic Transcoder でジョブの処理をスタートまたは終了したときや、ジョブの処理中に Elastic Transcoder で警告やエラーの状態が発生したときなど)、Amazon Simple Notification Service (Amazon SNS) トピックへメッセージを送信することができます。Elastic Transcoder でメッセージを送信するかどうかは任意に選択できます。メッセージを送信する場合は、メッセージの送信先となる SNS トピックを変更できます。

Amazon SNS にはさまざまな通知オプションが用意されており、Amazon Simple Queue Service (Amazon SQS) キューに Amazon SNS メッセージを送信することもできます。詳細については、「[Amazon Simple Notification Service デベロッパーガイド](#)」を参照してください。

次の手順では、コンソールを使用して通知を更新する方法を説明します。API を使用して通知を更新する方法については、「[パイプラインの通知を更新する](#)」を参照してください。

Elastic Transcoder コンソールを使用してパイプラインの通知を更新するには

1. にサインイン AWS マネジメントコンソールし、<https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. Elastic Transcoder コンソールのナビゲーションバーで、パイプラインを一時停止または再アクティブ化するリージョンを選択します。
3. 左側のナビゲーションペインで、[Pipelines] (パイプライン) をクリックします。
4. 通知を変更するパイプラインの横にあるチェックボックスをオンにします。
5. [Edit (編集)] をクリックします。
6. 値を適切に変更します。詳細については、「[Elastic Transcoder パイプラインの作成時に指定する設定](#)」を参照してください。
7. [保存] をクリックして変更を保存します。

## Elastic Transcoder でパイプラインの一覧表示と確認を行う

現在の AWS アカウントに関連付けられたパイプラインを一覧表示したり、指定したパイプラインの設定を確認したりすることができます。次の手順では、コンソールを使用してパイプラインを一覧表示する方法とパイプラインの設定を確認する方法を説明します。

API の使用方法については、次を参照してください。

- 現在の AWS アカウントのパイプラインを一覧表示するには、「[パイプラインを一覧表示する](#)」を参照してください。
- 指定したパイプラインの設定を取得するには、「[パイプラインを読み取る](#)」を参照してください。

Elastic Transcoder コンソールを使用してパイプラインの一覧表示やパイプラインの設定の確認を行う方法

1. にサインイン AWS マネジメントコンソールし、<https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. Elastic Transcoder コンソールのナビゲーションバーで、ジョブを一覧表示するリージョンを選択します。
3. コンソールのナビゲーションペイン (左側) で、[Pipelines] (パイプライン) をクリックします。右側のペインには、現在のアカウントに関連付けられたパイプラインが一覧表示されます。
4. パイプラインの詳細情報を表示するには、パイプラインの横にある  アイコンをクリックします。

## Elastic Transcoder パイプラインを削除する

パイプラインを削除するには、を使用する AWS マネジメントコンソール か、Elastic Transcoder Delete Pipeline API を使用します。次の手順では、コンソールを使用してパイプラインを削除する方法を説明します。API を使用してパイプラインを削除する方法については、「[パイプラインを削除する](#)」を参照してください。

### Note

未処理のジョブを含むパイプラインは削除できません。

Elastic Transcoder コンソールを使用してパイプラインを削除する方法

1. にサインイン AWS マネジメントコンソールし、<https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. Elastic Transcoder コンソールのナビゲーションバーで、削除するパイプラインが含まれているリージョンを選択します。

3. コンソールのナビゲーションペイン (左側) で、[Pipelines] (パイプライン) をクリックします。
4. 削除するパイプラインのチェックボックスをオンにします。
5. [Remove] をクリックします。

## Elastic Transcoder パイプラインの作成時に指定する設定

Elastic Transcoder コンソールを使用してパイプラインを作成するとき、以下の値を指定できます。

トピック

- [リージョン](#)
- [全般設定](#)
- [トランスコード済みのファイルやプレイリスト用に Amazon S3 Amazon S3 バケットを設定する](#)
- [サムネイルに関する Amazon S3 バケットの設定](#)
- [通知](#)

### リージョン

Elastic Transcoder では、現在使用しているリージョンでパイプラインが作成されます。ジョブを実行する予定のリージョンを選択します。

### 全般設定

パイプライン名

パイプラインの名前。この名前は AWS アカウント内で一意の名前にすることをお勧めしますが、一意にすることは強制ではありません。パイプラインの名前の最大長は 40 文字です。

入力バケット

変換するメディアファイルや、ウォーターマークに使用するグラフィックファイル (グラフィックファイルがある場合) を保存した Amazon S3 バケット。

IAM ロール

Elastic Transcoder でこのパイプラインのジョブのトランスコードを実行する際に使用するロールの IAM Amazon リソースネーム (ARN)。

## AWS KMS キー ARN

このパイプラインで使用する AWS Key Management Service(AWS KMS) キー。

s3 または s3-aws-kms のいずれかを Encryption:Mode として使用する場合は、デフォルトのキー (AWS-KMS キー) が自動的に作成されるため、キーにジョブを提供する必要はありません。AWS-KMS キーを提供する必要があるのは、デフォルト以外の AWS-KMS キーを使用する場合か、aes-pkcs7、aes-ctr、aes-gcm の Encryption:Mode を使用する場合があります。

## トランスコード済みのファイルやプレイリスト用に Amazon S3 Amazon S3 バケットを設定する

このセクションの設定を使用して、Amazon S3 バケット (パイプラインに送信されたジョブのトランスコード済みファイルやプレイリストが Elastic Transcoder により保存されます)、ファイルに割り当てる Amazon S3 ストレージクラス、ファイルへのアクセスを許可するユーザー、およびユーザーに付与するアクセス権のタイプを指定します。

### バケット

パイプラインに送信したジョブのトランスコード済みファイルやプレイリスト (該当する場合) が Elastic Transcoder により保存される Amazon S3 バケット。

### ストレージクラス

Elastic Transcoder により Amazon S3 バケットに保存されるトランスコード済みのファイルやプレイリストに割り当てられる、Amazon S3 ストレージクラス ([スタンダード] または [低冗長化])。詳細については、「Amazon Simple Storage Service ユーザーガイド」の「[低冗長化ストレージ](#)」を参照してください。

### 被付与者タイプ

変換結果ファイルやプレイリストへのアクセスを許可するユーザーやグループをどのように識別するかを指定します。被付与者のタイプを選択するとき、[\[Grantee Type\]](#) フィールドが表示されます。

- [正規]: [被付与者] の値は、AWS アカウントの正規ユーザー ID または Amazon CloudFront デイストリビューションのオリジンアクセスアイデンティティのいずれかになります。詳細については、「Amazon Simple Storage Service ユーザーガイド」の「[アクセスコントロールリスト \(ACL\) の概要](#)」を参照してください。CloudFront オリジンアクセスアイデンティティを使用して、Amazon S3 URL の代わりに CloudFront URL の使用を要求する方法の詳細については、

「Amazon CloudFront デベロッパーガイド」の「[オリジンアクセスアイデンティティを使用した Amazon S3 コンテンツへのアクセス制限](#)」を参照してください。

**⚠ Important**

正規ユーザー ID は AWS アカウント番号と同じではありません。

- [Email]: [Grantee] の値は、AWS アカウントの登録 E メールアドレスになります。
- [グループ]: [被付与者] の値は、事前定義されている Amazon S3 グループである [AllUsers]、[AuthenticatedUsers]、[LogDelivery] のいずれかになります。

### 被付与者

変換結果ファイルやプレイリストへのアクセスを許可する AWS のユーザーまたはグループ。ユーザーまたはグループを識別するには、AWS アカウントの正規ユーザー ID、CloudFront デイストリビューションのオリジンアクセスアイデンティティ、AWS アカウントの登録 E メールアドレス、または事前定義されている Amazon S3 グループを指定できます。詳細については、「[Grantee Type](#)」を参照してください。

### アクセス

[Grantee](#) で指定した AWS ユーザーに付与するアクセス許可。アクセス許可は、Elastic Transcoder によって Amazon S3 バケットに追加されたトランスコード済みファイルやプレイリストに関して付与されます。有効な値を次に示します。

- [Open/Download] (開く/ダウンロード): 被付与者は、Elastic Transcoder によって Amazon S3 バケットに追加されたトランスコード済みファイルやプレイリストのオブジェクトとメタデータを読み取ることができます。
- [View Permissions] (アクセス許可の表示): 被付与者は、Elastic Transcoder によって Amazon S3 バケットに追加されたトランスコード済みファイルやプレイリストのオブジェクト ACL を読み取ることができます。
- [Edit Permissions] (アクセス許可の編集): 被付与者は、Elastic Transcoder によって Amazon S3 バケットに追加されたトランスコード済みファイルやプレイリストの ACL を書き込むことができます。
- [Full Control] (完全制御): 被付与者は、Elastic Transcoder によって Amazon S3 バケットに追加されたトランスコード済みファイルやプレイリストのオブジェクトを読み取るアクセス許可、およびこれらのトランスコード済みファイルやプレイリストの ACL の表示と編集を行うアクセス許可を保持しています。

### Configure Amazon S3 Bucket for Transcoded Files and Playlists

**Bucket**  ⓘ

**Storage Class**  ⓘ

[+ Add Permission](#)

If output permissions are specified for a pipeline, the creator of the job is no longer assigned FullControl permissions by default. However, it is possible to assign FullControl to the job creator by adding that user's canonical ID to the permissions list.

**Grantee Type**  ⓘ

**Grantee**  ⓘ

**Access**

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ⓘ
Open/Download	View	Edit	Full	
Permission	Permission	Control		

[- Remove](#)

## サムネイルに関する Amazon S3 バケットの設定

このセクションの設定を使用して、Amazon S3 バケット (パイプラインに送信されたジョブのサムネイルが Elastic Transcoder により保存されます)、サムネイルに割り当てる Amazon S3 ストレージクラス、サムネイルへのアクセスを許可するユーザー、およびユーザーに付与するアクセス権のタイプを指定します。

### バケット

パイプラインに送信したジョブのサムネイルが Elastic Transcoder により保存される Amazon S3 バケット。

## ストレージクラス

Elastic Transcoder により Amazon S3 バケットに保存されるサムネイルに割り当てられる、Amazon S3 ストレージクラス ([スタンダード] または [低冗長化])。詳細については、「Amazon Simple Storage Service ユーザーガイド」の「[低冗長化ストレージ](#)」を参照してください。

## 被付与者タイプ

サムネイルへのアクセスを許可するユーザーやグループをどのように識別するかを指定します。被付与者のタイプを選択するとき、[\[Grantee Type\]](#) フィールドが表示されます。

- [正規]: [被付与者] の値は、AWS アカウントの正規ユーザー ID または Amazon CloudFront デイストリビューションのオリジンアクセスアイデンティティのいずれかになります。詳細については、「Amazon Simple Storage Service ユーザーガイド」の「[アクセスコントロールリスト \(ACL\) の概要](#)」を参照してください。CloudFront オリジンアクセスアイデンティティを使用して、Amazon S3 URL の代わりに CloudFront URL の使用を要求する方法の詳細については、「Amazon CloudFront デベロッパーガイド」の「[オリジンアクセスアイデンティティを使用した Amazon S3 コンテンツへのアクセス制限](#)」を参照してください。

### Important

正規ユーザー ID は AWS アカウント番号と同じではありません。

- [Email]: [Grantee] の値は、AWS アカウントの登録 E メールアドレスになります。
- [グループ]: [被付与者] の値は、事前定義されている Amazon S3 グループである [AllUsers]、[AuthenticatedUsers]、[LogDelivery] のいずれかになります。

## 被付与者

サムネイルへのアクセスを許可する AWS のユーザーまたはグループ。ユーザーまたはグループを識別するには、AWS アカウントの正規ユーザー ID、CloudFront デイストリビューションのオリジンアクセスアイデンティティ、AWS アカウントの登録 E メールアドレス、または事前定義されている Amazon S3 グループを指定できます。詳細については、「[Grantee Type](#)」を参照してください。

## アクセス

[\[Grantee\]](#) で指定した AWS ユーザーに付与するアクセス許可。アクセス許可は、Elastic Transcoder によって Amazon S3 バケットに追加されたサムネイルに関して付与されます。有効な値を次に示します。

- [Open/Download] (開く/ダウンロード): 被付与者は、Elastic Transcoder によって Amazon S3 バケットに追加されたサムネイルのオブジェクトとメタデータを読み取ることができます。
- [View Permissions] (アクセス許可の表示): 被付与者は、Elastic Transcoder によって Amazon S3 バケットに追加されたサムネイルのオブジェクト ACL を読み取ることができます。
- [Edit Permissions] (アクセス許可の編集): 被付与者は、Elastic Transcoder によって Amazon S3 バケットに追加されたサムネイルの ACL を書き込むことができます。
- [Full Control] (完全制御): 被付与者は、Elastic Transcoder によって Amazon S3 バケットに追加されたサムネイルのオブジェクトを読み取るアクセス許可、およびこれらのサムネイルの ACL の表示と編集を行うアクセス許可を保持しています。

### Configure Amazon S3 Bucket for Thumbnails

**Bucket**  ⓘ

**Storage Class**  ⓘ

[+ Add Permission](#)

If output permissions are specified for a pipeline, the creator of the job is no longer assigned FullControl permissions by default. However, it is possible to assign FullControl to the job creator by adding that user's canonical ID to the permissions list.

**Grantee Type**  ⓘ

**Grantee**  ⓘ

**Access**   View  Edit  Full ⓘ  
Open/Download Permission Permission Control

[- Remove](#)

## 通知

このセクションの設定を使用して、ジョブのステータスが変化したときに通知するように Elastic Transcoder を設定します。

### 進行中イベント

Elastic Transcoder でジョブの処理が開始されたときに通知する Amazon Simple Notification Service (Amazon SNS) トピック。

#### Important

通知を受信するには、Amazon SNS コンソールで新しいトピックにサブスクライブする必要があります。

Amazon SNS にはさまざまな通知オプションが用意されており、Amazon Simple Queue Service (Amazon SQS) キューに Amazon SNS メッセージを送信することもできます。詳細については、「[Amazon Simple Notification Service デベロッパーガイド](#)」を参照してください。

### 完了イベント

Elastic Transcoder でジョブの処理が終了したときに通知する Amazon SNS のトピック。

### 警告イベント

Elastic Transcoder で警告状態が発生したときに通知する Amazon SNS トピック。

### エラーイベント

Elastic Transcoder でエラー状態が発生したときに通知する Amazon SNS トピック。

▼ Notifications (Optional)

**On Progressing Event**

- No Notifications i
- Use an existing SNS topic
- Create a New SNS Topic

**On Warning Event**

- No Notifications i
- Use an existing SNS topic
- Create a New SNS Topic

**On Completion Event**

- No Notifications i
- Use an existing SNS topic
- Create a New SNS Topic

**On Error Event**

- No Notifications i
- Use an existing SNS topic
- Create a New SNS Topic

Cancel

Create Pipeline

# ジョブの使用

## トピック

- [Elastic Transcoder でジョブを作成する](#)
- [Elastic Transcoder でのジョブの一覧と設定の表示](#)
- [Elastic Transcoder ジョブのキャンセル](#)
- [Elastic Transcoder ジョブの作成時に指定する設定](#)

ジョブは、メディアファイルをある形式から別の形式に変換する処理を実行します。ジョブを作成するとき、Elastic Transcoder で変換を実行する際に必要となる情報を指定します。それらの情報には、変換するファイル、トランスコード済みファイルの名前、使用するプリセット (プリセットとは Elastic Transcoder で適用する設定を含んだテンプレートです) などがあります。

## Elastic Transcoder でジョブを作成する

ジョブは、AWS マネジメントコンソール または Elastic Transcoder ジョブの作成 API アクションを使用して作成できます。次の手順では、コンソールを使用してジョブを作成する方法を説明します。API を使用してジョブを作成する方法については、「[ジョブを作成する](#)」を参照してください。

ジョブのステータスが変化したとき (Elastic Transcoder でジョブの処理をスタートまたは終了したときや、警告またはエラーの条件が発生したときなど) に通知するように、Elastic Transcoder を設定できます。詳細については、「[Elastic Transcoder でパイプラインを作成する](#)」を参照してください。

ジョブを作成した後でジョブを更新することはできません。ジョブの設定を変更する必要がある場合は、そのジョブをキャンセルし、キャンセルしたジョブに基づいて新しいジョブを作成し、該当する値を更新して、新しいジョブを完成します。

### Note

ジョブを作成する前に、ジョブを管理するパイプライン (キュー) を作成する必要があります。パイプラインの作成に関する詳細については、「[Elastic Transcoder でパイプラインを作成する](#)」を参照してください。また、Elastic Transcoder のデフォルトのプリセットで提供される以外の設定を使用してファイルを変換する場合は、新しいプリセットを作成する

必要があります。プリセットの作成に関する詳細については、「[Elastic Transcoder でプリセットを作成する](#)」を参照してください。

## Elastic Transcoder コンソールを使用してジョブを作成する方法

1. にサインイン AWS マネジメントコンソールし、<https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. Elastic Transcoder コンソールのナビゲーションバーで、変換ジョブで使用するパイプラインに対応するリージョンを選択します。パイプラインは、パイプラインを作成したリージョンでのみ使用できます。
3. コンソールのナビゲーションペイン (左側) で、[Pipelines] (パイプライン) をクリックします。(ファイルのトランスコードに使用するジョブをパイプライン (キュー) に作成します。)
4. [Pipelines] (パイプライン) ページで、[Create New Job] (新しいジョブの作成) をクリックします。
5. 適切な値を入力します。各フィールドの詳細については「[Elastic Transcoder ジョブの作成時に指定する設定](#)」を参照してください。
6. [Create Job](新しいジョブの作成) をクリックします。ジョブは、作成するとすぐに開始されます。

### Note

ジョブが Access Denied エラーで失敗する場合は、Test Role API アクションを実行して、エラーの原因を判断することをお勧めします。詳細については、「[ロールのテスト](#)」を参照してください。

## Elastic Transcoder でのジョブの一覧と設定の表示

指定したパイプライン内のジョブや指定したステータスのジョブを一覧表示するには、Elastic Transcoder コンソールまたは適切な API アクションを使用します。また、個々のジョブの設定を確認することもできます。次の手順では、コンソールを使用してジョブを一覧表示する方法とジョブの設定を確認する方法を説明します。

**Note**

パイプラインに基づいてジョブの一覧を表示する場合、Elastic Transcoder では、そのパイプラインに対して過去 6 か月に作成したすべてのジョブが一覧表示されます。ステータスに基づいてジョブを一覧表示する場合、Elastic Transcoder では、過去 6 か月に作成したジョブのうち、現在は指定したステータスになっているすべてのジョブが一覧表示されます。

API の使用方法については、次を参照してください。

- 指定したパイプライン内のジョブを一覧表示するには、「[パイプラインに基づくジョブのリスト](#)」を参照してください。
- 指定したステータスになっているジョブを一覧表示するには、「[ステータスに基づくジョブのリスト](#)」を参照してください。
- 指定したジョブの設定を取得するには、「[ジョブの読み取り](#)」を参照してください。

**Note**

ジョブの出力を複数指定した場合 (例えば、1 つは Kindle Fire 用の出力、もう 1 つは Apple iPhone 4s 用の出力など)、ジョブを一覧表示するには Elastic Transcoder API を使用する必要があります。

Elastic Transcoder コンソールを使用してジョブの一覧表示やジョブの設定のチェックを行う方法

1. <https://console.aws.amazon.com/elasticbeanstalk/> で AWS マネジメントコンソールにサインインして Elastic Transcoder コンソールを開きます。
2. Elastic Transcoder コンソールのナビゲーションバーで、ジョブを一覧表示するリージョンを選択します。
3. コンソールのナビゲーションペイン ( 左側 ) で、[Jobs] (ジョブ) をクリックします
4. [Jobs] (ジョブ) ページで、適切な値を指定します。フィールドの詳細については、各フィールドの横にある  アイコンをクリックします。
5. [Search] (検索) をクリックします。

6. 検索結果に一覧表示されているジョブの詳細情報を表示するには、ジョブの横にある



アイコンをクリックします。

## Elastic Transcoder ジョブのキャンセル

ステータスがまだ [Submitted] (送信済み) になっているジョブをキャンセルできます。このステータスは、Elastic Transcoder ではファイルの変換がスタートされていないことを意味します。以下の手順では、Elastic Transcoder コンソールを使用してジョブをキャンセルする方法について説明します。

API を使用してジョブをキャンセルするには、Elastic Transcoder でジョブの処理がスタートされないように、対応するパイプライン一時停止し、ステータスが [Submitted] (送信済み) になっているジョブを一覧表示して、該当するジョブ ID を取得します。次に、キャンセルするジョブを示すジョブ ID を使用して、ジョブをキャンセルします。詳細については以下を参照してください。

- [パイプラインのステータスを更新する](#)
- [ステータスに基づくジョブのリスト](#)
- [ジョブをキャンセルする](#)

Elastic Transcoder コンソールを使用してジョブをキャンセルする方法

1. にサインイン AWS マネジメントコンソールし、<https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. Elastic Transcoder コンソールのナビゲーションバーで、ジョブをキャンセルするリージョンを選択します。
3. オプション。ただし、推奨: ジョブの送信先であるパイプラインを一時停止して、Elastic Transcoder でジョブの処理がスタートされないようにしてください。Elastic Transcoder でジョブの処理がスタートされると、ジョブはキャンセルできません。
  - a. 左側のナビゲーションペインで、[Pipelines] (パイプライン) をクリックします。
  - b. 一時停止するパイプラインの横にあるチェックボックスをオンにします。
  - c. [Pause] (一時停止) をクリックします。
4. コンソールのナビゲーションペインで、[Jobs] (ジョブ) をクリックします
5. [Jobs] (ジョブ) ページで、次の値を指定します。

[Search By]

[Status] (ステータス) をクリックします。

[Job Status]

[Submitted] (送信済み) を選択します。

 Note

キャンセルできるのは、ステータスが [Submitted] (送信済み) になっているジョブのみです。

[Order](順序) と [Number of Jobs] (ジョブ数) については、適切な値を入力してください。

6. [Search] (検索) をクリックします。
7. 検索結果で、キャンセルするジョブであるかどうかを確認するためにジョブの詳細が必要な場合は、ジョブの横にある  アイコンをクリックします。
8. ジョブをキャンセルするに、ジョブの横にあるチェックボックスをオンにして、[Cancel] (キャンセル) をクリックします。
9. ステップ 3 でパイプラインを一時停止した場合、ジョブの処理を再開できるように、パイプラインを再アクティブ化します。
  - a. ナビゲーションペインで、[Pipelines] (パイプライン) をクリックします。
  - b. 再アクティブ化するパイプラインの横にあるチェックボックスをオンにします。
  - c. [Activate] (アクティブにする) をクリックします。

## Elastic Transcoder ジョブの作成時に指定する設定

Elastic Transcoder コンソールを使用してジョブを作成するとき、以下の値を指定できます。

 Note

ジョブのステータスが変わったとき (Elastic Transcoder でジョブの処理をスタートまたは終了したときや、Elastic Transcoder で警告またはエラーの条件が発生したときなど) に通知す

るように、Elastic Transcoder を設定できます。詳細については、「[ジョブのステータスの通知](#)」を参照してください。

## トピック

- [リージョン](#)
- [全般設定](#)
- [入力詳細: パート 1](#)
- [入力詳細: パート 2、クリップとキャプションの設定](#)
- [出力詳細: パート 1](#)
- [出力詳細: パート 2](#)
- [出力詳細: パート 3、キャプションの設定](#)
- [\(オプション\) Output Encryption](#)
- [\( 動画/サムネイルのみ \) Watermarks](#)
- [\(FLAC/MP3/MP4 のみ\) アルバムアート](#)
- [\(オプション\) ユーザーメタデータ](#)
- [\(フラグメント化された MP4/MPEG-TS 出力のみ\) Playlist](#)
- [\(フラグメント化された MP4/MPEG-TS 出力のみ、オプション\) HLS Content Protection](#)
- [\( HLSv3 およびスムーズプレイリストのみ、オプション \) デジタル著作権管理](#)

## リージョン

Elastic Transcoder では、現在使用しているリージョンでジョブが作成されます。

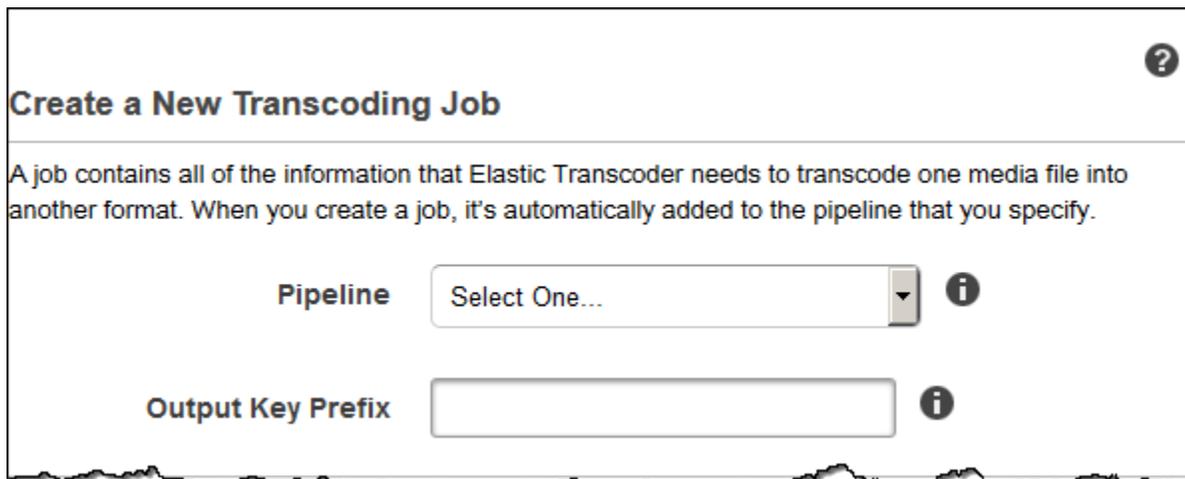
## 全般設定

### パイプライン

Elastic Transcoder で変換を行うときに使用するパイプラインの名前。パイプラインによって、いくつかの設定が決まります。それらの設定には、トランスコードするファイルを Elastic Transcoder で取得するための Amazon S3 バケットや、変換結果ファイルを Elastic Transcoder で出力するためのバケットなどがあります。

## 出力キープレフィックス

ジョブで作成されるすべてのファイル (出力ファイル、サムネイル、プレイリストなど) の名前の先頭に、Elastic Transcoder が追加する値 (値が指定されている場合)。値を指定する場合は、最初の文字よりも後の任意の場所に / を含める必要があります。これにより、Amazon S3 でのファイル管理が簡素化されます。



**Create a New Transcoding Job**

A job contains all of the information that Elastic Transcoder needs to transcode one media file into another format. When you create a job, it's automatically added to the pipeline that you specify.

**Pipeline**  ⓘ

**Output Key Prefix**  ⓘ

## 入力詳細: パート 1

このセクションの設定を使用して、入力ファイルに関する情報を指定します。複数の入力ファイルを指定すると、それらのファイルは Elastic Transcoder で変換されて 1 つの出力ファイルに結合されます。詳細については、「[クリップの結合](#)」を参照してください。

### 入力キー

変換するファイルの名前。ファイルの取得元となる Amazon S3 バケットを判別するため、Elastic Transcoder はジョブに対して指定したパイプラインの [Input Bucket] (入力バケット) フィールドを参照します。

ファイル名にプレフィックスが含まれている場合 (cooking/lasagna.mpg など)、キーにプレフィックスを含めます。指定したバケットにファイルが含まれていない場合、Elastic Transcoder からエラーが返されます。

### 入力復号化

入力ファイルの復号化に使用される暗号化設定 (存在する場合)。入力ファイルが暗号化されている場合は、Elastic Transcoder によるファイルの復号化で使用されるモードを指定する必要があります。

## 復号化モード (復号化に必須)

Elastic Transcoder でファイルの復号化に使用される特定の暗号化モード。

Elastic Transcoder では、以下のオプションがサポートされています。

- Amazon S3 サーバー側の暗号化: Amazon S3 により、ファイルの暗号化と復号化が処理されます。Elastic Transcoder で Amazon S3 バケットへのアクセス許可がある限り、何もする必要はありません。

詳細については、「[Amazon Simple Storage Service ユーザーガイド](#)」の「サーバー側の暗号化を使用したデータの保護」を参照してください。

- お客様が用意したキーを使用したクライアント側の暗号化: Elastic Transcoder では、お客様が用意したキーを使用した 3 種類の暗号化がサポートされています。
  - aes-cbc-pkcs7: オペレーションのパディング暗号ブロックモード。
  - aes-ctr: AES カウンターモード。
  - aes-gcm: AES Galois カウンターモード。認証された暗号化形式であるオペレーションモードで、ファイル、キー、または初期化ベクトルが改ざんされている場合に復号処理が失敗します。

AES 暗号化モードのいずれかを選択した場合は、次の 3 つの値も指定する必要があります (3 つすべてが base64 でエンコードされている必要があります)。

- 暗号化キー
- 暗号化キー MD5
- 暗号化初期化ベクトル

## 復号化キー (AES 復号化に必須)

データ暗号化キーは、メディアファイルの暗号化に使用されます。このキーは base64 でエンコードされますが、base64 でエンコードされる前の時点で、ビット長が、

96(AES-GCM のみ)、128、192、または 256 のいずれかである必要があります。

また、キーは を使用して暗号化する必要があります AWS KMS。詳細については、「[AWS Key Management Service デベロッパーガイド](#)」の「[データの暗号化と復号化](#)」を参照してください。

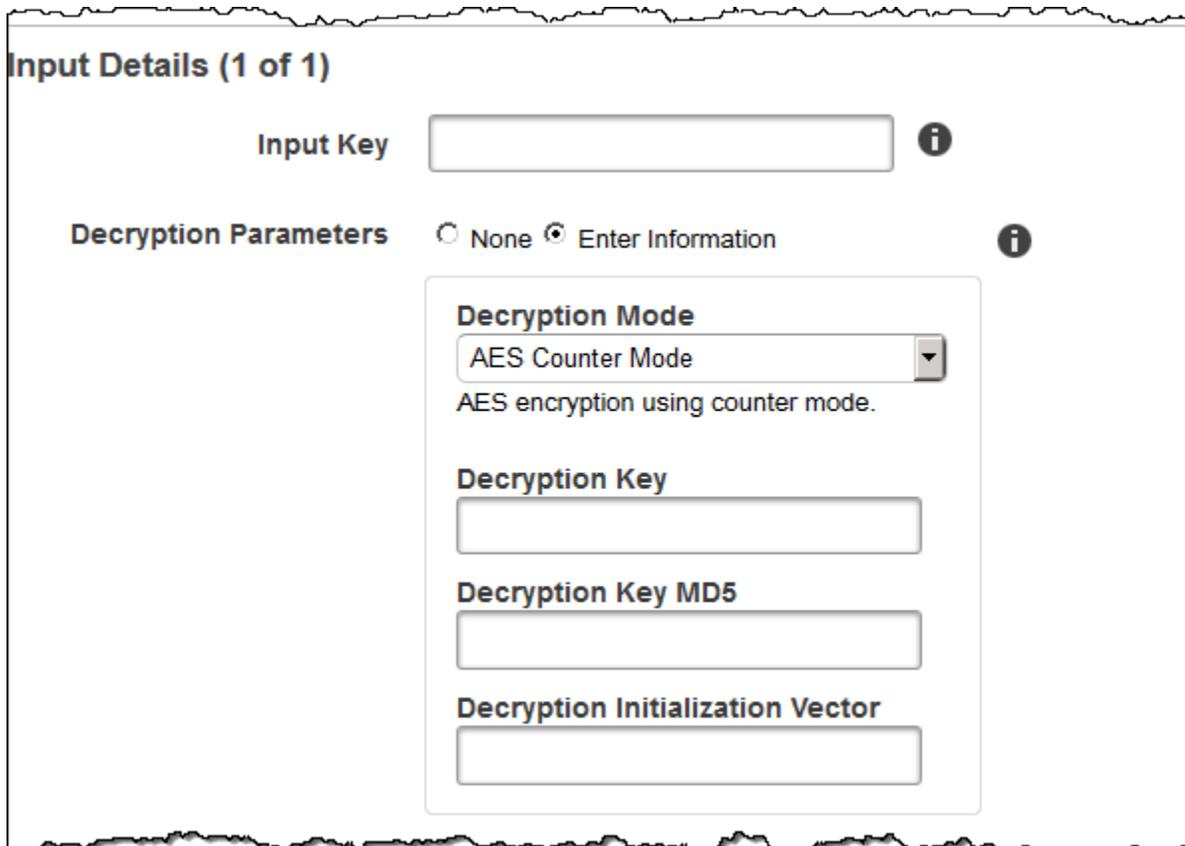
## 復号化キー MD5 (AES 復号化に必須)

入力ファイルの暗号化に使用するキー。Elastic Transcoder で、キーが転送中に破損していないことを確認するチェックサムとして使用されます。キー MD5 は base64 でエンコードされている必要があり、かつbase64 エンコードの前に正確に 16 バイトである必要があります。

## 復号化初期化ベクトル (AES 復号化に必須)

ランダムビットジェネレータによって作成された一連のランダムビット。各暗号化オペレーションに固有であり、入力ファイルの暗号化に使用します。初期化ベクトルは base64 でエンコードされている必要があり、かつbase64 エンコードの前に正確に 16 バイトである必要があります。

詳細については、「[初期化ベクトル](#)」を参照してください。



**Input Details (1 of 1)**

**Input Key**

**Decryption Parameters**  None  Enter Information

**Decryption Mode**  
AES Counter Mode  
AES encryption using counter mode.

**Decryption Key**

**Decryption Key MD5**

**Decryption Initialization Vector**

## 入力詳細: パート 2、クリップとキャプションの設定

このセクションの設定を使用して、入力ファイルのクリップとキャプションに関する情報を指定します。これらの設定は、複数の入力を指定した場合にのみ使用できます。

キャプションの詳細については、「[字幕](#)」を参照してください。

## (オプション) クリップの開始時間

入力ファイルの一部を抜粋して、出力ファイルを作成できます。[Clip Start Time] (クリップのスタート時間) は、クリップを開始する入力ファイル上の位置を示します。形式は、HH:mm:ss.SSS (最大値: 23:59:59.999、SSS は 1000 分の 1 秒部分) または sssss.SSS (最大値: 86399.999) を使用できます。値を指定しない場合、Elastic Transcoder では入力ファイルの先頭がスタート位置と見なされます。

## (オプション) クリップの再生時間

抜粋クリップの再生時間。形式は、HH:mm:ss.SSS (最大値: 23:59:59.999、SSS は 1000 分の 1 秒部分) または sssss.SSS (最大値: 86399.999) を使用できます。値を指定しない場合、Elastic Transcoder では、[Clip Start Time] (クリップのスタート時間) からファイルの終了までのクリップを作成します。

入力ファイルの再生時間よりも長い値を指定した場合、Elastic Transcoder では、[Clip Start Time] (クリップのスタート時間) からファイルの終了までを変換し、警告メッセージを返します。

## (ビデオのみ) キャプションマージポリシー

複数のキャプションが存在する場合の Elastic Transcoder での処理方法を決定するポリシー。

- MergeOverride: Elastic Transcoder では、埋め込みキャプションとサイドカーキャプションの両方が出力に変換されます。ある言語のキャプションが入力ファイルに埋め込まれていて、さらにサイドカーにも表示される場合、Elastic Transcoder ではサイドカーキャプションが使用され、その言語の埋め込みキャプションは無視されます。
- MergeRetain: Elastic Transcoder では、埋め込みキャプションとサイドカーキャプションの両方が出力に変換されます。ある言語のキャプションが入力ファイルに埋め込まれていて、さらにサイドカーにも表示される場合、Elastic Transcoder で埋め込みキャプションが使用され、その言語のサイドカーキャプションは無視されます。[Caption Source] (キャプションソース) が空の場合、Elastic Transcoder の出力ファイルではすべてのサイドカーキャプションが除外されます。
- Override: Elastic Transcoder では、[Caption Source] (キャプションソース) で指定したサイドカーキャプションのみを変換します。

## (ビデオのみ、オプション) 入力キー

Elastic Transcoder で変換して出力に含めるサイドカーキャプションファイルの名前。

## (ビデオのみ) 言語

キャプションの言語を示す文字列。次の形式のいずれかで指定します。

- 2 文字の ISO 639-1 コード (例: en (英語))
- 3 文字の ISO 639-2 コード (例: eng (英語))

ISO 言語コードの詳細については、「[ISO 639-1 コードのリスト](#)」を参照してください。

(ビデオのみ、オプション) TimeOffset

クリップの生成や関連するビデオファイルと同時にスタートされないキャプションでは、[TimeOffset] (時間オフセット) を使用して、キャプションを含める前にエンコードするビデオの長さを Elastic Transcoder に指示することができます。

タイムオフセットは、[+]*SS.sss* または [+]*HH:mm:SS.ss* の形式で指定します。

(ビデオのみ、オプション) ラベル

言語を選択するときにプレーヤーに表示されるキャプションのラベル。ここにはキャプションの言語名を、キャプションを表示する言語で指定することをお勧めします。

**Available Settings**  Clip  Input Captions ⓘ

**Clip Start Time**  ⓘ

**Clip Duration**  ⓘ

**Captions Merge Policy** Merge and Override ⓘ

**Caption Source**

[- Remove Caption Source](#)

**Input Key**  ⓘ

**Language**  ⓘ

**Time Offset**  ⓘ

**Label**  ⓘ

**Caption Decryption Parameters**  None  Enter Information ⓘ

**Decryption Mode**  
 ⓘ  
AES encryption using counter mode.

**Decryption Key**

**Decryption Key MD5**

**Decryption Initialization Vector**

[+ Add Caption Source](#)

[+ Add Another Input](#)

## 出力詳細: パート 1

このセクションの設定を使用して、出力ファイルに関する情報を指定します。

### プリセット

この出力に対して使用するプリセット。プリセットによって、変換時に Elastic Transcoder で使用される音声、ビデオ、サムネイルの各設定が決まります。

### (フラグメント化された MP4/MPEG-TS 出力のみ) セグメント再生時間

現在の出力に対して指定したプリセットで、[コンテナ] の値が [ts] (MPEG-TS) または [fmp4] (フラグメント化された MP4) である場合、[Segment Duration] (セグメント再生時間) は各セグメントのターゲット最長再生時間 (秒単位) になります。HLSv3 形式のプレイリストでは、各メディアセグメントは個別の .ts ファイルに保存されます。HLSv4、MPEG-DASH、および Smooth プレイリストでは、出力のメディアセグメントはすべて 1 つのファイルに保存されます。個々のセグメントの長さはそれぞれ異なるとしても、だいたい [Segment Duration] (セグメント再生時間) の長さになります。

有効な値の範囲は 1~60 秒です。動画の再生時間が [Segment Duration] (セグメント再生時間) で割り切ることができない場合、最後のセグメントの再生時間は、次の式の剰余の値になります。

$$\text{total length} / \text{Segment Duration}$$

Elastic Transcoder では、[Output Keys] (出力キー) で指定した各 HLS に対して出力固有のプレイリストを作成します。このジョブのマスタープレイリストに出力を追加するには、[Outputs in Master Playlist](#) にその出力を含めます。

Elastic Transcoder では、出力ビデオに関連付けられたすべてのキャプションにこのセグメント化が適用されます。

### 出力キー

Elastic Transcoder でトランスコード済みファイルおよびプレイリストに割り当てる名前。Elastic Transcoder によって、[Pipeline](#) で指定したパイプラインの[バケット] フィールドによって指定された Amazon S3 バケットに、1 つまたは複数のファイルが保存されます。名前の指定されたファイルがバケットに含まれている場合、出力は失敗します。ただし、同じジョブの別の出力は成功する場合があります。

ファイル名の形式は、コンテナタイプと、セグメントの再生時間が設定されているかどうかによって異なります。コンテナタイプが ts ではない場合や、セグメントの再生時間が指定され

ていない場合、出力ファイルの名前は [Output Key Prefix] (出力キープレフィックス) と [Output Key] (出力キー) を連結したものになります。

コンテナタイプが `ts` であり、セグメントの再生時間が指定されている場合、Elastic Transcoder は [Output Key] (出力キー) の値を使用して、出力のプレイリストと `.ts` ファイルの両方に次のように名前を付けます。

• プレイリスト:

- HLSv3: ファイル名は、[Output Key Prefix] (出力キープレフィックス) と [Output Key] (出力キー) を連結したものに、ファイル名拡張子 `.m3u8` を付加した名前になります。

Output Key PrefixOutput Key.m3u8

- HLSv4: ファイル名は、[Output Key Prefix] (出力キープレフィックス) と [Output Key] (出力キー) を連結したものに、ファイル名拡張子 `_v4.m3u8` を付加した名前になります。ビデオ出力では、[Output Key Prefix] (出力キープレフィックス) と [Output Key] (出力キー) を連結したものに、ファイル名拡張子 `_iframe.m3u8` を付加したファイル名で別のファイルが作成されます。

Output Key PrefixOutput Key\_v4.m3u8

Output Key PrefixOutput Key\_iframe.m3u8 (動画のみ)

• セグメント ( `.ts` ) ファイル:

- HLSv3: ファイル名は、[Output Key Prefix] (出力キープレフィックス) と [Output Key] (出力キー) を連結したものに、00000 から始まる 5 桁のシーケンシャルカウンタとファイル名拡張子 `.ts` を付加した名前になります。

Output Key PrefixOutput Key00000.ts

- HLSv4: ファイル名は、[Output Key Prefix] (出力キープレフィックス) と [Output Key] (出力キー) を連結したものに、ファイル名拡張子 `.ts` を付加した名前になります。

Output Key PrefixOutput Key.ts

セグメント化された `ts` 出力がマスタープレイリストに含まれていない場合、Elastic Transcoder はこの出力を HLSv3 として扱います。

**Note**

Elastic Transcoder は HLSv3 または HLSv4 プレイリストの出力に、関連するファイル拡張子を自動的に追加します。HLSv3 または HLSv4 プレイリスト出力の出力キーにファイル拡張子を含めると、ファイル名に 2 つの拡大子が付きます。

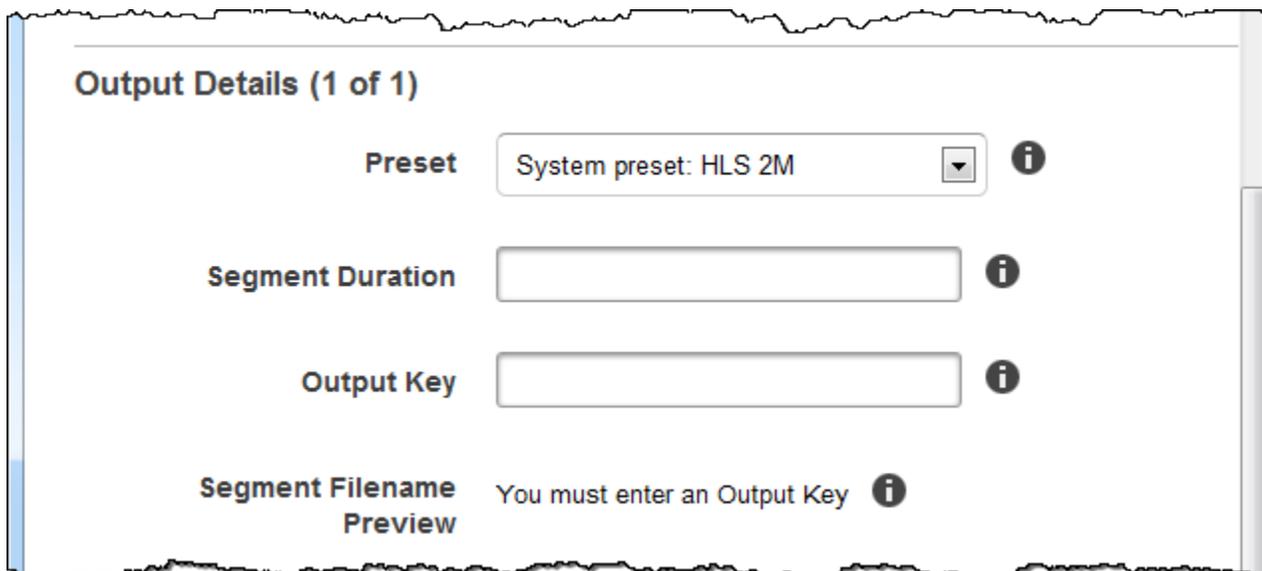
出力キープレフィックスによって、ジョブのすべてのファイルが Amazon S3 バケットにまとめられます。ジョブ内の出力ごとにファイルをまとめる場合は、[Output Key] (出力キー) の値にプレフィックスを含めます。次に例を示します。

OutputKeyPrefixiPhone/Key00000.ts

OutputKeyPrefixKindleFireHD/Key00000.ts

**セグメントファイル名のプレビュー**

出力キーについて入力した値に基づく、最初のセグメントファイルの名前。



**Output Details (1 of 1)**

**Preset** System preset: HLS 2M ⓘ

**Segment Duration** ⓘ

**Output Key** ⓘ

**Segment Filename Preview** You must enter an Output Key ⓘ

**出力詳細: パート 2**

このセクションの設定を使用して、出力ファイルに関する情報を指定します。

## (ビデオのみ) サムネイルの作成

Elastic Transcoder でビデオのサムネイルを作成する場合、[はい] を選択し、[Thumbnail Filename Pattern] (サムネイルファイル名パターン) フィールドでファイル名の形式を指定します。

### (ビデオのみ) サムネイルファイル名パターン

[Thumbnail Filename Pattern] (サムネイルファイル名パターン) で [はい] を選択した場合は、ファイル名の形式を指定します。次の値を任意の順序で指定できます。

- {count} ( 必須 ) : 00001 から始まる 5 桁の数値。指定のサムネイルが、変換結果ファイルの一連のサムネイルとして表示されることを示します。フィールドの任意の場所に {count} を含める必要があります。これを省略すると、Elastic Transcoder では、ファイル名の末尾 (ファイル名拡張子 (.jpg または .png) の直前) にカウントが自動的に追加されます。
- ( オプション ) Literal values: フィールドの任意の場所にリテラル値を追加できます。たとえば、ファイル名のプレフィックスや、{resolution} と {count} の間の区切り文字として指定できます。
- (オプション) {resolution}: Elastic Transcoder でファイル名に解像度を含める場合は、フィールドに {resolution} を追加します。

[Thumbnail Filename Preview] (サムネイルファイル名パターン) フィールドには、[Thumbnail Filename Pattern] に入力した値に基づいて、サムネイルのファイル名のサンプルが表示されます。

サムネイルを作成すると、Elastic Transcoder では、ファイルが .jpg または .png の形式で自動的に保存され、[Preset](#) で指定したプリセットに表示されます。Elastic Transcoder により、該当するファイル名拡張子も追加されます。

### (ビデオ/サムネイルのみ、オプション) 出力ローテーション

右回りの角度の値。Elastic Transcoder で、入力に関連して出力をローテーションさせる角度です。Elastic Transcoder で入力ファイルがローテーションされているかどうかを自動的に検出するには、auto を選択します。ただし、通常 Elastic Transcoder では、変換するファイルにローテーションメタデータが含まれている場合に出力をローテーションさせる必要があるかどうかを検出するだけです。

**Create Thumbnails**  No  Yes ⓘ

**Thumbnail Filename Pattern**  ⓘ

**Thumbnail Filename Preview** You must enter a Thumbnail Pattern ⓘ

**Thumbnail Encryption Parameters**  None  Enter Information ⓘ

**Encryption Mode**

AES Galois Counter Mode ⓘ

AES encryption using Galois counter mode.

**Auto generate encryption key**

This Pipeline does not have an AWS master key assigned to it. Assign a master key to the pipeline to use this mode. [Learn more](#)

**Output Rotation (Clockwise)** auto ⓘ

**Encryption Parameters**

To protect your HLS Outputs, add an HLSv3 or HLSv4 playlist along with Content Protection information.

**Available Settings**  Output Captions ⓘ

## 出力詳細: パート 3、キャプションの設定

このセクションの設定を使用して、出力ファイルのキャプションに関する情報を指定します。

キャプションの詳細については、「」を参照してください。

## (ビデオのみ) キャプション形式

指定した形式によって、Elastic Transcoder では出力用に埋め込みキャプションを生成するか、サイドカーキャプションを生成するかが決まります。この値を空白にすると、Elastic Transcoder はエラーを返します。

- 埋め込みキャプション形式: MP4 コンテナでは、mov-text および CEA-708 がサポートされています。MPEG-TS コンテナでは、CEA-708 がサポートされています。他のコンテナタイプでは、埋め込みキャプション形式はサポートされていません。

CEA-708 キャプションはストリームの H.264 SEI ユーザーデータに埋め込まれています。Elastic Transcoder では、出力ごとに 1 つのみの埋め込み形式がサポートされています。

- サイドカーキャプション形式: Elastic Transcoder では、dfxp、scc、srt、webvtt がサポートされています。Smooth プレイリストを持つ Fmp4 コンテナは dfxp のみをサポートし、Elastic Transcoder は拡張子 .ismt を持つファイルを作成します。MPEG-DASH プレイリストを持つ Fmp4 コンテナは webvtt のみをサポートし、Elastic Transcoder Elastic Transcoder は拡張子 .vtt を持つファイルを作成します。ttml または smpte-tt と互換性のあるキャプションが必要な場合は、出力フォーマットとして dfxp を指定してください。

## (ビデオのみ) キャプションファイル名パターン

キャプションファイル名のプレフィックス。 *description*-{language} の形式で指定します。

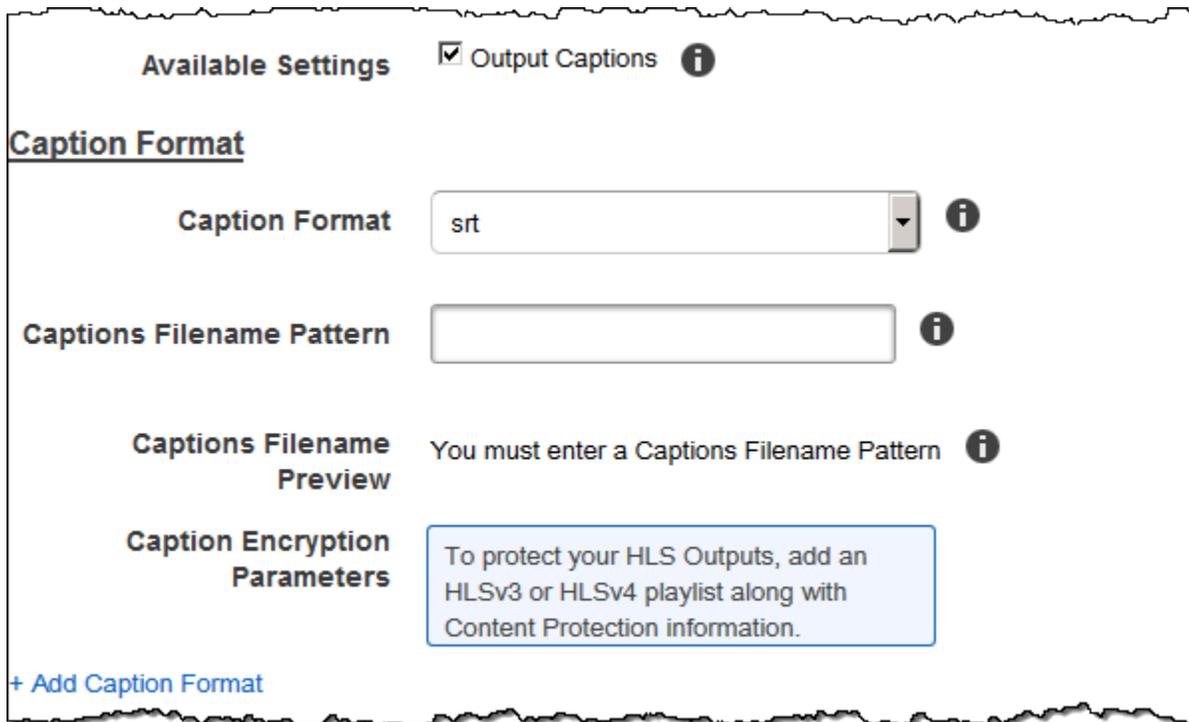
- *description* は、動画の説明です。
- {language} はリテラル値で、Elastic Transcoder により出力ファイル名のキャプションの言語を表す 2 文字または 3 文字のコードに置き換えられます。

ファイル名パターンに {language} を含めない場合、Elastic Transcoder では、##で指定した値に「{language}」が自動的に付加されます。Elastic Transcoder ではさらに、セグメントファイルの最後にカウントが自動的に追加されます。

たとえば、srt 形式に変換するとします。"Sydney-{language}-sunrise" と入力し、キャプションの言語が英語 (en) である場合、最初のキャプションファイルの名前は Sydney-en-sunrise00000.srt になります。

## (ビデオのみ) キャプションファイル名プレビュー

[Captions Filename Pattern] (キャプションファイル名パターン) に入力した値に基づく、最初のキャプションファイルの名前。プレビューでは、英語を表す ISO コード "en" を使用して、ファイル名で {language} がどのように表示されるかを示します。



## (オプション) Output Encryption

トランスコーディングジョブで使用するすべての出力ファイルについて、暗号化の設定を指定できます。これには、出力ファイル、および使用するウォーターマーク、サムネイル、アルバムアート、またはキャプションが含まれます。各ファイルに個別に暗号化設定を指定する必要があります。

### 出力暗号化

Elastic Transcoder で出力ファイルに適用する暗号化設定 (存在する場合)。暗号化の使用を選択した場合は、使用するモードを指定する必要があります。暗号化を使用しない場合、Elastic Transcoder は Amazon S3 バケットに暗号化されていないファイルを書き込みます。

### (ファイルレベルの暗号化に必須) 暗号化モード

Elastic Transcoder でファイルの復号化に使用される特定の暗号化モード。Elastic Transcoder では次の暗号化モードオプションがサポートされています:

- s3: Amazon S3 は、出力ファイルの暗号化に使用するキーを作成および管理します。

詳細については、「Amazon Simple Storage Service ユーザーガイド」の「[サーバー側の暗号化を使用したデータの保護](#)」を参照してください。

- s3-aws-kms: Amazon S3 呼び出し。ファイルの暗号化に使用されるキー AWS KMSを作成および管理します。s3-aws-kms を指定し、デフォルトのキーを使用しない場合は、使用する AWS-KMS キーをパイプラインに追加する必要があります。

詳細については、「Amazon Simple Storage Service ユーザーガイド」の「[AWS KMS で管理された暗号化キーによるサーバー側の暗号化を使用したデータの保護](#)」を参照してください。

- aes-cbc-pkcs7: オペレーションのパディング暗号ブロックモード。
- aes-ctr: AES カウンターモード。
- aes-gcm: AES Galois カウンターモード。認証された暗号化形式であるオペレーションモードで、ファイル、キー、または初期化ベクトルが改ざんされている場合に復号処理が失敗します。

AES 暗号化モードのいずれかを選択した場合は、次の 3 つの値も指定する必要があります (3 つすべてが base64 でエンコードされている必要があります)。

- 暗号化キー
- 暗号化キー MD5
- 暗号化初期化ベクトル

AES 暗号化モードのいずれかを選択し、Elastic Transcoder で 128 ビットの AES 暗号化キーを生成する場合は、暗号化キー、暗号化キー MD5、暗号化初期化ベクトルのいずれの値も指定しないでください。Elastic Transcoder でキーが生成されたら、ReadJob を呼び出してキーを取得できます。このキーは、CreateJobResponse オブジェクトには含まれません。

#### Important

AES モードでは、メディア固有のプライベート暗号化キーと暗号化されていないデータは AWS には保存されません。したがって、暗号化キーを安全に管理することが重要になります。キーを紛失すると、データを復号できなくなります。

### (オプション) 暗号化キー

Elastic Transcoder でキーが生成される場合は、このフィールドを空白にしておく必要があります。Elastic Transcoder でキーが生成されたら、[Read Job] (ジョブの読み取り) を呼び出してキーを取得できます。このキーは、[Create Job Response] (ジョブレスポンスの作成) オブジェクトには含まれません。

独自のキーの使用を選択した場合は、AWS KMS を使用してキーを暗号化する必要があります。このキーは base64 でエンコードされますが、base64 でエンコードされる前の時点で、ビット長が、

96(AES-GCM のみ)、128、192、または 256 のいずれかである必要があります。

Elastic Transcoder でキーが生成されるように設定した場合、Elastic Transcoder では、Create Job (ジョブの作成) レスポンスのこのフィールドが空白のままになります。生成されたデータ暗号化キーを取得するには、Read Job (ジョブの読み取り) リクエストを送信します。

によるキーの暗号化の詳細については AWS KMS、AWS Key Management Service デベロッパーガイドの「[データの暗号化と復号化](#)」を参照してください。

(暗号化キーが指定されている場合に必要) 暗号化キー MD5

Elastic Transcoder で出力ファイルの暗号化に使用するキーの MD5 ダイジェスト。Elastic Transcoder で、キーが転送中に破損していないことを確認するチェックサムとして使用されます。キー MD5 は base64 でエンコードされている必要があります、かつ base64 エンコードの前に正確に 16 バイトである必要があります。

Elastic Transcoder でキーが生成される場合は、このフィールドを空白にしておく必要があります。

(暗号化キーが指定されている場合に必要) 暗号化初期化ベクトル

ランダムビットジェネレーターによって作成された一連のランダムビット。各暗号化オペレーションに固有であり、Elastic Transcoder で出力ファイルの暗号化に使用します。初期化ベクトルは base64 でエンコードされている必要があります、かつ base64 エンコードの前に正確に 16 バイトである必要があります。

Elastic Transcoder でキーが生成される場合は、このフィールドを空白にしておく必要があります。

詳細については、「[Initialization Vector](#)」(初期化ベクトル) を参照してください。

**Output Details (1 of 1)**

**Preset**  ⓘ

**Output Key**  ⓘ

**Encryption Parameters**  None  Enter Information ⓘ

**Encryption Mode**

ⓘ

AES encryption using counter mode.

**Auto generate encryption key**

**Encryption Key**

This Pipeline does not have an AWS master key assigned to it. Assign a master key to the pipeline to use this mode. [Learn more](#)

**Encryption Key MD5**

**Encryption Initialization Vector**

**Available Settings**  Output Captions ⓘ

[+ Add Another Output](#)

## ( 動画/サムネイルのみ ) Watermarks

変換時に Elastic Transcoder でビデオに追加されるウォーターマークに関する情報。出力ごとに最大で 4 つのウォーターマークを指定できます。各ウォーターマークの設定は、現在の出力の [Presets] (プリセット) で指定したプリセット内に定義する必要があります。

ウォーターマークは、ジョブ出力でリストする順番で出力ビデオに追加されます。最初にリスト内の 1 番目のウォーターマークが出力ビデオに追加され、次にリスト内の 2 番目のウォーターマークが

追加されます。その結果、プリセットの設定が原因で、Elastic Transcoder ではすべてのウォーターマークが同じ場所に配置される場合、追加した 2 番目のウォーターマークは最初のウォーターマークを覆うように配置され、3 番目のウォーターマークは 2 番目のウォーターマークを覆うように配置され、4 番目のウォーターマークは 3 番目のウォーターマークを覆うように配置されていきます。

ウォーターマークの詳細については、「[ウォーターマーク](#)」を参照してください。

## プリセットウォーターマーク ID

変換時にビデオにウォーターマークを追加するために Elastic Transcoder で使用される、ウォーターマーク設定の ID。設定は、現在の出力の [Presets] (プリセット) で指定したプリセットに含まれています。そのプリセットでは、[Watermarks Id] の値によって、どの設定を使用するかが Elastic Transcoder に伝えられます。

## プリセットウォーターマーク ID の入力キー

ウォーターマークに使用する .png ファイルまたは .jpg ファイルの名前。指定したファイルが含まれている Amazon S3 バケットを判別するために、Elastic Transcoder では、[パイプライン] で指定されたパイプラインを調べます。そのパイプラインの Input Bucket (入力バケット) オブジェクトによって、バケットが特定されます。

ファイル名にプレフィックスが含まれている場合 ( logos/128x64.png など )、キーにプレフィックスを含めます。指定したバケットにファイルが含まれていない場合、Elastic Transcoder からエラーが返されます。

The screenshot shows the 'Watermarks' configuration page. At the top, there is a 'Preset Watermark Id' dropdown menu with 'Select One...' as the current selection. Below this, there is a '- Remove' link. The main configuration area is titled 'Input Key for Preset Watermark Id TopLeft' and contains a text input field with an information icon. Below that is the 'Watermark Decryption Parameters' section, which has two radio buttons: 'None' and 'Enter Information' (which is selected). This section contains a sub-panel with the following fields: 'Decryption Mode' (a dropdown menu showing 'AES Cipher-Block-Chaining with PKCS7'), 'Decryption Key' (a text input field), 'Decryption Key MD5' (a text input field), and 'Decryption Initialization Vector' (a text input field). At the bottom of the configuration area, there is a '+ Add Another Output' link.

## (FLAC/MP3/MP4 のみ) アルバムアート

出カファイルに関連付けるアルバムアート (ある場合)。

アートワークを削除またはアートワークを空にするには、[Artwork] を null に設定するか、[Merge Policy] を "Replace" に設定してから、空の [Artwork] 配列を使用します。

既存のアートワークを変更せずにそのまま使用するには、[Merge Policy] を "Prepend"、"Append"、または "Fallback" に設定してから、空の [Artwork] 配列を使用します。

**Note**

アルバムアートは flac、mp3、または mp4 コンテナの音声のみの出力で使用できます。

## アルバムアートマージポリシー

Elastic Transcoder でアルバムアートワークファイルが複数存在する場合の処理方法を決定するポリシー。

- [Replace]: 指定したアルバムアートによって既存のアルバムアートを置き換えます。
- [Prepend]: 指定したアルバムアートをすべての既存のアルバムアートの先頭に配置します。
- [Append]: 指定したアルバムアートをすべての既存のアルバムアートの後ろに配置します。
- [フォールバック]: 入力ファイルにアートワークが含まれている場合、Elastic Transcoder ではそのアートワークを出力で使用します。入力にアートワークが含まれていない場合、Elastic Transcoder では指定したアルバムアートファイルを使用します。

## アルバムアートのアートワーク

アルバムアートとして使用されるファイル。音声ファイルに最大 20 個のアートワークを関連付けることができます。

## アルバムアートの入力キー

アルバムアートとして使用されるファイルの名前。指定したファイルを含んでいる Amazon S3 バケットを判別するために、Elastic Transcoder では、[PipelineId] で指定されたパイプラインを調べます。そのパイプラインの [InputBucket] オブジェクトによって、バケットが特定されます。

ファイル名にプレフィックスが含まれている場合 ( cooking/pie.jpg など )、キーにプレフィックスを含めます。ファイルが指定したバケットに含まれていない場合、Elastic Transcoder はエラーを返します。

## アルバムアートの形式

アルバムアートの形式 (ある場合)。有効な形式は、jpg と png です。

## アルバムアートの最大幅

出力アルバムアートの最大幅 (ピクセル単位)。auto を指定すると、Elastic Transcoder ではデフォルト値として 600 が使用されます。数値を指定する場合は、32 以上 4096 以下の偶数を入力します。

## アルバムアートの最大高さ

出力アルバムアートの最大の高さ (ピクセル単位)。auto を指定すると、Elastic Transcoder ではデフォルト値として 600 が使用されます。数値を指定する場合は、32 以上 3072 以下の偶数を入力します。

## アルバムアートのサイジングポリシー

出力アルバムアートのスケーリングを制御する値。

- [Fit]: Elastic Transcoder で、[MaxWidth] (最大幅) または [MaxHeight] (最大高さ) で指定した値に一致し、かつ他方の値を超えないように、出力アートのスケーリングが行われます。
- [Fill]: Elastic Transcoder で、[MaxWidth] (最大幅) または [MaxHeight] (最大高さ) で指定した値に一致し、かつ他方の値に一致するかこれを超えるように、出力アートのスケーリングが行われます。Elastic Transcoder では、出力アートが中央に配置され、次に、最大値を超えるディメンション (該当する場合) がトリミングされます。
- [Stretch]: Elastic Transcoder では、[MaxWidth] (最大幅) と [MaxHeight] (最大高さ) に指定した値に一致するように、出力アートを引き伸ばします。入力アートと出力アートの相対的な比率が異なる場合、出力アートは歪みます。
- [Keep]: Elastic Transcoder で出力アートのスケーリングが行われません。入力アートのいずれかのディメンションが、[MaxWidth] (最大幅) と [MaxHeight] (最大高さ) に指定した値を超える場合、Elastic Transcoder により出力アートがトリミングされます。
- [ShrinkToFit]: Elastic Transcoder では、出力アートのディメンションが [MaxWidth] (最大幅) と [MaxHeight] (最大高さ) に指定した少なくともどちらか 1 つの値に一致し、かついずれの値も超えないように、出力アートが縮小されます。このオプションを指定すると、Elastic Transcoder で出力アートは拡大されません。
- [ShrinkToFill]: Elastic Transcoder では、出力アートのディメンションが [MaxWidth] (最大幅) と [MaxHeight] (最大高さ) に指定した少なくともどちらか 1 つの値に一致し、かつこれらの値を下回らないように、出力アートを縮小します。このオプションを指定すると、Elastic Transcoder で出力アートは拡大されません。

次の表は、出力アルバムアートで発生する可能性がある [SizingPolicy] の設定の影響を示しています。

SizingPolicy	出力アルバムアートが拡大される可能性がある	パディングポリシーが "Pad" の場合に出力アルバムアートが埋め込まれる可能性がある	出力アルバムアートが入力アルバムアートとは異なるピクセルアスペクト比になる可能性がある	出力アルバムアートがトリミングされる可能性がある
Fit	はい	はい		
Fill	はい			はい
[Stretch] (ストレッチ)	はい		はい	
Keep	はい	あり		はい
ShrinkToFit に設定します。		はい		
ShrinkToFill		はい		はい

## アルバムアートのパディングポリシー

[パディングポリシー] を Pad に設定すると、Elastic Transcoder で、出力アルバムアートの上下または左右 (あるいはその両方) に白いバーが追加される場合があります。これは、出力アートの合計サイズを [MaxWidth] (最大幅) と [MaxHeight] (最大高さ) に指定した値に一致させるためです。詳細については、「AlbumArt:Art:SizingPolicy」の表を参照してください。

### The art to be associated with the output file.

**Merge Policy**  ⓘ

[+ Add Artwork](#)

#### Artwork

**Input Key**  ⓘ

**Format**  ⓘ

**Max Width**  ⓘ

**Max Height**  ⓘ

**Sizing Policy**  ⓘ

**Padding Policy**  ⓘ

[- Remove Artwork](#)

**Artwork Decryption Parameters**  None  Enter Information i

**Decryption Mode**  
AES Counter Mode  
AES encryption using counter mode.

**Decryption Key**

**Decryption Key MD5**

**Decryption Initialization Vector**

[- Remove Artwork](#)

[+ Add Another Output](#)

## (オプション) ユーザーメタデータ

Elastic Transcoder ジョブに関連付けるユーザー定義のメタデータ。key/value ペアでメタデータを指定します。key/value ペアを使ってファイルの詳細情報を追跡することができます (例: Season 1: Episode 3)。

各ジョブに最大 10 個のキー/値ペアを追加できます。Elastic Transcoder では、key/value ペアが、指定したときの順序と同じ順序で返される保証はありません。

### メタデータキー

出力ファイルとともに返されるメタデータ key/value ペアのキー。各キーは 1-128 文字の一意の文字列でなければならず、各キーには以下のリストに記載されている文字しか使用できません。

- 0-9
- A-Z および a-z
- Space
- 次の記号: \_ . : / = + - % @

メタデータの整理や、メタデータの追加の 128 文字の保存、[値] に保存されたメタデータのラベリングを行う場合に、キーを番号方式として使用できます。値のメタデータのみを使用する場合は、key1 などのキーにスローアウェイ文字列を入れることができ、Elastic Transcoder からメタデータを取得するときにはそのキーを無視できます。

#### Important

ジョブ内のすべてのキーに一意的な文字列を指定する必要があります。ジョブ内の複数のキーに同じ文字列が使用されている場合、Elastic Transcoder では、そのキーを使用しているキー/値ペアのうちの 1 つだけが返されます。どの値が返されるのかを保証する方法はありません。

## メタデータ値

ジョブとともに返されるメタデータ key/value ペアの値。各値は 0-256 文字の文字列でなければならない。各値には以下のリストに記載されている文字しか使用できません。

- 0-9
- A-Z および a-z
- Space
- 次の記号: \_ . : / = + - % @



Key	Value
<input type="text"/>	<input type="text"/>

## (フラグメント化された MP4/MPEG-TS 出力のみ) Playlist

[プリセット] リストから、[コンテナ] の値が [ts] (MPEG-TS) または [fmp4] (フラグメント化された MP4) であるプリセットを選択した場合は、このセクションの設定を使用して、Elastic Transcoder で作成するマスタープレイリストに関する情報を指定します。プレイリスト形式ごとに複数のマスタープレイリストは作成しないことをお勧めします。

## マスタープレイリスト名

Elastic Transcoder でマスタープレイリストに割り当てる名前。名前に / 文字が含まれている場合、名前の最後の / より前の部分はすべてのプレイリスト名で同じである必要があります。複数のマスタープレイリストを作成する場合、各プレイリストには一意の名前を付ける必要があります。

### Note

Elastic Transcoder は、ファイル名に関連する拡張子を自動的に追加します ( HLSv3 プレイリストと HLSv4 プレイリストには .m3u8、MPEG-DASH プレイリストには .mpd、Smooth プレイリストには .ism と ismc )。[Master Playlist Name] にファイル拡張子を含めると、ファイル名の拡張子が 2 つになります。

セグメントの再生時間、クリップ、およびキャプションの各設定はいずれも、プレイリストのすべての出力で同じにする必要があります。[Smooth] プレイリストの場合、[Audio:Profile]、[Video:Profile]、および [Maximum Number of Frames Between keyframes] (キーフレーム間の最大フレーム数) に対する [Frame Rate] (フレームレート) の比率をすべての出力で同じにする必要があります。

## プレイリスト形式

プレイリストの形式。有効な形式は、[HLSv3]、[HLSv4]、[MPEG-DASH]、[Smooth] です。

## マスタープレイリストの出力

マスタープレイリストに含めるこのジョブの各出力に対する、[Output Key] フィールドの値。プレイリストに複数の出力を含める場合、すべての出力に対する [Segment Duration] の値は同じにする必要があります。

HLSv4 マスタープレイリストについては、出力プレイリスト内でリンクする音声入力とビデオ入力の組み合わせが Elastic Transcoder によって選択されます。デフォルトの再生操作として、最初の音声入と動画入力がリンクおよびレンダリングされます。このため、優先するデフォルトの再生操作を選択することができます。マスタープレイリストに含まれるその他の個々のプレイリストについては、最適な再生のための音声とビットレートの組み合わせが Elastic Transcoder によって選択されます。

Playlist (1 of 1) [- Remove Playlist](#)

Master Playlist Name

Playlist Format

Outputs in Master Playlist

Content Protection  None  HLS AES  PlayReady DRM

[+ Add Playlist](#)

## (フラグメント化された MP4/MPEG-TS 出力のみ、オプション) HLS Content Protection

### Note

HLS のコンテンツ保護ではファイル内の個々のセグメントが暗号化されますが、Elastic Transcoder では、HLS のコンテンツ保護とファイルレベルの暗号化の組み合わせをサポートしていません。

[プリセット] リストから、[コンテナ] の値が [ts] (MPEG-TS) または [fmp4] (フラグメント化された MP4) で、[Smooth] プレイリストであるプリセットを選択した場合、ファイルのストリーミングに対して暗号化設定を指定できます。データストリームの暗号化とファイルまたはキャプションの暗号化を組み合わせることはできません。データストリームを暗号化することを選択した場合は、このセクションの設定を使用して、データストリームの暗号化に関する情報を指定します。

HLS のコンテンツ保護を使用するには、セッションを認証できるウェブサーバー (Amazon Elastic Compute Cloud など)、ストリーミングメディアファイルを配信する方法 (Amazon CloudFront など)、暗号化されたストリーミングメディアを再生する方法 (プレイヤー対応ブラウザなど) が必要です。

### 方法

出力のコンテンツ保護方法。唯一の有効な値:

aes-128.

この値は、出力プレイリストの EXT-X-KEY メタデータタグの method 属性に書き込まれます。

## キー

Elastic Transcoder でキーが生成される場合は、このフィールドを空白にしておく必要があります。Elastic Transcoder でキーが生成されたら、ReadJob を呼び出してキーを取得できます。このキーは、CreateJobResponse オブジェクトには含まれません。

独自のキーの使用を選択した場合は、AWS KMS を使用してキーを暗号化する必要があります。このキーは base64 でエンコードされますが、base64 でエンコードされる前の時点で、ビット長が、

96(AES-GCM のみ)、128、192、または 256 のいずれかである必要があります。

Elastic Transcoder でキーが生成されるように設定した場合、Elastic Transcoder では、CreateJob レスポンスのこのフィールドが空白のままになります。生成されたデータ暗号化キーを取得するには、ReadJob リクエストを送信します。

によるキーの暗号化の詳細については AWS KMS、AWS Key Management Service デベロッパーガイドの [「データの暗号化と復号化」](#) を参照してください。

aes-128 の HLS コンテンツ保護方法を選択した場合、キーは 128 ビットでなければなりません。relative KeyStoragePolicy セットがある場合、Elastic Transcoder では、Amazon S3 サーバー側暗号化によりキーが Amazon S3 バケットに書き込まれます。

## キー MD5 (暗号化キーが支給されている場合は必須)

Elastic Transcoder で出力ファイルの暗号化に使用するキーの MD5 ダイジェスト。Elastic Transcoder で、キーが転送中に破損していないことを確認するチェックサムとして使用されます。キー MD5 は base64 でエンコードされている必要があります、かつ base64 エンコードの前に正確に 16 バイトである必要があります。

Elastic Transcoder でキーが生成される場合は、このフィールドを空白にしておく必要があります。

## 初期化ベクトル (暗号化キーが支給されている場合は必須)

ランダムビットジェネレーターによって作成された一連のランダムビット。各暗号化オペレーションに固有であり、Elastic Transcoder で出力ファイルの暗号化に使用します。初期化ベクトルは

base64 でエンコードされている必要があります、かつbase64 エンコードの前に正確に 16 バイトである必要があります。

Elastic Transcoder でキーが生成される場合は、このフィールドを空白にしておく必要があります。

詳細については、「[初期化ベクトル](#)」を参照してください。

## ライセンス取得 URL

HLS プレイリストの復号化に必要なライセンスキーの場所。この URL は絶対パスでなければならず、プレイリストファイルの EXT-X-キー メタデータタグの URI 属性で参照されます。例:

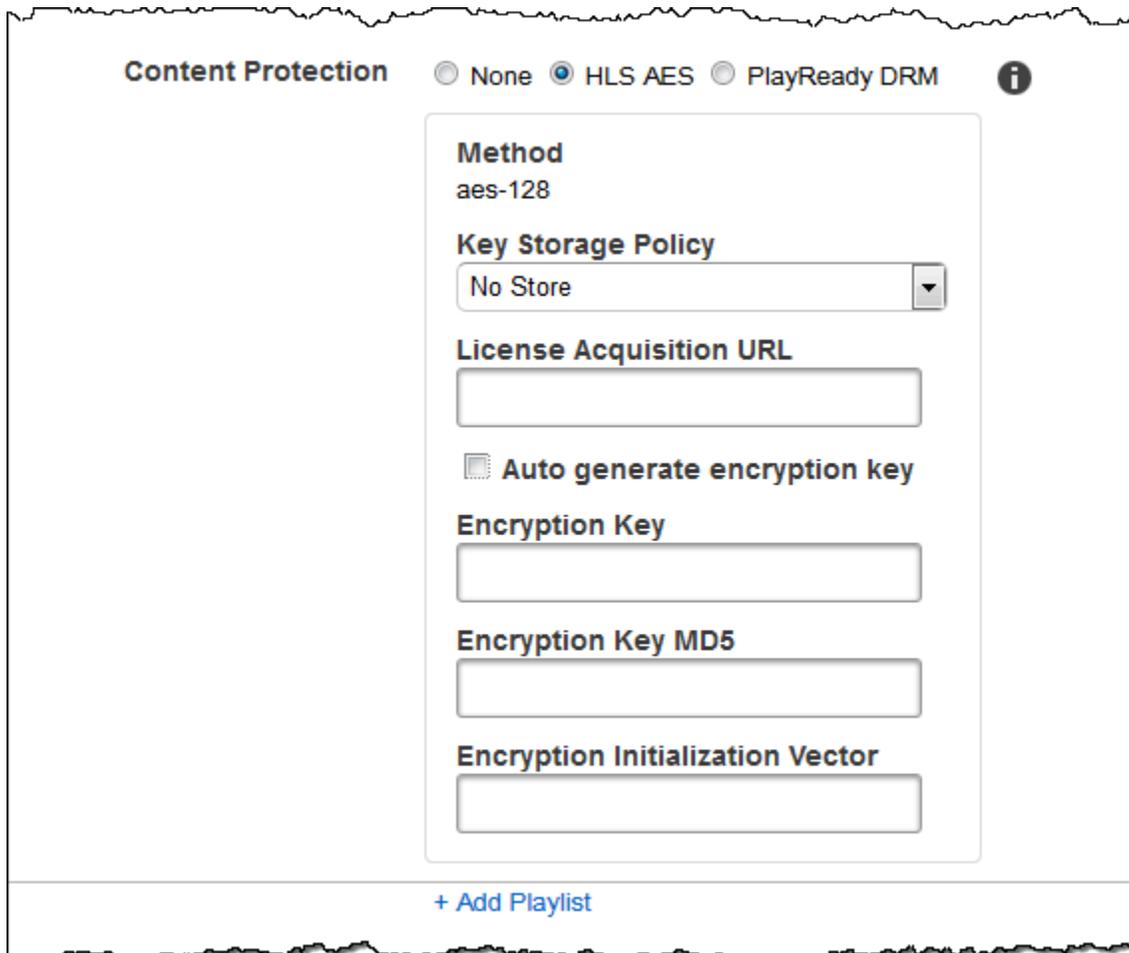
```
https://www.example.com/exampleKey/
```

## キーストレージポリシー

Elastic Transcoder で HLS ライセンスキーが Amazon S3 バケットに書き込まれるようにするかどうかを指定します。[WithVariantPlaylists] を選択すると、Elastic Transcoder により、関連付けられたプレイリストと同じ Amazon S3 バケットに暗号化されたキーが書き込まれます。

### Important

[NoStore] を選択した場合、Elastic Transcoder によりキーが保存されません。キーの保存先となる ライセンス取得 URL をユーザーに与えた時点で、キーの保存とユーザーへのキーの提供に関して責任を負うことになります。



**Content Protection**  None  HLS AES  PlayReady DRM i

**Method**  
aes-128

**Key Storage Policy**  
No Store

**License Acquisition URL**

Auto generate encryption key

**Encryption Key**

**Encryption Key MD5**

**Encryption Initialization Vector**

[+ Add Playlist](#)

## ( HLSv3 およびスムーズプレイリストのみ、オプション ) デジタル著作権管理

[プリセット] リストから、[コンテナ] の値が [fmp4] (フラグメント化された MP4) または [ts] (MPEG-TS) のいずれかで、[プレイリスト] のタイプが [HLSv3] または [Smooth] であるプリセットを選択した場合、ファイルに対して DRM 設定を指定できます。ファイルを保護するために DRM を使用することを選択した場合、このセクションにある設定を使用して、DRM 設定に関する情報を指定します。

DRM を使用するには、ライセンスを追跡および付与できるライセンスプロバイダサーバー、ユーザーを認証できるウェブサーバ、DRM 適用ファイルを暗号化するパッケージサーバー (Elastic Transcoder など)、メディアファイルを配信する方法 (Amazon CloudFront など)、および、DRM で保護されたメディアを再生する方法 (DRM 対応プレイヤーなど) が必要です。

**Note**

PlayReady DRM、HLS コンテンツ保護、および出力暗号化は、相互排他的なオプションです。

**(オプション) PlayReady DRM**

ファイルを視聴できるユーザーを制限するために使用する DRM 設定。これは、出力プレイリストに PlayReady DRM ヘッダーを含めることにより設定します。これは、アートワーク、キャプション、サムネイル、ウォーターマークには使用できません。PlayReady DRM では、aes-ctr 暗号化を使用してメディアファイルを暗号化します。

HLSv3 プレイリストに DRM を使用する場合、出力にはマスタープレイリストが必要です。

詳細については、「[デジタル著作権管理](#)」を参照してください。

**(DRM 保護に必須) DRM 形式**

出力プレイリストの DRM 形式。有効な形式は、discretix-3.0 と microsoft です。

Smooth タイプのプレイリストの場合は、microsoft を指定します。HLSv3 タイプのプレイリストの場合は、discretix-3.0 を指定します。

**(DRM 保護に必須) DRM キー**

DRM ライセンスプロバイダからそのファイル用として提供される DRM キー。このキーは base64 でエンコードされますが、base64 でエンコードされる前の時点で、ビット長が、

128、192、256

また、キーは を使用して暗号化する必要があります AWS KMS。詳細については、「AWS Key Management Service デベロッパーガイド」の「[データの暗号化と復号化](#)」を参照してください。

**(DRM 保護に必須) DRM キー ID**

DRM キーの ID。提供すべきキーを DRM ライセンスプロバイダが認識できるようにするものです。

キー ID はビッグエンディアンで指定する必要があります。Elastic Transcoder で、これはリトルエンディアンに変換された上で、PlayReady DRM ヘッダーに挿入されます。キー ID がライセン

サーバーからビッグエンディアンとリトルエンディアンのどちらで提供されているのかわからない場合は、DRM プロバイダに確認してください。

#### (DRM 保護に必須) DRM キー MD5

ファイルの DRM に使用するキーの MD5 ダイジェスト。Elastic Transcoder で、キーが転送中に破損していないことを確認するチェックサムとして使用されます。キー MD5 は base64 でエンコードされている必要があり、かつ base64 エンコードの前に正確に 16 バイトである必要があります。

#### (DRM 保護に必須) DRM 初期化ベクトル

ランダムビットジェネレーターによって作成された一連のランダムビット。各暗号化オペレーションに固有であり、Elastic Transcoder でファイルの暗号化に使用します。初期化ベクトルは base64 でエンコードされている必要があり、かつ base64 エンコードの前に正確に 8 バイト長である必要があります。初期化ベクトルが提供されていない場合は、Elastic Transcoder により生成されます。

詳細については、[「Initialization Vector」](#) (初期化ベクトル) を参照してください。

#### (DRM 保護に必須) DRM ライセンス取得 Url

DRM コンテンツの再生に必要なライセンスキーの場所。URL は絶対パスでなければならず、PlayReady ヘッダーによって参照されます。PlayReady ヘッダーは、スムーズストリーミング出力用のクライアントマニフェストの保護ヘッダー、および HLS プレイリスト出力用の EXT-X-DXDRM と EXT-XDXDRMINFO のメタデータタグで参照されます。URL の例は次のようになります:

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

**Content Protection**  None  HLS AES  PlayReady DRM ⓘ

**Format**  
Discretix 3.0

**License Acquisition URL**

**Key ID**

**Encryption Key**

**Encryption Key MD5**

**Encryption Initialization Vector**

[+ Add Playlist](#)

# プリセットの使用

## トピック

- [Elastic Transcoder でプリセットを作成する](#)
- [Elastic Transcoder でプリセットの一覧表示とプリセット設定の表示を行う](#)
- [Elastic Transcoder でプリセットに変更を加える](#)
- [Elastic Transcoder でプリセットを削除する](#)
- [Elastic Transcoder プリセットの作成時に指定する設定](#)

プリセットは、Elastic Transcoder によってトランスコーディングの処理に適用される設定を定義したテンプレートです。例えば、トランスコードしたファイルに必要なオーディオチャンネル数やビデオ解像度などを定義します。ジョブの作成時に、使用するプリセットを指定します。Elastic Transcoder には、多数の一般的な出力形式に対応したデフォルトプリセットがあります。その他の出力形式用にプリセットを追加することもできます。システムプリセットのリストについては、「[システムプリセット](#)」を参照してください。

## Elastic Transcoder でプリセットを作成する

プリセットは、AWS マネジメントコンソール または Elastic Transcoder の Create Preset API アクションを使用して作成できます。以下の手順では、コンソールを使用してプリセットを作成する方法について説明します。API を使用してプリセットを作成する方法については、「[プリセットを作成する](#)」を参照してください。

### Note

既存のプリセットを更新することはできません。プリセットの設定を変更する必要がある場合は、変更するプリセットに基づいて新しいプリセットを作成し、該当する値を更新して、新しいプリセットを保存します。

### Elastic Transcoder コンソールを使用してプリセットを作成する方法

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。

2. Elastic Transcoder コンソールのナビゲーションバーで、プリセットを作成するリージョンを選択します。
3. コンソールの左側のペインで、[Presets] (プリセット) をクリックします。
4. [Presets] (プリセット) ページで、[Create New Preset] (新しいプリセットを作成する) をクリックします。
5. 適切な値を入力します。各フィールドの詳細については「[Elastic Transcoder プリセットの作成時に指定する設定](#)」を参照してください。
6. [Create Preset] をクリックします。

## Elastic Transcoder でプリセットの一覧表示とプリセット設定の表示を行う

Elastic Transcoder に含まれているデフォルトのプリセットと、AWS リージョンに追加したプリセットを一覧表示するには、を使用する AWS マネジメントコンソール か、Elastic Transcoder List Presets API を使用します。また、個別のプリセットの設定を表示することもできます。以下の手順では、コンソールを使用してプリセットを一覧表示する方法とプリセットの設定を表示する方法について説明します。

API を使用してプリセットを一覧表示する方法については、「[プリセットのリスト](#)」を参照してください。API を使用してプリセットの設定を表示する方法については、「[プリセットを読み取る](#)」を参照してください。システムプリセットのリストについては、「[システムプリセット](#)」を参照してください。

### Elastic Transcoder コンソールを使用してプリセットを一覧表示する方法

1. にサインイン AWS マネジメントコンソール し、<https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. Elastic Transcoder コンソールのナビゲーションバーで、プリセットを一覧表示するリージョンを選択します。
3. コンソールの左側のペインで、[Presets] (プリセット) をクリックします。
4. 個々のプリセットの設定を表示するには、プリセットの横にある  アイコンをクリックします。

## Elastic Transcoder でプリセットに変更を加える

Elastic Transcoder では、既存のプリセットの設定を変更することはできません。この制限は、Elastic Transcoder に付属するデフォルトのプリセットにも、ユーザーが追加したプリセットにも適用されます。ただし、以下の手順で説明しているように、変更するプリセットのコピーを作成し、該当する設定を変更し、新しいプリセットを保存して、古いプリセットを削除することで、同じ結果を簡単に得られます。

### Elastic Transcoder コンソールを使用してプリセットに変更を加える方法

1. にサインイン AWS マネジメントコンソールし、<https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. Elastic Transcoder コンソールのナビゲーションバーで、変更するプリセットが含まれているリージョンを選択します。
3. コンソールの左側のペインで、[Presets] (プリセット) をクリックします。
4. 変更するプリセットのチェックボックスをオンにします。
5. [コピー] をクリックします。
6. 変更するプリセットのコピーで、該当する値を変更します。各フィールドの詳細については「[Elastic Transcoder プリセットの作成時に指定する設定](#)」を参照してください。
7. [Create Preset] をクリックします。
8. [Presets] (プリセット) ページに戻り、プリセットの古いバージョンのチェックボックスをオンにします。
9. [Remove] をクリックします。

## Elastic Transcoder でプリセットを削除する

AWS マネジメントコンソール を使用するか、Elastic Transcoder Delete Preset API を使用することで、AWS リージョンに追加したプリセットを削除できます。以下の手順では、コンソールを使用してプリセットを削除する方法について説明します。API を使用してプリセットを削除する方法については、「[プリセットを削除する](#)」を参照してください。

### Note

Elastic Transcoder に付属するシステムのプリセットを削除することはできません。

## Elastic Transcoder コンソールを使用してプリセットを削除する方法

1. <https://console.aws.amazon.com/elasticbeanstalk/> で AWS マネジメントコンソールにサインインして Elastic Transcoder コンソールを開きます。
2. Elastic Transcoder コンソールのナビゲーションバーで、削除するプリセットが含まれているリージョンを選択します。
3. コンソールの左側のペインで、[Presets] (プリセット) をクリックします。
4. 削除するプリセットのチェックボックスをオンにします。
5. [Remove] をクリックします。

## Elastic Transcoder プリセットの作成時に指定する設定

プリセットを作成するときは、以下の値を指定します。

### トピック

- [全般設定](#)
- [ビデオ設定](#)
- [ウォーターマークの設定](#)
- [オーディオ設定](#)
- [サムネイルの設定](#)

## 全般設定

### プリセットから始める

新しいプリセットの開始点として使用するプリセット。Elastic Transcoder では、新しいプリセットのフィールドに、選択したプリセットから値がコピーされます。Elastic Transcoder のデフォルトプリセットのいずれかを選択するか、独自に作成したプリセットを選択できます。

### 名前

プリセットの名前。この名前は AWS アカウント内で一意の名前にすることをお勧めしますが、一意にすることは強制ではありません。最大長は 40 文字です。

### 説明

プリセットの説明。最大長は 255 文字です。

## コンテナ

出力ファイルのコンテナタイプ。有効な値

は、flac、flv、fmp4、gif、mp2、mp3、mp4、mpg、mxf、oga、ogg、ts、wav、webm です。以下の表は、サポートされるコンテナコーデックの一覧です。

コンテナ	オーディオコーデック	ビデオコーデック
FLAC	flac	サポートされていません
FLV	AAC、mp3	H.264
FMP4	AAC	H.264
GIF	サポートされていません	gif
MP3	mp3	サポートされていません
MP4	AAC、mp3	H.264
MPG	mp2	mpeg2
MXF	pcm	mpeg2
OGA	flac、vorbis	サポートされていません
OGG	flac、vorbis	サポートされていません
TS	AAC、mp3	H.264
WAV	pcm	サポートされていません
WEBM	vorbis	vp8、vp9

?

## Create a New Transcoding Preset

---

A preset is a template that contains the settings that you want the 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.

**Start With Preset**  ⓘ

**Name**  ⓘ

**Description**  ⓘ

**Container**  ⓘ

---

### Available Settings

Available Settings  Video  Audio  Thumbnails ⓘ

## ビデオ設定

### Codec

出力ファイルのビデオコーデック。有効な値は、gif、H.264、mpeg2、vp8、vp9 です。以下は、コンテナとビデオコーデックの組み合わせの表です。

コンテナ	GIF	H.264	MPEG2	VP8	VP9
FLAC					
FLV		x			
GIF	x				

コンテナ	GIF	H.264	MPEG2	VP8	VP9
FMP4		x			
MP3					
MP4		x			
MPG			x		
MXF			x		
OGA					
OGG					
TS		x			
WEBM				x	x

H.264 ビデオ圧縮形式の詳細については、Wikipedia の [H.264/MPEG-4 AVC](#) に関するページを参照してください。

VP8 の詳細については、「[VP8](#)」を参照してください。VP9 の詳細については、「[VP9](#)」を参照してください。

#### (H.264/VP8 のみ) プロファイル

出力動画に使用するプロファイル。ビデオコーデックが [H.264] の場合、Elastic Transcoder では以下のプロファイルがサポートされています。

- baseline: このプロファイルは一般的に、ビデオ会議とモバイルアプリケーションに使用されます。
- main: このプロファイルは標準精細度デジタルテレビ放送に使用されます。
- high: このプロファイルは高精細度デジタルテレビ放送と Blu-ray ディスクに使用されます。

H.264 プロファイルの詳細については、Wikipedia のエントリ「H.264/MPEG-4 AVC」の「[プロファイル](#)」を参照してください。

ビデオコーデックが [VP8] の場合、Elastic Transcoder では 0、1、2、3 の値がサポートされています。

## (H.264 のみ) レベル

出力動画に使用する H.264 レベル。該当する値を選択します。

レベルの詳細については、Wikipedia のエントリ「H.264/MPEG-4 AVC」の「[レベル](#)」を参照してください。

## (H.264 のみ) リファレンスフレームの最大数

以前にデコードされたフレームの最大数。以降にフレームのデコーディングの基準として使用されます。入力した値が、[\[Max Height\]](#)、[\[Max Height\]](#)、[\[Level\]](#) に指定した値に基づく推奨値よりも大きい場合、推奨値を含むメッセージが Elastic Transcoder によって表示されます。Elastic Transcoder が実行する計算など、詳細については、「[プリセットを作成する](#)」トピックの「MaxReferenceFrames」を参照してください。

## (オプション。H.264/MPEG2/VP8/VP9 のみ) 最大ビットレート

出力動画の 1 秒あたりの最大キロビット数。16~62,500 のいずれかの値を指定します。

[ビットレート] で [auto] (自動) を指定すると、Elastic Transcoder で、出力ビデオの平均ビットレートとして入力ビデオのビットレートが使用されます。[Maximum Bit Rate] で、出力動画の最大ビットレートを設定できます。この設定は、ターゲットデバイスでサポートされている最大ビットレートが入力動画のビットレートよりも低い場合に便利です。最大ビットレートを下げると、動画の品質が低下することがあります。

## (オプション。H.264/MPEG2/VP8/VP9 のみ) バッファサイズ

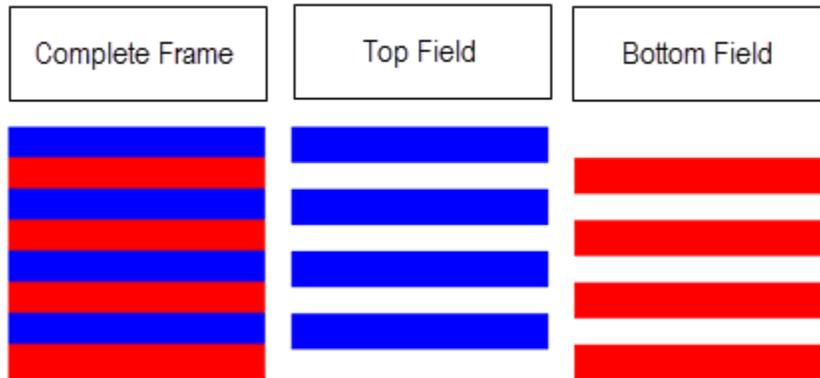
出力動画の x 秒あたりの最大キロビット数。この時間は一般的に 10 秒 (出力動画のコンテナタイプに MPEG-TS を使用しているときの標準的なセグメント再生時間) です。0 より大きい整数を指定します。[最大ビットレート] を指定し、[バッファサイズ] を省略した場合、Elastic Transcoder によって [バッファサイズ] が [最大ビットレート] の値の 10 倍に設定されます。

## (オプション。H.264/MPEG2 のみ) インターレースモード

出力ビデオのインターレースモード。

インターレースビデオでは、肉眼で 1 つのフレームに対して複数のイメージが残るように、2 つのフィールド (1 つは線 1 本おきにあるフィールド、もう 1 つは他の線にあるフィールド) をインターレースすることで、ビデオの知覚フレームレートを 2 倍にすることができます。インターレースを使用すると、ビデオの送信に必要な帯域幅が減りますが、イメージのぼやけやちらつきが発生する可能性があります。

2 つの行セットはフィールドと呼ばれ、インターレースフレームでは 2 つのイメージがフィールドに分割されます。



有効な値は、[Progressive] (インターレースなし。上から下へ)、[Top First] (上フィールドが最初)、[Bottom First] (下フィールドが最初)、[Auto] (自動) です。

[InterlaceMode] が指定されていない場合、Elastic Transcoder では出力に [Progressive] が使用されます。[Auto] を指定すると、Elastic Transcoder により出力のインターレースが行われます。

詳細については、Wikipedia の「[インターレース](#)」のページを参照してください。

(オプション。H.264/MPEG2 のみ) 色空間変換モード

Elastic Transcoder で出力ビデオに適用される色空間変換。色空間とは、色のレンダリング方法に関する情報を保存するためにコンピュータで使用されるアルゴリズムです。Bt.601 は標準解像度ビデオ向けの規格ですが、Bt.709 は高解像度ビデオ向けの規格です。

有効な値は、[なし]、[Bt709toBt601]、[Bt601toBt709]、[Auto] (自動) です。

[ColorSpaceConversionMode] の [Auto] (自動) を選択し、出力がインターレースされる場合、フレームレートは、23.97、24、25、29.97、50、60 のいずれかになり、SegmentDuration null になり、以下のグラフから解像度変更のうちの 1 つを使用することになり、Elastic Transcoder により以下の色空間変換が適用されます。

入力ファイル	入力解像度	出力解像度	適用される色空間変換
標準から HD へ	720x480	1920x1080	Bt601ToBt709
標準から HD へ	720x576	1920x1080	Bt601ToBt709

入力ファイル	入力解像度	出力解像度	適用される色空間変換
HD から標準へ	1920x1080	720x576	Bt709ToBt601
HD から標準へ	1920x1080	720x480	Bt709ToBt601

### Note

Elastic Transcoder では今後、ColorSpaceConversionMode Auto モードの動作が変更される可能性があります。プレイリスト内のすべての出力で同一のColorSpaceConversionMode を使用する必要があります。

ColorSpaceConversionMode を指定しない場合、Elastic Transcoder でファイルの色空間は変更されません。

出力ファイルにどの ColorSpaceConversionMode が適用されているか分からない場合は、ジョブレスポンスに含まれている AppliedColorSpaceConversion パラメータを確認してください。ジョブのレスポンスに AppliedColorSpaceConversion がない場合、ColorSpaceConversionMode が適用されていません。

色空間の詳細については、Wikipedia の「[色空間](#)」のページを参照してください。Bt.601 と Bt.709 の詳細については、Wikipedia の「[Rec. 601](#)」と「[Rec. 709](#)」。

### (MPEG2 のみ) クロマサブサンプリング

出力ビデオのクロマ (カラー) チャンネルのサンプリングパターン。有効な値は、[yuv420p] と [yuv422p] です。

yuv420p では水平線と垂直線のそれぞれ 1 本おきにクロマ情報がサンプリングされ、yuv422p では各水平線と水曲線 1 本おきにクロマ情報がサンプリングされます。

クロマサブサンプリングの詳細については、Wikipedia の「[クロマ・サブサンプリング](#)」のページを参照してください。

### (Gif のみ) ループカウント

出力 gif をループさせる回数。

有効な値は、[Infinite] (無限大) と、0 から 100 までの整数です。

## (H.264/MPEG2/VP8 のみ) キーフレーム間の最大フレーム数

キーフレーム間のフレームの最大数。[gif] 形式のコンテナには適用されません。キーフレームは、完全にエンコードされたフレームです。キーフレーム間のフレームは、2つのキーフレーム間の差異に基づいてエンコードされます。値は文字列として書式設定された整数です。有効な値は 1 (すべてのフレームがキーフレーム) ~ 100,000 のいずれかになります。値を大きくすると圧縮率は高くなりますが、動画品質が目に見えて低下する場合があります。

Smooth 出力では、[Frame Rate] を [Maximum Number of Frames Between Keyframes] に対して一定比率にする必要があります。これにより、ファイルの再生中に Smooth プレイリストを、異なる品質レベルに切り替えることができます。

たとえば、入力ファイルの [Frame Rate] が 30 で、[Maximum Number of Frames Between Keyframes] が 90 である場合があります。出力ファイルでは、比率は 1:3 になる必要があります。有効な出力では、[Frame Rates] が 30、25、10 の場合、[Maximum Number of Frames Between Keyframes] はそれぞれ 90、75、30 になります。

または、[Frame Rate] を [auto] に設定し、[Video Max Frame Rate] と [Maximum Number of Frames Between Keyframes] を同じ値にすることによってこれを実現できます。

キーフレームの詳細については、Wikipedia のエントリ「[動画圧縮のピクチャタイプ](#)」を参照してください。

## (H.264/MPEG2/VP8 のみ) キーフレーム間の固定フレーム数

[\[Fixed Number of Frames Between Keyframes\]](#) の固定値を使用するかどうか。

- [はい]: Elastic Transcoder で、キーフレーム間の距離 (1つのピクチャグループ (GOP、a group of pictures) 内のフレームの数) に [Maximum Number of Frames Between Keyframes](#) の値が使用されます。
- No: キーフレーム間の距離は可変になります。

### Note

[Fixed Number of Frames Between Keyframes] (キーフレーム間の固定フレーム数) は、fmp4 コンテナに対して true に設定する必要があります。

## Bit Rate

出力動画の動画ストリームのビットレート (キロビット/秒単位)。可変ビットレートまたは固定ビットレートのエンコーディングを設定できます。

- [Variable bit rate encoding] (可変ビットレートエンコーディング): [auto] (自動) を指定します。Elastic Transcoder によりビットレートが最適化され、出力のフレームごとに一貫した画質が維持されます。
- Constant bit rate encoding: ビットレートを指定します。

動画ビットレートの有効な値は、[Codec] で選択した値によって異なります。

- H.264: 有効な値は [Level] と [Profile] の値によって異なります。レベルとプロファイルの値としては、以下の表に示している H.264 に準拠した最大値以下の値を指定することをお勧めします。
- VP8: 以下の表は使用しないでください。[Level] が適用されるのは、ビデオコーデックが H.264 である場合のみです。

<u>Level</u>	最大動画ビットレート (キロビット/秒) : Baseline <u>Profile</u> と Main <u>Profile</u>	最大動画ビットレート (キロビット/秒) : High <u>Profile</u>
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

## フレームレート

出力動画の動画ストリームの 1 秒あたりのフレーム数。該当する値を選択します。

[auto] (自動) を指定した場合、Elastic Transcoder で、検出された入力ソースフレームレートが使用されます。フレームレートを指定した場合は、以下の計算を実行することをお勧めします。

フレームレート = 最大推奨デコーディング速度 (輝度サンプル/秒) / 動画最大幅 \* 動画最大高さ

ここで、フレームレート = 最大推奨デコーディング速度 (輝度サンプル/秒) / 動画最大幅 \* 動画最大高さは、[\[Level\]](#) に指定した値に基づく最大値 (次の表を参照) 以下の値になります。

### Note

入力した値が、[\[Max Width\]](#)、[\[Max Height\]](#)、[\[Level\]](#) に指定した値に基づく推奨値よりも大きい場合、推奨値を含むメッセージが Elastic Transcoder によって表示されます。

Smooth 出力では、[\[Frame Rate\]](#) を [\[Maximum Number of Frames Between Keyframes\]](#) に対して一定比率にする必要があります。例えば、入力ファイルで [\[Frame Rate\]](#) (フレームレート) に 30、[\[Maximum Number of Frames Between Keyframes\]](#) (キーフレーム間の最大フレーム数) に 90 を指定した場合、出力の [\[Frame Rate\]](#) (フレームレート) と [\[Maximum Number of Frames Between Keyframes\]](#) (キーフレーム間の最大フレーム数) の比率も 1:3 にする必要があります。

<a href="#">Level</a>	最大推奨のデコーディング速度 (輝度サンプル/秒)
1	380160
1b	380160
1.1	768000
1.2	1536000
1.3	3041280
2	3041280
2.1	5068800
2.2	5184000

Level	最大推奨のデコーディング速度 (輝度サンプル/秒)
3	10368000
3.1	27648000
3.2	55296000
4	62914560
4.1	62914560

## ビデオの最大フレームレート

[Frame Rate] (フレームレート) で auto を指定した場合、Elastic Transcoder では、出力ビデオのフレームレートに入力ビデオのフレームレートが使用されます (最大フレームレートまで)。[Video Max Frame Rate] (ビデオの最大フレームレート) を指定していない場合、Elastic Transcoder によってデフォルトの 30 が使用されます。

入力ビデオのフレームレートが出力ビデオの必要な最大フレームレートまたはデフォルトの最大フレームレートより大きいときに Elastic Transcoder によって使用される最大フレームレートを指定します。

Elastic Transcoder では、次の両方の条件を満たし、サポートされている最大のフレームレートが使用されます。

- フレームレートは、最大フレームレート以下である。
- フレームレートは、入力フレームレートを余りが出ないように均等に分割したものである。

例えば、入力ファイルのフレームレートが 50 で、[Video Max Frame Rate] (ビデオの最大フレームレート) の値として 30 を指定している場合、Elastic Transcoder が生成する出力ビデオのフレームレートは 25 フレーム/秒になります。これは、25 が 30 よりも小さく、50 を 25 で割ると 2 で割り切れるからです。

## Max Width

出力動画の最大幅 (ピクセル単位)。[auto] (自動) を指定した場合、Elastic Transcoder で、1920 (フル HD) がデフォルト値として使用されます。数値を指定する場合は、128 ~ 4096 のいずれかの偶数を入力します。

詳細については、「[Max Height](#)」を参照してください。

## Max Height

出力動画の最大高さ (ピクセル単位)。[auto] (自動) を指定した場合、Elastic Transcoder で、1080 (フル HD) がデフォルト値として使用されます。数値を指定する場合は、96~3072 のいずれかの偶数を入力します。

[Codec](#) の H.264 を指定した場合は、[Max Width] (最大幅) と [Max Height] (最大高さ) の値を指定することをお勧めします。これらの 2 つの値の積が、以下の表の該当する値以下になるようにするためです。

### Note

[Codec](#) の VP8 を指定した場合、以下の表は使用しないでください。[Level] (レベル) が適用されるのは、ビデオコーデックが H.264 である場合のみです。

値として個格納できません。 <a href="#">Level</a>	[Max Width] x [Max Height] の最大値
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

値として個格納できません。 <a href="#">Level</a>	[Max Width] x [Max Height] の最大値
4	2097152
4.1	2097152

## Sizing Policy

出力動画の拡大/縮小を制御するには、以下のいずれかの値を指定します。

- [Fit]: Elastic Transcoder で、[Max Width] (最大幅) または [Max Height] (最大高さ) で指定した値に一致し、かつ他方の値を超えないように、出力ビデオのスケールリングが行われます。
- [Fill]: Elastic Transcoder で、[Max Width] (最大幅) または [Max Height] (最大高さ) で指定した値に一致し、かつ他方の値に一致するかこれを超えるように、出力ビデオのスケールリングが行われます。Elastic Transcoder では、出力ビデオが中央に配置され、次に、最大値を超えるディメンション (該当する場合) がトリミングされます。
- [Stretch]: Elastic Transcoder では、[Max Width] (最大幅) と [Max Height] (最大高さ) に指定した値に一致するように、出力ビデオを引き伸ばします。入力動画と出力動画でアスペクト比が異なる場合、出力動画は歪みます。
- [Keep]: Elastic Transcoder で出力ビデオのスケールリングが行われません。入力ビデオのいずれかのディメンションが、[Max Width] (最大幅) と [Max Height] (最大高さ) に指定した値を超える場合、Elastic Transcoder により出力ビデオがトリミングされます。
- [ShrinkToFit]: Elastic Transcoder では、出力ビデオのディメンションが [Max Width] (最大幅) と [Max Height] (最大高さ) に指定した少なくともどちらか 1 つの値に一致し、かついずれの値も超えないように、出力ビデオが縮小されます。このオプションを指定すると、Elastic Transcoder で出力ビデオは拡大されません。
- [ShrinkToFill]: Elastic Transcoder では、出力ビデオのディメンションが [Max Width] (最大幅) と [Max Height] (最大高さ) に指定した少なくともどちらか 1 つの値に一致し、かつこれらの値を下回らないように、出力ビデオを縮小します。このオプションを指定すると、Elastic Transcoder で出力ビデオは拡大されません。

以下の表には、[Sizing Policy] の設定が出力動画に与える可能性のある影響を示しています。

SizingPolicy	出力動画が拡大される可能性がある	出力動画に余白が追加される可能性がある ([Padding Policy] が [Pad] の場合)	出力動画のピクセルアスペクト比が入力動画のものと異なる可能性がある	出力動画がクロップされる可能性がある
Fit	あり	あり		
Fill	あり			あり
[Stretch] (ストレッチ)	あり		あり	
Keep	あり	あり		あり
ShrinkToFit に設定します。		はい		
ShrinkToFill		あり		あり

## パディングポリシー

[PaddingPolicy] を [Pad] に設定すると、Elastic Transcoder によって出力ビデオの上下/左右に黒いバーが追加されることがあります。これは、指定した [Max Width] (最大幅) と [Max Height] (最大高さ) の値に出力ビデオの合計サイズを一致させるためです。詳細については、「[Sizing Policy](#)」の表を参照してください。

## アスペクト比の表示

Elastic Transcoder によって出力ビデオのメタデータに追加される値。[Display Aspect Ratio] (アスペクト比の表示) を [auto] (自動) に設定した場合、Elastic Transcoder によって、平方ピクセルになるアスペクト比が選択されます。別のオプションを選択すると、そのオプションで指定した値が Elastic Transcoder によって出力ビデオに設定されます。

**Video**

<b>Codec</b>	<input type="text" value="H.264"/>	<input type="button" value="v"/>	<input type="button" value="i"/>
<b>Profile</b>	<input type="text" value="high"/>	<input type="button" value="v"/>	<input type="button" value="i"/>
<b>Level</b>	<input type="text" value="3.1"/>	<input type="button" value="v"/>	<input type="button" value="i"/>
<b>Maximum Number of Reference Frames</b>	<input type="text" value="3"/>	<input type="button" value="v"/>	<input type="button" value="i"/>
<b>Max Bit Rate</b>	<input type="text" value="2056"/>		<input type="button" value="i"/>
<b>Buffer Size</b>	<input type="text" value="12336"/>		<input type="button" value="i"/>
<b>Interlaced Mode</b>	<input type="text" value="Progressive"/>	<input type="button" value="v"/>	<input type="button" value="i"/>
<b>Color Space Conversion Mode</b>	<input type="text" value="None"/>	<input type="button" value="v"/>	<input type="button" value="i"/>
<b>Maximum Number of Frames Between Keyframes</b>	<input type="text" value="60"/>		<input type="button" value="i"/>

<b>Maximum Number of Frames Between Keyframes</b>	<input type="text" value="15"/>	<b>i</b>
<b>Fixed Number of Frames Between Keyframes</b>	<input checked="" type="radio"/> Yes <b>i</b> <input type="radio"/> No	
<b>Bit Rate</b>	<input type="text" value="10000"/>	<b>i</b>
<b>Frame Rate</b>	<input type="text" value="29.97"/>	<b>i</b>
<b>Max Width</b>	<input type="text" value="1920"/>	<b>i</b>
<b>Max Height</b>	<input type="text" value="1080"/>	<b>i</b>
<b>Sizing Policy</b>	<input type="text" value="ShrinkToFill"/>	<b>i</b>
<b>Padding Policy</b>	<input type="text" value="Pad"/>	<b>i</b>
<b>Display Aspect Ratio</b>	<input type="text" value="16:9"/>	<b>i</b>

The image shows two screenshots of the Amazon Elastic Transcoder console, specifically the video settings section. The top screenshot shows settings for the mpeg2 codec, and the bottom screenshot shows settings for the gif codec.

**Video**

Codec	mpeg2	i
Max Bit Rate	6500	i
Buffer Size	65000	i
Interlaced Mode	TopFirst	i
Color Space Conversion Mode	Auto	i
Chroma Subsampling	yuv420p	i
Maximum Number of Frames Between Keyframes	12	i

**Video**

Codec	gif	i
Loop Count	Infinite	i

## ウォーターマークの設定

このセクションの設定を使用して、Elastic Transcoder でこのプリセットを使用してトランスコードされるビデオに重ねるグラフィックのサイズ、位置、スケール、不透明度を指定します。最大4つのウォーターマーク用に設定を指定できます。ウォーターマークは、トランスコードした動画の再生時間にわたって表示されます。

ウォーターマークには .png または .jpg 形式を使用できます。四角形以外のウォーターマークを表示する場合は、透明度がサポートされている .png 形式を使用します。

このプリセットを使用するジョブを作成するときに、Elastic Transcoder においてトランスコードされたビデオに含める .png または .jpg グラフィックを指定します。Elastic Transcoder では、対応するプリセットで指定したのと同数のウォーターマークを各ジョブ出力に指定する必要はありません。たとえば、プリセットで 4 つのウォーターマークの設定を指定し、ジョブ出力で 1 つのウォーターマークのみを指定することもできます。

グラフィックが歪まないようにウォーターマークを設定するには、[サイジングポリシー] を [Shrink to Fit] に設定し、[Maximum Width] (最大幅) と [Maximum Height] (最大高さ) の値を同じパーセンテージに設定します。グラフィックを元のサイズのまま表示する場合は、[Maximum Width] (最大幅) と [Maximum Height] (最大の高さ) を 100% に設定します。

詳細については、「[ウォーターマーク](#)」を参照してください。

## ID

1 つのウォーターマークの設定の一意の識別子。[Id] の値の最大長は 40 文字です。

## 最大幅

ウォーターマークの最大幅は以下のいずれかの形式になります。

- ##### px: 最小値は 16 ピクセルであり、最大値は [MaxHeight] の値です。
- #####%: 有効な値の範囲は 0~100 です。[Target] (ターゲット) の値を使用して、Elastic Transcoder により追加された黒いバーがある場合に Elastic Transcoder で計算に含めるかどうかを指定します。

値をピクセル単位で指定する場合は、[MaxHeight] の値以下にする必要があります。

## 最大高さ

ウォーターマークの最大高さは以下のいずれかの形式になります。

- ##### px: 最小値は 16 ピクセルであり、最大値は [MaxHeight] の値です。
- #####%: 有効な値の範囲は 0~100 です。[Target] (ターゲット) の値を使用して、Elastic Transcoder により追加された黒いバーがある場合に Elastic Transcoder で計算に含めるかどうかを指定します。

値をピクセル単位で指定する場合は、[MaxHeight] の値以下にする必要があります。

## サイジングポリシー

ウォーターマークの拡大/縮小を制御するには、以下のいずれかの値を指定します。

- [Fit]: Elastic Transcoder で、[Maximum Width] (最大幅) または [Maximum Height] (最大高さ) で指定した値に一致し、かつ他方の値を超えないように、ウォーターマークのスケールが行われます。
- [Stretch]: Elastic Transcoder では、[Maximum Width] (最大幅) と [Maximum Height] (最大高さ) に指定した値に一致するように、ウォーターマークを引き伸ばします。ウォーターマークと、[Maximum Width] および [Maximum Height] でアスペクト比が異なる場合、ウォーターマークは歪みます。
- [Shrink To Fit]: Elastic Transcoder では、ウォーターマークのディメンションが [Maximum Width] (最大幅) と [Maximum Height] (最大高さ) に指定した少なくともどちらか 1 つの値に一致し、かついずれの値も超えないように、ウォーターマークが縮小されます。このオプションを指定すると、Elastic Transcoder でウォーターマークは拡大されません。

### [Horizontal Alignment] (水平方向の配置)

ウォーターマークの水平位置。左端または右端に相対的にウォーターマークを配置するには、[Horizontal Offset] (水平オフセット) にゼロ以外の値を指定します。

- Left: ウォーターマークの左端が動画の左端に揃えられます。
- Right: ウォーターマークの右端が動画の右端に揃えられます。
- Center: ウォーターマークは動画の左端と右端の中央に配置されます。

### 水平オフセット

ウォーターマークの水平方向の位置を [Horizontal Alignment] で指定した位置からどれくらいオフセットさせるかを指定します。

- ##### px: 最小値は 0 ピクセルであり、最大値は [MaxHeight] の値です。
- ##### %: 有効な値の範囲は 0~100 です。

例えば、[Horizontal Alignment] (水平配置) で [左] を指定し、[Horizontal Offset] (水平オフセット) で [5px] を指定すると、ウォーターマークの左側が、出力ビデオの左境界から 5 ピクセル分表示されます。

[HorizontalOffset] が有効なのは、[Horizontal Alignment] が [Left] または [Right] のときのみです。

オフセットの指定によりウォーターマークの表示範囲がビデオの左端または右端を越えている場合、Elastic Transcoder によって黒いバーが追加されていないと、ウォーターマークはトリミングされます。Elastic Transcoder によって黒いバーが追加されていると、ウォーターマークの表

示範囲はその黒いバー内にも及びます。ウォーターマークの表示範囲が黒いバーを越えている場合、ウォーターマークはクロップされます。

[Target] (ターゲット) の値を使用して、Elastic Transcoder により追加された黒いバーがある場合に Elastic Transcoder でオフセット計算に含めるかどうかを指定します。

## 垂直配置

ウォーターマークの垂直位置。上端または下端に相対的にウォーターマークを配置するには、[Vertical Offset] (垂直オフセット) にもゼロ以外の値を指定します。

- Top: ウォーターマークの上端が動画の上端に揃えられます。
- Bottom: ウォーターマークの下端が動画の下端に揃えられます。
- Center: ウォーターマークは動画の上端と下端の中央に配置されます。

## 垂直オフセット

ウォーターマークの垂直方向の位置を [Vertical Alignment] で指定した位置からどれくらいオフセットさせるかを指定します。

- ##### px: 最小値は 0 ピクセルであり、最大値は [Maximum Height] の値です。
- #####%: 有効な値の範囲は 0~100 です。

例えば、[Vertical Alignment] (垂直配置) で [上] を指定し、[Vertical Offset] (垂直オフセット) で [5px] を指定すると、ウォーターマークの上部が、出力ビデオの上部境界から 5 ピクセル分表示されます。

[Vertical Offset] が有効なのは、[Vertical Alignment] が [Top] または [Bottom] のときのみです。

オフセットの指定によりウォーターマークの表示範囲がビデオの上端または下端を越えている場合、Elastic Transcoder によって黒いバーが追加されていないと、ウォーターマークはトリミングされます。Elastic Transcoder によって黒いバーが追加されていると、ウォーターマークの表示範囲はその黒いバー内にも及びます。ウォーターマークの表示範囲が黒いバーを越えている場合、ウォーターマークはクロップされます。

[Target] (ターゲット) の値を使用して、Elastic Transcoder により追加された黒いバーがある場合に Elastic Transcoder でオフセット計算に含めるかどうかを指定します。

## [Opacity] (不透明度)

動画でウォーターマークが表示される位置でそのウォーターマークをどれくらい不透明にするかを示すパーセンテージ。有効な値は 0 (ウォーターマークは完全に透明) から 100 (ウォーターマークは完全に不透明) です。

Elastic Transcoder はで透明の .png グラフィックがサポートされています。透過 .png を使用した場合は、[Opacity] の値として 0 を指定したかのように、ウォーターマークから動画は透けて表示されます。.jpg ファイル形式では、透明度はサポートされていません。

## [Target] (ターゲット)

ウォーターマーク設定の [Horizontal Offset] (水平オフセット)、[Vertical Offset] (垂直オフセット)、[Maximum Width] (最大幅)、[Maximum Height] (最大高さ) で指定した値が Elastic Transcoder でどのように解釈されるかを決定する値。

- [Content] (コンテンツ): [Horizontal Offset] (水平オフセット) の値と [Vertical Offset] (垂直オフセット) の値がビデオの境界線に基づいて計算されます (Elastic Transcoder によって追加された黒いバーはあっても除外されます)。

さらに、[Maximum Width] (最大幅) と [Maximum Height] (最大高さ) は、パーセントで指定した場合、ビデオの境界線に基づいて計算されます (Elastic Transcoder によって追加された黒いバーはあっても除外されます)。

- [Frame] (フレーム): [Horizontal Offset] (水平オフセット) の値と [Vertical Offset] (垂直オフセット) の値がビデオの境界線に基づいて計算されます (Elastic Transcoder によって追加された黒いバーがあれば包含されます)。

さらに、[Maximum Width] (最大幅) と [Maximum Height] (最大高さ) は、パーセントで指定した場合、ビデオの境界線に基づいて計算されます (Elastic Transcoder によって追加された黒いバーがあれば包含されます)。

## Watermarks

[+ Add Watermark](#)

**Id**  ⓘ

**Max Width**  ⓘ

**MaxHeight**  ⓘ

**Sizing Policy**  ⓘ

**Horizontal Align**  ⓘ

**Horizontal Offset**  ⓘ

**Vertical Align**  ⓘ

**Vertical Offset**  ⓘ

**Opacity**  ⓘ

**Target**  ⓘ

[- Remove](#)

## オーディオ設定

### Codec

出力ファイルのオーディオコーデック。有効な値は、AAC、flac、mp2、mp3、pcm、vorbis です。以下は、コンテナとビデオコーデックの有効な組み合わせの表です。

コンテナ	AAC	FLAC	MP2	MP3	PCM	Vorbis
FLAC		x				
FLV	x			x		
GIF						
FMP4	x					
MP3				x		
MP4	x			x		
MPG			x			
MXF					x	
OGA		x				x
OGG		x				x
TS	x			x		
WAV					x	
WEBM						x

### (AAC のみ) プロファイル

[Audio:Codec] で AAC を指定した場合、出力ファイルに対して AAC プロファイルを選択します。Elastic Transcoder では、以下のプロファイルがサポートされています。

- [auto] (自動): [auto] (自動) を指定すると、出力ファイル用に選択したビットレートに基づいて Elastic Transcoder によりプロファイルが選択されます。
- AAC-LC: 最も一般的な AAC プロファイル。ビットレートが 64 kbps より高い場合に使用します。詳細については、「[Advanced Audio Coding](#)」を参照してください。
- HE-AAC: 旧式のプレーヤーやデバイスによってはサポートされていない場合があります。ビットレートが 40 ~ 80 kbps の場合に使用します。詳細については、「[High-Efficiency Advanced Audio Coding](#)」を参照してください。
- HE-AACv2: 旧式のプレーヤーやデバイスによってはサポートされていない場合があります。ビットレートが 48 kbps 未満の場合に使用します。詳細については、「[High-Efficiency Advanced Audio Coding](#)」を参照してください。

[Smooth] プレイリストのすべての出力は、[Profile] に対して同じ値である必要があります。

 Note

AAC プロファイルの追加前に任意のプリセットを作成した場合、それらのプリセットには Elastic Transcoder によって AAC-LC プロファイルが使用されます。

AAC の詳細については、Wikipedia のエントリ「MPEG-4 Part 3」の「[オーディオプロファイル](#)」を参照してください。

(オプション。FLAC/PCM のみ) ビット深度

サンプルのビット深度とは、オーディオサンプルに含まれる情報のビット数です。ビット深度が高いほど音質が良くなりますが、ファイルは大きくなります。

FLAC コーデックの有効な値は、16 と 24 です。

PCM コーデックの有効な値は、8、16、24、32 です。

(オプション、PCM のみ) PCM ビット署名

オーディオサンプルを負の数と正の数 (符号付き) で表現するか、または正の数のみで表現します (符号なし)。

有効な値は、[Signed] (署名済み) と [Unsigned] (未署名) です。

最も一般的な値は [Signed] (署名済み) です。

(オプション。PCM のみ) PCM ビット順序

PCM サンプルのビットが保存される順序。

サポートされる値は、LittleEndian です。

## サンプルレート

出力ファイルのオーディオストリームのサンプルレート (Hz 単位)。Elastic Transcoder によって入力ファイルのサンプルレートが自動的に検出され、その値が出力ファイルに使用されるようにする場合は、[auto] (自動) を選択します。サンプルレートを指定する場合は、該当する値を選択します。

## Bit Rate

出力ファイルのオーディオストリームのビットレート (キロビット/秒単位)。64 ~ 320 のいずれかの整数を入力します。

## Channels

出力ファイル内のオーディオチャンネルの数。以下の値が有効です。

auto, 0, 1, 2

1 つのチャンネルには、単一スピーカーで再生された情報が保持されます。例えば、2 つのチャンネルのステレオトラックでは、1 つのチャンネルが左のハンドスピーカーに送られ、もう 1 つのチャンネルが右のハンドスピーカーに送られます。出力チャンネルはトラックに編成されます。Elastic Transcoder によって入力ファイルのオーディオチャンネル数が自動的に検出され、その値が出力ファイルに使用されるようにする場合は、auto を選択します。

指定したチャンネル値	入力トラックとチャンネル	出力サウンドのタイプ
auto	任意の入力	最大 8 つの入力チャンネルを通過する。
0	任意の入力	出力からオーディオが省略される。
1	少なくとも 1 つのチャンネル	モノラルサウンド。
2	任意の入力	2 つの同一のモノラルチャンネルまたはステレオ。トラックの詳細については、「 <a href="#">Audio Packing Mode</a> 」を参照してください。

デジタルオーディオの仕組みについては、「[デジタル音声](#)」を参照してください。Elastic Transcoder でチャンネルとトラックを編成する方法については、「[Audio Packing Mode](#)」を参照してください。

### (MXF と PCM のみ) オーディオパッキングモード

オーディオチャンネルとトラックを整理する方法。[Channels] (チャンネル) を使用して出力のチャンネル数を指定し、[Audio Packing Mode] (オーディオパッキングモード) を使用してトラック数およびトラックとチャンネルの関係を指定します。オーディオパッキングモードを指定しない場合、Elastic Transcoder では SingleTrack が使用されます。

以下の値が有効です。

SingleTrack、OneChannelPerTrack、および OneChannelPerTrackWithMosTo8Tracks  
シングルトラック

Elastic Transcoder では、出力用のトラックが 1 つ作成されます。そのトラックには最大で 8 つのチャンネルを設けることができます。mxf ではないすべてのコンテナに対して SingleTrack を使用します。

指定したチャンネル値	入力トラックとチャンネル	出力トラックとチャンネル
0	任意の入力	出力からオーディオが省略される。
1、2、または auto	オーディオ入力なし	出力からオーディオが省略される
1	オーディオを含むあらゆる入力	1トラック1チャンネル (必要に応じてダウンミックスされる)
2	1トラック1チャンネル	1トラック同一2チャンネル
2 または auto	2トラック各1チャンネル	1トラック2チャンネル
2 または auto	1トラック2チャンネル	1トラック2チャンネル
2	1トラック複数チャンネル	1トラック2チャンネル

指定したチャンネル値	入力トラックとチャンネル	出力トラックとチャンネル
auto	1トラック 1チャンネル	1トラック 1チャンネル
auto	1トラック複数チャンネル	1トラック複数チャンネル

### (MXF のみ) 各トラック 1 チャンネル

Elastic Transcoder では、出力の各チャンネルに対して新しいトラックが 1 つずつ作成されます。出力には最大で 8 つのシングルチャンネルトラックを含めることができます。

指定したチャンネル値	入力トラックとチャンネル	出力トラックとチャンネル
0	任意の入力	出力からオーディオが省略される
1、2、または auto	オーディオ入力なし	出力からオーディオが省略される
1	オーディオを含むあらゆる入力	1トラック 1チャンネル (必要に応じてダウンミックスされる)
2	1トラック 1チャンネル	それぞれ 1 つの同一チャンネルがある 2 つのトラック
2 または auto	2トラック各 1チャンネル	2トラック各 1チャンネル
2 または auto	1トラック 2チャンネル	2トラック各 1チャンネル
2	1トラック複数チャンネル	2トラック各 1チャンネル
auto	1トラック 1チャンネル	1トラック 1チャンネル
auto	1トラック複数チャンネル	それぞれ 1 つのチャンネルがある最大 8 つのトラック

## (MXF のみ) 各 MOSトラック 1 チャンネル (最大 8 トラック)

Elastic Transcoder では、出力用のシングルチャンネルトラックが最大で 8 つ作成されます。入力チャンネルのオーディオデータが含まれていないトラックはすべて音なし (MOS、Mit Out Sound) トラックです。

指定したチャンネル値	入力トラックとチャンネル	出力トラックとチャンネル
0	任意の入力	出力からオーディオが省略される
1、2、または auto	オーディオ入力なし	出力からオーディオが省略される
1	オーディオを含むあらゆる入力	1 トラック 1 チャンネル (必要に応じてダウンミックスされる)、+ 7 MOS トラック
2	1 トラック 1 チャンネル	2 トラック各 1 同一チャンネル、+ 6 MOS トラック
2 または auto	2 トラック各 1 チャンネル	2 トラック各 1 チャンネル、+ 6 MOS トラック
2 または auto	1 トラック 2 チャンネル	2 トラック各 1 チャンネル、+ 6 MOS トラック
2	1 トラック複数チャンネル	2 トラック各 1 チャンネル、+ 6 MOS トラック
auto	1 トラック 1 チャンネル	1 トラック 1 チャンネル、+ 7 MOS トラック
auto	1 トラック複数チャンネル	最大 8 トラック各 1 チャンネル、+ 8 MOS トラック (すべてに 8 トラックが存在するまで)

チャンネルとトラックの詳細については、「[デジタル音声](#)」を参照してください。

**Audio**

Codec pcm ⓘ

Profile auto ⓘ

Bit Depth Select One... ⓘ

Sample Rate auto ⓘ

Channels auto ⓘ

Audio Packing Mode Select One... ⓘ

## サムネイルの設定

### 形式

サムネイルの形式 (該当する場合)。有効な形式は [jpg] と [png] です。

ジョブを作成するときに、Elastic Transcoder によってサムネイルが作成されるようにするかどうかを指定します。詳細については、トピック「[Elastic Transcoder ジョブの作成時に指定する設定](#)」の「[Thumbnail Filename Pattern](#)」を参照してください。

### インターバル

サムネイル間のおおよその秒数。整数値を指定します。1 つのサムネイルから次のサムネイルまでの実際の間隔は数秒異なる場合があります。

### Max Height

サムネイルの最大高さ (ピクセル単位)。[auto] (自動) を指定した場合、Elastic Transcoder で、1080 (フル HD) がデフォルト値として使用されます。数値を指定する場合は、32 ~ 3072 のいずれかの偶数を入力します。

## Max Width

サムネイルの最大幅 (ピクセル単位)。[auto] (自動) を指定した場合、Elastic Transcoder で、1920 (フル HD) がデフォルト値として使用されます。数値を指定する場合は、32 ~ 4096 のいずれかの偶数を入力します。

## Sizing Policy

サムネイルの拡大/縮小を制御するには、以下のいずれかの値を指定します。

- [Fit]: Elastic Transcoder で、サムネイルの [Max Width] (最大幅) 設定または [Max Height] (最大高さ) 設定で指定した値に一致し、かつ他方の値を超えないように、サムネイルのスケールリングが行われます。
- [Fill]: Elastic Transcoder で、サムネイルの [Max Width] (最大幅) 設定または [Max Height] (最大高さ) 設定で指定した値に一致し、かつ他方の値に一致するかこれを超えるように、サムネイルのスケールリングが行われます。Elastic Transcoder では、サムネイルのイメージが中央に配置され、次に、最大値を超えるディメンション (該当する場合) がトリミングされます。
- [Stretch]: Elastic Transcoder では、サムネイルの [Maximum Width] (最大幅) と [Maximum Height] (最大高さ) の設定で指定した値に一致するように、サムネイルが引き伸ばされます。入力動画とサムネイルでアスペクト比が異なる場合、サムネイルは歪みます。
- [Keep]: Elastic Transcoder でサムネイルがスケールリングされません。入力ビデオのいずれかのディメンションが、サムネイルの [Max Width] (最大幅) と [Max Height] (最大高さ) 設定で指定した値を超える場合、Elastic Transcoder によりサムネイルがトリミングされます。
- [ShrinkToFit]: Elastic Transcoder では、サムネイルのディメンションが、サムネイルの [MaxWidth] (最大幅) と [MaxHeight] (最大高さ) で指定した少なくともどちらか 1 つの値に一致し、かついずれの値も超えないように、サムネイルが縮小されます。このオプションを指定すると、Elastic Transcoder でサムネイルは拡大されません。
- [ShrinkToFill]: Elastic Transcoder では、サムネイルのディメンションが [MaxWidth] (最大幅) と [MaxHeight] (最大高さ) に指定した少なくともどちらか 1 つの値に一致し、かつこれらの値を下回らないように、サムネイルを縮小します。このオプションを指定すると、Elastic Transcoder でサムネイルは拡大されません。

以下の表には、[Sizing Policy] の設定がサムネイルに与える可能性のある影響を示しています。

SizingPolicy	サムネイルが拡大される可能性がある	サムネイルに余白が追加される可能性がある ([Padding Policy] が [Pad] の場合)	サムネイルのピクセルアスペクト比が入力動画のものと異なる可能性がある	サムネイルがクロップされる可能性がある
Fit	あり	あり		
Fill	あり			あり
[Stretch] (ストレッチ)	あり		あり	
Keep	あり	あり		あり
ShrinkToFit に設定します。		はい		
ShrinkToFill		あり		あり

## パディングポリシー

[PaddingPolicy] (パディングポリシー) を [Pad] (パディングあり) に設定すると、サムネイルの合計サイズが指定した [Max Width] (最大幅) の値と [Max Height] (最大高さ) の値に一致するように、Elastic Transcoder によってサムネイルの上下/左右に黒いバーが追加される場合があります。詳細については、「[Sizing Policy](#)」の表を参照してください。

## Thumbnails

**Format**   

**Interval**  

**Max Width**  

**Max Height**  

**Sizing Policy**   

**Padding Policy**   

[Cancel](#) [Create Preset](#)

## Amazon S3 バケット内のファイルの整理

Amazon Elastic Transcoder を Amazon S3 と統合すれば、入力ファイルと出力ファイルを保存できます。Amazon S3 の 100 バケットというストレージ制限は、Elastic Transcoder ファイルの整理方法と管理方法に影響します。

Amazon S3 で作業を行う場合、[通常使用するファイルシステム](#)を模倣することをお勧めします。そのためには、Elastic Transcoder で OutputKeyPrefix を使用して、出力ファイルにファイルパスを追加します。ファイルシステムを模倣すれば、Amazon S3 API の[プレフィックスと区切り記号のパラメータ](#)を使用してファイルを検索できます。

たとえば、多くの顧客用に動画を取り込み、それらの動画を変換して複数の一般的なデバイスで使用できるようにするユーザー生成コンテンツ ( UGC ) サイトを作成するとします。動画を保存するには、次のようにして入力バケットと出力バケットを整理できます。

```
Input:  /amzn-s3-demo-bucket/input/movie.mp4
Output: /amzn-s3-demo-bucket/output/movie/<format>/movie.mp4
```

これにより、動画を個別の形式で保存できるようになり、動画を簡単に検索することができます。たとえば、2 人の顧客がおり、サービスを利用してさまざまな動画を複数の形式に変換するとします。次のようにバケットを整理できます。

Customer 1:

```
/amzn-s3-demo-bucket/<1>/output/movie/avi/movie.avi
/amzn-s3-demo-bucket/<1>/output/movie/mkv/movie.mkv
/amzn-s3-demo-bucket/<1>/output/movie/hls400k/movie.m3u8
- playlist file for the hls400k version of movie
/amzn-s3-demo-bucket/<1>/output/movie/hls400k/movie.ts
- video file for the hls400k version of movie
/amzn-s3-demo-bucket/<1>/output/movie/hls400k/movie.png
- thumbnail file for the hls400k version of movie
/amzn-s3-demo-bucket/<1>/output/myOthermovie/hls1000k/movie.m3u8
- playlist file for the hls1000k version of movie
/amzn-s3-demo-bucket/<1>/output/myOthermovie/hls1000k/movie.ts
- video file for the hls1000k version of movie
/amzn-s3-demo-bucket/<1>/output/myOthermovie/hls1000k/movie.png
- thumbnail file for the hls1000k version of movie
```

Customer 2:

```
/amzn-s3-demo-bucket/<2>/output/somemovie/avi/somemovie.avi
/amzn-s3-demo-bucket/<2>/output/somemovie/mkv/somemovie.mkv
/amzn-s3-demo-bucket/<2>/output/somemovie/hls400k/somemovie.m3u8
- playlist file for the hls400k version of somemovie
/amzn-s3-demo-bucket/<2>/output/somemovie/hls400k/somemovie.ts
- video file for the hls400k version of somemovie
/amzn-s3-demo-bucket/<2>/output/somemovie/hls400k/somemovie.png
- thumbnail file for the hls400k version of somemovie
/amzn-s3-demo-bucket/<2>/output/myOthermovie/hls1000k/movie.m3u8
- playlist file for the hls1000k version of movie
/amzn-s3-demo-bucket/<2>/output/myOthermovie/hls1000k/movie.ts
- video file for the hls1000k version of movie
/amzn-s3-demo-bucket/<2>/output/myOthermovie/hls1000k/movie.png
- thumbnail file for the hls1000k version of movie
```

OutputKeyPrefix の最後に "/" を使用することをお勧めします。これにより、OutputKeyPrefix の最後の部分と OutputKey が混在しなくなります。そのようにしないと、ファイルは次のようになります。

```
/amzn-s3-demo-bucket/output/<movie title>/mkvmovie.mkv
```

また、入力メディアを各リージョンで 1 つのバケットに保存し、変換結果メディアとサムネイルを同じリージョンの別のバケットに保存することもお勧めします。この方法によって、Amazon S3 バケットと Elastic Transcoder の間でリージョン間データ転送料金が発生するのを防ぐことができます。

#### Note

1 秒あたり 100 件を超えるリクエストが Amazon S3 バケットにアクセスすると予想される場合は、システムを設計するときに、「[Amazon S3 のパフォーマンスに関する考慮事項](#)」に従うことをお勧めします。

## コンテンツの保護

このセクションでは、Elastic Transcoder リソースへのアクセスを制御する方法、ファイルを保管時に暗号化する方法、ファイルにデジタル著作権管理 (DRM) を適用する方法について説明します。リソースへのアクセス権限を管理すると、パイプラインやプリセットリソースの更新権限や削除権限は与えずに、ジョブの作成タスクなどをいずれかの IAM ロールに認めることができます。保管中のファイルを暗号化すると、特に機密性の高いファイルや厳密に管理されたファイルに対して、さらにレベルのコンテンツ保護が提供されます。一方、DRM を使用すると、リソースに対する AWS アクセス許可によって提供されるレベルを超えて、ファイルを再生できるユーザーをさらに制御できます。

IAM アクセスコントロールは、ファイルにアクセスできるのは誰か、パイプラインやプリセットなどのリソースに影響を与えることができるのは誰かを制御する場合に使用します。ファイル暗号化 (保管時の暗号化) は、ファイルを暗号化した状態で保管する場合に使用します。また、HLS と DRM は、誰がファイルを再生できるのかを制御する場合は使用します。

セキュリティのベストプラクティスの詳細については、「[IAM ベストプラクティス](#)」ガイドを参照してください。

### トピック

- [Elastic Transcoder に対するアクセス権限の制御](#)
- [データ暗号化のオプション](#)

## Elastic Transcoder に対するアクセス権限の制御

Amazon Elastic Transcoder を使用すると、AWS Identity and Access Management (IAM) を使用して Elastic Transcoder でユーザーが実行できる操作を制御したり、Elastic Transcoder が必要とする他の AWS サービスへの Elastic Transcoder のアクセスを制御したりできます。アクセスの制御には、IAM ポリシーを使用します。このポリシーは、IAM ユーザー、IAM グループ、またはロールに関連付けることができるアクセス許可を集めたものです。

### トピック

- [Elastic Transcoder に対するアクセス権限の制御](#)
- [Elastic Transcoder パイプラインのサービスロール](#)

## Elastic Transcoder に対するアクセス権限の制御

アクセス権限を認めるには、ユーザー、グループ、またはロールにアクセス許可を追加します。

- のユーザーとグループ AWS IAM アイデンティティセンター:

アクセス許可セットを作成します。「AWS IAM アイデンティティセンター ユーザーガイド」の「[権限設定を作成する](#)」の手順に従ってください。

- IAM 内で、ID プロバイダーによって管理されているユーザー:

ID フェデレーションのロールを作成します。詳細については「IAM ユーザーガイド」の「[サードパーティー ID プロバイダー \(フェデレーション\) 用のロールを作成する](#)」を参照してください。

- IAM ユーザー:

- ユーザーが担当できるロールを作成します。手順については「IAM ユーザーガイド」の「[IAM ユーザーのロールの作成](#)」を参照してください。
- (お奨めできない方法) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加します。詳細については「IAM ユーザーガイド」の「[ユーザー \(コンソール\) へのアクセス権限の追加](#)」を参照してください。

Elastic Transcoder の他の AWS サービスへのアクセスを制御するには、サービスロールを作成できます。それは、パイプラインを作成するときに割り当てて、Elastic Transcoder 自体にトランスコーディングにともなうタスクを実行するためのアクセス権限を与える IAM ロールです。

のロールを作成するには AWS のサービス (IAM コンソール)

1. にサインイン AWS マネジメントコンソールし、<https://console.aws.amazon.com/iam/> で IAM コンソールを開きます。
2. IAM コンソールのナビゲーションペインで、[ロール]、[ロールを作成] を選択します。
3. 信頼できるエンティティタイプで、AWS のサービス を選択します。
4. [サービスまたはユースケース] でサービスを選択し、次にユースケースを選択します。ユースケースは、サービスに必要な信頼ポリシーを含める定義になります。
5. [次へ] を選択します。
6. [アクセス許可ポリシー] では、オプションは選択したユースケースによって異なります。
  - サービスがロールのアクセス許可を定義している場合、アクセス許可ポリシーを選択することはできません。

- 制限されたアクセス許可ポリシーのセットから選択します。
  - すべてのアクセス許可ポリシーから選択します。
  - アクセス許可ポリシーを選択するのではなく、ロールの作成後にポリシーを作成し、そのポリシーをロールにアタッチします。
7. (オプション) [アクセス許可の境界](#)を設定します。このアドバンスド機能は、サービスロールで使用できますが、サービスにリンクされたロールではありません。
- a. [アクセス許可の境界の設定] セクションを開き、[アクセス許可の境界を使用してロールのアクセス許可の上限を設定する] を選択します。
- IAM には、アカウントの AWS 管理ポリシーとカスタマー管理ポリシーのリストが含まれます。
- b. アクセス許可の境界として使用するポリシーを選択します。
8. [次へ] を選択します。
9. [ロール名] では、オプションはサービスによって異なります。
- サービスでロール名が定義されている場合、ロール名を編集することはできません。
  - サービスでロール名のプレフィックスが定義されている場合、オプションのサフィックスを入力できます。
  - サービスでロール名が定義されていない場合、ロールに名前を付けることができます。

**⚠ Important**

ロールに名前を付けるときは、次のことに注意してください。

- ロール名は 内で一意である必要があり AWS アカウント、大文字と小文字を区別することはできません。

例えば、**PRODROLE** と **prodrole** の両方の名前で作成することはできません。ロール名がポリシーまたは ARN の一部として使用される場合、ロール名は大文字と小文字が区別されます。ただし、サインインプロセスなど、コンソールにロール名がユーザーに表示される場合、ロール名は大文字と小文字が区別されません。

- 他のエンティティがロールを参照する可能性があるため、ロールを作成した後にロール名を編集することはできません。

10. (オプション) [説明] にロールの説明を入力します。

11. (オプション) ロールのユースケースとアクセス許可を編集するには、[ステップ 1: 信頼されたエンティティを選択] または [ステップ 2: アクセス権限を追加] のセクションで [編集] を選択します。
12. (オプション) ロールの識別、整理、検索を簡単にするには、キーと値のペアとしてタグを追加します。IAM でのタグの使用の詳細については、IAM ユーザーガイドの「[AWS Identity and Access Management リソースのタグ](#)」を参照してください。
13. ロールを確認したら、[Create role] (ロールを作成) を選択します。

ユーザーロールとサービスロールがトランスコーディングプロセスでいずれも重要であることを示す例として、Elastic Transcoder では Amazon S3 バケットからファイルを取得するためにサービスロールが必要になり、そのトランスコードしたファイルを別の Amazon S3 バケットに保存したとき、一方で、ユーザーは Elastic Transcoder によるジョブの作成を許可する IAM ロールが必要な場合があります。

IAM の詳細については、『[IAM ユーザーガイド](#)』を参照してください。サービスロールの詳細については、「[AWS サービスのロールの作成](#)」を参照してください。

## Elastic Transcoder のポリシーの例

ユーザーが、パイプラインの作成やジョブの実行など、Elastic Transcoder の管理機能を実行するためには、そのユーザーに関連付けることができるポリシーが必要です。このセクションでは、ポリシーの作成方法と、Elastic Transcoder のオペレーションと Elastic Transcoder が依存する関連サービスのオペレーションへのアクセス権限を制御するための 3 つのポリシーを示します。AWS アカウントのユーザーに、すべての Elastic Transcoder オペレーションまたはそれらのサブセットへのアクセスを許可できます。

IAM ユーザーポリシーの管理の詳細については、『IAM ユーザーガイド』の「[IAM ポリシーの管理](#)」を参照してください。

JSON ポリシーエディタでポリシーを作成するには

1. にサインイン AWS マネジメントコンソールし、<https://console.aws.amazon.com/iam/> で IAM コンソールを開きます。
2. 左側のナビゲーションペインで、[ポリシー] を選択します。  
  
初めて [ポリシー] を選択する場合には、[管理ポリシーによろこそ] ページが表示されます。[今すぐ始める] を選択します。
3. ページの上部で、[ポリシーを作成] を選択します。

4. [ポリシーエディタ] セクションで、[JSON] オプションを選択します。
5. JSON ポリシードキュメントを入力するか貼り付けます。IAM ポリシー言語の詳細については、「[IAM JSON ポリシーリファレンス](#)」を参照してください。
6. [ポリシーの検証](#)中に生成されたセキュリティ警告、エラー、または一般警告をすべて解決してから、[次へ] を選択します。

**Note**

いつでも [Visual] と [JSON] エディタオプションを切り替えることができます。ただし、[Visual] エディタで [次へ] に変更または選択した場合、IAM はポリシーを再構成して visual エディタに合わせて最適化することがあります。詳細については、「IAM ユーザーガイド」の「[ポリシーの再構成](#)」を参照してください。

7. (オプション) でポリシーを作成または編集するときに AWS マネジメントコンソール、テンプレートで使用できる JSON または YAML ポリシー CloudFormation テンプレートを生成できません。

これを行うには、ポリシーエディタで [アクション] を選択し、次に [CloudFormation テンプレートを生成] を選択します。詳細については CloudFormation、「AWS CloudFormation ユーザーガイド」の「[AWS Identity and Access Management リソースタイプのリファレンス](#)」を参照してください。

8. ポリシーにアクセス権限を追加し終えたら、[次へ] を選択します。
9. [確認と作成] ページで、作成するポリシーの [ポリシー名] と [説明] (オプション) を入力します。[このポリシーで定義されているアクセス許可] を確認して、ポリシーによって付与されたアクセス許可を確認します。
10. (オプション) タグをキー - 値のペアとしてアタッチして、メタデータをポリシーに追加します。IAM でのタグの使用の詳細については、IAM ユーザーガイドの[AWS Identity and Access Management 「リソースのタグ」](#)を参照してください。
11. [Create Policy (ポリシーを作成)] をクリックして、新しいポリシーを保存します。

## Elastic Transcoder と Amazon S3 への読み取り専用アクセス権限を与える

次のポリシーでは、Elastic Transcoder リソースに対する読み取り専用のアクセス権限と Amazon S3 のリストオペレーションのアクセス権限を与えます。このポリシーは、トランスコードされたファイルの検索や表示のためのアクセス権限や、IAM アカウントで利用できるバケットを表示するアクセス権限のみが必要で、リソースやファイルの更新、作成、削除権限は不要なユーザーに適しています。

す。また、このポリシーでは、IAM アカウントで利用できるすべてのパイプライン、プリセット、ジョブのリストが可能です。特定のバケットへのアクセス権限を制限するには、「[VPC 内のリソースへのアクセス権限を制限する](#)」を参照してください。

## JSON

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

## ジョブの作成アクセス権限を与える

次のポリシーでは、アカウントに関連付けられたすべての Elastic Transcoder リソースのリストと取得、ジョブやプリセットの作成や変更、Amazon S3 と Amazon SNS のリストオペレーション使用のためのアクセス権限を与えます。

このポリシーは、トランスコーディング設定の変更や、プリセットやジョブの作成や削除に適しています。このポリシーでは、パイプライン、Amazon S3 バケット、Amazon SNS 通知の作成、更新、または削除はできません。

## JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "elastictranscoder:Read*",
        "elastictranscoder:List*",
        "elastictranscoder:*Job",
        "elastictranscoder:*Preset",
        "s3:List*",
        "sns:List*"
    ],
    "Resource": "*"
}
]
```

## 制御可能なアクセス権限による Elastic Transcoder オペレーション

すべての Elastic Transcoder オペレーションのリストを次に示します。

```
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
```

## VPC 内のリソースへのアクセス権限を制限する

オペレーション ( アクション ) へのアクセスの制限に加えて、特定のジョブ、パイプライン、プリセットへのアクセス権限を制限できます。これは、リソースレベルのアクセス権限の付与と呼ばれます。

Elastic Transcoder リソースのサブセットへのアクセス権を制限または許可するには、ポリシーのリソース要素にリソースの ARN を記述します。Elastic Transcoder の ARN の一般的な形式は次のようになります。

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

#####、#####、####、ID の各変数を有効な値に置き換えます。有効な値は次のとおりです。

- **region**: リージョンの名前。リージョンのリストについては、[こちら](#)を参照してください。すべてのリージョンを含めるには、ワイルドカード (\*) を使用します。値を指定する必要があります。
- **account**: AWS アカウントの ID。値を指定する必要があります。
- **####**: Elastic Transcoder リソースのタイプ (preset、pipeline、または job)。
- **ID**: 特定のプリセット、パイプライン、またはジョブの ID、または現在の AWS アカウントに関連付けられている指定されたタイプのすべてのリソースを示す \*。

たとえば、次の ARN は、アカウント 111122223333 の us-east-2 リージョンにあるすべてのプリセットリソースを指定します。

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

リソースの ARN を検索するには、拡大鏡のアイコン (



) をクリックします。このアイコンは、パイプライン、プリセット、またはジョブのコンソールページのリソース名の横にあります。

詳細については、『IAM ユーザーガイド』の「[リソース](#)」を参照してください。

### リソースを制限するポリシーの例

次のポリシーは、Amazon S3 の amzn-s3-demo-bucket という名前のバケットへのアクセス権限、Elastic Transcoder 内のすべてに対するリストと読み取りのアクセス権限、および example\_pipeline という名前のパイプラインでジョブを作成するための認可を付与します。

このポリシーは、利用可能なファイルやリソースを確認し、それらのリソースを使用して独自のトランスコーディングジョブを作成する必要がある SDK および CLI ユーザーに適しています。このポリシーは、リソースの更新や削除、ジョブ以外のリソースを作成、またはここで指定されているもの以外のリソースの操作は許可されません。また、コンソールユーザーには適用されません。

## Elastic Transcoder パイプラインのサービスロール

トランスコーディングジョブを管理するパイプラインを作成する場合、IAM サービスロールを指定する必要があります。IAM サービスロールには、トランスコーディング用のパイプラインで使用されるアクセス権限を指定するポリシーがあります。

パイプラインのロールを指定するときは、以下の 2 つのオプションがあります。

- デフォルトのロールを使用します。これには、Elastic Transcoder でのトランスコーディングに必要なアクセス権限のみが含まれます。Elastic Transcoder コンソールを使用してパイプラインを作成する場合は、最初のパイプラインを作成するときにデフォルトのロールが自動的に作成されるように選択できます。デフォルトのロールを含め、IAM サービスロールを作成するには、管理権限が必要です。
- 既存のロールを選択します。この場合、IAM であらかじめロールを作成し、ファイルをトランスコードするための十分なアクセス許可を Elastic Transcoder に付与するポリシーをそのロールに添付しておく必要があります。このロールは他の AWS サービスに使用する場合にも便利です。

### パイプラインのデフォルトの IAM ロール

Elastic Transcoder によって作成されたデフォルトのロールで、これにより Elastic Transcoder で次のオペレーションが実行されます。

- トランスコーディング用に Amazon S3 バケットからファイルを取得する。
- 任意の Amazon S3 バケットの内容を一覧表示する。
- トランスコードしたファイルを Amazon S3 バケットに保存する。
- Amazon S3 マルチパートアップロードを作成する。
- 任意の SNS トピックに通知を発行する。

このポリシーにより、Elastic Transcoder では以下のいずれかのオペレーションの実行が禁止されます。

- Amazon SNS の削除オペレーションを実行するか、トピックのポリシーステートメントを追加または削除します。
- Amazon S3 バケットまたは項目の削除オペレーションを実行するか、バケットポリシーの追加、削除、または変更します。

デフォルトのロールのアクセス (アクセス権限) ポリシーの定義は、次のようになります。

JSON

```
{
  "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": "*"
    }
  ]
}
```

## Elastic Transcoder サービスにリンクされたロールに関してサポートされているリージョン

Elastic Transcoder では、サービスにリンクされたロールの以下のリージョンにおける使用がサポートされています。

リージョン名	リージョンアイデンティティ	Elastic Transcoder におけるサポート
米国東部 (バージニア北部)	us-east-1	はい
米国東部 (オハイオ)	us-east-2	なし
米国西部 (北カリフォルニア)	us-west-1	はい
米国西部 (オレゴン)	us-west-2	はい
アジアパシフィック (ムンバイ)	ap-south-1	はい
アジアパシフィック (大阪)	ap-northeast-3	いいえ
アジアパシフィック (ソウル)	ap-northeast-2	なし
アジアパシフィック (シンガポール)	ap-southeast-1	はい
アジアパシフィック (シドニー)	ap-southeast-2	はい
アジアパシフィック (東京)	ap-northeast-1	はい
カナダ (中部)	ca-central-1	なし
欧州 (フランクフルト)	eu-central-1	なし
欧州 (アイルランド)	eu-west-1	はい
欧州 (ロンドン)	eu-west-2	なし
欧州 (パリ)	eu-west-3	なし
南米 (サンパウロ)	sa-east-1	なし

## データ暗号化のオプション

ファイルを Amazon S3 に保存 (保管) する一方で、トランスコーディングジョブで使用する入力ファイルや出力ファイルをすべて暗号化することによって、Elastic Transcoder のデータを保護できます。これには、入力ファイル、出力ファイル、サムネイル、キャプション、入力ウォーターマーク、または入力アルバムアートが含まれます。プレイリストとメタデータは暗号化されません。

パイプライン、Amazon S3 バケット、AWS Key Management Service キーを含むジョブのすべてのリソースは、同じ AWS リージョンに存在する必要があります。

### トピック

- [暗号化オプション](#)
- [Elastic Transcoder AWS KMS での の使用](#)
- [HLS のコンテンツ保護](#)
- [デジタル著作権管理](#)

## 暗号化オプション

Elastic Transcoder では、2 つの主要な暗号化オプションがサポートされています。

- Amazon S3 のサーバー側の暗号化: AWS がお客様の暗号化プロセスを管理します。例えば、Elastic Transcoder は Amazon S3 を呼び出し、Amazon S3 はデータを暗号化し、データセンター内のディスクに保存して、ダウンロードするときにデータを復号化します。

デフォルトでは、Amazon S3 バケットは暗号化されたファイルと暗号化されていないファイルの両方を受け入れますが、暗号化されたファイルだけを受け入れるように Amazon S3 バケットを設定できます。Elastic Transcoder が Amazon S3 バケットにアクセスできる限り、アクセス権限を変更する必要はありません。

Amazon S3 のサーバー側の暗号化の詳細については、「Amazon Simple Storage Service ユーザーガイド」の「[サーバー側の暗号化を使用したデータの保護](#)」を参照してください。AWS KMS キーの詳細については、「AWS Key Management Service デベロッパーガイド」の「[AWS Key Management Service とは](#)」を参照してください。

**Note**

AWS-KMS キーを使用するための追加料金は不要です。詳細については、「[AWS Key Management Service の料金](#)」を参照してください。

- お客様が用意したキーによるクライアント側の暗号化: Elastic Transcoder では、お客様が用意した暗号化キーを使用して、入力ファイル (お客様によって暗号化済みのもの) を復号することや、Amazon S3 に保存する前に出力ファイルを暗号化することができます。この場合、暗号化キーや関連ツールはお客様が管理してください。

Elastic Transcoder でクライアント提供のキーを使用してファイルをトランスコードする場合、ジョブリクエストには、ファイルの暗号化に使用した AWS KMS暗号化されたキー、チェックサムとして使用されるキーの MD5、および出力ファイルの暗号化時に Elastic Transcoder で使用する初期化ベクトル (またはランダムビットジェネレーターによって作成された一連のランダムビット) を含める必要があります。

Elastic Transcoder は、KMS AWS KMS キーで暗号化されたお客様が用意したキーのみを使用でき、Elastic Transcoder には KMS キーを使用するためのアクセス許可を付与する必要があります。キーを暗号化するには、次の情報を含む暗号化呼び出しを使用して AWS KMS プログラムで呼び出す必要があります。

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

**Important**

プライベート暗号化キーと暗号化されていないデータは AWS には保存されません。したがって、暗号化キーを安全に管理することが重要です。キーを紛失すると、データを復号できなくなります。

キーを使用するためのアクセス許可を Elastic Transcoder に付与する方法については、「[Elastic Transcoder AWS KMS での使用](#)」を参照してください。

データ暗号化の詳細については、[AWS KMS API リファレンス](#)と「[データの暗号化と復号化](#)」を参照してください。コンテキストについての詳細は、『AWS Key Management Service デベロッパーガイド』の「[暗号化コンテキスト](#)」を参照してください。

お客様が提供するキーの詳細については、『Amazon Simple Storage Service ユーザーガイド』の「[お客様が用意した暗号化キーによるサーバー側の暗号化を使用したデータの保護](#)」を参照してください。

Elastic Transcoder コンソールを使用してファイルの復号化と暗号化を行う際に必要な設定については、「[\(オプション\) Output Encryption](#)」を参照してください。Elastic Transcoder API を使用してファイルの復号化と暗号化を行う際に必要な設定については、暗号化要素から始まる [ジョブを作成する](#) API アクションを参照してください。

## Elastic Transcoder AWS KMS での の使用

Elastic Transcoder で AWS Key Management Service ( AWS KMS ) を使用して、データの暗号化に使用される暗号化キーを作成および管理できます。使用する Elastic Transcoder を設定する前に AWS KMS、次のものがが必要です。

- Elastic Transcoder のパイプライン
- Elastic Transcoder パイプラインに関連付けられた IAM ロール
- AWS KMS キー
- AWS KMS キーの ARN

次の手順は、既存のリソースを識別する方法または新しいリソースを作成する方法を示します。

### Elastic Transcoder AWS KMS の使用準備

パイプラインを作成するには

- 「[Elastic Transcoder でパイプラインを作成する](#)」のステップを実行してください。

パイプラインに関連付けられた IAM ロールを特定する方法

1. にサインイン AWS マネジメントコンソールし、<https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. ナビゲーションペインで、[Pipelines] (パイプライン) をクリックします。

3. パイプラインの名の横の拡大鏡アイコンをクリックします。
4. [Permissions] セクションをクリックして展開します。
5. IAM ロールの情報を書き留めます。Elastic Transcoder によって作成されたデフォルトのロールを使用している場合、ロールは Elastic\_Transcoder\_Default\_Role になります。

AWS KMS キーを作成するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. 「[キーの作成](#)」の手順を行います。

AWS KMS キーの ARN を識別するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで、[Encryption Keys] をクリックします。
3. リージョンのドロップダウンリストで、キーとパイプラインがあるリージョンを選択します。
4. 使用するキーをクリックします。
5. ARN を書き留めます。

コンソールを使用して AWS KMS キーを作成できますが、暗号化 API と復号 APIs を使用して AWS KMS キーでデータを暗号化または復号する必要があります。詳細については、「[データの暗号化と復号](#)」を参照してください。

## Elastic Transcoder と の接続 AWS KMS

パイプライン、IAM ロール、AWS KMS キーを取得したら、使用するキーをパイプラインに伝え、使用できる IAM ロールをキーに伝える必要があります。

AWS KMS キーをパイプラインに追加するには

1. <https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. AWS KMS キーを使用するパイプラインを選択し、**編集** をクリックします。
3. [Encryption] セクションをクリックして展開し、[AWS KMS Key ARN] セクションで [Custom] を選択します。
4. AWS KMS キーの ARN を入力し、**保存** をクリックします。

## IAM ロールを AWS KMS キーに追加するには

パイプラインに関連付けられた IAM ロールを使用して AWS KMS キーを作成しなかった場合は、以下の手順で追加できます。

1. <https://console.aws.amazon.com/kms> で AWS KMS コンソールを開きます。
2. リージョンのドロップダウンリストで、キーとパイプラインを作成したときに選択したリージョンを選択します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. 右側の [カスタマー管理型のキー] で、使用するキーの名前を選択します。
5. [Key users (キーのユーザー)] セクションで、[Add (追加)] を選択します。
6. [Add key users (キーのユーザーを追加)] ページで、パイプラインに関連付けられたロールを検索し、検索結果からロールを選択して [Add (追加)] を選択します。

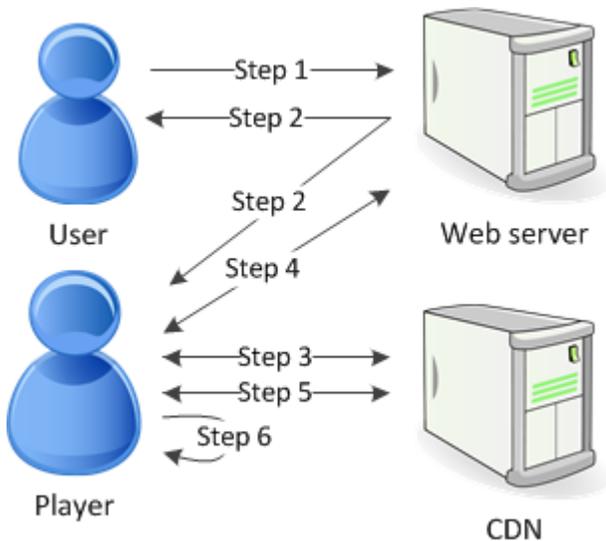
Elastic Transcoder パイプラインで AWS KMS キーを使用できるようになりました。

## HLS のコンテンツ保護

HTTP Live Streaming (HLS) は、ストリーミングを最適化するためにメディアファイルをセグメント化するプロトコルです。HLS によって、メディアプレーヤーでは、再生時にネットワーク接続でサポートされている最高品質の解像度でセグメントを再生できます。

Elastic Transcoder を使用して、ストリーミングされるメディアファイルのセグメントを暗号化し、暗号化されたセグメントをインターネットで送信して再生時に復号化できます。これにより、メディアコンテンツが保護され、承認されたユーザーのみがメディアファイルの暗号化されたセグメントを表示できます。

ここでは、HLS のコンテンツ保護を含むメディアファイルの再生プロセスの概要を示します。



1. ウェブページにアクセスしたユーザーがウェブサーバーで認証され、ユーザーのブラウザでセッション Cookie が設定されます。
2. ユーザーがウェブサーバーからプレーヤーをロードします。
3. プレーヤーはコンテンツ配信ネットワーク (CDN) からマスタープレイリストを取得します。マスタープレイリストは、メディアファイルで利用可能なビットレートと解像度を提供します。
4. プレーヤーはウェブサーバーを呼び出します。ウェブサーバーは、セッション Cookie を検証し、ユーザーがコンテンツの表示を許可されていることを確認して、データ復号キーを返します。
5. プレーヤーは、プレイリストを選択し、CDN から関連するメディアセグメントを取得します。
6. プレーヤーはデータキーを使用してセグメントを復号し、メディアの再生を開始します。

#### **i** Note

HLS のコンテンツ保護を使用してストリーミングファイルのセグメントを暗号化するか、ファイル全体を暗号化することができます。両方の暗号化を使用することはできません。したがって、HLS のコンテンツ保護と個々のファイル保護の両方を選択しないでください。

## HLS のコンテンツ保護のキー

HLS のコンテンツ保護を Elastic Transcoder で使用するには、2 つのタイプのキーが必要になります。

- AWS KMS key - Elastic Transcoder パイプラインに関連付けられたキー
- データキー - Elastic Transcoder ジョブに関連付けられたキー

HLS コンテンツ保護を使用するには AWS KMS key、が必要です。KMS キーは、インターネットで送信する前にデータキーを暗号化するために使用されます。すべてのトランスコーディングジョブで使用する KMS キーを 1 個作成することをお勧めします。キーの作成と設定の詳細については、[Elastic Transcoder AWS KMS での の使用](#) を参照してください。

データキーは、メディアファイルの暗号化に使用されます。同じコンテンツのすべてのバリエーションとセグメントは、同じデータキーを使用して暗号化されます。データキーを指定しなかった場合、Elastic Transcoder によってデータキーが生成されます。

## HLS で保護されるコンテンツのストリーミング

HLS で保護されたコンテンツを配信するには、以下のものがが必要です。

- 暗号化されたメディアファイルとデータキーを保存する場所。Amazon S3 にファイルを保存し、DynamoDB などのデータベースでキーを保護することをお勧めします。DynamoDB の詳細については、「Amazon DynamoDB デベロッパーガイド」の「[Amazon DynamoDB とは](#)」を参照してください。
- ( オプション ) ファイルをストリーミングするコンテンツ配信ネットワーク (CDN)。CDN の詳細については、「Amazon CloudFront デベロッパーガイド」の「[CloudFrontの開始方法](#)」を参照してください。
- ユーザーの認証と承認を実行し、データ暗号化キーを安全に保存できるアプリケーション。Amazon EC2 を使用してこのアプリケーションを実行できます。詳細については、[Amazon EC2 ユーザーガイド](#) の Amazon EC2 のセットアップ」(Windows ユーザーの場合) または [Amazon EC2 ユーザーガイド](#) の「[Amazon EC2 のセットアップ](#)」(Linux ユーザーの場合) を参照してください。Amazon EC2
- 暗号化された HLS ファイルを復号化できるプレーヤー。詳細については、「[Http Live Streaming](#)」を参照してください。

## 暗号化されたストリーミングコンテンツの作成

HLS のコンテンツ保護用のファイルを準備するには、新しいパイプラインか、既存のパイプラインに KMS キーを関連付ける必要があります。

指定した KMS キーでパイプラインを設定する方法については、「[Elastic Transcoder AWS KMS での使用](#)」を参照してください。

以下の手順は、Elastic Transcoder コンソールの使用によって、HLS のコンテンツ保護用にファイルを暗号化する方法を示しています。

ファイルに HLS のコンテンツ保護を使用するには

1. <https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. ナビゲーションペインで、[Jobs] (ジョブ) をクリックし、新しいジョブを作成します。詳細については、「[Elastic Transcoder でジョブを作成する](#)」を参照してください。
3. [Output Details] の [Presets (プリセット)] ドロップダウンリストで、[HLS] プリセットを選択します。
4. [Encryption Parameters] を [None] に設定したままにします。
5. [Playlists] で、[Add Playlist] をクリックし、プレイリストのタイプとして [HLSv3] または [HLSv4] のいずれかを選択します。
6. [Content Protection] で、[Enter Information] を選択します。

a. 独自のキーを管理するには、[Key Storage Policy (キーストレージポリシー)] で、[No Store] を選択します。[License Acquisition Url] で、データキーを保存する場所への絶対パスを入力します。例:

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

[No Store] を選択し、安全な Amazon S3 バケットまたは DynamoDB などのデータベースにキーを保存することをお勧めします。

b. 公開 Amazon S3 バケットにキーを保存するには、[Key Storage Policy (キーストレージポリシー)] で With Variant Playlists を選択します。Elastic Transcoder では、プレイリストファイルと同じバケットにデータキーが書き込まれます。

#### Important

[With Variant Playlists] を使用して保存されたキーは、パブリックバケットに書き込まれます。実際のキーでは、[No Store] を使用します。

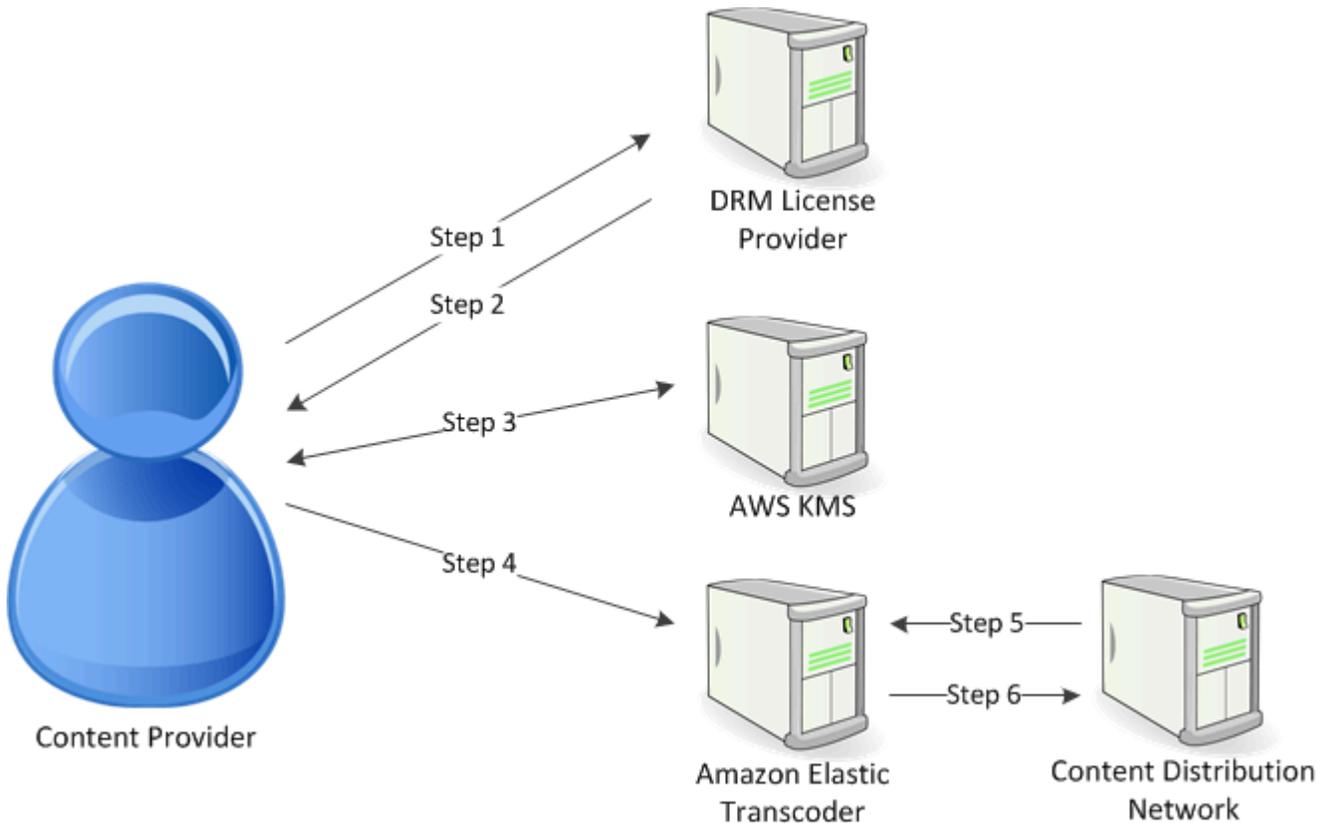
**Note**

No Store を選択すると、データキーは、Elastic Transcoder によりジョブオブジェクトの一部として返されますが、保存はされません。データキーは、お客様が保存する必要があります。

## デジタル著作権管理

デジタル著作権管理 (DRM) でプレイリストへのアクセスを制限して、認証されたユーザーのみコンテンツが見られるようにできます。

DRM は、他社の DRM ライセンスサーバーで認証されたユーザーにファイルの再生を制限することで機能します。DRM は、ファイルのセグメントに DRM ヘッダーを含めたりパッケージ化することでその機能を発揮します。DRM のパッケージには、ライセンスサーバーに接続するために必要な情報やファイルを再生するために必要なすべての暗号化情報が含まれています。ファイルがパッケージ化されると、Amazon S3 や CloudFront などのサービス、または Akamai などの非AWS サービスであるコンテンツ配信ネットワークに送信されます。ファイルが配信された後、ファイルの再生以前にライセンスサーバーからライセンスを確認する必要があります。DRM は、コンテンツ保護のためのプロトコルのセット全体を指定することによって、標準の暗号化以上の効果を実現します。次の図は、DRM の機能の基本フローを示しています。



Elastic Transcoder は、DRM のパッケージング (このプロセスの最後の 3 つのステップ) をユーザーに代わって処理します。

1. コンテンツプロバイダ (お客様) がコンテンツキー ID を使用して DRM ライセンスプロバイダを呼び出し、コンテンツキーを生成します。
2. ライセンスプロバイダはキー ID を使用してコンテンツキーを生成し、生成したコンテンツキーとライセンス取得用 URL をコンテンツプロバイダに返します。
3. コンテンツプロバイダーは AWS KMS を呼び出してコンテンツキーを暗号化し、暗号化されたコンテンツキー AWS KMS を返します。
4. コンテンツプロバイダでは、暗号化されたコンテンツキー、キー ID、およびライセンス取得用 URL を使用して、パッケージサーバー (Elastic Transcoder) を呼び出します。
5. パッケージサーバーは保護するファイルを取得し、ファイルに DRM 情報をバンドルします。
6. パッケージサーバーは DRM で保護されたファイルをディストリビューションサーバーに送信し、ディストリビューションサーバーがファイルを配信します。

## トピック

- [PlayReady DRM](#)

## PlayReady DRM

Elastic Transcoder では、PlayReady DRM を使用した DRM がサポートされます。これにより、メディアコンテンツが保護され、承認されたユーザーのみがメディアファイルを表示できます。PlayReady で保護されたファイルの認証と再生のフローは、特定のプレーヤーおよびライセンスサーバー設定によって異なります。

### デジタル著作権管理のキー

Elastic Transcoder で DRM を使用するには、2 つのタイプのキーと 1 つのキー ID が必要になります。

- コンテンツキー - DRM ライセンスサーバーから取得するキー (Elastic Transcoder ジョブと関連付けられます)
- キー ID - DRM ライセンスサーバーから取得する ID (ライセンスサーバーはこの ID を使用して、ファイルの復号に必要なコンテンツキーを特定します)
- AWS KMS key — Elastic Transcoder パイプラインに関連付けられた AWS KMS キー

DRM を使用するには、コンテンツキーが必要です。コンテンツキーは、メディアファイルの暗号化に使用されます。同じコンテンツのすべてのバリエーションとセグメントは、同じコンテンツキーを使用して暗号化されます。Elastic Transcoder ではこのキーは生成されないため、DRM ライセンスサーバーから取得する必要があります。また、Elastic Transcoder が保護されたプレイリスト内にキーをパッケージできるように、キーの ID も必要です (それにより、プレーヤーはライセンスサーバーからコンテンツキーを取得できます)。

DRM を使用するには、AWS KMS キーが必要です。AWS KMS キーは、インターネット経由で送信される前にコンテンツキーを暗号化するために使用されます。すべてのトランスコードジョブで使用する AWS KMS キーを 1 つ作成することをお勧めします。AWS KMS キーの作成と設定の詳細については、「」を参照してください[Elastic Transcoder AWS KMS での の使用](#)。

### DRM で保護されるコンテンツのストリーミング

DRM コンテンツを配信するには、以下のものがが必要です。

- DRM コンテンツキーを生成して保存するための DRM ライセンス プロバイダ。
- 暗号化されたメディアファイルを保存する場所。ファイルを Amazon S3 に保存することをお勧めします。

- ( オプション ) ファイルをストリーミングするコンテンツ配信ネットワーク (CDN)。詳細については、「Amazon CloudFront デベロッパーガイド」の「[CloudFront の開始方法](#)」を参照してください。
- ユーザーを認証し、承認する機能を持ったアプリケーション。Amazon EC2 を使用してこのアプリケーションを実行できます。詳細については、[Amazon EC2 ユーザーガイド](#)のAmazon EC2 のセットアップ」(Windows ユーザーの場合) または[Amazon EC2 ユーザーガイド](#)の「[Amazon EC2 のセットアップ](#)」(Linux ユーザーの場合) を参照してください。Amazon EC2
- DRM で保護された、暗号化されているファイルを再生する機能を持ったプレーヤー。

#### Note

PlayReady DRM を適用した HLS を使用するには、独自のプレーヤーが必要です。

## DRM で保護されたストリーミングコンテンツの作成

DRM 用にファイルを準備するには、AWS KMS を新規または既存のパイプラインに関連付ける必要があります。

指定した を使用してパイプラインを設定するには、AWS KMS 「」を参照してください[Elastic Transcoder AWS KMS での の使用](#)。

以下の手順は、Elastic Transcoder コンソールの使用によって、DRM 用にファイルを暗号化する方法を示しています。

ファイルで DRM を使用するには

1. <https://console.aws.amazon.com/elastictranscoder/> で Elastic Transcoder コンソールを開きます。
2. ナビゲーションペインで、[Jobs] (ジョブ) をクリックし、新しいジョブを作成します。詳細については、「[Elastic Transcoder でジョブを作成する](#)」を参照してください。
3. [Encryption Parameters] を [None] に設定したままにします。
4. [Playlists] で、[Add Playlist] をクリックし、プレイリストのタイプとして [Smooth] または [HLSv3] のいずれかを選択します。
5. [PlayReady DRM] では [Enter Information] を選択します。
  - a. [License Acquisition Url (ライセンス取得 URL)] に、DRM サーバーの絶対パスを入力します。

- b. [Key ID (キー ID)] に、コンテンツキー ID を入力します。
  - c. [Encryption Key (暗号化キー)] に、base64 エンコードされたコンテンツキーを入力します。
  - d. [Encryption Key MD5 (暗号化キーの MD5)] に、コンテンツキーの base64 エンコードされた MD5 ハッシュを入力します。
- 例 [Encryption Initialization Vector (暗号化初期化ベクトル)] に、base64 エンコードされた初期化ベクトルを入力します。

 Note

コンテンツキーは、Elastic Transcoder によりジョブオブジェクトの一部として返されますが、保存はされません。お客様の責任において、DRM サーバーにコンテンツキーを保存する必要があります。

## ジョブのステータスの通知

Elastic Transcoder には、ジョブのステータスが変わると通知する機能があります。Elastic Transcoder では、以下のステータスに変わると通知が送信されるように設定できます。

- Progressing: Elastic Transcoder によるパイプライン内のジョブの処理が開始されました。
- Complete: Elastic Transcoder によるパイプライン内のジョブの処理が完了しました。
- Warning: Elastic Transcoder によるパイプライン内のジョブの処理中に警告が発生しました。
- Error: Elastic Transcoder によるパイプライン内のジョブの処理中にエラーが発生しました。

Elastic Transcoder から、Amazon Simple Notification Service (Amazon SNS) を使用して通知が送信されます。Amazon SNS には、HTTP エンドポイント、E メールアドレス、Amazon Simple Queue Service (Amazon SQS) キューへのメッセージ送信機能など、さまざまな通知オプションが用意されています。これらのオプションとその他のオプションの詳細については、「[Amazon Simple Notification Service デベロッパー ガイド](#)」を参照してください。

通知はイベント駆動型のアプリケーションの設計に便利です。通知を使用してジョブの完了を判断すると、ポーリングが不要になります。したがって、ポーリングが原因となることのある Limit Exceeded の例外 (HTTP ステータスコード 429) が発生しなくなります。

通知はパイプラインの作成または更新時に設定します。そのパイプラインにジョブを送信するたびに、Elastic Transcoder によって該当する通知が送信されます。

### Important

通知を変更すると、その変更はすぐに反映されます。通知の変更後に送信するジョブだけでなく、すでに送信したが Elastic Transcoder によってまだ処理が開始されていないジョブにも、変更は適用されます。

Elastic Transcoder API を使用して、ジョブのステータスの変更を通知するように Elastic Transcoder を設定する方法

1. 1 つ以上の Amazon SNS トピックを作成し、各トピックにサブスクライブします。詳細については、[Amazon Simple Notification Service ドキュメント](#)を参照してください。

使用する Amazon SNS トピックがすでにある場合は、このステップを省略できます。

2. パイプラインを作成するか、既存のパイプラインを更新します。変わったときに通知の対象にするステータス ([Progressing]、[Complete]、[Warning]、[Error]) ごとに、ステップ 1 で作成した該当する Amazon SNS トピックを指定します。詳細については「[パイプラインを作成する](#)」、「[パイプラインを更新する](#)」または「[パイプラインの通知を更新する](#)」を参照してください。
3. Elastic Transcoder にジョブを送信することで、[Progressing] ステータスと [Complete] ステータスについての通知をテストします。

Elastic Transcoder コンソールを使用して、通知を設定することもできます。コンソールを使用する場合は、パイプラインの作成または更新時に Amazon SNS トピックを作成できます。ただし、Amazon SNS コンソール、API、または CLI を使用して、新しいトピックにサブスクライブする必要はあります。

Elastic Transcoder によってステータスの変更の通知が送信される時、通知で返されるメッセージは以下の JSON 形式になります。メッセージには、ジョブのステータスだけでなく、各出力のステータスも含まれます。

```
{
  "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|Complete|Warning|Error"
    },
    {...}
  ],
  "playlists": [
    {
      applicable job playlists settings
    }
  ],
}
```

```
"userMetadata": {  
  "metadata key": "metadata value"  
}  
}
```

## クリップの結合

Elastic Transcoder で複数の入力ファイルの部分 (クリップ) を結合して 1 つの出力を作成できるようになりました。これにより、別々の録画セッションや試合のハイライトなど、複数の異なる入力から 1 つの出力を作成できます。

複数の入力のクリップを結合するには、ジョブの作成時に複数の入力を指定します。Elastic Transcoder では、ジョブリクエストで指定した順にクリップが結合されます。複数の入力を結合しないで変換する場合は、入力別にジョブを作成します。

指定する入力ごとに、開始時間と再生時間を指定することで、各入力の必要な部分のみを出力として結合できます。出力の先頭と末尾を切り詰める場合は、最初と最後の入力のクリップ設定を指定します。たとえば、出力の先頭のトレーラーと出力の末尾のクレジットを削除する場合は、トレーラーを削除する最初の入力の開始時間と再生時間を指定し、クレジットを削除する最後のクリップの開始時間と再生時間を指定します。

Elastic Transcoder でファイルを結合できるように、ファイル間で音声とビデオの設定を統一します。例えば、複数の入力間でビデオのアスペクト比が異なる場合は、プリセットでパディングポリシーやサイジング ポリシーを適用し、Elastic Transcoder で一貫したアスペクト比で出力を生成できるようにします。これらのポリシーとアスペクト比の関係については、「[サイズ設定のポリシーとアスペクト比](#)」を参照してください。また、動画はすべての入力に含まれているか、どの入力にも含まれていないこと、音声もすべての入力に含まれているか、どの入力にも含まれていないことが必要です。同様に、キャプション設定も統一する必要があります。入力間で共通する言語のキャプションには、同じ言語の値を指定します。そうしないと、同じキャプショントラックの一部とみなされません。

ウォーターマークの配置は、プリセットにコンテンツターゲットまたはフレーム ターゲットがあるかどうかによって決まります。コンテンツターゲットでは、ウォーターマークがオリジナルの入力のコンテンツに従って配置され、パディングポリシーで追加されたパディングは無視されます。たとえば、パディングが不要なクリップでウォーターマークが左上にある場合、そのクリップのウォーターマークは動画のエッジから指定した距離だけ離して配置されます。パディングが必要なクリップの場合、ウォーターマークはパディングのエッジから指定した距離だけ離して配置されます。フレームターゲットでは、ウォーターマークが出力のコンテンツに従って配置されます。ウォーターマークは、常に動画のエッジから指定した距離だけ離して配置されます。ウォーターマークの詳細については、「[ウォーターマークの設定](#)」を参照してください。

Elastic Transcoder ですべての入力が結合された後で、サムネイルが作成されます。

# 字幕

キャプションに対してある形式から別の形式へのトランスコードが実行されるように Elastic Transcoder を設定できます。Elastic Transcoder では、次の 2 種類のキャプションがサポートされます。

- **埋め込み:** 埋め込みキャプションはオーディオやビデオと同じファイルに含まれます。Elastic Transcoder では、埋め込みキャプションが言語ごとに 1 つだけサポートされます。

有効な入力形式は CEA-608 (EIA-608、最初の空白でないチャンネルのみ)、CEA-708 (EIA-708、最初の空白でないチャンネルのみ)、mov-text などです。

有効な出力形式は mov-text (MP4 のみ)、CEA-708 (MPEG-TS、MP4、29.97 および 30 フレーム/秒のみ) などです。CEA-708 キャプションはストリームの H.264 SEI ユーザーデータに埋め込まれています。

Elastic Transcoder では、埋め込み形式が出力ごとに 1 つだけサポートされます。

- **サイドカー:** サイドカーキャプションは、オーディオや動画のデータとは別のメタデータファイルに保存されます。サイドカーキャプションには、ビデオファイルとサイドカーファイルの関係を解釈できるプレイヤーが必要です。Elastic Transcoder では、言語あたり 1 つのみのサイドカーキャプションがサポートされ、ファイルあたり最大 20 のサイドカーキャプショントラックがサポートされます。

有効な入力形式は DFXP (最初の div 要素のみ)、EBU-TT、SCC、SMPT、SRT、TTML (最初の div 要素のみ)、WebVTT などです。

有効な出力形式は DFXP (最初の div 要素のみ)、SCC、SRT、WebVTT などです。

TTML または SMPTE-TT 互換のキャプションが必要な場合は、出力形式として DFXP を指定します。

Elastic Transcoder では OCR (光学式文字認識) がサポートされません。また、キャプションの有効な入力として画像を使用することはできず、オーディオのみのトランスコーディングには利用できません。Elastic Transcoder によるトランスコード処理中、テキストの書式設定 (イタリック体など) は維持されません。

Elastic Transcoder コンソール使用時のキャプションのトランスコードについては、「[出力詳細: パート 3、キャプションの設定](#)」を参照してください。Elastic Transcoder API 使用時のキャプショ

ンのトランスコードについては、キャプション要素で始まる [ジョブを作成する](#) API アクションのドキュメントを参照してください。

埋め込みファイルの詳細については、Wikipedia ページ「[字幕 \(キャプション\)](#)」を参照してください。

サイドカーファイルの詳細については、Wikipedia ページ「[メタデータプラットフォーム](#)」と「[サイドカーファイル](#)」を参照してください。

# ウォーターマーク

変換を実行するときに、ビデオ上に最大 4 つまでのグラフィックを重ねるように Elastic Transcoder を設定できます。このようなグラフィックはウォーターマークと呼ばれます。たとえば、放送中に画面上に表示されるテレビ局のロゴがウォーターマークです。

ウォーターマークは、変換された動画の再生中、指定したサイズ、位置、スケール、不透明度に基づいて表示されます。不透明度は、ウォーターマークの濃さを調整します。完全な透明 (ウォーターマークが表示されない状態) から完全な不透明 (動画の一部がウォーターマークによって覆い隠されている状態) までの調整ができます。

ウォーターマークに使用するグラフィックには、.png または .jpg 形式のものを使用できます。長方形以外のウォーターマークを表示する場合は、透明度をサポートする .png 形式を使用してください。グラフィックは、変換するビデオを保存するバケットと同じ Amazon S3 バケットに保存します。パイプラインの設定で、このバケット (入力バケット) を指定します。

ウォーターマークを追加するには、プリセットとジョブの両方で設定を指定します。

- プリセットには、最大 4 つのウォーターマークに関するサイズ、位置、不透明度の設定が含まれます。設定の各グループには ID を割り当てます。ジョブを作成するとき、この ID を使用することで、どのグラフィックに対してどのプリセットの設定を使用するかを制御できます。

Elastic Transcoder コンソールを使用してプリセットに対してウォーターマークの設定を指定する方法については、「[Elastic Transcoder でプリセットを作成する](#)」を参照してください。Elastic Transcoder API を使用して設定を指定する方法については、「[プリセットを作成する](#)」を参照してください。

- ジョブでは、ウォーターマークとして使用する .png または .jpg 画像、および各ウォーターマークに対して使用するプリセットに含まれる設定のグループを指定します。ジョブを作成し、ウォーターマークの設定を含んでいるプリセットを指定するとき、プリセットで定義したウォーターマークの設定の一部またはすべてに対してグラフィックを追加することができます。また、これらの設定のいずれに対してもグラフィックを追加しないこともできます。

Elastic Transcoder コンソールを使用してジョブに対してウォーターマークの設定を指定する方法については、「[Elastic Transcoder でジョブを作成する](#)」を参照してください。Elastic Transcoder API を使用して設定を指定する方法については、「[ジョブを作成する](#)」を参照してください。

## デジタル音声

デジタル音声は、保存、転送、およびチャンネルとして再生することができます。また、ストリームとも呼ばれるトラック単位でのマルチチャンネルの保存、転送および再生もできます。マルチチャンネルやトラックは、サラウンドサウンド、ステレオや多国語といった追加情報をファイルに含めることができます。たとえば、6つのチャンネルのサラウンドサウンドトラックおよび2つのチャンネルのステレオトラックがあるファイルを作成できます。Elastic Transcoder を使用して、ファイル内のトラックやチャンネル数を変更できます。ファイルにサラウンドサウンドの6チャンネルトラックのみがある場合には、Elastic Transcoder を使用してステレオの2チャンネルトラックのファイルを作成することができます。

シングルチャンネルは、単一スピーカーで再生された情報を保持します。たとえば、2つのチャンネルのステレオトラックは、1つのチャンネルを左スピーカーに送り、もう1つのチャンネルを右スピーカーに送ります。トラックを使って異なるチャンネルの組み合わせが選択できます。たとえば、1つのトラックには2つのチャンネル、そしてもう1つのトラックには6つのチャンネルがある2つのトラックのファイルを保持できます。ファイルを再生すると、ステレオスピーカーシステムには2チャンネルトラックが使用され、サラウンドスピーカーシステムの場合には6チャンネルトラックが使われることとなります。

すべてのトラックに音声情報があるわけではありません。音声のないトラックはミットアウトサウンド (MOS) トラックと呼ばれます。MOS トラックとサウンド編集プログラムを使って、ファイルに音響効果や音楽を挿入することができます。出力ファイルが MXF コンテナを使用する場合、Elastic Transcoder を使ってファイルに MOS トラックを追加できます。

Elastic Transcoder はデフォルトで単一トラック仕様となっていますが、トラック内のチャンネル数を選択できます。2つのトラックまで選択でき、また入力と同じチャンネル数に合わせることもできます。MXF コンテナでファイルのトランスコードを行うと、Elastic Transcoder で出力用に複数のトラックを作成することができます。

Elastic Transcoder のプリセットを使用して、以下を行うことができます。

- 出力から音声の削除
- モノラルからマルチチャンネルへの複製
- ステレオからモノラルへの組み合わせ (ダウンミックス)
- サラウンドサウンドからステレオまたはモノラルへのダウンミックス
- 現存するチャンネルへの移行
- ( MXF のみ ) MOS トラックを出力に追加

Elastic Transcoder を使用した音声のトランスコードについては、「[Channels](#)」を参照してください。

## システムプリセット

ジョブに使用するプリセットの Id オブジェクトの値。プリセットによって、変換時に Elastic Transcoder で使用される音声、ビデオ、サムネイルの各設定が決まります。作成したプリセットを使用するには、プリセットの作成時に Elastic Transcoder によって応答で返されたプリセット ID を指定します。

### Note

AAC プロファイルの追加前に任意のプリセットを作成した場合、それらのプリセットには Elastic Transcoder によって AAC-LC プロファイルが使用されます。

Elastic Transcoder に用意されているシステムプリセットのいずれかを使用するには、以下の ID を使用します (「[プリセットのリスト](#)」の手順を使用しても、これらの ID を取得できます)。

説明	PresetId
Audio AAC - 256k	1351620000001-100110
Audio AAC - 160k	1351620000001-100120
Audio AAC - 128k	1351620000001-100130
Audio AAC - 64k (Audio:CodecOptions:Profile には [auto] を使用)	1351620000001-100141
Audio MP3 - 320k	1351620000001-300010
Audio MP3 - 192k	1351620000001-300020
Audio MP3 - 160k	1351620000001-300030
Audio MP3 - 128k	1351620000001-300040
音声 WAV 44100 Hz、16 ビット	1351620000001-300300
音声 WAV 44100 Hz、8 ビット	1351620000001-300200

説明	PresetId
Amazon Kindle Fire HDX	1351620000001-100150
Amazon Kindle Fire HD 8.9	1351620000001-100090
Amazon Kindle Fire HD	1351620000001-100080
Apple TV 3G、Roku HD/2 XD	1351620000001-100060
Apple TV 2G	1351620000001-100050
FLAC - CD	1351620000001-300110
フル HD 1080i50	1351620000001-100180
フル HD 1080i60	1351620000001-100190
フル HD 1080i50 - XDCAM422	1351620000001-100230
フル HD 1080i60 - XDCAM422	1351620000001-100220
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 (Audio:CodecOptions:Profile には [auto] を使用)	1351620000001-000061
Gif (アニメーション)	1351620000001-100200
HLS v3 (Apple HTTP Live Streaming)、2メガビット/秒	1351620000001-200010

説明	PresetId
HLS v3 および v4 (Apple HTTP Live Streaming)、2 メガビット/秒、動画のみ	1351620000001-200015
HLS v3 (Apple HTTP Live Streaming)、1.5 メガビット/秒	1351620000001-200020
HLS v3 および v4 (Apple HTTP Live Streaming)、1.5 メガビット/秒、動画のみ	1351620000001-200025
HLS v3 (Apple HTTP Live Streaming)、1 メガビット/秒	1351620000001-200030
HLS v3 および v4 (Apple HTTP Live Streaming)、1 メガビット/秒、動画のみ	1351620000001-200035
HLS v3 (Apple HTTP Live Streaming)、600 キロビット/秒	1351620000001-200040
HLS v3 および v4 (Apple HTTP Live Streaming)、600 キロビット/秒、動画のみ	1351620000001-200045
HLS v3 (Apple HTTP Live Streaming)、400 キロビット/秒	1351620000001-200050
HLS v3 および v4 (Apple HTTP Live Streaming)、400 キロビット/秒、動画のみ	1351620000001-200055
HLS v3 および v4 音声、160k	1351620000001-200060
HLS v3 および v4 音声、64k (Audio:CodecOptions:Profile に auto を使用)	1351620000001-200071
iPhone 5、iPhone 4S、iPad 4G/3G、iPad mini、Samsung Galaxy S2/S3/Tab 2	1351620000001-100020
iPod touch、iPhone 3/1、iPod classic	1351620000001-100040
MPEG-DASH Audio 128 k	1351620000001-500060

説明	PresetId
MPEG-DASH ビデオ 600 k	1351620000001-500050
MPEG-DASH ビデオ 1200 k	1351620000001-500040
MPEG-DASH ビデオ 2400 k	1351620000001-500030
MPEG-DASH ビデオ 4800 k	1351620000001-500020
NTSC	1351620000001-100160
PAL	1351620000001-100170
スムーズストリーミング 3 メガビット/秒	1351620000001-400010
スムーズストリーミング 2 メガビット/秒	1351620000001-400020
スムーズストリーミング 1.5 メガビット/秒	1351620000001-400030
スムーズストリーミング 1 メガビット/秒	1351620000001-400040
スムーズストリーミング 800 キロビット/秒	1351620000001-400050
スムーズストリーミング 600 キロビット/秒	1351620000001-400060
スムーズストリーミング 500 キロビット/秒	1351620000001-400070
スムーズストリーミング 400 キロビット/秒	1351620000001-400080
Web: Facebook、SmugMug、Vimeo、YouTube	1351620000001-100070
ウェブ: Flash ビデオ	1351620000001-100210
Webm 720p	1351620000001-100240
Webm VP9 720p	1351620000001-100250
Webm VP9 360p	1351620000001-100260

以下のシステムプリセットはサポートされなくなっており、使用はお勧めしません。

説明	非推奨 PresetId	推奨 PresetId
Audio AAC - 64k (Audio:CodecOptions:Profile には [AAC-LC] を使用)	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	1351620000001-100080
Amazon Kindle Fire	1351620000001-100100	1351620000001-100080
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
Generic 320x240 (Audio:CodecOptions:Profile には [AAC-LC] を使用)	1351620000001-000060	1351620000001-000061
Generic 320x240	1351620000000-000060	1351620000001-000060
HLS オーディオ、64k (Audio:CodecOptions)	1351620000001-200070	1351620000001-200071

説明	非推奨 PresetId	推奨 PresetId
:Profile には [AAC-LC] を使用)		
iPhone 4、iPod touch 5G/4G、iPad 2G/1G	1351620000001-100010	1351620000001-100020
iPhone4S	1351620000000-100020	1351620000001-100020
iPhone4	1351620000000-100010	1351620000001-100020
iPhone 3GS	1351620000001-100030	1351620000001-100020
iPhone3GS	1351620000000-100030	1351620000001-100020
iPod Touch	1351620000000-100040	1351620000001-100040
Web	1351620000000-100070	1351620000001-100070

## サイズ設定のポリシーとアスペクト比

選択した [サイジングポリシー] は、次の表に示すように、Elastic Transcoder によって出カイメージに適用されるスケーリングに影響します。

Sizing Policy	出カイメージは 拡張される場合 があります	パディングポリ シーが「Pad」 の場合、出カイ メージには余白 が入る場合があ ります	出カイメージは 入カイメージと 異なるピクセル のアスペクト比 となる場合があ ります	出カイメージは クロップされる 場合があります
Fit	あり	あり		
Fill	あり			あり
[Stretch] (スト レッチ)	あり		あり	
Keep		あり		あり
ShrinkToFit に設 定します。		はい		
ShrinkToFill		あり		あり

## サムネイルのアスペクト比

次の表では、[サイジングポリシー]、[パディングポリシー]、[Max Height] (最大高さ)、[Max Width] (最大幅) が出カイメージの変更にどのように影響するかについて示しています

トピック

- [Fit](#)
- [Fill](#)
- [\[Stretch\] \(ストレッチ\)](#)
- [Keep](#)

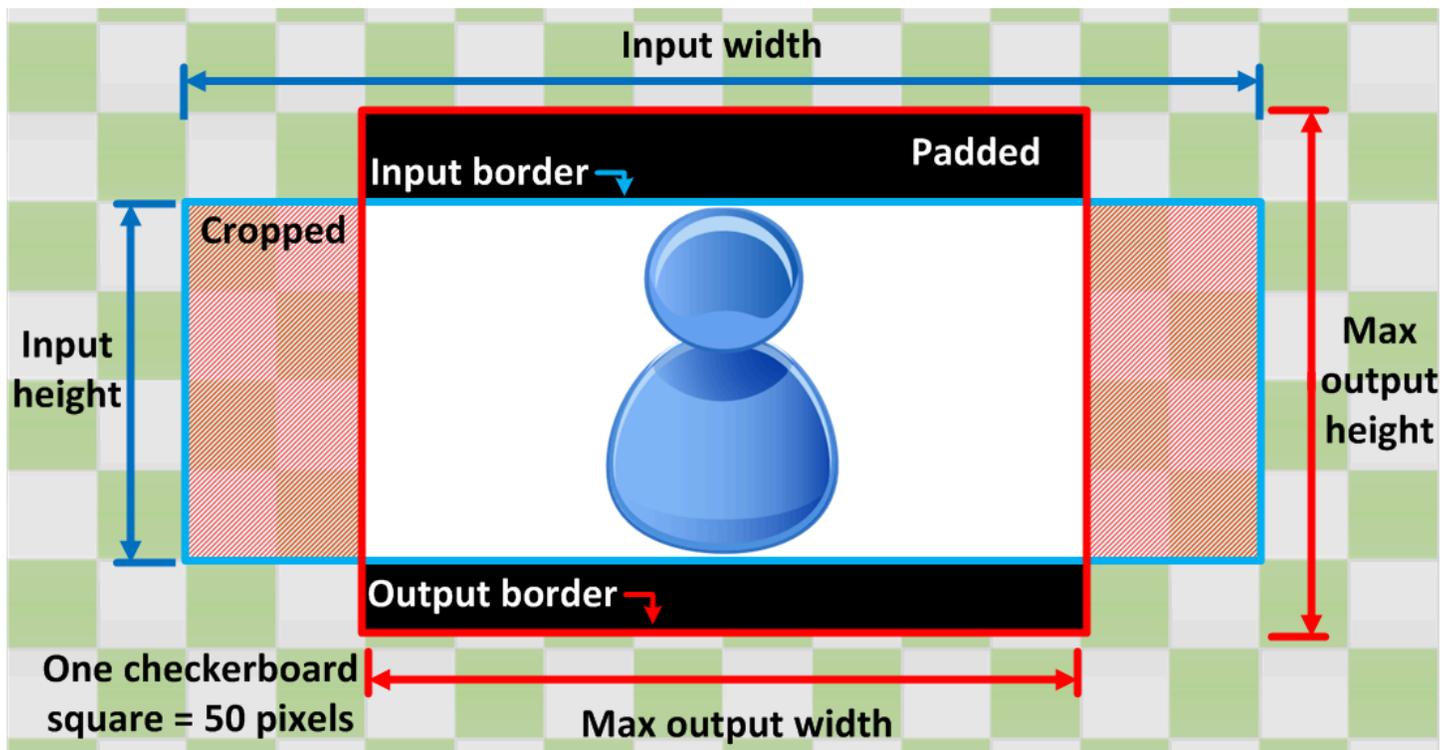
- [フィットするまで縮小する](#)
- [圧縮して充填する](#)

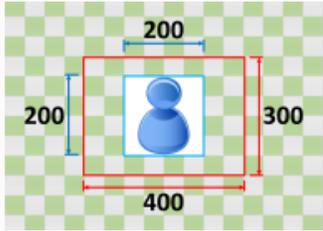
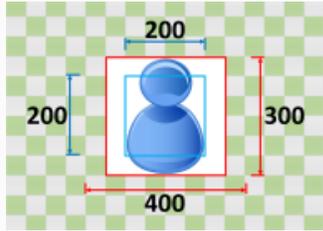
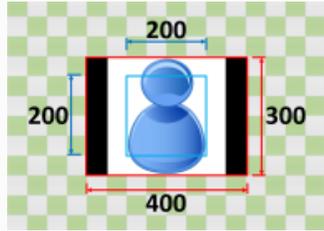
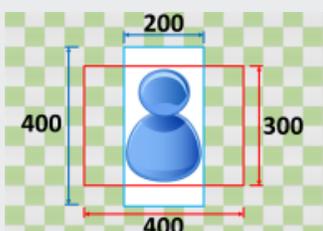
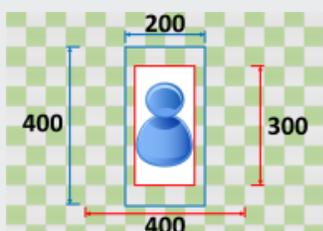
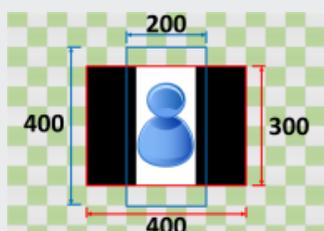
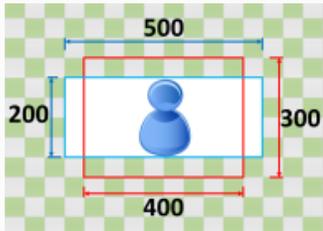
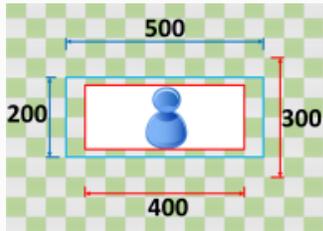
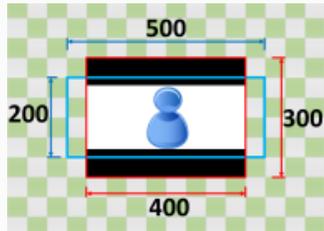
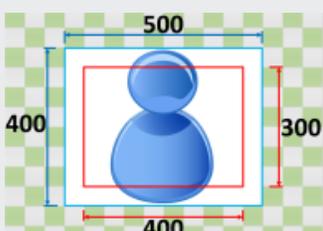
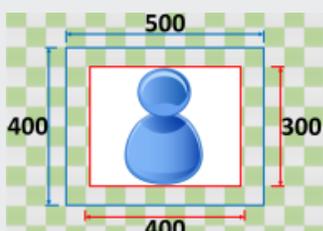
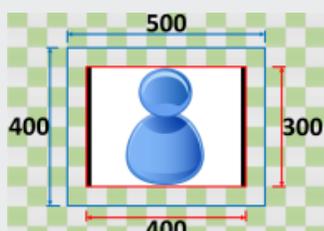
## Fit

サイジングポリシーに対して [Fit] (適合) を選択すると、入力ファイルが出力イメージのディメンション内に収まる (出力イメージのディメンションを超えない) まで、Elastic Transcoder により入力ファイルのスケールが調整されます。

例えば、入力ファイルが 200 ピクセル x 200 ピクセルで、300 ピクセル x 400 ピクセルの出力イメージが欲しい場合、Elastic Transcoder により、ファイルのサイズが 300 ピクセル x 300 ピクセルに拡大され、パディングポリシーがファイルに適用されます。パディングポリシーに対して [Unpadded] (パディングなし) を選択すると、Elastic Transcoder により 300 ピクセル x 300 ピクセルのファイルが出力として返されます。[Padded] (パディングあり) を選択した場合には、Elastic Transcoder により出力の両側に 50 ピクセルのパディングが追加され、300 ピクセル x 400 ピクセルファイルが返されます。

キー



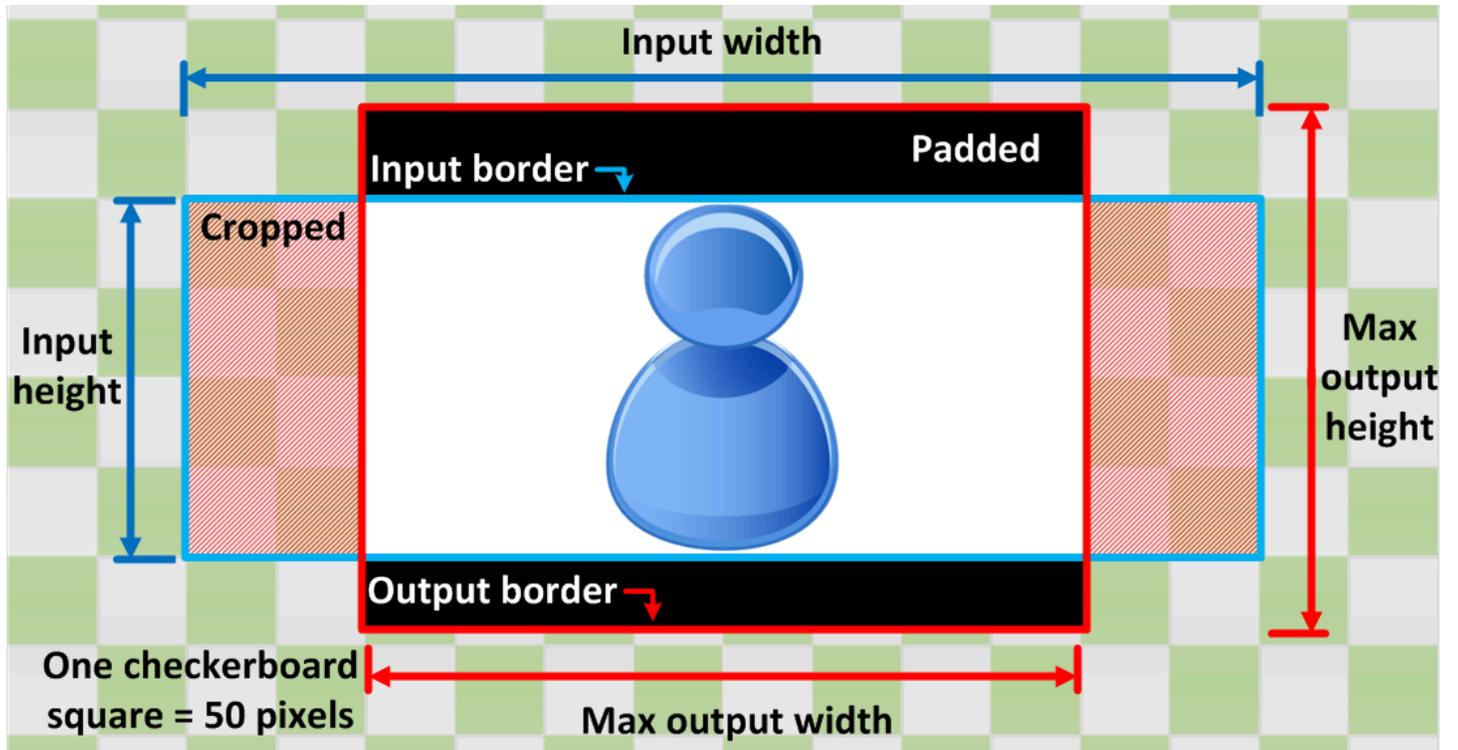
条件	Input	出力 : NoPad	出力 : Pad
入力幅 [<] 出力の最大幅 入力縦 [<] 出力の最大縦			
入力幅 [<] 出力の最大幅 入力縦 [>] 出力の最大縦			
入力幅 [>] 出力の最大幅 入力縦 [<] 出力の最大縦			
入力幅 [>] 出力の最大幅 入力縦 [>] 出力の最大縦			

## Fill

サイジングポリシーに対して [Fill] (充填) を選択すると、Elastic Transcoder により、入力ファイルが出カイメージのディメンション内に収まるように入力ファイルが調整され、出カイメージのディメンションを超えた部分がトリミングされます。

例えば、入力ファイルが 200 ピクセル x 200 ピクセルであり 300 ピクセル x 400 ピクセルの出カイメージが欲しい場合、Elastic Transcoder により、入力のサイズが 400 ピクセル x 400 ピクセルに拡大され、上部と下部の 50 ピクセルがトリミングされ、300 ピクセル x 400 ピクセルファイルが返されます。Elastic Transcoder では、フィルポリシーにパディングは使用されません。

キー



条件	Input	出力 : NoPad	出力 : Pad
入力幅 [ $<$ ] 出力の最大幅 入力縦 [ $<$ ] 出力の最大縦			
入力幅 [ $>$ ] 出力の最大幅 入力縦 [ $>$ ] 出力の最大縦			

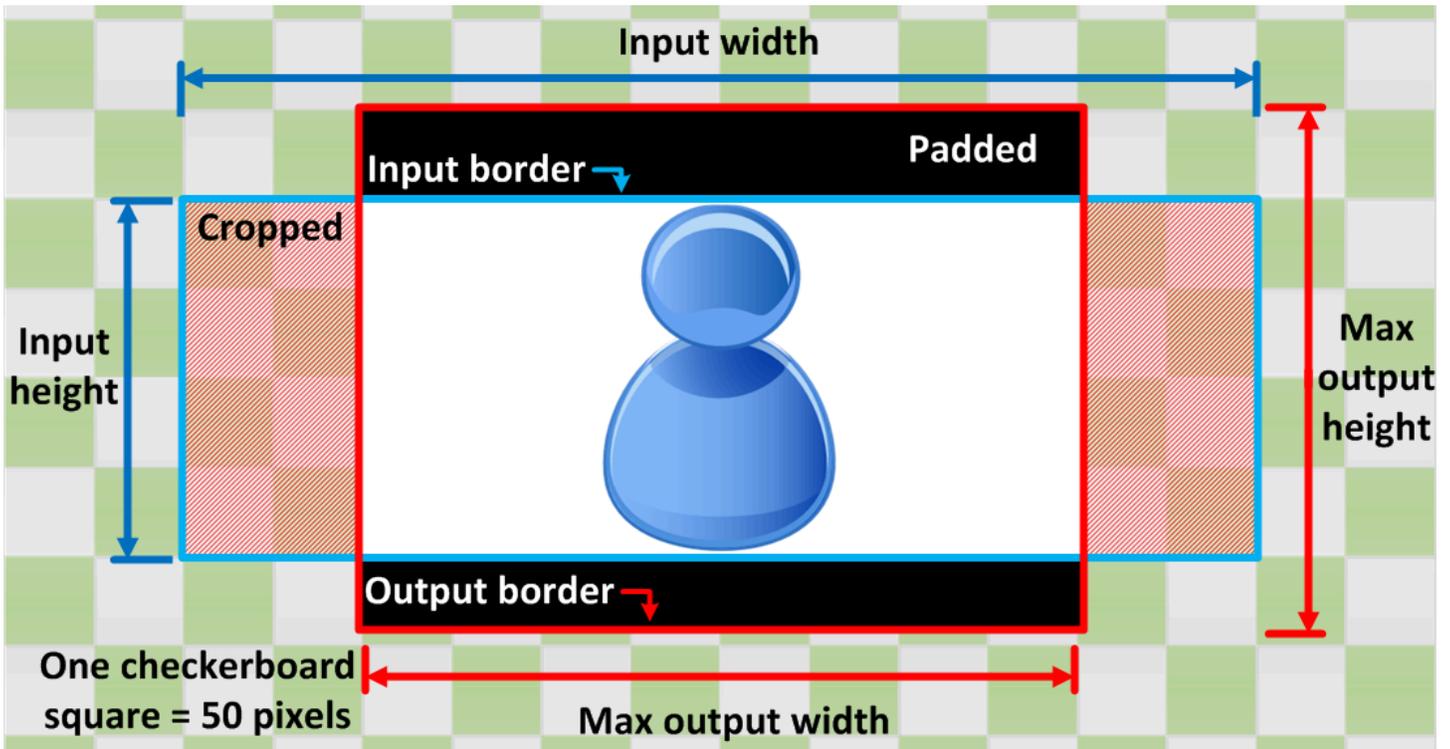
条件	Input	出力 : NoPad	出力 : Pad
入力幅 [>] 出力の最大幅 入力縦 [<] 出力の最大縦			
入力幅 [>] 出力の最大幅 入力縦 [>] 出力の最大縦			

## [Stretch] (ストレッチ)

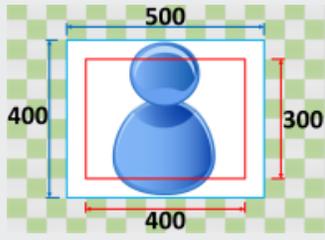
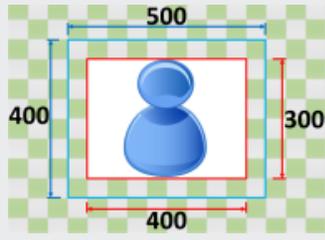
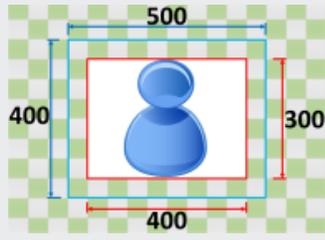
[サイジングポリシー] で [引き伸ばし] を選択した場合、Elastic Transcoder により、出力ファイルのディメンションと一致するまで入力ファイルが拡大または縮小されます。

例えば、入力ファイルが 200 ピクセル x 200 ピクセルで、300 ピクセル x 400 ピクセルの出力イメージが欲しい場合、Elastic Transcoder により入力のサイズが 300 ピクセル x 400 ピクセルに拡大されるため、出力イメージのバランスがゆがみます。Elastic Transcoder では、引き伸ばしポリシーにパディングとトリミングは使用されません。

キー



条件	Input	出力 : NoPad	出力 : Pad
入力幅 [ $\leq$ ] 出力の最大幅 入力縦 [ $\leq$ ] 出力の最大縦			
入力幅 [ $\leq$ ] 出力の最大幅 入力縦 [ $>$ ] 出力の最大縦			
入力幅 [ $>$ ] 出力の最大幅 入力縦 [ $\leq$ ] 出力の最大縦			

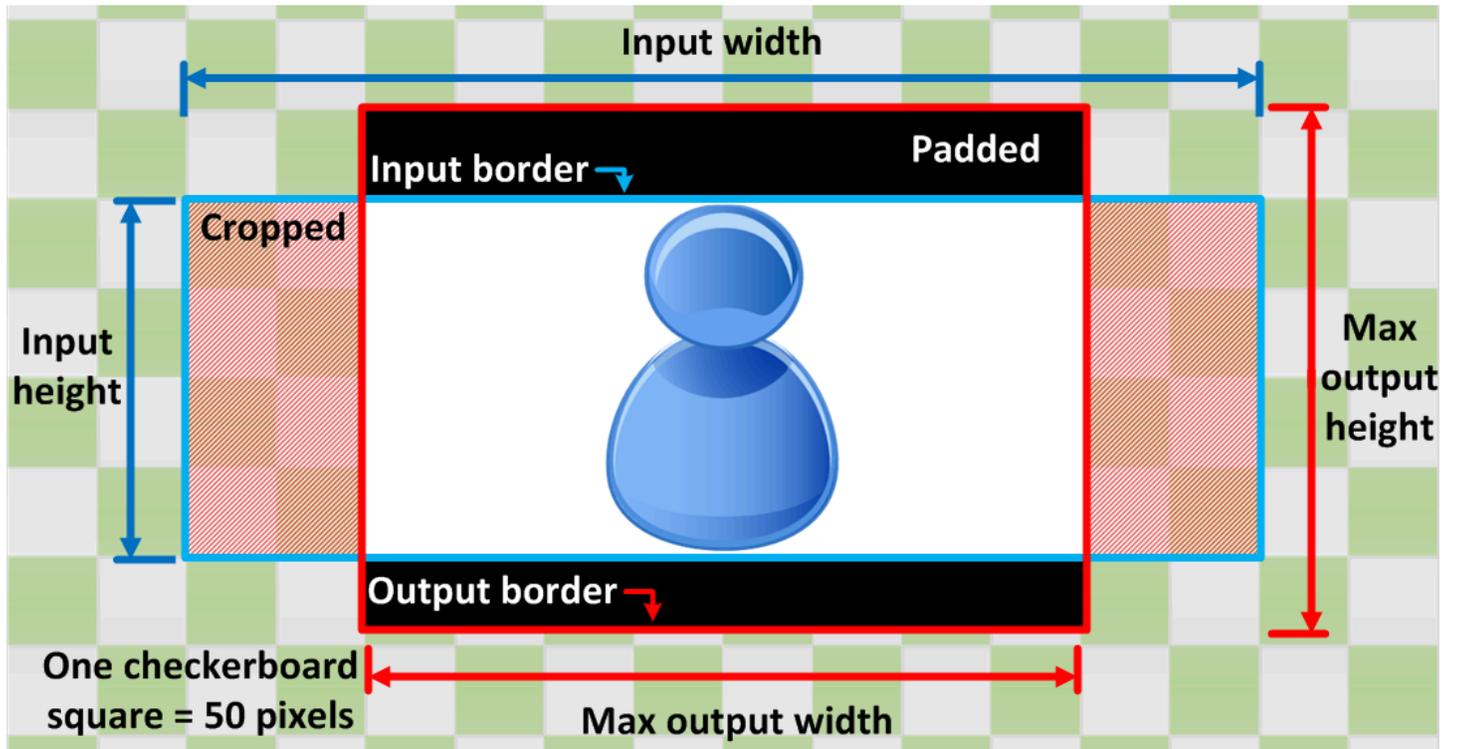
条件	Input	出力 : NoPad	出力 : Pad
入力幅 [>] 出力の最大幅 入力縦 [>] 出力の最大縦			

## Keep

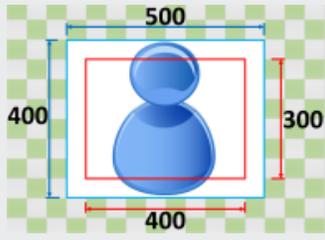
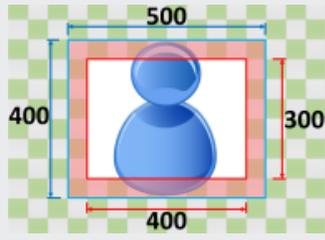
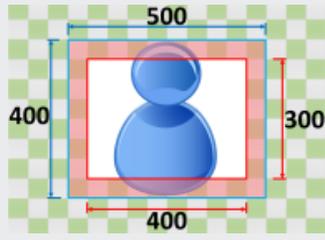
サイジングポリシーに対して [保持] を選択した場合、Elastic Transcoder で入力ファイルのスケールが調整されません。Elastic Transcoder では、入力ファイルが出力イメージのディメンションと一致するまで、入力ファイルのトリミングまたはパディングが行われます。

例えば、入力ファイルが 400 ピクセル x 200 ピクセルで、300 ピクセル x 300 ピクセルの出力イメージが欲しい場合、Elastic Transcoder により、上部と下部の 100 ピクセルがトリミングされ、パディングポリシーが両サイドに適用されます。パディングポリシーに対して [Unpadded] (パディングなし) を選択すると、Elastic Transcoder により 300 ピクセル x 200 ピクセルの出力ファイルが返されます。[Padded] (パディングあり) を選択した場合、Elastic Transcoder により 300 ピクセル x 300 ピクセルのファイルが返されます。

キー



条件	Input	出力 : NoPad	出力 : Pad
入力幅 [ $<$ ] 出力の最大幅 入力縦 [ $<$ ] 出力の最大縦			
入力幅 [ $<$ ] 出力の最大幅 入力縦 [ $>$ ] 出力の最大縦			
入力幅 [ $>$ ] 出力の最大幅 入力縦 [ $<$ ] 出力の最大縦			

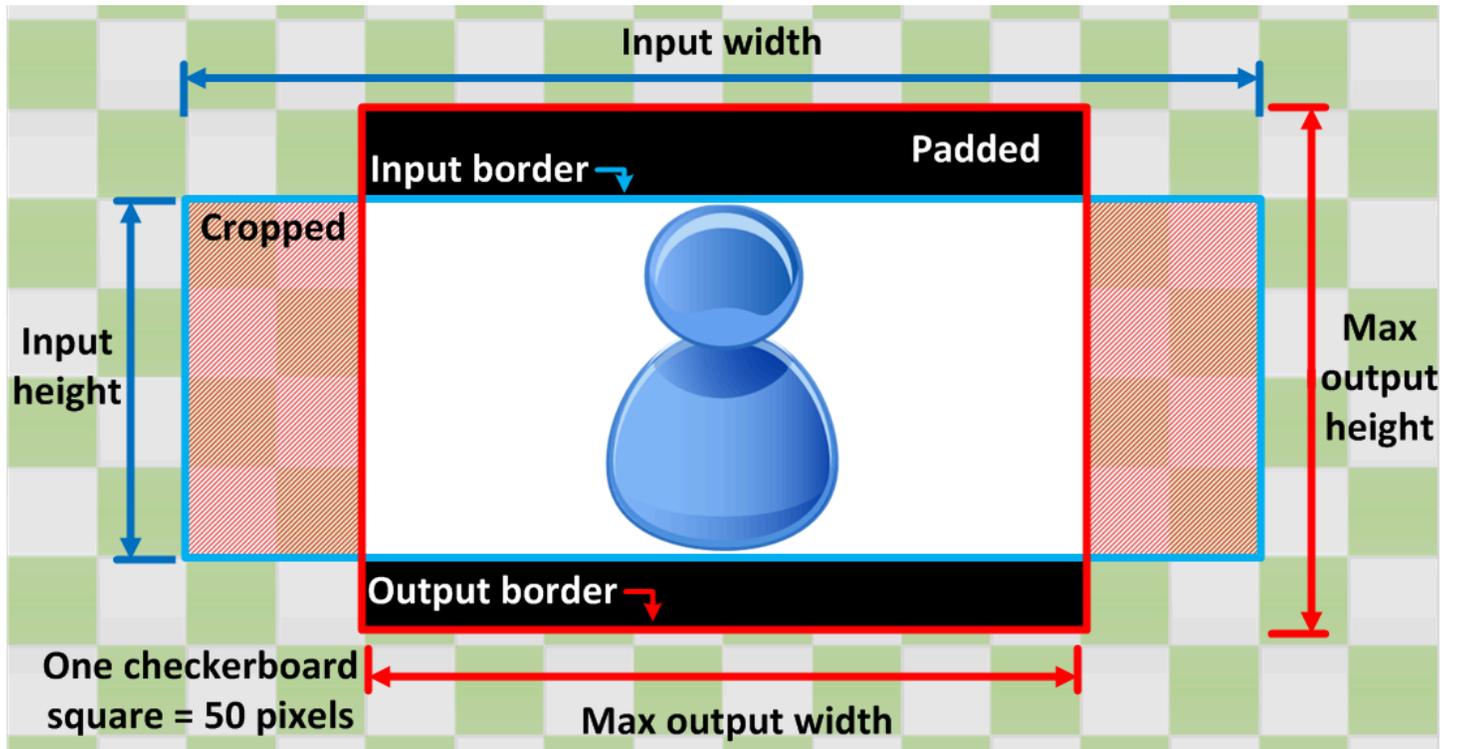
条件	Input	出力 : NoPad	出力 : Pad
入力幅 [>] 出力の最大幅 入力縦 [>] 出力の最大縦			

## フィットするまで縮小する

サイジングポリシーに対して [Shrink to Fit] (適合するまで縮小) を選択すると、入力ファイルが出力ファイルのディメンション内に収まる (どの部分も出カイメージのディメンションを超えない) まで、Elastic Transcoder により入力ファイルのサイズが縮小されます。入力ファイルが出カイメージより小さい場合、Elastic Transcoder によりファイルのサイズが拡大されません。

例えば、入力ファイルが 400 ピクセル x 400 ピクセルで、200 ピクセル x 300 ピクセルの出カイメージが欲しい場合、Elastic Transcoder により、入力が 200 ピクセル x 200 ピクセルに縮小され、パディングポリシーが適用されます。パディングポリシーに対して [Unpadded] (パディングなし) を選択すると、Elastic Transcoder により 200 x 200 ピクセルのファイルが出力として返されます。[Padded] (パディングあり) を選択した場合には、Elastic Transcoder により出力の両側に 50 ピクセルのパディングが追加され、300 ピクセル x 300 ピクセルファイルが返されます。

キー



条件	Input	出力 : NoPad	出力 : Pad
入力幅 [ $<$ ] 出力の最大幅 入力縦 [ $<$ ] 出力の最大縦			
入力幅 [ $<$ ] 出力の最大幅 入力縦 [ $>$ ] 出力の最大縦			
入力幅 [ $>$ ] 出力の最大幅 入力縦 [ $<$ ] 出力の最大縦			

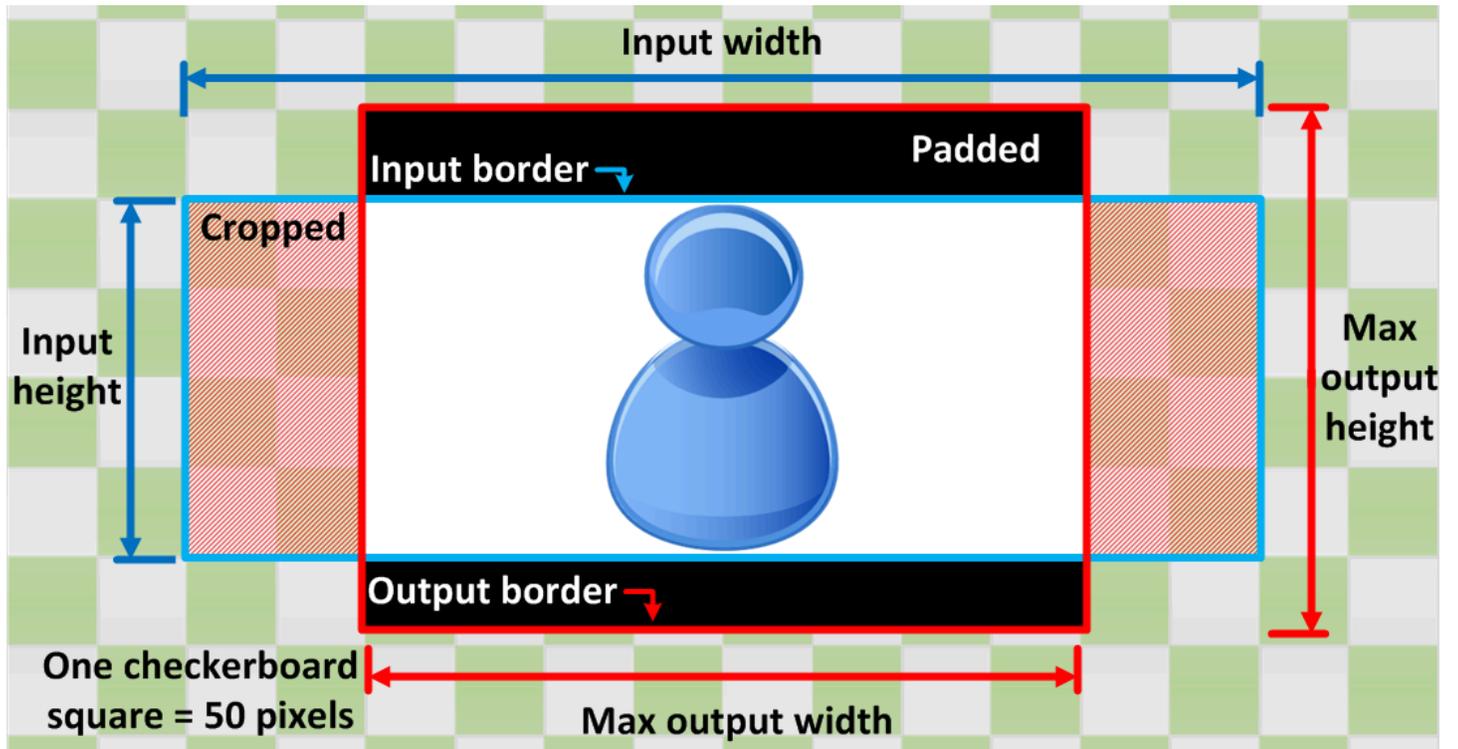
条件	Input	出力 : NoPad	出力 : Pad
入力幅 [>] 出力の最大幅 入力縦 [>] 出力の最大縦			

## 圧縮して充填する

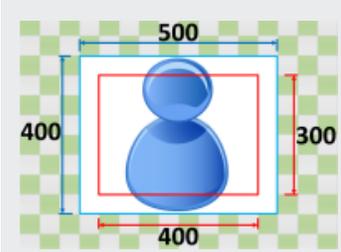
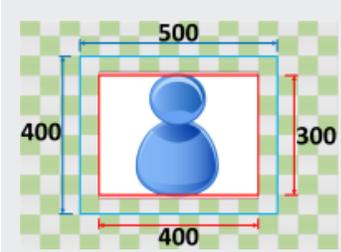
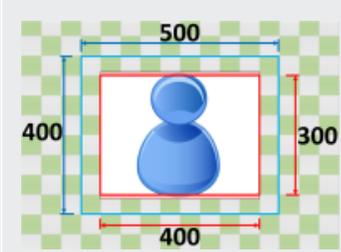
サイジングポリシーに対して [Shrink to Fill] (充填されるまで縮小) を選択すると、Elastic Transcoder により、出カイメージのディメンションが入カファイルで埋まるまで入カファイルのサイズが縮小され、出カイメージのディメンション内に収まらない部分がトリミングされ、パディングポリシーが適用されます。出カイメージが入カファイルより大きい場合、Elastic Transcoder で入カファイルのサイズが拡大されません。

例えば、入カファイルが 400 ピクセル x 200 ピクセルで、200 ピクセル x 300 ピクセルの出カイメージが欲しい場合、Elastic Transcoder により、ファイルの両サイドの 100 ピクセルがトリミングされ、パディングポリシーがファイルの上部と下部に適用されます。パディングポリシーに対して [Unpadded] (パディングなし) を選択すると、Elastic Transcoder により 200 ピクセル x 200 ピクセルの出カファイルが返されます。[Padded] (パディングあり) を選択した場合、Elastic Transcoder により 200 ピクセル x 300 ピクセルのファイルが返されます。

キー



条件	Input	出力 : NoPad	出力 : Pad
入力幅 [ $<$ ] 出力の最大幅 入力縦 [ $<$ ] 出力の最大縦			
入力幅 [ $<$ ] 出力の最大幅 入力縦 [ $>$ ] 出力の最大縦			
入力幅 [ $>$ ] 出力の最大幅 入力縦 [ $<$ ] 出力の最大縦			

条件	Input	出力 : NoPad	出力 : Pad
入力幅 [>] 出力の最大幅 入力縦 [>] 出力の最大縦			

# Elastic Transcoder のモニタリング

モニタリングは、AWS ソリューションの信頼性、可用性、Elastic Transcoder のパフォーマンスを維持する上で重要な役割を果たします。マルチポイント障害が発生した場合は、その障害をより簡単にデバッグできるように、AWS ソリューションのすべての部分からモニタリングデータを収集する必要があります。ただし、Elastic Transcoder のモニタリングをスタートする前に、以下の質問に対する回答を含めたモニタリング計画を作成する必要があります。

- モニタリングの目的は何ですか？
- どのリソースをモニタリングしますか？
- どのくらいの頻度でこれらのリソースをモニタリングしますか？
- どのモニタリングツールを利用しますか？
- 誰がモニタリングタスクを実行しますか？
- 問題が発生したときに誰が通知を受け取りますか？

次のステップでは、さまざまなタイミングと負荷条件でパフォーマンスを測定することにより、お客様の環境で通常の Elastic Transcoder のパフォーマンスのベースラインを確定します。Elastic Transcoder のモニタリングでは、過去のモニタリングデータを保存し、現在のパフォーマンスデータと比較することで、パフォーマンスの通常パターンと異常パターンを特定し、問題に対処する方法を考案できます。

例えば、Amazon EC2 を使用している場合は、インスタンスの CPU 使用率、ディスク I/O、ネットワーク使用率をモニタリングできます。確立したベースラインからパフォーマンスが外れた場合は、インスタンスの再設定または最適化を行って CPU 使用率の抑制、ディスク I/O の改善、またはネットワークトラフィックの低減を行うことが必要な場合があります。

ベースラインを確立するには、少なくとも、次の項目をモニタリングする必要があります。

- 完了したジョブ
- エラーのあるジョブ
- StandbyTime
- エラー
- Throttles

トピック

- [モニタリングツール](#)
- [Amazon CloudWatch でのモニタリング](#)

## モニタリングツール

AWS では、Elastic Transcoder のモニタリングに使用できるさまざまなツールを提供しています。これらのツールの一部はモニタリングを行うように設定できますが、一部のツールは手動による介入が必要です。モニタリングタスクをできるだけ自動化することをお勧めします。

### 自動モニタリングツール

以下の自動化されたモニタリングツールを使用して、Elastic Transcoder を監視し、問題が発生したときにレポートできます。

- Amazon CloudWatch アラーム - 指定した期間にわたって単一のメトリクスをモニタリングし、複数の期間にわたる特定のしきい値に対するメトリクスの値に基づいて 1 つ以上のアクションを実行します。アクションは、Amazon Simple Notification Service (Amazon SNS) のトピックまたは Amazon EC2 Auto Scaling のポリシーに送信される通知です。CloudWatch アラームは、特定の状態にあるという理由だけでアクションを呼び出すことはありません。状態が変更され、指定された期間維持されている必要があります。詳細については、「[Amazon CloudWatch でのモニタリング](#)」を参照してください。
- Amazon CloudWatch Logs – AWS CloudTrail またはその他のソースのログファイルのモニタリング、保存、アクセスを行います。詳細については、「Amazon CloudWatch ユーザーガイド」の「[ログファイルのモニタリング](#)」を参照してください。
- AWS CloudTrail ログモニタリング – アカウント間でログファイルを共有し、CloudWatch Logs に送信CloudWatch CloudTrail ログファイルをリアルタイムでモニタリングし、Java でログ処理アプリケーションを書き込み、CloudTrail による配信後にログファイルが変更されていないことを確認します。詳細については、「AWS CloudTrail ユーザーガイド」の「[CloudTrail ログファイルの使用](#)」を参照してください。

### 手動モニタリングツール

Elastic Transcoder のモニタリングでもう 1 つ重要な点は、CloudWatch のアラームの対象外のディメンションを手動でモニタリングすることです。Elastic Transcoder、CloudWatch、その他の AWS コンソールのダッシュボードには、AWS 環境の状態が一目でわかるように表示されます。CloudTrail のログ ファイルをチェックすることもお勧めします。

- Elastic Transcoder ダッシュボードには、以下が表示されます。
  - パイプラインとそのステータス
- CloudWatch ホームページには、次の内容が表示されます。
  - 現在のアラームとステータス
  - アラームとリソースのグラフ
  - サービスのヘルスステータス

また、CloudWatch を使用して以下のことを行えます。

- 重視するサービスをモニタリングするための[カスタマイズしたダッシュボード](#)を作成します
- メトリクスデータをグラフ化して、問題のトラブルシューティングを行い、傾向を確認する
- AWS リソースのすべてのメトリクスを検索して、参照する
- 問題があることを通知するアラームを作成/編集する

## Amazon CloudWatch でのモニタリング

CloudWatch を使用してパイプラインとオペレーションをモニタリングすることで、Elastic Transcoder から生データを収集し、リアルタイムに近い読み取り可能なメトリクスに加工することができます。これらの統計は 2 週間記録されるため、履歴情報にアクセスしてウェブアプリケーションまたはサービスの動作をよりの確に把握することができます。デフォルトでは、Elastic Transcoder のメトリクスデータは 1 分間隔で自動的に CloudWatch に送信されます。詳細については、Amazon CloudWatch ユーザーガイドの [Amazon CloudWatch](#)、[Amazon CloudWatch Events](#)、[Amazon CloudWatch Logs とは](#) をご覧ください。

トピック

- [Elastic Transcoder のメトリクスとディメンション](#)
- [Elastic Transcoder メトリクスの使用方法](#)
- [CloudWatch アラームを作成して Elastic Transcoder をモニタリングする](#)

## Elastic Transcoder のメトリクスとディメンション

ジョブを作成するとき、Elastic Transcoder により以下のメトリクスとディメンションが 1 分間隔で CloudWatch に送信されます。Elastic Transcoder のメトリクスを表示するには、以下の手順を使用できます

CloudWatch コンソールを使用してメトリクスを表示するには

メトリクスはまずサービスの名前空間ごとにグループ化され、次に各名前空間内のさまざまなディメンションの組み合わせごとにグループ化されます。

1. CloudWatch コンソール (<https://console.aws.amazon.com/cloudwatch/>) を開きます。
2. 必要に応じてリージョンを変更します。ナビゲーションバーから、AWS リソースがあるリージョンを選択します。詳細については、「[リージョンとエンドポイント](#)」を参照してください。
3. ナビゲーションペインでメトリクスを選択します。
4. [カテゴリ別の CloudWatch メトリクス] ペインの Elastic Transcoder のメトリクスカテゴリで、メトリクスカテゴリを選択します。次に、上のペインで下にスクロールしてメトリクスを一覧表示します。

AWS CLI を使用してメトリクスを表示するには

- コマンドプロンプトで、次のコマンドを使用します。

```
aws cloudwatch list-metrics --namespace "AWS/ElasticTranscoder"
```

CloudWatch には Elastic Transcoder に関する以下のメトリクスが表示されます。

## Elastic Transcoder のディメンションとメトリクス

Elastic Transcoder から Amazon CloudWatch に送信されるメトリクスとディメンションを以下に示します。

### Elastic Transcoder メトリクス

AWS/ElasticTranscoder 名前空間には、次のメトリクスが含まれます。

メトリクス	説明
Billed HD Output	パイプラインの HD 出力の課金可能な秒数。  有効なディメンション: Pipelineld  単位: 秒

メトリクス	説明
Billed SD Output	パイプラインの SD 出力の課金可能な秒数。 有効なディメンション: PipelineId 単位: 秒
Billed Audio Output	パイプラインのオーディオ出力の課金可能な秒数。 有効なディメンション: PipelineId 単位: 秒
Jobs Completed	このパイプラインで完了したジョブの数。 有効なディメンション: PipelineId 単位: 数
Jobs Errored	指定した入力バケットにないファイルのトランスコードを行うためのリクエストなど、無効な入力のために失敗したジョブの数。 有効なディメンション: PipelineId 単位: 数
Outputs per Job	Elastic Transcoder によりジョブに対して作成された出力の数。 有効なディメンション: PipelineId 単位: 数
Standby Time	Elastic Transcoder でジョブのトランスコードがスタートされるまでの秒数。 有効なディメンション: PipelineId 単位: 秒

メトリクス	説明
Errors	<p>ジョブ ID を含まないジョブステータスへのリクエストなど、無効なオペレーション パラメーターによって発生したエラーの数。</p> <p>有効なディメンション: Operation</p> <p>単位: 数</p>
Throttles	<p>Elastic Transcoder でオペレーションが自動的に調整された回数。</p> <p>有効なディメンション: Operation</p> <p>単位: 数</p>

## Elastic Transcoder メトリクスのディメンション

Elastic Transcoder メトリクスは、Elastic Transcoder 名前空間を使用し、以下のディメンションのメトリクスを提供しています。

ディメンション	説明
PipelineId	パイプラインの ID。このディメンションを指定すると、Elastic Transcoder パイプラインのためにリクエストしたデータがフィルタリングされます。
Operation	このディメンションを指定すると、Elastic Transcoder から提供される API のためにリクエストしたデータがフィルタリングされます。

## Elastic Transcoder メトリクスの使用方法

Elastic Transcoder によってレポートされるメトリクスが提供する情報は、さまざまな方法で分析できます。以下のリストは、メトリクスの一般的な利用方法をいくつか示しています。ここで紹介するのは使用開始するための提案事項であり、総括的な一覧ではありません。

- [ジョブが開始するまでの待機時間を追跡することはできますか？](#)
- [エラーのためジョブが失敗したときに、通知されるようにするには？](#)
- [ジョブが完了したときに、通知されるようにするには？](#)

## CloudWatch アラームを作成して Elastic Transcoder をモニタリングする

アラームの状態が変わったら、Amazon SNS メッセージを送信する Amazon CloudWatch のアラームを作成することができます。アラームは、指定期間にわたって単一のメトリクスを監視し、指定したしきい値に対応したメトリクスの値に基づいて、期間数にわたって1つ以上のアクションを実行します。アクションは、Amazon SNS のトピックまたはオートスケーリングのポリシーに送信される通知です。アラームは、持続している状態変化に対してのみアクションを呼び出します。CloudWatch のアラームは、メトリクスが特定の状態にあるだけではアクションを呼び出しません。アクションを呼び出すには、指定した期間継続している必要があります。

ジョブが開始するまでの待機時間を追跡することはできますか？

Standby Time メトリクスを使用して、パイプラインでジョブのトランスコーディングを開始するまでにかかる時間を追跡できます。

1. にサインイン AWS マネジメントコンソールし、<https://console.aws.amazon.com/cloudwatch/> で CloudWatch コンソールを開きます。
2. アラームの作成(アラームの作成) を選択します。これにより、[Create Alarm Wizard] が起動します。
3. [AWS/ElasticTranscoder Metrics] (AWS/ElasticTranscoder メトリクス) を選択し、Elastic Transcoder メトリクスをスクロールして [Standby Time] (スタンバイ時間) メトリクスを見つけます。アラームを作成する [Standby Time] (スタンバイ時間) メトリクスを選択し、[Next] をクリックします。
4. [Name] (名前)、[Description] (説明)、[Whenever] (いつでも) のそれぞれにメトリクスの値を入力します。
5. [アクション] で、既存の Amazon SNS トピックを選択します。

Amazon SNS 通知の詳細については、「Amazon Simple Notification Service デベロッパーガイド」の「[トピックの作成](#)」を参照してください。

6. アラームの作成(アラームの作成) を選択します。

## エラーのためジョブが失敗したときに、通知されるようにするには？

Job Errored メトリクスを使用して、無効な入力パラメーターによってパイプラインで失敗したジョブの数を追跡できます。

1. にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/cloudwatch/> で CloudWatch コンソールを開きます。
2. アラームの作成(アラームの作成) を選択します。これにより、[Create Alarm Wizard] が起動します。
3. [AWS/ElasticTranscoder Metrics] (AWS/ElasticTranscoder メトリクス) を選択し、Elastic Transcoder メトリクスをスクロールして [Job Errored] (エラーが発生したジョブ) メトリクスを見つけます。アラームを作成する [Job Errored] (エラーが発生したジョブ) メトリクスを選択し、[Next] をクリックします。
4. [Name] (名前)、[Description] (説明)、[Whenever] (いつでも) のそれぞれにメトリクスの値を入力します。
5. [アクション] で、既存の Amazon SNS トピックを選択します。

Amazon SNS 通知の詳細については、「Amazon Simple Notification Service デベロッパーガイド」の「[トピックの作成](#)」を参照してください。

6. アラームの作成(アラームの作成) を選択します。

## ジョブが完了したときに、通知されるようにするには？

Jobs Completed メトリクスを使用して、パイプラインがトランスコードされたジョブの数を追跡できます。

1. にサインイン AWS マネジメントコンソール し、 <https://console.aws.amazon.com/cloudwatch/> で CloudWatch コンソールを開きます。
2. アラームの作成(アラームの作成) を選択します。これにより、[Create Alarm Wizard] が起動します。
3. [AWS/ElasticTranscoder Metrics] (AWS/ElasticTranscoder メトリクス) を選択し、Elastic Transcoder メトリクスをスクロールして [Jobs Completed] (完了したジョブ) メトリクスを見つけます。アラームを作成する [Jobs Completed] (完了したジョブ) メトリクスを選択し、[Next] をクリックします。
4. [Name] (名前)、[Description] (説明)、[Whenever] (いつでも) のそれぞれにメトリクスの値を入力します。

5. [アクション] で、既存の Amazon SNS トピックを選択します。

Amazon SNS 通知の詳細については、「Amazon Simple Notification Service デベロッパーガイド」の「[トピックの作成](#)」を参照してください。

6. アラームの作成(アラームの作成) を選択します。

# を使用した Elastic Transcoder API コールのログ記録 AWS CloudTrail

Elastic Transcoder は AWS CloudTrail、Elastic Transcoder のユーザー、ロール、または サービスによって実行されたアクションを記録する AWS サービスであると統合されています。CloudTrail は、Elastic Transcoder コンソールからの呼び出しと Elastic Transcoder API オペレーションへのコード呼び出しを含む、Elastic Transcoder API コールをイベントとしてキャプチャします。証跡を作成する場合は、Elastic Transcoder のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信を有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [イベント履歴] で最新のイベントを表示できます。CloudTrail で収集された情報を使用して、Elastic Transcoder に対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。

CloudTrail の詳細については、「[AWS CloudTrail ユーザーガイド](#)」を参照してください。

## CloudTrail の Elastic Transcoder 情報

CloudTrail は、AWS アカウントの作成時にアカウントで有効になります。Elastic Transcoder でアクティビティが発生すると、そのアクティビティはイベント履歴の他の AWS サービスイベントとともに CloudTrail イベントに記録されます。AWS アカウントで最近のイベントを表示、検索、ダウンロードできます。詳細については、「[CloudTrail イベント履歴でのイベントの表示](#)」を参照してください。

Elastic Transcoder のイベントなど、AWS アカウントのイベントの継続的な記録については、証跡を作成します。証跡により、ログファイルを CloudTrail で Amazon S3 バケットに配信できます。デフォルトでは、コンソールで追跡を作成するときに、追跡がすべてのリージョンに適用されます。証跡は、AWS パーティション内のすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集されたイベントデータをより詳細に分析し、それに基づく対応を行うように他の AWS サービスを設定できます。詳細については、以下を参照してください。

- [証跡を作成するための概要](#)
- [CloudTrail がサポートするサービスと統合](#)
- [CloudTrail 用 Amazon SNS 通知の構成](#)
- 「[複数のリージョンから CloudTrail ログファイルを受け取る](#)」および「[複数のアカウントから CloudTrail ログファイルを受け取る](#)」

すべての Elastic Transcoder アクションは CloudTrail によって記録され、[API リファレンス](#) で文書化されます。たとえば、[CreatePipeline]、[CreateJob]、[CreatePreset] セクションの呼び出しは、CloudTrail ログファイルにエントリを生成します。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。同一性情報は次の判断に役立ちます。

- リクエストが、ルートと IAM ユーザー認証情報のどちらを使用して送信されたか。
- リクエストがロールまたはフェデレーションユーザーの一時的なセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、[CloudTrail userIdentity 要素](#)を参照してください。

## Elastic Transcoder ログファイルエントリの理解

「トレイル」は、指定した Amazon S3 バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルは、単一か複数のログエントリを含みます。イベントはあらゆるソースからの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストのパラメータなどの情報が含まれます。CloudTrail ログファイルは、公開 API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次は CreateJob、ListJobsByStatus、ReadJob、DeletePreset および DeletePipeline のアクションを示す CloudTrail ログエントリの例です。

```
{
  "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-east-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-east-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-east-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-east-2",
"sourceIPAddress": "192.0.2.17",
"userAgent": "aws-sdk-ruby/1.39.0 ruby/1.9.3 x86_64-linux",
"requestParameters": {
  "id": "1412018849233-f2czl1r"
},
"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-east-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-east-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 サンプルコード

このセクションでは、Java、Ruby、PHP、Python 用の AWS SDK を使用して以下のオペレーションを実行する方法を示すサンプルコードを取り上げています。

- Amazon Elastic Transcoder で HLS プレイリストを作成する
- Elastic Transcoder でジョブを作成する
- Amazon Simple Queue Service (Amazon SQS) キュー をポーリングする
- Amazon Simple Notification Service (Amazon SNS) からの通知を解析する
- Amazon SNS からの通知を処理する

これらのオペレーションは、Elastic Transcoder のベストプラクティスを表しています。Amazon SQS キューのポーリングでは、ロングポーリングを使用でき、通常のポーリングよりも効率的なスケーリングが可能です。Amazon SNS 通知の設定では、Elastic Transcoder から Amazon SQS キューへのメッセージの配信が可能です。

サンプルコードは、HTTP ライブストリーミング (HLS、HTTP Live Streaming) 用のトランスコード方法を示しています。また、Amazon SNS 通知を Amazon SQS キューから削除してから処理する方法も示しています。

使用している言語に対応した SDK がある場合は、API ではなく、SDK の使用を推奨します。SDK を使用するほうが認証プロセスはよりシンプルです。また、SDK は開発環境と容易に統合されるため、関連するコマンドへのアクセスが簡単です。

## トピック

- [AWS SDK サンプルコードの背景情報](#)
- [環境をセットアップする](#)
- [Java サンプル](#)
- [Ruby サンプル](#)
- [PHP サンプル](#)
- [Python サンプル](#)

# AWS SDK サンプルコードの背景情報

## HTTP Live Streaming ( HLS )

HTTP ライブストリーミング (HLS、HTTP Live Streaming) 対応プレイヤーにアダプティブビットレートコンテンツを配信するには、HLS 出力を使用します。HLS 対応デバイスの例としては、Android 4 以降のデバイス、iOS デバイス、QuickTime や VLC などのデスクトップメディアプレイヤー、jwplayer などのブラウザプレイヤーがあります。アダプティブビットレートを使用すると、クライアント接続の質に基づいて、配信されるコンテンツの質を自動的に調整できます。

## 通知

Elastic Transcoder の ReadJob API をポーリングしてジョブのステータスを追跡するには、送信済みジョブごとに継続的に ReadJob を呼び出す必要があります。この方法では、トランスコードジョブの増加に合わせた調整ができません。この問題は、ジョブステータス追跡のためのイベント駆動型メカニズムを持つ Amazon SNS に対し、Elastic Transcoder から通知を発行することで解決できます。

Elastic Transcoder の各通知は、Message フィールド内の JSON オブジェクトとして送信されます。通知自体は JSON 形式で送信されるため、メッセージ文字列内の Elastic Transcoder 通知をエスケープする必要があります。Elastic Transcoder 通知の形式と内容の詳細については、[「ジョブのステータスの通知」](#) セクションを参照してください。

ジョブステータス通知を処理するためにカスタムコードを記述する場合は、次のベストプラクティスに従ってください。

- ハンドラはべき等である必要があります。通知が複数回配信される可能性があります。
- ハンドラは順不同通知に対応している必要があります。通知が順不同に配信される可能性があります。
- ハンドラはすべてのジョブの通知を処理できる必要があります。特定のジョブの通知が特定のワーカーに配信されることを保証する手段はありません。
- ハンドラは短いオペレーションである必要があります。可視性タイムアウトの前に、すべてのメッセージが処理されて削除される必要があります。可視性タイムアウトが 15 秒で、Amazon SQS から最大 5 件のメッセージが返される場合、各メッセージが 3 秒以内に処理されてキューから削除される必要があります。処理にかかる時間がこれより長い場合、未処理のメッセージはタイムアウトして別のワーカーに配信されます。

Java、Python、および Ruby の場合、通知トピックにサブスクライブされている Amazon SQS キューのポーリングによって通知を消費することをお勧めします。Amazon SQS ではロングポーリングメカニズムを採用しているため、Amazon SQS キューのポーリングは、スケーラブルなジョブ通知消費手段となります。また、Amazon SQS により、ホストに障害が発生した場合や高い負荷がかかっているときの可用性とスケーリングが簡素化されており、通常は特別な ACL 設定は必要ありません。

Apache で実行されている PHP の場合、エンドポイントを Amazon SNS トピックに直接サブスクライブすることをお勧めします。Amazon SNS はユーザーに通知を直接プッシュできないため、エンドポイントを公開する必要があります。

## 環境をセットアップする

サンプルコードを実行するには、AWS 環境のセットアップが完了している必要があります。このセクションでは、セットアップ手順について説明し、Elastic Transcoder の最適な動作に必要な AWS リソースの作成方法を示します。

### トピック

- [Java 環境をセットアップする](#)
- [Ruby 環境をセットアップする](#)
- [PHP 環境をセットアップする](#)
- [Python 環境をセットアップする](#)

## Java 環境をセットアップする

このセクションでは、Java 環境をセットアップする手順について説明します。

### トピック

- [AWS SDK for Java をセットアップする](#)
- [Amazon S3 入出力バケットを作成する](#)
- [ジョブステータス通知を受信するための Amazon SNS トピックを作成する](#)
- [ジョブステータス通知のためにポーリング用の Amazon SQS キューを作成する](#)
- [Amazon SNS トピックへ Amazon SQS キューをサブスクライブする](#)
- [Elastic Transcoder パイプラインを作成する](#)

- [スケーラブルな方法でコンテンツを配信するための Amazon CloudFront デистриビューションを作成する](#)
- [Amazon S3 バケットポリシーを変更する](#)

サンプルコードには、以下の 2 つの主要な前提があります。

- サンプルは Java バージョン 1.6 以上で動作するように記述する。
- サンプルは Eclipse と AWS Toolkit for Eclipse を使用して実行する。

## AWS SDK for Java をセットアップする

これらのサンプルは AWS Toolkit for Eclipse の使用を前提としています。[AWS SDK for Java](#) と以下の Jackson JSON プロセッサ JAR ファイルの両方が必要です。

- Jackson Core
- Jackson Databind
- Jackson Annotations

Jackson JSON プロセッサによってジョブステータス通知が処理されます。

代わりに、Maven を使用してこれらの依存関係を管理する場合は、pom.xml ファイルに以下のスニペットを追加できます。

バージョンプロパティ:

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

依存関係:

```
<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>
```

詳細については、[AWS SDK for Java のドキュメント](#)を参照してください。

## Amazon S3 入出力バケットを作成する

1. [Amazon S3 コンソール](#)を開きます。
2. コンソールで、[バケットの作成] をクリックします。
3. [バケット名] フィールドで、希望する入力バケット名を入力します。
4. Amazon S3 コンソールで、[バケットの作成] を再びクリックします。
5. [バケット名] フィールドで、希望する出力バケット名を入力します。

## ジョブステータス通知を受信するための Amazon SNS トピックを作成する

1. [\[Amazon SNS コンソール\]](#) を開きます。
2. コンソールで、[Create and Add] (作成と追加) をクリックして [Create New Topic] (新しいトピックの作成) を選択します。
3. [Topic Name] (トピック名) フィールドに **ets-sample-topic** を入力してから、[Create Topic] (トピックの作成) をクリックします。
4. ets-sample-topic の ARN を記録します。

## ジョブステータス通知のためにポーリング用の Amazon SQS キューを作成する

1. [Amazon SQS コンソール](#) を開きます。

2. コンソールで、[新しいキューの作成] をクリックします。
3. [Queue Name] (キュー名) フィールドに **ets-sample-queue** を入力してから、[Create Queue] (キューの作成) をクリックします。
4. Amazon SQS キューの URL を記録します。

## Amazon SNS トピックへ Amazon SQS キューをサブスクライブする

1. [Amazon SQS コンソール](#) を開きます。
2. コンソールで、キューのリストから [ets-sample-queue] を選択します。
3. [Queue Actions] (キューのアクション) で、[Subscribe Queue to Amazon SNS Topic] (キューを Amazon SNS トピックにサブスクライブする) を選択します。
4. [Choose a Topic] (トピックの選択) で [ets-sample-topic] を選択してから、[Subscribe] (サブスクライブ) をクリックします。

キューがトピックに正常にサブスクライブされたことを確認するメッセージが表示されます。

## Elastic Transcoder パイプラインを作成する

1. [Amazon Elastic Transcoder コンソール](#) を開きます。
2. コンソールで、[新しいパイプラインの作成] をクリックします。
3. [パイプライン名] フィールドで、**ets-sample-pipeline** を入力します。
4. [入力バケット] フィールドで、入力バケットの名前を入力します。
5. IAM ロールは **Console Default Role** のままにしておきます。
6. [Configure Amazon S3 Bucket for Transcoded Files and Playlists] (トランスコード済みファイルとプレイリスト用に Amazon S3 バケットを設定する) で、出力バケットの名前を入力します。
7. [Configure Amazon S3 Bucket for Transcoded Files and Playlists] (トランスコード済みファイルとプレイリスト用に Amazon S3 バケットを設定する) で、[standard storage class] (標準的なストレージクラス) を選択します。
8. [Notifications] (通知) ドロップダウンメニューを展開します。
9. 4 つのイベントタイプすべてに対して、[Use an existing SNS topic] (既存の SNS トピックを使用) を選択し、[Select a Topic] (トピックの選択) で **ets-sample-topic** を選択します。
10. [Create Pipeline (パイプラインの作成)] をクリックします。

## 11. Elastic Transcoder パイプライン ID を記録します。

### スケーラブルな方法でコンテンツを配信するための Amazon CloudFront ディストリビューションを作成する

1. [Amazon CloudFront コンソール](#)を開きます。
2. コンソールのナビゲーションペインで[Private Content] (プライベートコンテンツ) を展開して、[Origin Access Identity] (オリジンアクセスアイデンティティ) をクリックします。
3. [オリジンアクセスアイデンティティの作成] をクリックします。
4. [Create] (作成) をクリックします。
5. オリジンアクセスアイデンティティの作成時に生成される Amazon S3 正規ユーザー ID を記録します。
6. ナビゲーションペインで、[ディストリビューション] をクリックします。
7. [Create Distribution] をクリックします。
8. [Web] (ウェブ) を選択して [Continue] (続行) をクリックします。
9. [Origin Settings] (オリジン設定) で、Amazon S3 出力バケットを [Origin Domain Name] (オリジンドメイン名) として入力します。
10. [Origin ID] (オリジン ID) フィールドに **S3-transcoder-sample-output** を入力します。
11. [Restrict Bucket Access] (バケットアクセスの制限) で、[はい] を選択します。
12. [Create Distribution] (ディストリビューションの作成) をクリックし、ディストリビューションのドメイン名を記録します。

### Amazon S3 バケットポリシーを変更する

1. [Amazon S3 コンソール](#)を開きます。
2. 出力バケットの横にある拡大鏡アイコンをクリックしてバケットのプロパティを表示します。
3. [Permissions] (アクセス許可) を展開して [Add bucket policy] (バケットポリシーの追加) をクリックします。
4. 以下のポリシーステートメントを入力し、[CloudFront-oai-s3-canonical-user-id] を **your recorded Amazon S3 canonical user ID** に、[s3-output-bucket-name] を **the name of your output bucket** に置き換えます。
5. [保存] をクリックします。

```
{
  "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>/*"
  }]
}
```

「[Java サンプル](#)」に進みます。

## Ruby 環境をセットアップする

このセクションでは、Ruby 環境をセットアップする手順について説明します。

### トピック

- [AWS SDK for Ruby をセットアップする](#)
- [Amazon S3 入出力バケットを作成する](#)
- [ジョブステータス通知を受信するための Amazon SNS トピックを作成する](#)
- [ジョブステータス通知のためにポーリング用の Amazon SQS キューを作成する](#)
- [Amazon SNS トピックへ Amazon SQS キューをサブスクライブする](#)
- [Elastic Transcoder パイプラインを作成する](#)
- [スケーラブルな方法でコンテンツを配信するための Amazon CloudFront デイストリビューションを作成する](#)
- [Amazon S3 バケットポリシーを変更する](#)

サンプルコードには、以下の 1 つの主要な前提があります。

- サンプルは Ruby バージョン 1.9 と互換性があるように記述する。

## AWS SDK for Ruby をセットアップする

AWS SDK for Ruby は [こちら](#) から入手できます。この SDK は以下のコマンドで Ruby gem を使用して簡単にインストールできます。

```
gem install aws-sdk
```

詳細については、[AWS SDK for Ruby のドキュメント](#)を参照してください。

## Amazon S3 入出力バケットを作成する

1. [Amazon S3 コンソール](#)を開きます。
2. コンソールで、[バケットの作成] をクリックします。
3. [バケット名] フィールドで、希望する入力バケット名を入力します。
4. Amazon S3 コンソールで、[バケットの作成] を再びクリックします。
5. [バケット名] フィールドで、希望する出力バケット名を入力します。

## ジョブステータス通知を受信するための Amazon SNS トピックを作成する

1. [\[Amazon SNS コンソール\]](#) を開きます。
2. コンソールで、[Create and Add] (作成と追加) をクリックして [Create New Topic] (新しいトピックの作成) を選択します。
3. [Topic Name] (トピック名) フィールドに **ets-sample-topic** を入力してから、[Create Topic] (トピックの作成) をクリックします。
4. ets-sample-topic の ARN を記録します。

## ジョブステータス通知のためにポーリング用の Amazon SQS キューを作成する

1. [Amazon SQS コンソール](#) を開きます。
2. コンソールで、[新しいキューの作成] をクリックします。
3. [Queue Name] (キュー名) フィールドに **ets-sample-queue** を入力してから、[Create Queue] (キューの作成) をクリックします。
4. Amazon SQS キューの URL を記録します。

## Amazon SNS トピックへ Amazon SQS キューをサブスクライブする

1. [Amazon SQS コンソール](#) を開きます。
2. コンソールで、キューのリストから [ets-sample-queue] を選択します。

3. [Queue Actions] (キューのアクション) で、[Subscribe Queue to Amazon SNS Topic] (キューを Amazon SNS トピックにサブスクライブする) を選択します。
4. [Choose a Topic] (トピックの選択) で [ets-sample-topic] を選択してから、[Subscribe] (サブスクライブ) をクリックします。

キューがトピックに正常にサブスクライブされたことを確認するメッセージが表示されます。

## Elastic Transcoder パイプラインを作成する

1. [Amazon Elastic Transcoder コンソール](#)を開きます。
2. コンソールで、[新しいパイプラインの作成] をクリックします。
3. [パイプライン名] フィールドで、**ets-sample-pipeline** を入力します。
4. [入力バケット] フィールドで、入力バケットの名前を入力します。
5. IAM ロールは **Console Default Role** のままにしておきます。
6. [Configure Amazon S3 Bucket for Transcoded Files and Playlists] (トランスコード済みファイルとプレイリスト用に Amazon S3 バケットを設定する) で、出力バケットの名前を入力します。
7. [Configure Amazon S3 Bucket for Transcoded Files and Playlists] (トランスコード済みファイルとプレイリスト用に Amazon S3 バケットを設定する) で、[standard storage class] (標準的なストレージクラス) を選択します。
8. [Notifications] (通知) ドロップダウンメニューを展開します。
9. 4 つのイベントタイプすべてに対して、[Use an existing SNS topic] (既存の SNS トピックを使用) を選択し、[Select a Topic] (トピックの選択) で **ets-sample-topic** を選択します。
10. [Create Pipeline (パイプラインの作成)] をクリックします。
11. Elastic Transcoder パイプライン ID を記録します。

## スケーラブルな方法でコンテンツを配信するための Amazon CloudFront ディストリビューションを作成する

1. [Amazon CloudFront コンソール](#)を開きます。
2. コンソールのナビゲーションペインで[Private Content] (プライベートコンテンツ) を展開して、[Origin Access Identity] (オリジンアクセスアイデンティティ) をクリックします。
3. [オリジンアクセスアイデンティティの作成] をクリックします。

4. [Create] (作成) をクリックします。
5. オリジンアクセスアイデンティティの作成時に生成される Amazon S3 正規ユーザー ID を記録します。
6. ナビゲーションペインで、[ディストリビューション] をクリックします。
7. [Create Distribution] をクリックします。
8. [Web] (ウェブ) を選択して [Continue] (続行) をクリックします。
9. [Origin Settings] (オリジン設定) で、Amazon S3 出力バケットを [Origin Domain Name] (オリジンドメイン名) として入力します。
10. [Origin ID] (オリジン ID) フィールドに **S3-transcoder-sample-output** を入力します。
11. [Restrict Bucket Access] (バケットアクセスの制限) で、[はい] を選択します。
12. [Create Distribution] (ディストリビューションの作成) をクリックし、ディストリビューションのドメイン名を記録します。

## Amazon S3 バケットポリシーを変更する

1. [Amazon S3 コンソール](#)を開きます。
2. 出力バケットの横にある拡大鏡アイコンをクリックしてバケットのプロパティを表示します。
3. [Permissions] (アクセス許可) を展開して [Add bucket policy] (バケットポリシーの追加) をクリックします。
4. 以下のポリシーステートメントを入力し、[CloudFront-oai-s3-canonical-user-id] を **your recorded Amazon S3 canonical user ID** に、[s3-output-bucket-name] を **the name of your output bucket** に置き換えます。
5. [保存] をクリックします。

```
{
  "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>/*"
    }
  ]
}
```

```
}
```

[「Ruby サンプル」](#)に進みます。

## PHP 環境をセットアップする

このセクションでは、PHP 環境をセットアップする手順について説明します。

### トピック

- [サンプルコードをインストールする](#)
- [AWS SDK for PHP をセットアップする](#)
- [Amazon S3 入出力バケットを作成する](#)
- [ジョブステータス通知を受信するための Amazon SNS トピックを作成する](#)
- [Amazon SNS にサーバーをサブスクライブする](#)
- [Elastic Transcoder パイプラインを作成する](#)
- [スケーラブルな方法でコンテンツを配信するための Amazon CloudFront ディストリビューションを作成する](#)
- [Amazon S3 バケットポリシーを変更する](#)

サンプルコードには、以下の 3 つの主要な前提があります。

- PHP サンプルは Apache サーバー内で実行する。
- Apache サーバーは Amazon SNS 通知を受け取るためにインターネットからの入力を受け付ける。
- サンプルは PHP バージョン 5.4 以上で動作するように記述する。

### サンプルコードをインストールする

PHP サンプルコードを使用するには、まず、そのコードをダウンロードしてインストールする必要があります。

- 以下のサンプルコードをダウンロードします。
  - [HLS サンプルコード](#)
  - [通知サンプルコード](#)
- サンプルコードを解凍します。

- Apache サーバーの DocumentRoot の下にコードを配置します。
- ダウンロードしたサンプルコードの .zip ファイルを削除します。
- AWS SDK インストールへのパスを更新します。

#### Note

HlsJobCreationSample.php および JobStatusNotificationsSample.php で、autoload.php へのパスを更新する必要があります。phar インストールを使用する場合は、ダウンロードした aws.phar ファイルへのパスを使用できます。必要なすべての依存関係がこのファイルに含まれています。

## AWS SDK for PHP をセットアップする

AWS SDK for PHP は [こちら](#) で見つかります。このチュートリアルでは、phar インストールを使用することをお勧めします。ただし、長期的なプロジェクトでは、コンポーザーを使用したほうが管理は効率的です。

詳細については、[AWS SDK for PHP のドキュメント](#) を参照してください。

## Amazon S3 入出力バケットを作成する

1. [Amazon S3 コンソール](#) を開きます。
2. コンソールで、[バケットの作成] をクリックします。
3. [バケット名] フィールドで、希望する入力バケット名を入力します。
4. Amazon S3 コンソールで、[バケットの作成] を再びクリックします。
5. [Bucket Name] (バケット名) フィールドで、希望する出力バケット名を入力します。

## ジョブステータス通知を受信するための Amazon SNS トピックを作成する

1. [Amazon SNS コンソール](#) を開きます。
2. コンソールで、[Create and Add] (作成と追加) をクリックして [Create New Topic] (新しいトピックの作成) を選択します。
3. [Topic Name] (トピック名) フィールドに **ets-sample-topic** を入力してから、[Create Topic] (トピックの作成) をクリックします。

4. `ets-sample-topic` の ARN を記録します。

## Amazon SNS にサーバーをサブスクライブする

1. PHP サンプルコードがインストールされていることを確認します。
2. [\[Amazon SNS コンソール\]](#) を開きます。
3. [AdditionalActions] で、[Create Subscription] をクリックします。
4. サーバーの設定に基づいて [HTTP] または [HTTPS] を選択します。
5. [Endpoint] で、サーバーのエンドポイントを入力します。エンドポイントのパスは `JobStatusNotificationsSampleNotificationHandler.php` を参照している必要があります。
6. [Subscribe] をクリックします。これにより、PHP エンドポイントにサブスクリプションリクエストが送信されます。

PHP サンプルコードによってサブスクリプションリクエストが自動的に処理され、サブスクリプションが確認されます。サブスクリプションリクエストと応答は `/tmp/subscribe_requests.txt` に書き込まれます。

## Elastic Transcoder パイプラインを作成する

1. [Amazon Elastic Transcoder コンソール](#) を開きます。
2. コンソールで、[新しいパイプラインの作成] をクリックします。
3. [パイプライン名] フィールドで、**ets-sample-pipeline** を入力します。
4. [入力バケット] フィールドで、入力バケットの名前を入力します。
5. IAM ロールは **Console Default Role** のままにしておきます。
6. [Configure Amazon S3 Bucket for Transcoded Files and Playlists] (トランスコード済みファイルとプレイリスト用に Amazon S3 バケットを設定する) で、出力バケットの名前を入力します。
7. [Configure Amazon S3 Bucket for Transcoded Files and Playlists] (トランスコード済みファイルとプレイリスト用に Amazon S3 バケットを設定する) で、[standard storage class] (標準的なストレージクラス) を選択します。
8. [Notifications] (通知) ドロップダウンメニューを展開します。
9. 4 つのイベントタイプすべてに対して、[Use an existing SNS topic] (既存の SNS トピックを使用) を選択し、[Select a Topic] (トピックの選択) で **ets-sample-topic** を選択します。

10. [Create Pipeline (パイプラインの作成)] をクリックします。
11. Elastic Transcoder パイプライン ID を記録します。

## スケーラブルな方法でコンテンツを配信するための Amazon CloudFront ディストリビューションを作成する

1. [Amazon CloudFront コンソール](#)を開きます。
2. コンソールのナビゲーションペインで[Private Content] (プライベートコンテンツ) を展開して、[Origin Access Identity] (オリジンアクセスアイデンティティ) をクリックします。
3. [オリジンアクセスアイデンティティの作成] をクリックします。
4. [Create] (作成) をクリックします。
5. オリジンアクセスアイデンティティの作成時に生成される Amazon S3 正規ユーザー ID を記録します。
6. ナビゲーションペインで、[ディストリビューション] をクリックします。
7. [Create Distribution] をクリックします。
8. [Web] (ウェブ) を選択して [Continue] (続行) をクリックします。
9. [Origin Settings] (オリジン設定) で、Amazon S3 出力バケットを [Origin Domain Name] (オリジンドメイン名) として入力します。
10. [Origin ID] (オリジン ID) フィールドに **S3-transcoder-sample-output** を入力します。
11. [Restrict Bucket Access] (バケットアクセスの制限) で、[はい] を選択します。
12. [Create Distribution] (ディストリビューションの作成) をクリックし、ディストリビューションのドメイン名を記録します。

## Amazon S3 バケットポリシーを変更する

1. [Amazon S3 コンソール](#)を開きます。
2. 出力バケットの横にある拡大鏡アイコンをクリックしてバケットのプロパティを表示します。
3. [Permissions] (アクセス許可) を展開して [Add bucket policy] (バケットポリシーの追加) をクリックします。
4. 以下のポリシーステートメントを入力し、[CloudFront-oai-s3-canonical-user-id] を **your recorded Amazon S3 canonical user ID** に、[s3-output-bucket-name] を **the name of your output bucket** に置き換えます。
5. [保存] をクリックします。

```
{
  "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>/*"
  }]
}
```

「[PHP サンプル](#)」に進みます。

## Python 環境をセットアップする

このセクションでは、Python 環境をセットアップする手順について説明します。

### トピック

- [AWS SDK for Python をセットアップする](#)
- [Amazon S3 入出力バケットを作成する](#)
- [ジョブステータス通知を受信するための Amazon SNS トピックを作成する](#)
- [ジョブステータス通知のためにポーリング用の Amazon SQS キューを作成する](#)
- [Amazon SNS トピックへ Amazon SQS キューをサブスクライブする](#)
- [Elastic Transcoder パイプラインを作成する](#)
- [スケーラブルな方法でコンテンツを配信するための Amazon CloudFront デイストリビューションを作成する](#)
- [Amazon S3 バケットポリシーを変更する](#)

サンプルコードには、以下の 1 つの主要な前提があります。

- サンプルは Python バージョン 2.7 と互換性があるように記述する。

## AWS SDK for Python をセットアップする

AWS SDK for Python は [こちら](#) で見つかります。この SDK は以下のコマンドで PIP を使用しても簡単にインストールできます。

```
pip-2.7 install boto
```

詳細については、[AWS SDK for Python のドキュメント](#)を参照してください。

## Amazon S3 入出力バケットを作成する

1. [Amazon S3 コンソール](#)を開きます。
2. コンソールで、[バケットの作成] をクリックします。
3. [バケット名] フィールドで、希望する入力バケット名を入力します。
4. Amazon S3 コンソールで、[バケットの作成] を再びクリックします。
5. [Bucket Name] (バケット名) フィールドで、希望する出力バケット名を入力します。

## ジョブステータス通知を受信するための Amazon SNS トピックを作成する

1. [Amazon SNS コンソール](#)を開きます。
2. コンソールで、[Create and Add] (作成と追加) をクリックして [Create New Topic] (新しいトピックの作成) を選択します。
3. [Topic Name] (トピック名) フィールドに **ets-sample-topic** を入力してから、[Create Topic] (トピックの作成) をクリックします。
4. ets-sample-topic の ARN を記録します。

## ジョブステータス通知のためにポーリング用の Amazon SQS キューを作成する

1. [Amazon SQS コンソール](#)を開きます。
2. コンソールで、[新しいキューの作成] をクリックします。
3. [Queue Name] (キュー名) フィールドに **ets-sample-queue** を入力してから、[Create Queue] (キューの作成) をクリックします。
4. Amazon SQS キューの URL を記録します。

## Amazon SNS トピックへ Amazon SQS キューをサブスクライブする

1. [Amazon SQS コンソール](#)を開きます。
2. コンソールで、キューのリストから [ets-sample-queue] を選択します。

3. [Queue Actions] (キューのアクション) で、[Subscribe Queue to Amazon SNS Topic] (キューを Amazon SNS トピックにサブスクライブする) を選択します。
4. [Choose a Topic] (トピックの選択) で [ets-sample-topic] を選択してから、[Subscribe] (サブスクライブ) をクリックします。

キューがトピックに正常にサブスクライブされたことを確認するメッセージが表示されます。

## Elastic Transcoder パイプラインを作成する

1. [Amazon Elastic Transcoder コンソール](#)を開きます。
2. コンソールで、[新しいパイプラインの作成] をクリックします。
3. [パイプライン名] フィールドで、**ets-sample-pipeline** を入力します。
4. [入力バケット] フィールドで、入力バケットの名前を入力します。
5. IAM ロールは **Console Default Role** のままにしておきます。
6. [Configure Amazon S3 Bucket for Transcoded Files and Playlists] (トランスコード済みファイルとプレイリスト用に Amazon S3 バケットを設定する) で、出力バケットの名前を入力します。
7. [Configure Amazon S3 Bucket for Transcoded Files and Playlists] (トランスコード済みファイルとプレイリスト用に Amazon S3 バケットを設定する) で、[standard storage class] (標準的なストレージクラス) を選択します。
8. [Notifications] (通知) ドロップダウンメニューを展開します。
9. 4 つのイベントタイプすべてに対して、[Use an existing SNS topic] (既存の SNS トピックを使用) を選択し、[Select a Topic] (トピックの選択) で **ets-sample-topic** を選択します。
10. [Create Pipeline (パイプラインの作成)] をクリックします。
11. Elastic Transcoder パイプライン ID を記録します。

## スケーラブルな方法でコンテンツを配信するための Amazon CloudFront ディストリビューションを作成する

1. [Amazon CloudFront コンソール](#)を開きます。
2. コンソールのナビゲーションペインで[Private Content] (プライベートコンテンツ) を展開して、[Origin Access Identity] (オリジンアクセスアイデンティティ) をクリックします。
3. [オリジンアクセスアイデンティティの作成] をクリックします。

4. [Create] (作成) をクリックします。
5. オリジンアクセスアイデンティティの作成時に生成される Amazon S3 正規ユーザー ID を記録します。
6. ナビゲーションペインで、[ディストリビューション] をクリックします。
7. [Create Distribution] をクリックします。
8. [Web] (ウェブ) を選択して [Continue] (続行) をクリックします。
9. [Origin Settings] (オリジン設定) で、Amazon S3 出力バケットを [Origin Domain Name] (オリジンドメイン名) として入力します。
10. [Origin ID] (オリジン ID) フィールドに **S3-transcoder-sample-output** を入力します。
11. [Restrict Bucket Access] (バケットアクセスの制限) で、[はい] を選択します。
12. [Create Distribution] (ディストリビューションの作成) をクリックし、ディストリビューションのドメイン名を記録します。

## Amazon S3 バケットポリシーを変更する

1. [Amazon S3 コンソール](#)を開きます。
2. 出力バケットの横にある拡大鏡アイコンをクリックしてバケットのプロパティを表示します。
3. [Permissions] (アクセス許可) を展開して [Add bucket policy] (バケットポリシーの追加) をクリックします。
4. 以下のポリシーステートメントを入力し、[CloudFront-oai-s3-canonical-user-id] を **your recorded Amazon S3 canonical user ID** に、[s3-output-bucket-name] を **the name of your output bucket** に置き換えます。
5. [保存] をクリックします。

```
{
  "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>/*"
    }
  ]
}
```

```
}
```

「[Python サンプル](#)」に進みます。

## Java サンプル

すべての Java サンプルには、以下の 2 つの前提があります。

- サンプルは Java バージョン 1.6 以上で動作するように記述する。
- サンプルは Eclipse と AWS Toolkit for Eclipse を使用して実行する。

## HLS サンプル

### トピック

- [Java HLS サンプルをダウンロードする](#)
- [サンプルコードの構成要素](#)
- [タスク](#)

このサンプルは、アダプティブビットレートストリームの再生に使用できる HLS ジョブと HLS プレイリストファイルを作成する方法を示しています。

### Java HLS サンプルをダウンロードする

サンプルコードは[こちら](#)からダウンロードできます。

### サンプルコードの構成要素

Java サンプルコードには以下の構成要素があります。

- HlsJobCreationSample クラス

### タスク

サンプルを実行するには、以下の手順を実行します。

1. 「[Java 環境をセットアップする](#)」の手順に従って環境をセットアップします。

2. サンプルコードをダウンロードします。
3. Eclipse で、`com/amazonaws/services/elastictranscoder/samples/HlsJobCreationSample.java` を開きます。
4. `PIPELINE_ID` と `INPUT_KEY` を適切な値に置き換えます。
5. Eclipse でサンプルを実行します。

## 通知サンプル

### トピック

- [通知の構文例](#)
- [Java 通知サンプルをダウンロードする](#)
- [サンプルコードの構成要素](#)
- [タスク](#)

### 通知の構文例

```
{
  "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\" :
    \"135162000001-000020\",\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" : "E1SqJW3ZksCPViYGTayI/p+LjdF2HB42iJlIJRJ+jWzWwygXdiJXvZX194qhd/
tLln1lxPqijjivb5RWu7n5yzdZwbod6lpLwyZ2TfWM6dZt570zsG3GbdTxgqwVsztVSyWCYhcV8f
+CrT3IQrfrU3Me/SRYVUqrSUXSsu4Ls7A2q9mosG7v0Sn+3Z1rAa9+Rf6QmkfAg05UX0sLyA
```

```
+I2hFqTu5oAGDK4Cm6FHuIwV+oYJXNBbGWuS7tD6mRNwrYvPB1UvBLXx9m3CbqSXv5Yoj39vglv
+1djtaLA3GpwX+B1hHx8QT373lgmmsmGDRWhSQretp0TWDYb81PV2K0bg==" ,
  "SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationService-
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 通知サンプルをダウンロードする

サンプルコードは[こちら](#)からダウンロードできます。

### サンプルコードの構成要素

Java サンプルコードには以下の構成要素があります。

- Notification クラス: Java から通知を利用するために、Jackson JSON ライブラリを使用して ジョブステータス通知を POJO に逆シリアル化します。Notification クラスは、Amazon SQS キューから取得するメッセージのモデルを実装します。
- JobStatusNotification クラス: JobStatusNotification クラスは、Elastic Transcoder 通知のモデルを実装します。
- SqsQueueNotificationWorker クラス: SqsQueueNotificationWorker クラスは、個別のスレッドで開始されて、Amazon SQS をポーリングし、ジョブステータス通知を処理します。このクラスは、メッセージを受け取り、受け取った通知ごとにすべての登録済みのハンドラーを呼び出した後、キューからメッセージを削除します。
- JobStatusNotificationHandler インターフェイス: JobStatusNotificationHandler インターフェイスは、通知のカスタム処理のために実装できます。
- JobStatusNotificationsSample クラス: JobStatusNotificationsSample クラスは、ジョブを作成し、ジョブの完了を待ちます。ジョブが終了状態になると、キューワーカーがシャットダウンされ、アプリケーションは終了します。

## タスク

サンプルを実行するには、以下の手順を実行します。

1. 「[Java 環境をセットアップする](#)」の手順に従って環境をセットアップします。
2. サンプルコードをダウンロードします。

3. Java プロジェクトのソースディレクトリにサンプルコードを解凍します。
4. Eclipse で、`com/amazonaws/services/elastictranscoder/samples/JobStatusNotificationsSample.java` を開きます。
5. `PIPELINE_ID`、`SQS_QUEUE_URL`、`INPUT_KEY` を適切な値に置き換えます。
6. Eclipse でサンプルを実行します。

## Ruby サンプル

すべての Ruby サンプルには、以下の 1 つの主要な前提があります。

- サンプルは Ruby バージョン 1.9 と互換性があるように記述する。

## HLS サンプル

トピック

- [Ruby HLS サンプルをダウンロードする](#)
- [サンプルコードの構成要素](#)
- [タスク](#)

このサンプルは、アダプティブビットレートストリームの再生に使用できる HLS ジョブと HLS プレイリストファイルを作成する方法を示しています。

### Ruby HLS サンプルをダウンロードする

サンプルコードは[こちら](#)からダウンロードできます。

### サンプルコードの構成要素

Ruby サンプルコードには以下の構成要素があります。

- `HlsJobCreationSample.rb` ファイル

### タスク

サンプルを実行するには、以下の手順を実行します。

1. 「[Ruby 環境をセットアップする](#)」の手順に従って環境をセットアップします。

2. サンプルコードをダウンロードします。
3. Ruby プロジェクトのソースディレクトリにサンプルコードを解凍します。
4. `HlsJobCreationSample.rb` を編集し、`pipeline_id` と `input_key` を適切な値に置き換えます。
5. ターミナルウィンドウから、サンプルコードを解凍したディレクトリに移動し、以下のように実行します。

```
$ruby HlsJobCreationSample.rb
```

## 通知サンプル

### トピック

- [通知の構文例](#)
- [Ruby 通知サンプルをダウンロードする](#)
- [サンプルコードの構成要素](#)
- [タスク](#)

### 通知の構文例

```
{
  "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" : "E1SqJW3ZksCPViYGTayI/p+LjdF2HB42iJlIJRj+jWzWwygXdiJXvZX194qhd/
tLln1lxPqijjivb5RWu7n5yzdZwbod6lpLwyZ2TfWM6dZt570zsG3GbdTxgqwVsztVSyWCYhcV8f
+CrT3IQrfrU3Me/SRYVUqrSUXXsu4Ls7A2q9mosG7v0Sn+3Z1rAa9+Rf6QmkfAg05UX0sLyA
+I2hFqTu5oAGDK4Cm6FHUwV+oYJXNBbGWuS7tD6mRNwrYvPB1UvBLXx9m3CbqSXv5Yoj39vglv
+1djtaLA3GpwX+B1hHx8QT3731gmmSmGDRWhSQretp0TWDYb81PV2K0bg==",
"SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationService-
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 通知サンプルをダウンロードする

サンプルコードは[こちら](#)からダウンロードできます。

### サンプルコードの構成要素

Ruby サンプルコードには以下の構成要素があります。

- SqsQueueNotificationWorker クラス: SqsQueueNotificationWorker クラスは、Amazon SQS をポーリングし、通知ごとにすべての登録済みのハンドラーを呼び出した後、キューからメッセージを削除します。ワーカーは別のスレッドで実行されます。Ruby の実装で「グリーン」スレッドを使用している場合は、一度に 1 つのスレッドのみが実行されます (真のマルチプロセッシングではありません)。
- スクリプト JobStatusNotificationsSample.rb: このスクリプトは、Elastic Transcoder ジョブを作成し、Amazon SQS ワーカーを開始して、ジョブの完了を待ちます。登録済みのハンドラーからの指示により、Amazon SQS ワーカーは、作成されたジョブの処理が完了すると停止します。通知の処理のために、add\_handlers メソッドが呼び出されて、1 つの引数を受け取るハンドラーとして Proc または lambda が渡されます。受け取った通知ごとに、すべての登録済みのハンドラーが呼び出されて、ハンドラーへの入力引数として通知が渡されます。

## タスク

サンプルを実行するには、以下の手順を実行します。

1. 「[Ruby 環境をセットアップする](#)」の手順に従って環境をセットアップします。
2. サンプルコードをダウンロードします。
3. Ruby プロジェクトのソースディレクトリにサンプルコードを解凍します。

4. `JobStatusNotificationsSample.rb` を編集し、`pipeline_id`、`sq3_queue_url`、`input_key` を適切な値に置き換えます。
5. ターミナルウィンドウから、サンプルコードを解凍したディレクトリに移動し、以下のように実行します。

```
$ruby JobStatusNotificationsSample.rb
```

## PHP サンプル

すべての PHP サンプルには、以下の 3 つの主要な前提があります。

- PHP サンプルは Apache サーバー内で実行する。
- Apache サーバーは Amazon SNS 通知を受け取るためにインターネットからの入力を受け付ける。
- サンプルは PHP バージョン 5.4 以上で動作するように記述する。

## HLS サンプル

トピック

- [PHP HLS サンプルをダウンロードする](#)
- [サンプルコードの構成要素](#)
- [タスク](#)

このサンプルは、アダプティブビットレートストリームの再生に使用できる HLS ジョブと HLS プレリストファイルを作成する方法を示しています。

### PHP HLS サンプルをダウンロードする

サンプルコードは [こちら](#) からダウンロードできます。

### サンプルコードの構成要素

PHP サンプルコードには以下の構成要素があります。

- `HlsJobCreationSample.php` ファイル

## タスク

サンプルを実行するには、以下の手順を実行します。

1. 「[PHP 環境をセットアップする](#)」の手順に従って環境をセットアップします。
2. サンプルコードをダウンロードします。
3. ブラウザで、`http://<your-endpoint>/transcoder-samples/HlsJobCreationSample.php` ページをロードします。
4. パイプライン ID と入力キーを入力し、ジョブ作成用のフォームを送信します。

## 通知サンプル

### トピック

- [通知の構文例](#)
- [PHP 通知サンプルをダウンロードする](#)
- [サンプルコードの構成要素](#)
- [タスク](#)

### 通知の構文例

```
{
  "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" : "ElSqJW3ZksCPViYGTayI/p+LjdF2HB42iJlIJRJ+jWzWwygXdiJXvZX194qhd/
tLln1lxPqijjivb5RWu7n5yzdZwbod6lpLwyZ2TfWM6dZt570zsG3GbdTxgqwVsztVSyWCYhcV8f
+CrT3IQrfrU3Me/SRYVUqrSUXXsu4Ls7A2q9mosG7v0Sn+3Z1rAa9+Rf6QmkfAg05UX0sLyA
+I2hFqTu5oAGDK4Cm6FHUwV+oYJXNBbGWuS7tD6mRNwrYvPB1UvBLXx9m3CbqSXv5Yoj39vglv
+1djtaLA3GpwX+B1hHx8QT373lgmmsmGDRWhSQretp0TWDYb81PV2K0bg==",
"SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationService-
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 通知サンプルをダウンロードする

サンプルコードは[こちら](#)からダウンロードできます。

### サンプルコードの構成要素

PHP サンプルコードには以下の構成要素があります。

- ファイル `JobStatusNotificationsSampleNotificationHandler.php`: 通知がサーバーに POST 送信されると、通知ハンドラーが `/tmp/<job-id>` にあるファイルにステータスを再び書き込みます。
- ファイル `JobStatusNotificationsSample.php`: 通知ハンドラが `/tmp/<job-id>` にあるファイルに再び書き込んだ後、`JobStatusNotificationsSample.php` は、渡されたクエリ文字列内の ID で指定された `/tmp/<job-id>` ステータスファイルをロードします。

## タスク

サンプルを実行するには、以下の手順を実行します。

1. 「[PHP 環境をセットアップする](#)」の手順に従って環境をセットアップします。
2. サンプルコードをダウンロードします。
3. PHP プロジェクトのソースディレクトリにサンプルコードを解凍します。
4. ブラウザで、`http://<your-endpoint>/transcoder-samples/JobStatusNotificationsSample.php` ページをロードします。
5. トランスコードするパイプラインの ID と入力キーを入力し、[Create Job] ボタンを押します。

# Python サンプル

すべての Python サンプルには、以下の 1 つの主要な前提があります。

- サンプルは Python バージョン 2.7 と互換性があるように記述する。

## HLS サンプル

トピック

- [Python HLS サンプルをダウンロードする](#)
- [サンプルコードの構成要素](#)
- [タスク](#)

このサンプルは、アダプティブビットレートストリームの再生に使用できる HLS ジョブと HLS プレイリストファイルを作成する方法を示しています。

### Python HLS サンプルをダウンロードする

サンプルコードは[こちら](#)からダウンロードできます。

### サンプルコードの構成要素

Python サンプルコードには以下の構成要素があります。

- HlsJobCreationSample.py ファイル

### タスク

サンプルを実行するには、以下の手順を実行します。

1. 「[Python 環境をセットアップする](#)」の手順に従って環境をセットアップします。
2. サンプルコードをダウンロードします。
3. サンプルコードを解凍します。
4. HlsJobCreationSample.py を編集し、pipeline\_id と input\_key を適切な値に置き換えます。
5. ターミナルウィンドウから、サンプルコードを解凍したディレクトリに移動し、以下のように実行します。

```
$python HlsJobCreationSample.py
```

## 通知サンプル

### トピック

- [通知の構文例](#)
- [Python 通知サンプルをダウンロードする](#)
- [サンプルコードの構成要素](#)
- [タスク](#)

### 通知の構文例

```
{
  "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\" :
    \"135162000001-000020\",\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" : "E1SqJW3ZksCPViYGTayI/p+LjdF2HB42iJlIJRj+jWzWwygXdiJXvZX194qhd/
tLln1lxPqijjivb5RWu7n5yZdZwbod6lpLwyZ2TfWM6dZt570zsG3GbdTxgqwVsztVSYhCYhcV8f
+CrT3IQrfrU3Me/SRYVUqrSUXSsu4Ls7A2q9mosG7v0Sn+3Z1rAa9+Rf6QmkfAg05UX0sLyA
+I2hFqTu5oAGDK4Cm6FHUwV+oYJXNBbGwuS7tD6mRNwrYvPB1UvBLXx9m3CbqSXv5Yoj39vglv
+1djtaLA3GpwX+B1hHx8QT3731gmmsmGDRWhSQretp0TWDYb81PV2K0bg==",
  "SigningCertURL" : "https://sns.us-east-1.amazonaws.com/SimpleNotificationService-
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 通知サンプルをダウンロードする

サンプルコードは[こちら](#)からダウンロードできます。

### サンプルコードの構成要素

Python サンプルコードには以下の構成要素があります。

- `SqsQueueNotificationWorker.py` クラス: `SqsQueueNotificationWorker` クラスは、個別のプロセスで開始されて、Amazon SQS をポーリングし、通知を処理します。これにより、Python で真のマルチプロセッシングが可能になります。このクラスは、メッセージを受け取り、受け取った通知ごとにすべての登録済みのハンドラーを呼び出した後、キューからメッセージを削除します。このクラスには、Elastic Transcoder 通知の処理用フレームワークを実装するための `JobStatusNotificationHandler` クラスの定義も含まれています。このクラスは継承できるため、ジョブ処理用のメソッドをカスタムのメソッドで上書きできます。これは、Python のダックタイピングの規格に従っていませんが、このサンプルの目的では、ハンドラーの正式な定義が可能になります。
- `JobStatusNotificationSample.py` スクリプト: `JobStatusNotificationSample.py` スクリプトは、Elastic Transcoder でジョブを作成し、ジョブの完了を待ちます。ジョブが完了すると、キューのポーリングプロセスを停止し、終了します。ハンドラーがポーリング先のキューとは別のプロセスで実行されるため、値オブジェクトのマルチプロセッシングによるプロセス間同期が必要になります。

## タスク

サンプルを実行するには、以下の手順を実行します。

1. 「[Python 環境をセットアップする](#)」の手順に従って環境をセットアップします。
2. サンプルコードをダウンロードします。
3. ローカルフォルダーにサンプルコードを解凍します。
4. `JobStatusNotificationsSample.py` を編集し、`pipeline_id`、`sqs_queue_url`、`input_key` を適切な値に置き換えます。
5. ターミナルウィンドウから、サンプルコードを解凍したディレクトリに移動し、以下のように実行します。

```
$python JobStatusNotificationsSample.py
```

# API リファレンス

使用している言語に対応した AWS SDK がある場合は、API を使用して作業を行うのではなく、SDK を利用してください。SDK を利用すると、認証が簡素化され、開発環境との統合が容易になり、Elastic Transcoder コマンドに簡単にアクセスすることができます。AWS SDK の詳細 (環境のセットアップ方法、SDK ドキュメントへのリンク、サンプルコードなど) については、「[AWS SDK サンプルコード](#)」を参照してください。

## トピック

- [AWS SDK の使用](#)
- [Elastic Transcoder に対する HTTP リクエストの作成](#)
- [リクエストへの署名](#)
- [Elastic Transcoder でのエラー処理](#)
- [パイプラインのオペレーション](#)
- [ジョブのオペレーション](#)
- [プリセットのオプション](#)

## AWS SDK の使用

使用している言語に対応した AWS SDK がある場合は、API を使用して作業を行うのではなく、SDK を利用してください。SDK を利用すると、認証が簡素化され、開発環境との統合が容易になり、Elastic Transcoder コマンドに簡単にアクセスすることができます。AWS SDK の詳細 (環境のセットアップ方法、SDK ドキュメントへのリンク、サンプルコードなど) については、「[AWS SDK サンプルコード](#)」を参照してください。

## Elastic Transcoder に対する HTTP リクエストの作成

Elastic Transcoder REST リクエストは、RFC 2616 で定義されている HTTPS リクエストです。詳細については、<http://www.ietf.org/rfc/rfc2616.txt> を参照してください。このセクションでは、Elastic Transcoder REST リクエストの構造について説明します。実行できるアクションの詳細については、「[パイプラインのオペレーション](#)」、「[ジョブのオペレーション](#)」、「[プリセットのオプション](#)」を参照してください。

通常の REST アクションは、Elastic Transcoder への 1 つの HTTPS リクエストの送信と、レスポンスの待機で構成されます。すべての HTTP リクエストと同様に、Elastic Transcoder への REST リク

エストにはリクエストメソッド、URI、リクエストヘッダーが含まれており、クエリ文字列またはリクエストボディが含まれている場合もあります。レスポンスには HTTP ステータスコードとレスポンスヘッダーが含まれており、レスポンス本文が含まれている場合もあります。

## トピック

- [HTTP ヘッダーの内容](#)
- [HTTP リクエストボディ](#)
- [HTTP レスポンス](#)

## HTTP ヘッダーの内容

Elastic Transcoder では、HTTP リクエストのヘッダーに次の情報を入力する必要があります。

### Host ( 必須 )

リソースが作成される場所を指定する Elastic Transcoder エンドポイント。値は、リージョンの指名エンドポイントにする必要があります。Amazon S3 バケットと、Elastic Transcoder のパイプラインやジョブに対しては同じエンドポイントを使用することをお勧めします。異なるエンドポイントを使用すると、Amazon S3 バケットを含んでいるリージョンと Elastic Transcoder のエンコードが実行されるリージョンの間で転送されるデータについて追加料金が発生します。また、データ転送にかかる時間が原因で、変換結果ファイルへのアクセスが遅延される場合があります。

サポートされている Elastic Transcoder エンドポイントの一覧については、「[アマゾン ウェブ サービス全般リファレンス](#)」の「リージョンとエンドポイント」を参照してください。

リージョン間の料金の詳細については、「[Amazon S3 料金表](#)」の「データ転送料金表」を参照してください。

### x-amz-date または Date ( 必須 )

Authorization ヘッダーに含まれる署名を作成するのに使用された日付。ISO 8601 の標準形式に基づいて UTC 時間で日付を指定します (例: X-Amz-Date: 20130613T203622Z)。

x-amz-date または Date のどちらかを含める必要があります。(一部の HTTP クライアントライブラリでは、Date ヘッダーを設定することができません)。x-amz-date ヘッダーがある場合、リクエストの認証時に Date ヘッダーは無視されます。

タイムスタンプは、リクエストが受け取られた時点の AWS システム時刻の 15 分以内に入っている必要があります。このようにしないと、リクエストは RequestExpired エラーコードで失敗し、任意のユーザーがリクエストを再現できなくなります。

#### Authorization (必須)

リクエスト認証に必要な情報。このヘッダーの作成方法の詳細については、「[リクエストへの署名](#)」を参照してください。

#### Content-Type (条件付き)

JSON とバージョンを指定します (例: Content-Type: application/x-amz-json-1.0)。

条件: POST リクエストの場合は必須です。

#### Content-Length (条件付き)

RFC 2616 に基づくメッセージの長さ (ヘッダーなし)。

条件: リクエストボディ自体に情報が含まれる場合、必須です (このヘッダーは、ほとんどのツールキットで自動的に追加されます)。

次に、パイプラインを作成する HTTP リクエストのサンプルヘッダーの例を示します。

```
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/elastictranscoder/aws4_request,SignedHeaders=host;x-amz-date;x-amz-target,Signature=145b1567ab3c50d929412f28f52c45dbf1e63ec5c66023d232a539a4afd11fd9
content-type: application/x-amz-json-1.0
content-length: 231
connection: Keep-Alive
```

## HTTP リクエストボディ

多くの Elastic Transcoder API アクションでは、リクエストの本文に JSON 形式のデータを含める必要があります。JSON は Elastic Transcoder スキーマに準拠しています。

### Note

リクエストボディの JSON 値は文字列です。

## Example リクエスト

次のサンプルリクエストでは、簡単な JSON ステートメントを使用して、sample.mp4 という名前のファイルを変換するジョブを作成し、そのファイルを sams-birthday.mp4 として保存します。

```
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 レスポンス

すべての Elastic Transcoder API アクションにおいて、レスポンスに JSON 形式のデータが含まれます。JSON は Elastic Transcoder スキーマに準拠しています。

### Note

レスポンスの JSON 値は文字列です。

HTTP レスポンスの重要なヘッダーと、それらをアプリケーション内で扱う方法 (該当する場合) を示します。

## HTTP/1.1

このヘッダーにはステータスコードが続きます。ステータスコード 200 は、オペレーションが成功したことを示します。エラーコードについては、「[API のエラーコード \(クライアントエラーとサーバーエラー\)](#)」を参照してください。

タイプ: 文字列

### x-amzn-RequestId

Elastic Transcoder によって作成された値であり、リクエストを一意に識別します (例: K2QH8DNOU907N97FNA2GDLL8OBVV4KQNSO5AEMVJF66Q9ASUAAJG)。Elastic Transcoder で問題が発生した場合、AWS ではこの値を使用して、問題をトラブルシューティングします。これらの値をログに記録することをお勧めします。

タイプ: 文字列

### Content-Length

レスポンス本文の長さ (バイト単位)。

型: 文字列

### 日付

Elastic Transcoder が応答した日時 (例: Sun, 25 Mar 2012 12:00:00 GMT)。日付の形式は、RFC 2616 のセクション 3.3 で指定されている完全な日付の形式のいずれかであることが必要です。

タイプ: 文字列

## リクエストへの署名

使用する言語の SDK が AWS に用意されている場合は、その SDK を使用することをお勧めします。AWS SDK を使用するほうが、Elastic Transcoder API を使用するよりも、リクエストの署名プロセスがきわめてシンプルで、大幅な時間の節約になります。また、SDK は開発環境と容易に統合されるため、関連するコマンドへのアクセスが簡単です。

Elastic Transcoder では、リクエストに署名することで、送信するすべてのリクエストを認証する必要があります。リクエストに署名するには、暗号化ハッシュ関数を使用してデジタル署名を計算し

ます。この関数は入力に基づいてハッシュ値を返します。入力には、リクエストのテキスト、およびシークレットアクセスキーが含まれます。ハッシュ関数から返されるハッシュ値をリクエストに署名として含めます。署名は、リクエストの Authorization ヘッダーの一部です。

Elastic Transcoder は、リクエストを受け取ると、リクエストの署名に使用されたものと同じハッシュ関数と入力を使用して署名を再計算します。再計算された署名とリクエスト内の署名が一致した場合、Elastic Transcoder はリクエストを処理します。それ以外の場合、リクエストは拒否されます。

Elastic Transcoder では、[AWS 署名バージョン 4](#) を使用した認証がサポートされています。署名の計算プロセスは 3 つのタスクに分けることができます。

- [タスク 1: 正規リクエストを作成する](#)

『<https://docs.aws.amazon.com/general/latest/gr/sigv4-create-canonical-request.html>』の

「Amazon Web Services 全般のリファレンスタスク 1: 署名バージョン 4 の正規リクエストを作成する」で説明されているように、正規形式で HTTP リクエストを作成します。

- [タスク 2: 署名対象の文字列を作成する](#)

暗号化ハッシュ関数への入力値の 1 つとして使用する文字列を作成します。署名文字列と呼ばれる文字列は、ハッシュアルゴリズムの名前、要求日付、認証情報スコープの文字列、および前のタスクで正規化されたリクエストを結合したものです。認証情報スコープの文字列自体は、日付、リージョン、およびサービス情報を結合したものです。

X-Amz-Credential パラメータには、以下の情報を指定します。

- リクエストの送信先であるエンドポイントのコード ( us-east-1 など )。Elastic Transcoder のリージョンとエンドポイントのリストについては、Amazon Web Services 全般のリファレンスの「[リージョンとエンドポイント](#)」の章を参照してください。エンドポイントのコードを指定するときは、elastictranscoder. と .amazonaws.com の間の部分のみを含めます。
- サービスの省略形としての elastictranscoder

例:

```
X-Amz-Credential=AKIAIOSFODNN7EXAMPLE/20130501/us-east-1/elastictranscoder/aws4_request
```

- [タスク 3: 署名を作成する](#)

2 つの入力文字列 (署名文字列と派生キー) を受け付ける暗号化ハッシュ関数を使用して、リクエストの署名を作成します。シークレットアクセスキーから開始し、認証情報スコープの文字列を使

用して一連のハッシュベースのメッセージ認証コード (HMAC) を作成することで、派生キーが計算されます。

## Elastic Transcoder でのエラー処理

### トピック

- [API のエラーコード \(クライアントエラーとサーバーエラー\)](#)
- [ジョブ処理中のエラー](#)
- [エラーの捕捉](#)
- [エラーの再試行とエクスポネンシャルバックオフ](#)

リクエストを送信して Elastic Transcoder API からの応答を取得するとき、以下の 2 種類の API エラーが発生する場合があります。

- クライアントエラー: クライアントエラーは 4xx HTTP 応答コードで示されます。クライアントエラーは、Elastic Transcoder によってクライアントリクエストについての問題 (認証の違反や必須パラメータの不足など) が検出されたことを示します。リクエストを再度送信する前に、クライアントアプリケーションで問題を修正します。
- サーバーエラー: サーバーエラーは 5xx HTTP 応答コードで示され、Amazon が解決する必要がある場合があります。リクエストは、成功するまで再送信/再試行することができます。

各 API エラーについて、Elastic Transcoder によって以下の値が返されます。

- ステータスコード (400 など)
- エラーコード (ValidationException など)
- エラーメッセージ (Supplied AttributeValue is empty, must contain exactly one of the supported datatypes など)

クライアントとサーバーのエラーについて Elastic Transcoder によって返されるエラーコードのリストについては、「[API のエラーコード \(クライアントエラーとサーバーエラー\)](#)」を参照してください。

さらに、Elastic Transcoder によるジョブの処理中にエラーが発生する場合があります。詳細については、「[ジョブ処理中のエラー](#)」を参照してください。

## API のエラーコード ( クライアントエラーとサーバーエラー )

HTTP ステータスコードは、特定のオペレーションが成功したかどうかを示しています。

応答コード 200 は、オペレーションが成功したことを示します。その他のエラーコードは、クライアントエラー ( 4xx ) またはサーバーエラー ( 5xx ) を示します。

以下の表に、Elastic Transcoder によって返されるエラーを示します。一部のエラーは、同じリクエストを再試行することで解決されます。この表は、連続的な再試行によって解決する可能性が高いエラーを示しています。[Retry (リトライ)] 列の値は次のことを示しています。

- Yes: 同じリクエストを再び送信します。
- No: 新しいリクエストの送信前にクライアント側で問題を解決します。

リクエストの再試行の詳細については、「[エラーの再試行とエクスポネンシャルバックオフ](#)」を参照してください。

HTTP ステータスコード	エラーコード	メッセージ	原因	再試行
400	Conditional Check Failed Exception	条件付きリクエストが失敗しました。	例: 期待値がシステムに格納されている値に一致しませんでした。	いいえ
400	Incomplete Signature Exception	リクエストの署名が AWS 基準に適合しません。	リクエスト内の署名に、必要なすべての要素が含まれていませんでした。「 <a href="#">HTTP ヘッダーの内容</a> 」を参照してください。	いいえ
403	Missing Authentication Token Exception	The request must contain a valid (registered) AWS Access Key ID.	必要な x-amz-security-token がリクエストに含まれていませんでした。「 <a href="#">Elastic Transcoder に対する HTTP リ</a>	いいえ

HTTP ステータスコード	エラーコード	メッセージ	原因	再試行
			<a href="#">クエストの作成</a> を参照してください。	
400	Validation Exception	各種の値。	リクエストで 1 つ以上の値が見つからないか無効でした。たとえば、空の値があるか、最大許容の値よりも大きい値があります。	いいえ
403	AccessDeniedException	<ul style="list-style-type: none"> <li>Deleting a system preset is not allowed: account=&lt;accountId&gt;, presetId=&lt;presetId&gt;。</li> <li>一般的な認証の失敗。クライアントがリクエストに正しく署名しませんでした。「<a href="#">リクエストへの署名</a>」を参照してください。</li> </ul>	システムプリセットを削除しようとしたか、Elastic Transcoder API 呼び出し時の署名が無効であったか、ユーザーにこのオペレーションの実行許可がありません。	いいえ

HTTPステータスコード	エラーコード	メッセージ	原因	再試行
404	ResourceNotFound Exception	<ul style="list-style-type: none"> <li>The specified &lt;resource&gt; could not be found: &lt;resourceId&gt;。</li> <li>The specified job was not found: account=&lt;accountId&gt;, jobId=&lt;jobId&gt;。</li> <li>The specified pipeline was not found: account=&lt;accountId&gt;, pipelineId=&lt;pipelineId&gt;</li> <li>The specified preset was not found: account=&lt;accountId&gt;, presetId=&lt;presetId&gt;</li> </ul>	例: ジョブを追加しようとしているパイプラインが存在しないか、まだ作成中です。	いいえ
409	ResourceInUse Exception	<ul style="list-style-type: none"> <li>The &lt;resource&gt; was already in use: accountId=&lt;accountId&gt;, resourceId=resourceId&gt;。</li> <li>The pipeline contains active jobs: account=&lt;accountId&gt;, pipeline=&lt;pipelineId&gt;。</li> </ul>	例: 現在使用中のパイプラインを削除しようとした。	いいえ

HTTP ステータス コード	エラーコード	メッセージ	原因	再試行
429	Limit Exceeded Exception	<ul style="list-style-type: none"> <li>The account already has the maximum number of pipelines allowed: account=&lt;accountId&gt;, maximum number of pipelines=&lt;maximum&gt;</li> <li>The account already has the maximum number of presets allowed: account=&lt;accountId&gt;, maximum number of presets=&lt;maximum&gt;</li> <li>The account already has the maximum number of jobs per pipeline in the backlog: account=&lt;accountId&gt;, maximum number of jobs in backlog for pipeline=&lt;maximum&gt;</li> </ul>	<p>現在の AWS アカウントが Elastic Transcoder オブジェクトの制限を超えました。詳細については、「<a href="#">Elastic Transcode「パイプライン、ジョブ、プリセットの数の制限」</a>」を参照してください。</p>	

HTTP ステータスコード	エラーコード	メッセージ	原因	再試行
429	Provisioned Throughput Exceeded Exception	プロビジョニングされたスループットが許容されている最大値を超えました。	<p>例: リクエストの頻度が多すぎます。Elastic Transcoder の AWS SDK は、この例外を受け取ったリクエストを自動的に再試行します。リクエストは最終的に成功しますが、再試行キューが大きすぎて終了しない場合もあります。リクエストの頻度を少なくしてください。詳細については、「<a href="#">エラーの再試行とエクスポネンシャルバックオフ</a>」を参照してください。</p> <p>ポーリングによりリクエストのステータスを調べている場合は、通知を使用してステータスを調べることを検討してください。詳細については、「<a href="#">ジョブのステータスの通知</a>」を参照してください。</p>	はい

HTTP ステータスコード	エラーコード	メッセージ	原因	再試行
429	Throttling Exception	リクエストの速度が、許容されているスループットを超えています。	リクエスト (新しいジョブの作成リクエストなど) の送信が速すぎます。  ポーリングによりリクエストのステータスを調べている場合は、通知を使用してステータスを調べることを検討してください。詳細については、「 <a href="#">ジョブのステータスの通知</a> 」を参照してください。	はい
500	内部エラー	リクエストの処理中にサーバーで内部エラーが発生しました。	リクエストの処理中にサーバーでエラーが発生しました。	はい
500	Internal Server Error	リクエストの処理中にサーバーで内部エラーが発生しました。	リクエストの処理中にサーバーでエラーが発生しました。	はい
500	Internal Service Exception		リクエストの処理中にサーバーで予期せぬエラーが発生しました。	はい
500	Service Unavailable Exception	サービスが現在利用できないかビジー状態です。	リクエストの処理中にサーバーで予期しないエラーが発生しました。	はい

## エラー応答のサンプル

以下の HTTP 応答は、inputBucket の値が null であり、有効な値でなかったことを示しています。

```
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"}
```

## ジョブ処理中のエラー

ジョブの処理中に Elastic Transcoder でエラーが発生したときは、以下の 2 つの方法でエラーがレポートされます。

- ジョブステータスと出カステータス: Elastic Transcoder は、違反した出力の `Job:Status` オブジェクトと `Outputs:Status` オブジェクトを `Error` に設定します。さらに、Elastic Transcoder によって、違反した出力の `Outputs:StatusDetail` JSON オブジェクトが、違反を説明する値に設定されます。
- SNS 通知: Elastic Transcoder で処理中にエラーが発生した場合に SNS 通知が送信されるようにパイプラインを設定した場合、Elastic Transcoder により JSON オブジェクトが次の形式で通知に含められます。

```
{
  "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|Complete|Warning|Error"
    },
  ],
}
```

```

    {...}
  ],
  "playlists": [
    {
      applicable job playlists settings
    }
  ],
  "userMetadata": {
    "metadata key": "metadata value"
  }
}

```

値として 個格納 できま す。 <b>error</b>	値として個格納でき ます。 <b>messageDe tails</b>	原因
1,000	検証エラー	ジョブの処理中、Elastic Transcoder によってリクエスト内の 1 つ以上の値が無効であると判断されました。
1001	依存関係のエラー	プレイリストの依存関係の 1 つ以上でエラーが発生したため、Elastic Transcoder はプレイリストを作成できませんでした。
2000	Cannot Assume Role	Elastic Transcoder は、このジョブのパイプラインの Role オブジェクトで指定された AWS Identity and Access Management ロールを引き受けることはできません。
3000	Unclassified Storage Error	
3001	Input Does Not Exist	このジョブの Input:Key オブジェクトで指定した名前のファイルは存在しません。そのファイルは、このジョブのパイプラインの InputBucket オブジェクトで指定した Amazon S3 バケットに存在します。
3002	Output Already Exists	このジョブの Outputs:Key (または Output:Key ) オブジェクトで指定した名前のファイルはすでに存在し

値として 個格納 できま す。 <b>error</b>	値として個格納でき ます。 <b>messageDe tails</b>	原因
		ます。そのファイルは、このジョブのパイプラインの OutputBucket オブジェクトで指定した Amazon S3 バケツに存在することはできません。
3003	Does Not Have Read Permission	このジョブに使用したパイプラインの Role オブジェクトで指定した IAM ロールに、トランスコードするファイルを Amazon S3 バケツから読み取るためのアクセス許可がありません。
3004	Does Not Have Write Permission	このジョブに使用したパイプラインの Role オブジェクトで指定した IAM ロールに、トランスコードしたファイルまたはサムネイルファイルを Amazon S3 バケツに書き込むためのアクセス許可がありません。
3005	Bucket Does Not Exist	指定された S3 バケツが存在しません: bucket={1}
3006	Does Not Have Write Permission	キーがバケツと同じリージョンにないため、Elastic Transcoder では key={1} を bucket={2} に書き込むことができませんでした。
4000	Bad Input File	このジョブの Input:Key オブジェクトで指定したファイルの形式は、Elastic Transcoder では現在サポートされていません。
4001	Bad Input File	このジョブの Input:Key オブジェクトで指定したファイルの幅 x 高さが最大許容の幅 x 高さを超えています。
4002	Bad Input File	このジョブの Input:Key オブジェクトで指定したファイルのサイズが最大許容のサイズを超えています。
4003	Bad Input File	このジョブの Outputs:Watermarks:InputKey オブジェクトのいずれかで指定したファイルが Elastic Transcoder で解釈されませんでした。

値として 個格納 できま す。 <b>error</b>	値として個格納でき ます。 <b>messageDe tails</b>	原因
4004	Bad Input File	このジョブの <code>Outputs:Watermarks:InputKey</code> オブジェクトのいずれかで指定したファイルの幅 x 高さが最大許容の幅 x 高さを超えています。
4005	Bad Input File	{1} オブジェクトのいずれかに指定したファイルのサイズが最大許容のサイズを超えています: <code>bucket={2}</code> 、 <code>key={3}</code> 、 <code>size{4}</code> 、 <code>max size={5}</code>
4006	Bad Input File	Elastic Transcoder で入力ファイルのトランスコードが実行されませんでした。この形式はサポートされていません。
4007	Unhandled Input File	Elastic Transcoder によって、一般的にサポートされているファイルのタイプが検出されましたが、ファイルが正しく処理されませんでした。このエラーによってサポートケースが自動的に開かれ、Amazon が問題の原因調査を開始しました。
4008	Bad Input File	<p>根本的な原因はプリセットと入力ファイルとの不一致です。以下に例を示します。</p> <ul style="list-style-type: none"> <li>• プリセットにはオーディオ設定が含まれているが、入力ファイルにはオーディオが含まれていない。</li> <li>• プリセットには動画設定が含まれているが、入力ファイルには動画が含まれていない。</li> </ul>
4009	Bad Input File	Elastic Transcoder でアルバムアートのすべてを出力ファイルに挿入できませんでした。アートワークストリームの最大数を超えています。
4010	Bad Input File	<code>AlbumArt:Artwork:InputKey</code> に指定したグラフィックファイルが Elastic Transcoder で解釈されませんでした。

値として 個格納 できま す。error	値として個格納でき ます。messageDe tails	原因
4011	Bad Input File	Elastic Transcoder で埋め込みアートワークストリームが検出されましたが、解釈されませんでした。
4012	Bad Input File	AlbumArt:Artwork に指定した画像が最大許容の幅 x 高さ (4,096 x 3,072) を超えています。
4013	Bad Input File	埋め込みアートワークの幅 x 高さが最大許容の幅 x 高さ (4,096 x 3,072) を超えています。
4014	Bad Input	クリップの開始時間に指定した値が入力ファイルの終了時間より後になっています。Elastic Transcoder で出力ファイルを作成できませんでした。
4015	Bad Input	生成されたセグメントが一致しなかったため、Elastic Transcoder によりマニフェストファイルが生成されませんでした。
4016	Bad Input	Elastic Transcoder で、{2} を使用して {1} から入力ファイルを復号できませんでした。
4017	Bad Input	AES キーは {2} ビット暗号化キーを使用して暗号化されています。AES では、128、192、256 ビットの暗号化キーのみをサポートしています。MD5={1}
4018	Bad Input	MD5={1} を使用して暗号化されたキーが Elastic Transcoder で復号できませんでした。
4019	Bad Input	Elastic Transcoder で、KMS キーの ARN {0} を使用してデータキーを生成できませんでした。
4020	Bad Input	AES-128 暗号化では、キーが 128 ビットである必要があります。MD5={1}、{2} ビット。

値として 個格納 できま す。 <b>error</b>	値として個格納でき ます。 <b>messageDe tails</b>	原因
4021	Bad Input	PlayReady DRM では、キーが 128 ビットである必要があります。 MD5={1}、 strength={2} ビット。
4022	Bad Input	指定されたメディアファイル {1} 個の結合サイズが最大許容サイズを超えています: bucket={2}、 size={3}。
4023	Bad Input	連結のために指定された {1} 個の入力ファイルから、指定されたプリセットで一貫した解像度の出力が作成されません。異なる PaddingPolicy、SizingPolicy、MaxWidth、MaxHeight 設定のプリセットを使用してください。
4024	Bad Input	連結のために指定された {1} 個の入力ファイルから、指定されたプリセットで一貫した解像度のサムネイルが作成されません。サムネイルの異なる PaddingPolicy、SizingPolicy、MaxWidth、MaxHeight 設定のプリセットを使用してください。
4025	Bad Input	少なくとも 1 つのメディアファイル (入力 #{1}) が他のファイルと一致しません。すべてのメディアファイルに動画が含まれているか、どのメディアファイルにも動画が含まれていないことが必要です。
4026	Bad Input	少なくとも 1 つのメディアファイル (入力 #{1}) が他のファイルと一致しません。すべてのメディアファイルに音声が含まれているか、どのメディアファイルにも音声が含まれていないことが必要です。
4100	Bad Input File	Elastic Transcoder によって埋め込みキャプショントラックが検出されましたが、解釈されませんでした。

値として 個格納 できま す。 <b>error</b>	値として個格納でき ます。 <b>messageDe tails</b>	原因
4101	Bad Input File	Amazon S3 bucket={1}、key={2} の指定のキャプションファイルが Elastic Transcoder で解釈されませんでした。
4102	Bad Input File	指定のキャプションファイルが UTF-8 でエンコードされていなかったため Elastic Transcoder で解釈されませんでした: Amazon S3 bucket={1}、key={2}
4103	Bad Input File	キャプショントラックの一部がキャプショントラックの最大数 ({1}) を超えたため、Elastic Transcoder で処理されませんでした。
4104	Bad Input File	指定の出力に {1} 個の埋め込みキャプションオプションが含まれていたため、Elastic Transcoder によってマスタープレイリストが生成されませんでした。最大個数は 4 です。
4105	Bad Input File	CEA-708 でフレームレート {1} はサポートされていないため、Elastic Transcoder でキャプショントラックが埋め込まれませんでした。フレームレート [29.97, 30] のみがサポートされています。
4106	Bad Input File	形式 {1} では {2} キャプショントラックしかサポートされていないため、Elastic Transcoder によってキャプショントラックが埋め込まれませんでした。
9000	Internal Service Error	
9001	Internal Service Error	
9999	Internal Service Error	

## エラーの捕捉

アプリケーションをスムーズに実行するには、エラーを見つけ、エラーに対応するロジックを組み込む必要があります。1つの一般的なアプローチとしては、try ブロックまたは if-then ステートメントにリクエストを実装する方法が挙げられます。

AWS SDK は独自に再試行とエラーチェックを実行します。いずれかの AWS SDK の使用中にエラーが発生した場合は、エラーコードと説明が表示されます。また Request ID の値も表示されます。Request ID の値は、Elastic Transcoder のサポートによって問題のトラブルシューティングを行うために役立ちます。

次の例では、AWS SDK for Java を使用して try ブロック内の項目を削除し、catch ブロックを使用してエラーに対応しています。この場合、リクエストが失敗したと警告されます。この例では AmazonServiceException クラスを使用して、Request ID を含むオペレーションエラーに関する情報を取り出しています。さらにこの例では、その他の理由でリクエストが失敗した場合のために、AmazonClientException クラスも使用されています。

```
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());
}
```

## エラーの再試行とエクスポネンシャルバックオフ

DNS サーバー、スイッチ、ロードバランサーなど、ネットワークの多数のコンポーネントが、特定のリクエストの存続期間中どこでもエラーを生成する可能性があります。

ネットワーク環境でこれらのエラー応答を処理する通常の方法は、クライアントアプリケーションで再試行を実装することです。この技術は、アプリケーションの信頼性を向上させ、開発者の運用コストを削減します。

Elastic Transcoder をサポートする各 AWS SDK には自動再試行ロジックが実装されています。AWS SDK for Java は自動的にリクエストを再試行します。再試行は、ClientConfiguration クラスを使用して設定できます。たとえば、ウェブページが最小のレイテンシーで再試行なしでリクエストを実行する場合などに、再試行ロジックを停止させることがあります。再試行を無効にするには、ClientConfiguration クラスを使用し、maxErrorRetry の 0 値を指定します。

AWS SDK を使用していない場合は、サーバーエラー (5xx) を受け取る元のリクエストを再試行する必要があります。ただし、クライアントエラー (4xx、ThrottlingException または ProvisionedThroughputExceededException 以外) は、再試行する前にリクエスト自体を修正して問題を解決する必要があることを示しています。

### Note

ポーリングによりリクエストのステータスを調べているとき、Elastic Transcoder によって HTTP ステータスコード 429 (エラーコード Provisioned Throughput Exceeded Exception または Throttling Exception) が返される場合は、ポーリングの代わりに通知を使用してステータスを調べることを検討してください。詳細については、「[ジョブのステータスの通知](#)」を参照してください。

単純な再試行に加えて、効果的なフロー制御を行うために、エクスポネンシャルバックオフアルゴリズムを使用することをお勧めします。エクスポネンシャルバックオフの背後にある考え方は、連続したエラー応答の再試行間の待機時間を徐々に長く使用することです。たとえば、最初の再試行前に 1 秒間、2 回目の再試行前に 4 秒間、3 回目の再試行前に 16 秒間というように、待機時間を指定します。ただし、リクエストが 1 分後に成功しなかった場合、問題はハードリミットであり、リクエストの頻度でない可能性があります。たとえば、最大許可のパイプライン数に達した可能性があります。1 分程度で再試行が停止するように最大回数を設定します。

次に、再試行ロジックが含まれたワークフローを示します。このワークフローロジックでは、最初にそのエラーがサーバーエラー (5xx) であるかどうか判别されます。エラーがサーバーエラーである場合は、コードによって元のリクエストが再試行されます。

```
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
```

## パイプラインのオペレーション

### トピック

- [パイプラインを作成する](#)
- [パイプラインを一覧表示する](#)
- [パイプラインを読み取る](#)
- [パイプラインを更新する](#)
- [パイプラインのステータスを更新する](#)
- [パイプラインの通知を更新する](#)
- [パイプラインを削除する](#)
- [ロールのテスト](#)

パイプラインは、変換ジョブを管理するキューです。ジョブを作成するとき、ジョブを追加するパイプラインを指定します。Elastic Transcoder によってパイプライン内のジョブの処理が開始されます。処理は、キューにジョブを追加した順序で行われます。

このセクションでは、Elastic Transcoder API を使用してパイプラインに対して実行できるオペレーションについて説明します。Elastic Transcoder コンソールを使用して同じオペレーションを実行する方法など、パイプラインの詳細については、「[パイプラインの使用](#)」を参照してください。

## パイプラインを作成する

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

### 説明

パイプラインを作成するには、`/2012-09-25/pipelines/` リソースに POST リクエストを送信します。

### リクエスト

### 構文

```
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 endpoint/
    elastictranscoder/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",
    "Complete": "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"
}
}
```

## リクエストパラメータ

このオペレーションはリクエストパラメータを使用しません。

## リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

## リクエスト本文

リクエストボディの JSON 文字列には、CreatePipeline オペレーションの入力オブジェクトが含まれます。入力オブジェクトの詳細については、「[Elastic Transcoder パイプラインの作成時に指定する設定](#)」を参照してください。

## レスポンス

### 構文

```
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",
  "Complete":"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": [
  {
    "Code": "6000|6001|6002|6003|6004|6005|6006|6007|6008",
    "Message": "The code message"
  },
  {...}
]
}
```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

パイプラインを作成すると、リクエストで指定した値が Elastic Transcoder によって返されます。詳細については、「[リクエスト本文](#)」を参照してください。

また、以下の値も Elastic Transcoder によって返されます。

### (自動) Id

パイプラインの識別子。この値を使用して、ジョブやプリセットの作成など、さまざまな操作を実行するパイプラインを特定します。

### (自動) Status

パイプラインの現在のステータス:

- Active: パイプラインでジョブが処理されています。
- Paused: 現在、パイプラインでジョブが処理されていません。

## (自動) Warnings

他のリージョンでリソースを使用するパイプラインを作成すると、Elastic Transcoder により 1 つ以上の警告が返されます。パイプラインは作成されますが、処理時間が長くなりリージョン間の料金が発生する可能性があります。警告は次の形式になります。

CodeCode — 警告コード。

Message – 警告コードに関連付けられたメッセージ。

返すことができる警告コードとメッセージは次のとおりです。

### 6000

入力バケットとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、入力バケットとパイプラインに同じリージョンを使用してください。

### 6001

ContentConfig バケットとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、ContentConfig バケットとパイプラインに同じリージョンを使用してください。

### 6002

ThumbnailConfig バケットとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、ThumbnailConfig バケットとパイプラインに同じリージョンを使用してください。

### 6003

進捗イベントの SNS 通知トピックとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、SNS 通知トピックとパイプラインに同じリージョンを使用してください。

## 6004

警告イベントの SNS 通知トピックとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、SNS 通知トピックとパイプラインに同じリージョンを使用してください。

## 6005

完了イベントの SNS 通知トピックとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、SNS 通知トピックとパイプラインに同じリージョンを使用してください。

## 6006

エラーイベントの SNS 通知トピックとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、SNS 通知トピックとパイプラインに同じリージョンを使用してください。

## 6007

このパイプラインに対して指定された AWS KMS キーと ContentConfig バケットが異なるリージョンに存在するため、s3-aws-kms 暗号化モードを使用した出力が失敗します。s3-aws-kms 暗号化モードを使用するには、KMS キーと ContentConfig バケットに同じリージョンを使用してください。

## 6008

このパイプラインに対して指定された AWS KMS キーと ThumbnailConfig バケットが異なるリージョンに存在するため、s3-aws-kms 暗号化モードを使用した出力が失敗します。s3-aws-kms 暗号化モードを使用するには、KMS キーと ThumbnailConfig バケットに同じリージョンを使用してください。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

## 例

以下のリクエスト例では、Default という名前のパイプラインを作成しています。

## リクエスト例

```
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 endpoint/
elastictranscoder/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": "",
    "Complete": "",
    "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"
}
}
```

## レスポンス例

```
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": {
      "Complete": "",
      "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."
  },
  {...}
]
}
```

## パイプラインを一覧表示する

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

### 説明

現在の AWS アカウントに関連付けられたパイプラインのリストを取得するには、`/2012-09-25/pipelines/` リソースに GET リクエストを送信します。

## リクエスト

### 構文

現在の AWS アカウントに関連付けられたすべてのパイプラインの情報を取得するには、以下の GET リクエストを送信します。

```
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 endpoint/
    elastictranscoder/aws4_request,
    SignedHeaders=host;x-amz-date;x-amz-target,
    Signature=calculated-signature
```

### リクエストパラメータ

このオペレーションでは、次のリクエストパラメータを使用します。Elastic Transcoder によりすべてのパイプラインが返されます。

### 昇順

パイプラインを送信日時の古い順に一覧表示するには、true を入力します。パイプラインを送信日時の新しい順に一覧表示するには、false を入力します。

### PageToken

Elastic Transcoder から返される結果のページが複数にわたる場合、後続の GET リクエストで PageToken を使用することで、それぞれの結果のページを連続したページとして取得できます。

### リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

### リクエスト本文

この操作にリクエストボディはありません。

## レスポンス

### 構文

```
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 process the job",
        "Complete":"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"
},
{...}
],
"NextPageToken": value for accessing the next page of results|null
}

```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

レスポンス本文の JSON 文字列は、パイプラインの作成時に指定した値を返します。各オブジェクトの詳細については、「[Elastic Transcoder パイプラインの作成時に指定する設定](#)」を参照してください。

また、以下の値も Elastic Transcoder によって返されます。

## (自動) Id

パイプラインの識別子。この値を使用して、ジョブやプリセットの作成など、さまざまな操作を実行するパイプラインを特定します。

## (自動) Status

パイプラインの現在のステータス:

- Active: パイプラインでジョブが処理されています。
- Paused: 現在、パイプラインでジョブが処理されていません。

## (自動) NextPageToken

結果の 2 ページ目以降のページ (存在する場合) にアクセスするために使用する値。パイプラインが 1 ページに収まる場合、または結果の最後のページに達したとき、NextPageToken の値は null になります。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

## 例

以下のリクエスト例では、現在の AWS アカウントに関連付けられたパイプラインのリストを取得しています。

### リクエスト例

```
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 endpoint/
    elastictranscoder/aws4_request,
    SignedHeaders=host;x-amz-date;x-amz-target,
    Signature=calculated-signature
```

## レスポンス例

```
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":"111111111111-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":"","",
        "Complete":"","",
        "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": "222222222222-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": "",
    "Complete": "",
    "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"
}
]
}
```

## パイプラインを読み取る

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

### 説明

パイプラインの詳細情報を取得するには、`/2012-09-25/pipelines/pipelineId` リソースに GET リクエストを送信します。

### リクエスト

### 構文

```
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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
```

## リクエストパラメータ

このオペレーションでは、次のリクエストパラメーターを使用します。

### pipelineId

詳細情報を取得するパイプラインの識別子。

### リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

### リクエスト本文

この操作にリクエストボディはありません。

## レスポンス

### 構文

```
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 process the job",
      "Complete":"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"
},
"(##) Warnings": [

```

```
{
  {
    "Code": "6000|6001|6002|6003|6004|6005|6006|6007|6008",
    "Message": "The code message"
  },
  {...}
]
```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

レスポンス本文の JSON 文字列は、パイプラインの作成時に指定した値を返します。各オブジェクトの詳細については、「[Elastic Transcoder パイプラインの作成時に指定する設定](#)」を参照してください。

Elastic Transcoder には、該当する場合、以下の値と警告も含まれます。

### (自動) Id

パイプラインの識別子。この値を使用して、ジョブやプリセットの作成など、さまざまな操作を実行するパイプラインを特定します。

### (自動) Status

パイプラインの現在のステータス:

- Active: パイプラインでジョブが処理されています。
- Paused: 現在、パイプラインでジョブが処理されていません。

### (自動) Warnings

他のリージョンでリソースを使用するパイプラインを作成すると、Elastic Transcoder により 1 つ以上の警告が返されます。パイプラインは作成されますが、処理時間が長くなりリージョン間の料金が発生する可能性があります。警告は次の形式になります。

## コード

Message – 警告コードに関連付けられたメッセージ。

以下は有効な警告コードとそのメッセージの一覧です。

#### 6000

入力バケットとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、入力バケットとパイプラインに同じリージョンを使用してください。

#### 6001

ContentConfig バケットとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、ContentConfig バケットとパイプラインに同じリージョンを使用してください。

#### 6002

ThumbnailConfig バケットとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、ThumbnailConfig バケットとパイプラインに同じリージョンを使用してください。

#### 6003

進捗イベントの SNS 通知トピックとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、SNS 通知トピックとパイプラインに同じリージョンを使用してください。

#### 6004

警告イベントの SNS 通知トピックとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、SNS 通知トピックとパイプラインに同じリージョンを使用してください。

#### 6005

完了イベントの SNS 通知トピックとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、SNS 通知トピックとパイプラインに同じリージョンを使用してください。

## 6006

エラーイベントの SNS 通知トピックとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、SNS 通知トピックとパイプラインに同じリージョンを使用してください。

## 6007

このパイプラインに対して指定された AWS KMS キーと ContentConfig バケットが異なるリージョンに存在するため、s3-aws-kms 暗号化モードを使用した出力が失敗します。s3-aws-kms 暗号化モードを使用するには、KMS キーと ContentConfig バケットに同じリージョンを使用してください。

## 6008

このパイプラインに対して指定された AWS KMS キーと ThumbnailConfig バケットが異なるリージョンに存在するため、s3-aws-kms 暗号化モードを使用した出力が失敗します。s3-aws-kms 暗号化モードを使用するには、KMS キーと ThumbnailConfig バケットに同じリージョンを使用してください。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

## 例

以下のリクエスト例では、ID が 111111111111-abcde1 であるパイプラインを取得しています。

### リクエスト例

```
GET /2012-09-25/pipelines/111111111111-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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
```

## レスポンス例

```
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":"111111111111-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":"","",
      "Complete":"","",
      "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."
  },
  {...}
]
}
```

## パイプラインを更新する

### トピック

- [説明](#)
- [値を更新および削除する方法](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

### 説明

パイプラインの設定を更新するには、`/2012-09-25/pipelines/pipelineId` リソースに PUT リクエストを送信します。

**⚠ Important**

パイプラインの設定を変更すると、その変更はすぐに有効になります。設定の変更後に送信するジョブだけでなく、すでに送信したが Elastic Transcoder によってまだ処理が開始されていないジョブにも、変更は適用されます。

## 値を更新および削除する方法

値を更新するには、たとえば以下のように、新しい値を指定します。

```
"GranteeType":"Canonical"
```

値を変更しないで現在のままにするには、以下のいずれかの操作を実行します。

- 現在の値を指定する。
- null 値を指定する。これは個々のオブジェクト ("Role":null など) と配列 ("ContentConfig":null など) の両方に適用されます。
- リクエストから JSON オブジェクトを省く。これは個々のオブジェクト (Role など) と配列 (ContentConfig など) の両方に適用されます。

現在の値を削除するには、空の値を指定します。たとえば、通知を削除するには、"Progressing":"" のように空の文字列を指定します。配列のすべての値を削除するには、"Permissions":[] のように空の配列を指定します。

例えば、ジョブの処理中に Elastic Transcoder で警告が発生したときの通知が不要になり、通知のその他のいずれの設定も変更しない場合は、リクエストで Notifications を以下のように指定します。

```
...  
"Notifications":{  
  "Warning":""  
},  
...
```

## リクエスト

### 構文

```
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 endpoint/
elastictranscoder/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",
    "Complete": "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"
}
```

## リクエストパラメータ

このオペレーションはリクエストパラメータを使用しません。

## リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

## リクエスト本文

リクエストボディの JSON 文字列には、UpdatePipeline オペレーションの入力オブジェクトが含まれます。入力オブジェクトの詳細については、「[Elastic Transcoder パイプラインの作成時に指定する設定](#)」を参照してください。

## ID

パイプラインの識別子。この値を使用して、ジョブやプリセットの作成など、さまざまな操作を実行するパイプラインを特定します。

## ステータス

パイプラインの現在のステータス:

- Active: パイプラインでジョブが処理されています。
- Paused: 現在、パイプラインでジョブが処理されていません。

## レスポンス

### 構文

```
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",
    "Complete": "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": [
    {
      "Code": "6000|6001|6002|6003|6004|6005|6006|6007|6008",
      "Message": "The code message"
    },
    {...}
  ]
}

```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

パイプラインを作成すると、リクエストで指定した値が Elastic Transcoder によって返されます。詳細については、「[リクエスト本文](#)」を参照してください。

Elastic Transcoder には、該当する場合、以下の警告も含まれます。

### (自動) Warnings

他のリージョンでリソースを使用するパイプラインを作成すると、Elastic Transcoder により 1 つ以上の警告が返されます。パイプラインは作成されますが、処理時間が長くなりリージョン間の料金が発生する可能性があります。警告は次の形式になります。

## コード

Message – 警告コードに関連付けられたメッセージ。

以下は有効な警告コードとそのメッセージの一覧です。

### 6000

入力バケットとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、入力バケットとパイプラインに同じリージョンを使用してください。

### 6001

ContentConfig バケットとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、ContentConfig バケットとパイプラインに同じリージョンを使用してください。

### 6002

ThumbnailConfig バケットとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、

リージョン間の料金の発生を回避するには、ThumbnailConfig バケットとパイプラインに同じリージョンを使用してください。

#### 6003

進捗イベントの SNS 通知トピックとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、SNS 通知トピックとパイプラインに同じリージョンを使用してください。

#### 6004

警告イベントの SNS 通知トピックとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、SNS 通知トピックとパイプラインに同じリージョンを使用してください。

#### 6005

完了イベントの SNS 通知トピックとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、SNS 通知トピックとパイプラインに同じリージョンを使用してください。

#### 6006

エラーイベントの SNS 通知トピックとパイプラインが異なるリージョンに存在するため、パイプライン内のジョブの処理時間が長くなり、追加料金が発生する可能性があります。処理時間を短縮し、リージョン間の料金の発生を回避するには、SNS 通知トピックとパイプラインに同じリージョンを使用してください。

#### 6007

このパイプラインに対して指定された AWS KMS キーと ContentConfig バケットが異なるリージョンに存在するため、s3-aws-kms 暗号化モードを使用した出力が失敗します。s3-aws-kms 暗号化モードを使用するには、KMS キーと ContentConfig バケットに同じリージョンを使用してください。

#### 6008

このパイプラインに対して指定された AWS KMS キーと ThumbnailConfig バケットが異なるリージョンに存在するため、s3-aws-kms 暗号化モードを使用した出力が失敗します。s3-aws-kms 暗号化モードを使用するには、KMS キーと ThumbnailConfig バケットに同じリージョンを使用してください。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

### 例

以下のリクエスト例では、ID が 111111111111-abcde1 であるパイプラインを更新しています。

#### リクエスト例

```
PUT /2012-09-25/pipelines/111111111111-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 endpoint/
    elastictranscoder/aws4_request,
    SignedHeaders=host;x-amz-date;x-amz-target,
    Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
  "Id": "111111111111-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": "",
    "Complete": "",
    "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"
}
```

## レスポンス例

```
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": "",
    "Complete": "",
    "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."
    },
    {...}
  ]
}
```

## パイプラインのステータスを更新する

### トピック

- [説明](#)

- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

## 説明

パイプラインを一時停止するか再びアクティブにするには (パイプライン内のジョブを停止するか再開するには)、パイプラインのステータスを更新します。/2012-09-25/pipelines/*pipelineId*/status リソースに POST リクエストを送信します。

パイプラインのステータスの変更は、1 つ以上のジョブをキャンセルする場合に便利です。ジョブは Elastic Transcoder による処理の開始後にキャンセルすることはできません。ジョブを送信した先のパイプラインを一時停止する場合は、キャンセルするジョブの ID を取得して Delete Job リクエストを送信するための時間が余分にかかります。

## リクエスト

### 構文

```
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 endpoint/
    elastictranscoder/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"
}
```

## リクエストパラメータ

このオペレーションでは、次のリクエストパラメーターを使用します。

## pipelineId

一時停止または再びアクティブにするパイプラインの識別子。

## リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

## リクエスト本文

リクエストの本文内の JSON 文字列には、以下のオブジェクトが含まれます。

## ステータス

パイプラインの新しいステータス:

- Active: パイプラインが有効になり、ジョブの処理が開始されます。
- Paused: パイプラインが無効になり、ジョブの処理が停止します。

## レスポンス

### 構文

```
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"
}
```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

パイプラインのステータスを更新すると、リクエストで指定した値が Elastic Transcoder によって返されます。詳細については、「[リクエスト本文](#)」を参照してください。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

## 例

以下のリクエスト例では、ID が 111111111111-abcde1 であるパイプラインを有効にしています。

### リクエスト例

```
POST /2012-09-25/pipelines/111111111111-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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
  "Status":"Active"
}
```

### レスポンス例

```
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":"111111111111-abcde1",
  "Status":"Active"
```

```
}
```

## パイプラインの通知を更新する

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

### 説明

パイプラインの Amazon Simple Notification Service (Amazon SNS) 通知のみを更新するには、`/2012-09-25/pipelines/pipelineId/notifications` リソースに POST リクエストを送信します。

#### Important

通知を変更すると、その変更はすぐに反映されます。通知の変更後に送信するジョブだけでなく、すでに送信したが Elastic Transcoder によってまだ処理が開始されていないジョブにも、変更は適用されます。

### リクエスト

#### 構文

```
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 endpoint/
elastictranscoder/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",  
    "Complete": "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"  
  }  
}
```

## リクエストパラメータ

このオペレーションでは、次のリクエストパラメーターを使用します。

### PipelineId

通知設定を変更するパイプラインの識別子。

## リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

## リクエスト本文

リクエストボディの JSON 文字列には、次のオブジェクトが含まれます。

### ID

更新するパイプラインの ID。

### Notifications:Progressing

Elastic Transcoder でこのパイプラインのジョブの処理が開始されたときに通知する Amazon Simple Notification Service (Amazon SNS) のトピック ARN。これは、トピックの作成時に Amazon SNS によって返された ARN です。詳細については、[Amazon Simple Notification Service デベロッパーガイド](#)の「トピックの作成」を参照してください。

**⚠ Important**

通知を受信するには、Amazon SNS コンソールで新しいトピックにサブスクライブする必要があります。

Amazon SNS にはさまざまな通知オプションが用意されており、Amazon Simple Queue Service キューに Amazon SNS メッセージを送信することもできます。詳細については、「[Amazon Simple Notification Service デベロッパーガイド](#)」を参照してください。

**Notifications:Complete**

Elastic Transcoder でこのパイプラインのジョブの処理が終了したときに通知する Amazon Simple Notification Service (Amazon SNS) のトピック ARN。これは、トピックの作成時に Amazon SNS によって返された ARN です。

**Notifications:Warning**

Elastic Transcoder でこのパイプラインのジョブの処理中に警告が発生したときに通知する Amazon Simple Notification Service (Amazon SNS) のトピック ARN。これは、トピックの作成時に Amazon SNS によって返された ARN です。

**Notifications:Error**

Elastic Transcoder でこのパイプラインのジョブの処理中にエラーが発生したときに通知する Amazon Simple Notification Service (Amazon SNS) のトピック ARN。これは、トピックの作成時に Amazon SNS によって返された ARN です。

## レスポンス

### 構文

```
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",  
    "Complete": "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"  
  }  
}  
}
```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

パイプラインの通知を更新すると、リクエストで指定した値が Elastic Transcoder によって返されます。詳細については、「[リクエスト本文](#)」を参照してください。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

## 例

以下のリクエスト例では、パイプラインの通知を更新しています。

### リクエスト例

```
POST /2012-09-25/pipelines/111111111111-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 endpoint/  
    elastictranscoder/aws4_request,  
    SignedHeaders=host;x-amz-date;x-amz-target,  
    Signature=calculated-signature  
Content-Length: number of characters in the JSON string  
{  
  "Id": "111111111111-abcde1",
```

```
"Notifications":{
  "Progressing":"","
  "Complete":"","
  "Warning":"","
  "Error":"arn:aws:sns:us-east-1:111222333444:ETS_Errors"
}
```

## レスポンス例

```
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":"111111111111-abcde1",
  "Notifications":{
    "Progressing":"","
    "Complete":"","
    "Warning":"","
    "Error":"arn:aws:sns:us-east-1:111222333444:ETS_Errors"
  }
}
```

## パイプラインを削除する

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

### 説明

パイプラインを削除するには、`/2012-09-25/pipelines/pipelineId` リソースに DELETE リクエストを送信します。使用されたことのないパイプラインまたは現在使用されていない (アクティブ

なジョブが含まれていない) パイプラインのみを削除できます。削除するパイプラインが現在使用されている場合は、Delete Pipeline によってエラーが返されます。

## リクエスト

### 構文

```
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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
```

### リクエストパラメータ

このオペレーションでは、次のリクエストパラメーターを使用します。

#### pipelineId

削除するパイプラインの識別子。

### リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

### リクエスト本文

この操作にリクエストボディはありません。

## レスポンス

### 構文

```
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"
}
```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

応答の本文には、以下の JSON オブジェクトが含まれます。

### Success (成功)

パイプラインが正常に削除された場合、Success の値は true です。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

## 例

以下のリクエスト例では、111111111111-abcde1 という名前のパイプラインを削除しています。

### リクエスト例

```
DELETE /2012-09-25/pipelines/111111111111-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 endpoint/
                elastictranscoder/aws4_request,
```

```
SignedHeaders=host;x-amz-date;x-amz-target,  
Signature=calculated-signature
```

## レスポンス例

```
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"  
}
```

## ロールのテスト

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

### 説明

パイプラインの設定をテストして、Elastic Transcoder がジョブの作成と処理を実行できるかどうかを確認するには、`/2012-09-25/roleTests` リソースに POST リクエストを送信します。

### リクエスト

#### 構文

```
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 endpoint/
elastictranscoder/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"
  ]
}
```

## リクエストパラメータ

このオペレーションはリクエストパラメータを使用しません。

## リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

## リクエスト本文

リクエストボディの JSON 文字列には、次のオブジェクトが含まれます。

### InputBucket

トランスコードするメディアファイルを保存した Amazon S3 バケット。Test Role はこのバケットから読み取りを試行します。

### OutputBucket

Elastic Transcoder でトランスコード済みファイルを保存する Amazon S3 バケット。Test Role はこのバケットから読み取りを試行します。

## ロール

Elastic Transcoder でジョブのトランスコードに使用するロールの IAM Amazon リソースネーム (ARN)。Test Role は指定したロールの引き受けを試みます。

## トピック

Test Role によるテスト通知の送信先となる 1 つ以上の Amazon Simple Notification Service (Amazon SNS) トピックの ARN。Amazon SNS 通知を使用していない場合は、空のリストを指定できます。

## レスポンス

### 構文

```
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"
}
```

### レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

### レスポンス本文

パイプラインの設定をテストすると、Elastic Transcoder により次の値が返されます。

### メッセージ

Success の値が false の場合、Messages には失敗したテストを説明する 1 つ以上メッセージの配列が含まれます。

### Success (成功)

オペレーションが成功した場合、この値は true になります。それ以外の場合、値は false になります。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

### 例

#### リクエスト例

```
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 endpoint/
elastictranscoder/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"]
}
```

#### レスポンス例

```
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"  
}
```

## ジョブのオペレーション

### トピック

- [ジョブを作成する](#)
- [パイプラインに基づくジョブのリスト](#)
- [ステータスに基づくジョブのリスト](#)
- [ジョブの読み取り](#)
- [ジョブをキャンセルする](#)

ジョブは、トランスコーディングの処理を実行します。各ジョブが1つのファイルを1つの形式に変換します。たとえば、1つのメディアファイルを6つの異なる形式に変換する場合は、6つのジョブを作成します。

このセクションでは、Elastic Transcoder API を使用してジョブに対して実行できるオペレーションについて説明します。Elastic Transcoder コンソールを使用して同じオペレーションを実行する方法など、ジョブの詳細については、「[ジョブの使用](#)」を参照してください。

## ジョブを作成する

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

### 説明

ジョブを作成するには、/2012-09-25/jobs リソースに POST リクエストを送信します。ジョブを作成するとすぐにジョブが開始されます。

**Note**

ジョブのステータスが変化するとき (Elastic Transcoder でジョブの処理をスタートまたは終了したときや、Elastic Transcoder で警告またはエラーの条件が発生したときなど) に通知するように、Elastic Transcoder を設定できます。詳細については、「[パイプラインを作成する](#)」を参照してください。

## リクエスト

### 構文

```
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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
  "Inputs":[{
    "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"
    },
    "TimeSpan":{
      "StartTime": "starting place of the clip, in
HH:mm:ss.SSS or ssss.SSS",
      "Duration": "duration of the clip, in HH:mm:ss.SSS
or ssss.SSS"
    },
    "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|aac|asf|avi|divx|flv|m4a|mkv|mov|mp2|mp3|
  mp4|mpeg|mpeg-ps|mpeg-ts|mxfl|ogg|vob|wav|webm",
"InputCaptions": {
  "MergePolicy": "MergeOverride|MergeRetain|Override",
  "CaptionSources": [
    {
      "Key": "name of the input caption file",
      "Encryption": {
        "Mode": "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"
    },
    {...}
  ]
}
},
{...}
]
"OutputKeyPrefix": "prefix for file names in Amazon S3 bucket",
"Outputs": [{
  "Key": "name of the transcoded file",
  "Encryption": {
    "Mode": "s3|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|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|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"
            },
            {...}
        ]
    },
    {...}
],
"Captions": {
    "CaptionFormats": [
        {
            "Format": "cea-708|dfxp|mov-text|scc|srt|webvtt",
            "Pattern": "myCaption/file-language",

```

```

        "Encryption":{
            "Mode":"s3|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|MPEG-DASH|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 acquisition url",
        "KeyStoragePolicy":"NoStore|WithVariantPlaylists"
    },
    "PlayReadyDrm":{
        "Format":"microsoft|discretix-3.0",
        "Key":"encrypted and base64-encoded DRM key",
        "KeyId":"id of the DRM key",
        "KeyMd5":"base64-encoded key digest",
        "InitializationVector":"base64-encoded
            initialization vector",
        "LicenseAcquisitionUrl":"license acquisition url"
    }
},
{...}],
"UserMetadata": {
    "Key":"Value",
    "Second user metadata key":"Second user metadata value"
}

```

```
  },  
  "PipelineId": "pipeline to use for transcoding"  
}
```

## リクエストパラメータ

このオペレーションはリクエストパラメータを使用しません。

## リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

## リクエスト本文

リクエストボディの JSON 文字列には、CreateJob オペレーションの入力オブジェクトが含まれます。入力オブジェクトの詳細については、「[Elastic Transcoder ジョブの作成時に指定する設定](#)」を参照してください。

## レスポンス

### 構文

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",  
    "Inputs": [ {  
      "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"  
      },  
      "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"
  },
  "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|aac|asf|avi|divx|flv|m4a|mkv|mov|mp2|mp3|
               mp4|mpeg|mpeg-ps|mpeg-ts|mxfl|ogg|vob|wav|webm",
  "InputCaptions": {
    "MergePolicy": "MergeOverride|MergeRetain|Override",
    "CaptionSources": [
      {
        "Key": "name of the input caption file",
        "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"
        "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"
        },
      },
    ],
  },
  {...}
]
}
},
{...}],
"OutputKeyPrefix": "prefix for file names in Amazon S3 bucket",
"Outputs": [{
  "Id": "sequential counter",
  "Key": "name of the transcoded file",
  "Encryption": {
    "Mode": "s3|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|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|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"
      },
      {...}
    ]
  }
]
```

```

    },
    {...}
  ],
  "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": {
    "CaptionFormats": [
      {
        "Format": "cea-708|dfxp|mov-text|scc|srt|webvtt",
        "Pattern": "myCaption/file-language",
        "Encryption": {
          "Mode": "s3||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|MPEG-DASH|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 acquisition url",
        "KeyStoragePolicy": "NoStore|WithVariantPlaylists"
    },
    "PlayReadyDrm": {
        "Format": "microsoft|discretix-3.0",
        "Key": "encrypted and base64-encoded DRM key",
        "KeyId": "id of the DRM key",
        "KeyMd5": "base64-encoded key digest",
        "InitializationVector": "base64-encoded
            initialization vector",
        "LicenseAcquisitionUrl": "license acquisition url"
    }
},
{...}
],
"UserMetadata": {
    "key1": "First user metadata value",
    "key2": "Second user metadata value"
},
"PipelineId": "pipeline to add the job to",
"Status": "Submitted|Progressing|Complete|Canceled|Error"
}
}
```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

ジョブを作成すると、リクエストで指定した値が Elastic Transcoder によって返されます。詳細については、「[Elastic Transcoder ジョブの作成時に指定する設定](#)」を参照してください。

また、以下の値も Elastic Transcoder によって返されます。

### (自動) Id

Elastic Transcoder でジョブに割り当てられた識別子。この値を使用して、ジョブの設定の取得やジョブの削除を行います。

### (自動) Outputs:Id

1 から始まるシーケンシャルカウンター。現在のジョブから出力間の出力を識別します。Output 構文ではこの値は常に 1 になります。

### (自動) Outputs:Duration

出力ファイルの継続時間 (秒)。端数は切り上げられます。

### (自動) Outputs:Width

出力ファイルの幅 (ピクセル単位)。

### (自動) Outputs:Height

出力ファイルの高さ (ピクセル)。

### (自動) Outputs:Status

ジョブ内の 1 つの出力のステータス。ジョブに対して出力を 1 つだけ指定した場合、Outputs:Status は常に Job:Status と同一になります。複数の出力を指定した場合:

- Elastic Transcoder で最初の出力の処理が開始されるまで、すべての出力の Job:Status と Outputs:Status が Submitted になります。
- Elastic Transcoder で最初の出力の処理が開始されると、その出力の Outputs:Status と Job:Status の両方が Progressing に変わります。各出力については、Outputs:Status の値は、Elastic Transcoder で出力の処理が開始されるまで Submitted のままになります。
- Job:Status は、すべての出力が終了ステータス (Complete または Error) に達するまで Progressing のままになります。
- すべての出力が終了ステータスに達すると、すべての出力の Outputs:Status が Complete である場合のみ、Job:Status が Complete に変わります。1 つまたは複数の出力の Outputs:Status が Error になっている場合は、Job:Status の終了ステータスも Error になります。

Status の値は、Submitted、Progressing、Complete、Canceled、Error のいずれかになります。

### (自動) Outputs:StatusDetail

Outputs:Status を詳しく説明した情報。

### (自動) Outputs:AppliedColorSpaceConversion

Elastic Transcoder で ColorSpaceConversionMode とともにプリセットを使用して出力ファイルのトランスコードを行う場合、AppliedColorSpaceConversion パラメータは、使用され

る変換を示します。プリセットで定義されている `ColorSpaceConversionMode` がない場合、このパラメータはジョブ応答に含まれません。

### (自動) Status

ジョブに対して複数の出力を指定した場合のジョブ全体のステータス。Elastic Transcoder でジョブの処理が開始された時点で `Job:Status` の値が `Progressing` に変わり、Elastic Transcoder でがすべての出力の処理が完了するまでそれは変わりません。処理が完了すると、`Job:Status` は、`Complete`、または、いずれかの出力が失敗した場合は `Error` に変わります。

ジョブに対して出力を 1 つだけ指定した場合、`Job:Status` は `Outputs:Status` と同一になります。

`Job:Status` の値は、`Submitted`、`Progressing`、`Complete`、`Canceled`、`Error` のいずれかになります。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

### Note

ジョブが `Access Denied` エラーで失敗する場合は、`Test Role API` アクションを実行して、エラーの原因を判断することをお勧めします。詳細については、「[ロールのテスト](#)」を参照してください。

## 例

以下のリクエスト例では、2 つの出力を指定したジョブを作成しています。

### リクエスト例

```
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 endpoint/
elastictranscoder/aws4_request,
    SignedHeaders=host;x-amz-date;x-amz-target,
    Signature=calculated-signature
Content-Length: number of characters in the JSON string
{
  "Inputs":[{
    "Key":"recipes/lasagna.mp4",
    "FrameRate":"auto",
    "Resolution":"auto",
    "AspectRatio":"auto",
    "Interlaced":"auto",
    "Container":"mp4",
    "InputCaptions":{
      "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"
        }
      ]
    }
  ]
}
"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":{
```

```
    "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"
```

```
}
```

## レスポンス例

```
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":"33333333333333-abcde3",
    "Inputs":[{
      "Key":"recipes/lasagna.mp4",
      "FrameRate":"auto",
      "Resolution":"auto",
      "AspectRatio":"auto",
      "Interlaced":"auto",
      "Container":"mp4",
      "InputCaptions":{
        "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"
          }
        ]
      }
    ]
  }
},
  "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":{"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":""
},
{
  "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":"111111111111-abcde1",
"Status":"Progressing"
}
}
```

## パイプラインに基づくジョブのリスト

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

## 説明

パイプラインに割り当てたジョブのリストを取得するには、`/2012-09-25/jobsByPipeline/pipelineId` リソースに GET リクエストを送信します。パイプラインに基づいてジョブのリストを取得すると、Elastic Transcoder では、指定したパイプラインに対して最近作成したジョブが一覧表示されます。

## リクエスト

### 構文

現在パイプラインに含まれているジョブに関する情報を取得するには、次の GET リクエストを送信します。

```
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 endpoint/
elastictranscoder/aws4_request,
    SignedHeaders=host;x-amz-date;x-amz-target,
    Signature=calculated-signature
```

## リクエストパラメータ

このオペレーションでは、次のリクエストパラメータを使用します。Elastic Transcoder は、指定したパイプラインに含まれる過去 6 か月のジョブを返します。

### PipelineId

指定したパイプラインに追加したジョブに関する情報を取得するには、そのパイプラインの Id オブジェクトの値を指定します。

### 昇順

ジョブを、送信された日時に基づいて古い順に一覧表示するには、`true` を入力します。ジョブを新しい順に一覧表示するには、`false` を入力します。

## PageToken

Elastic Transcoder から返される結果のページが複数にわたる場合、後続の GET リクエストで PageToken を使用することで、それぞれの結果のページを連続したページとして取得できます。

## リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

## リクエスト本文

このオペレーションではリクエストボディを使用しません。

## レスポンス

### 構文

```
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",
      "Inputs": [
        {
          "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"
          },
          "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"
          }
        }
      ]
    }
  ]
}
```

```

"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|aac|asf|avi|divx|flv|m4a|mkv|mov|mp2|mp3|
mp4|mpeg|mpeg-ps|mpeg-ts|mxfl|ogg|vob|wav|webm",
"DetectedProperties": {
  "Width": "video width in pixels",
  "Height": "video height in pixels",
  "FrameRate": "video frame rate in fps",
  "FileSize": "file size in bytes",
  "DurationMillis": "file duration in milliseconds"
},
"InputCaptions": {
"MergePolicy": "MergeOverride|MergeRetain|Override",
  "CaptionSources": [
    {
      "Key": "name of the input caption file",
      "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"
      "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"
      }
    },
    {
      ...
    }
  ]
},
{...}],
"OutputKeyPrefix": "prefix for file names in Amazon S3 bucket",
"Outputs": [{
  "Id": "sequential counter",
  "Key": "name of the transcoded file",
  "Encryption": {
    "Mode": "s3|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|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|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"
    },
    {...}
]
},
{...}],
"Duration": "duration in seconds",
"DurationMillis": "duration in milliseconds",
"Height": "height in pixels",
"Width": "width in pixels",
"FrameRate": "frame rate in fps",
"FileSize": "file size in bytes",
"Status": "Submitted|In Progress|Complete|Error",
"StatusDetail": "detail associated with Status",
"Captions": {
    "CaptionFormats": [
        {
            "Format": "cea-708|dfxp|mov-text|scc|srt|webvtt",
            "Pattern": "myCaption/file-language",
            "Encryption": {
                "Mode": "s3|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|MPEG-DASH|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 acquisition url",
      "KeyStoragePolicy":"NoStore|WithVariantPlaylists"
    },
    "PlayReadyDrm":{
      "Format":"microsoft|discretix-3.0",
      "Key":"encrypted and base64-encoded DRM key",
      "KeyId":"id of the DRM key",
      "KeyMd5":"base64-encoded key digest",
      "InitializationVector":"base64-encoded
        initialization vector",
      "LicenseAcquisitionUrl":"license acquisition url"
    }
  },
  {...}
],
"UserMetadata":
{
  "Key":"Value",
  "Second user metadata key":"Second user metadata value"
},
"PipelineId":"PipelineId for the job",
"Status":"Submitted|Progressing|Complete|Canceled|Error",
"Timing":{
  "SubmitTimeMillis":"job submitted time in epoch milliseconds",
  "StartTimeMillis":"job start time in epoch milliseconds",
  "FinishTimeMillis":"job finish time in epoch milliseconds"
}
},
{...}
],
"NextPageToken":value for accessing the next page of results|null
}

```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

レスポンス本文には、検索条件を満たすジョブごとに 1 つの要素が含まれ、各要素にはジョブの作成時に指定した値が含まれます。ジョブ値の詳細については、「[Elastic Transcoder ジョブの作成時に指定する設定](#)」を参照してください。

また、以下の値も Elastic Transcoder によって返されます。

### (自動) Id

Elastic Transcoder でジョブに割り当てられた識別子。この値を使用して、ジョブのプリセットの設定を取得したり、削除したりできます。

### (自動) Inputs:DetectedProperties

検出された入力ファイルプロパティ。Elastic Transcoder では、入力ファイルからこれらの値が識別されます。

### (自動) Inputs:Width

検出された入力ファイル幅 (ピクセル)。

### (自動) Inputs:Height

検出された入力ファイル高さ (ピクセル)。

### (自動) Inputs:FrameRate

検出された入力ファイルフレームレート (フレーム/秒)。

### (自動) Inputs: FileSize

検出された入力ファイルサイズ (バイト)。

### (自動) Inputs:DurationMillis

検出された入力ファイル継続時間 (ミリ秒)。

### (自動) Outputs:Id

1 から始まるシーケンシャルカウンター。現在のジョブから出力間の出力を識別します。Output 構文ではこの値は常に 1 になります。

### (自動) Outputs:Duration

出力ファイルの継続時間 (秒)。端数は切り上げられます。

**(自動) Outputs:DurationMillis**

出力ファイルの継続時間 (ミリ秒)。

**(自動) Outputs:Width**

出力ファイルの幅 (ピクセル単位)。

**(自動) Outputs:Height**

出力ファイルの高さ (ピクセル単位)。

**(自動) Outputs:FrameRate**

出力ファイルのフレームレート (フレーム/秒)。

**(自動) Outputs:FileSize**

出力ファイルのファイルサイズ (バイト)。

**(自動) Outputs:Status**

ジョブ内の 1 つの出力のステータス。ジョブに対して出力を 1 つだけ指定した場合、Outputs:Status は常に Job:Status と同一になります。複数の出力を指定した場合:

- Elastic Transcoder で最初の出力の処理が開始されるまで、すべての出力の Job:Status と Outputs:Status が Submitted になります。
- Elastic Transcoder で最初の出力の処理が開始されると、その出力の Outputs:Status と Job:Status の両方が Progressing に変わります。各出力については、Outputs:Status の値は、Elastic Transcoder で出力の処理が開始されるまで Submitted のままになります。
- Job:Status は、すべての出力が終了ステータス (Complete または Error) に達するまで Progressing のままになります。
- すべての出力が終了ステータスに達すると、すべての出力の Outputs:Status が Complete である場合のみ、Job:Status が Complete に変わります。1 つまたは複数の出力の Outputs:Status が Error になっている場合は、Job:Status の終了ステータスも Error になります。

Status の値は、Submitted、Progressing、Complete、Canceled、Error のいずれかになります。

**(自動) Outputs:StatusDetail**

Outputs:Status を詳しく説明した情報。

### (自動) Outputs:AppliedColorSpaceConversion

Elastic Transcoder で ColorSpaceConversionMode とともにプリセットを使用して出力ファイルのトランスコードを行う場合、AppliedColorSpaceConversion パラメータは、使用される変換を示します。プリセットで定義されている ColorSpaceConversionMode がいない場合、このパラメータはジョブ応答に含まれません。

### (自動) Status

ジョブに対して複数の出力を指定した場合のジョブ全体のステータス。Elastic Transcoder でジョブの処理が開始された時点で Job:Status の値が Progressing に変わり、Elastic Transcoder でがすべての出力の処理が完了するまでそれは変わりません。処理が完了すると、Job:Status は、Complete、または、いずれかの出力が失敗した場合は Error に変わります。

ジョブに対して出力を 1 つだけ指定した場合、Job:Status は Outputs:Status と同一になります。

Job:Status の値は、Submitted、Progressing、Complete、Canceled、Error のいずれかになります。

### (自動) Timing

ジョブのタイミングに関する詳細。

#### (自動) Timing:SubmitTimeMillis

ジョブが Elastic Transcoder に送信された時間 (エポックミリ秒単位)。

#### (自動) Timing:StartTimeMillis

ジョブがトランスコーディングを開始した時間 (エポックミリ秒単位)。

#### (自動) Timing:FinishTimeMillis

ジョブがトランスコーディングを完了した時間 (エポックミリ秒単位)。

エポックタイムの詳細については、Wikipedia の「[Epoch \(computing\)](#)」のページを参照してください。

### (自動) NextPageToken

結果の 2 ページ目以降のページ (存在する場合) にアクセスするために使用する値。指定したパイプラインのジョブが 1 ページに収まる場合、または結果の最後のページに達したとき、NextPageToken の値は null になります。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

### 例

次のサンプルリクエストでは、ジョブを作成します。

#### リクエスト例

次のサンプルリクエストでは、パイプライン ID 111111111111-abcde1 に過去 6 か月に割り当てたすべてのジョブのリストを取得します。

```
GET /2012-09-25/jobsByPipeline/111111111111-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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
```

#### レスポンス例

```
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": "333333333333-abcde3",
      "Inputs": [
        {
          "Key": "cooking/lasagna.mp4",
          "FrameRate": "auto",
          "Resolution": "auto",
          "AspectRatio": "auto",
          "Interlaced": "auto",
```

```
"Container": "mp4",
"InputCaptions": {
  "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"
    }
  ],
},
"DetectedProperties": {
  "Width": "1280",
  "Height": "720",
  "FrameRate": "30.00",
  "FileSize": "5872000",
  "DurationMillis": "1003000"
},
"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",
    "DurationMillis": "1003000",
    "Width": "1280",
    "Height": "720",
    "FrameRate": "30.00",
    "FileSize": "5872000",
```

```
"Status": "Submitted",
"StatusDetail": "Job has been received.",
"Captions": {
  "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",
  "DurationMillis": "1003000",
  "Width": "1920",
  "Height": "1080",
  "FrameRate": "30.00",
  "FileSize": "4718600",
  "Status": "Submitted",
  "StatusDetail": "Job has been received.",
  "AppliedColorSpaceConversion": "None"
},
{
  "PipelineId": "1111111111111-abcde1",
  "Timing": {
    "SubmitTime": "1427212800000",
```

```
    "StartTime": "1427212856000",
    "FinishTime": "1427212875000"
  }
},
{
  "Id": "4444444444444444-abcde4",
  "Input": {
    "Key": "cooking/baked-ziti.mp4",
    "FrameRate": "auto",
    "Resolution": "auto",
    "AspectRatio": "auto",
    "Interlaced": "auto",
    "Container": "mp4",
    "DetectedProperties": {
      "Width": "1280",
      "Height": "720",
      "FrameRate": "30.00",
      "FileSize": "5872000",
      "DurationMillis": "1003000"
    }
  },
  "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",
      "DurationMillis": "1003000",
      "Width": "1280",
      "Height": "720",
      "FrameRate": "30.00",
      "FileSize": "4718600",
      "Status": "Complete",
      "StatusDetail": ""
    }
  ],
}
```

```
    "UserMetadata":
      {
        "Food type":"Italian",
        "Cook book":"recipe notebook"
      },
    "PipelineId":"11111111111111-abcde1",
    "Timing":{
      "SubmitTime":"1427212800000",
      "StartTime":"1427212856000",
      "FinishTime":"1427212875000"
    }
  },
  "NextPageToken":null
}
```

## ステータスに基づくジョブのリスト

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

### 説明

指定したステータスに該当する ジョブのリストを取得するために GET リクエストを `/2012-09-25/jobsByStatus/Submitted` リソースに送信します。Elastic Transcoder では、直近に作成したジョブと、現在は指定した状態になっているジョブの一覧が表示されます。

### リクエスト

#### 構文

指定したステータスに該当しており、現在の AWS アカウントに関連付けられた過去 6 か月のジョブに関する情報を取得するには、次の GET リクエストを送信します。

```
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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
```

## リクエストパラメータ

このオペレーションでは、次のリクエストパラメータを使用します。Elastic Transcoder によって、指定したステータスに該当する過去 6 か月のジョブがすべて返されます。

### ステータス

指定したステータスに該当しており、現在の AWS アカウントに関連付けられた過去 6 か月のすべてのジョブに関する情報を取得するには、ステータス (Submitted、Progressing、Complete、Canceled、Error) を指定します。

### 昇順

ジョブを、送信された日時に基づいて古い順に一覧表示するには、true を入力します。ジョブを新しい順に一覧表示するには、false を入力します。

### PageToken

Elastic Transcoder から返される結果のページが複数にわたる場合、後続の GET リクエストで PageToken を使用することで、それぞれの結果のページを連続したページとして取得できます。

## リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

## リクエスト本文

このオペレーションではリクエストボディを使用しません。

## レスポンス

### 構文

```
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",
      "Inputs": [ {
        "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"
        },
        "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"
        },
        "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|aac|asf|avi|divx|flv|m4a|mkv|mov|mp2|mp3|
          mp4|mpeg|mpeg-ps|mpeg-ts|mxfl|ogg|vob|wav|webm",
        "DetectedProperties": {
          "Width": "video width in pixels",
          "Height": "video height in pixels",
          "FrameRate": "video frame rate in fps",
          "FileSize": "file size in bytes",
          "DurationMillis": "file duration in milliseconds"
        },
        "InputCaptions": {
          "MergePolicy": "MergeOverride|MergeRetain|Override",
          "CaptionSources": [
```

```

    {
      "Key": "name of the input caption file",
      "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"
      "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"
      },
    },
    {...}
  ]
}
},
{...}],
"OutputKeyPrefix": "prefix for file names in Amazon S3 bucket",
"Outputs": [{
  "Id": "sequential counter",
  "Key": "name of the transcoded file",
  "Encryption": {
    "Mode": "s3||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||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|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"
      },
      {...}
    ]
  },
  {...}],
"Duration": "duration in seconds",
"DurationMillis": "duration in milliseconds",
"Height": "height in pixels",
"Width": "width in pixels",
"FrameRate": "frame rate in fps",
"FileSize": "file size in bytes",
"Status": "Submitted|In Progress|Complete|Error",
"StatusDetail": "detail associated with Status",
"Captions": {
  "CaptionFormats": [

```

```

        {
            "Format": "cea-708|dfxp|mov-text|scc|srt|webvtt",
            "Pattern": "myCaption/file-language",
            "Encryption": {
                "Mode": "s3||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|MPEG-DASH|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 acquisition url",
            "KeyStoragePolicy": "NoStore|WithVariantPlaylists"
        },
        "PlayReadyDrm": {
            "Format": "microsoft|discretix-3.0",
            "Key": "encrypted and base64-encoded DRM key",
            "KeyId": "id of the DRM key",
            "KeyMd5": "base64-encoded key digest",
            "InitializationVector": "base64-encoded
                initialization vector",

```

```
        "LicenseAcquisitionUrl": "license acquisition url"
      }
    },
    {...}
  ],
  "UserMetadata": {
    "Key": "Value",
    "Second user metadata key": "Second user metadata value"
  },
  "PipelineId": "PipelineId for the job",
  "Status": "Submitted|Progressing|Complete|Canceled|Error",
  "Timing": {
    "SubmitTimeMillis": "job submitted time in epoch milliseconds",
    "StartTimeMillis": "job start time in epoch milliseconds",
    "FinishTimeMillis": "job finish time in epoch milliseconds"
  }
},
{...}
],
"NextPageToken": value for accessing the next page of results|null
}
```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

レスポンス本文には、検索条件を満たすジョブごとに1つの要素が含まれ、各要素にはジョブの作成時に指定した値が含まれます。ジョブ値の詳細については、「[Elastic Transcoder ジョブの作成時に指定する設定](#)」を参照してください。

また、以下の値も Elastic Transcoder によって返されます。

### (自動) Id

Elastic Transcoder でジョブに割り当てられた識別子。この値を使用して、ジョブのプリセットの設定を取得したり、削除したりできます。

**(自動) Inputs:DetectedProperties**

検出された入力ファイルプロパティ。Elastic Transcoder では、入力ファイルからこれらの値が識別されます。

**(自動) Inputs:Width**

検出された入力ファイル幅 (ピクセル)。

**(自動) Inputs:Height**

検出された入力ファイル高さ (ピクセル)。

**(自動) Inputs:FrameRate**

検出された入力ファイルフレームレート (フレーム/秒)。

**(自動) Inputs: FileSize**

検出された入力ファイルサイズ (バイト)。

**(自動) Inputs:DurationMillis**

検出された入力ファイル継続時間 (ミリ秒)。

**(自動) Outputs:Id**

1 から始まるシーケンシャルカウンター。現在のジョブから出力間の出力を識別します。Output 構文ではこの値は常に 1 になります。

**(自動) Outputs:Duration**

出力ファイルの継続時間 (秒)。端数は切り上げられます。

**(自動) Outputs:DurationMillis**

出力ファイルの継続時間 (ミリ秒)。

**(自動) Outputs:Width**

出力ファイルの幅 (ピクセル単位)。

**(自動) Outputs:Height**

出力ファイルの高さ (ピクセル単位)。

**(自動) Outputs:FrameRate**

出力ファイルのフレームレート (フレーム/秒)。

### (自動) Outputs:FileSize

出力ファイルのファイルサイズ (バイト)。

### (自動) Outputs:Status

ジョブ内の 1 つの出力のステータス。ジョブに対して出力を 1 つだけ指定した場合、Outputs:Status は常に Job:Status と同一になります。複数の出力を指定した場合:

- Elastic Transcoder で最初の出力の処理が開始されるまで、すべての出力の Job:Status と Outputs:Status が Submitted になります。
- Elastic Transcoder で最初の出力の処理が開始されると、その出力の Outputs:Status と Job:Status の両方が Progressing に変わります。各出力については、Outputs:Status の値は、Elastic Transcoder で出力の処理が開始されるまで Submitted のままになります。
- Job:Status は、すべての出力が終了ステータス (Complete または Error) に達するまで Progressing のままになります。
- すべての出力が終了ステータスに達すると、すべての出力の Outputs:Status が Complete である場合のみ、Job:Status が Complete に変わります。1 つまたは複数の出力の Outputs:Status が Error になっている場合は、Job:Status の終了ステータスも Error になります。

Status の値は、Submitted、Progressing、Complete、Canceled、Error のいずれかになります。

### (自動) Outputs:StatusDetail

Outputs:Status を詳しく説明した情報。

### (自動) Outputs:AppliedColorSpaceConversion

Elastic Transcoder で ColorSpaceConversionMode とともにプリセットを使用して出力ファイルのトランスコードを行う場合、AppliedColorSpaceConversion パラメータは、使用される変換を示します。プリセットで定義されている ColorSpaceConversionMode がいない場合、このパラメータはジョブ応答に含まれません。

### (自動) Status

ジョブに対して複数の出力を指定した場合のジョブ全体のステータス。Elastic Transcoder でジョブの処理が開始された時点で Job:Status の値が Progressing に変わり、Elastic Transcoder ですべての出力の処理が完了するまでそれは変わりません。処理が完了すると、Job:Status は、Complete、または、いずれかの出力が失敗した場合は Error に変わります。

ジョブに対して出力を 1 つだけ指定した場合、`Job:Status` は `Outputs:Status` と同一になります。

`Job:Status` の値は、`Submitted`、`Progressing`、`Complete`、`Canceled`、`Error` のいずれかになります。

#### (自動) Timing

ジョブのタイミングに関する詳細。

#### (自動) Timing:SubmitTimeMillis

ジョブが Elastic Transcoder に送信された時間 (エポックミリ秒単位)。

#### (自動) Timing:StartTimeMillis

ジョブがトランスコーディングを開始した時間 (エポックミリ秒単位)。

#### (自動) Timing:FinishTimeMillis

ジョブがトランスコーディングを完了した時間 (エポックミリ秒単位)。

エポックタイムの詳細については、Wikipedia の「[Epoch \(computing\)](#)」のページを参照してください。

#### (自動) NextPageToken

結果の 2 ページ目以降のページ (存在する場合) にアクセスするために使用する値。指定したパイプラインのジョブが 1 ページに収まる場合、または結果の最後のページに達したとき、`NextPageToken` の値は `null` になります。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

## 例

次のサンプルリクエストでは、ジョブを作成します。

### リクエスト例

次のサンプルリクエストでは、過去 6 か月に作成したジョブのうち、`Complete` ステータスに該当するすべてのジョブのリストを取得します。

```
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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
```

## レスポンス例

```
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",
          "InputCaptions": {
            "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"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

```
    }
  ]
},
"DetectedProperties":{
  "Width":"1280",
  "Height":"720",
  "FrameRate":"30.00",
  "FileSize":"5872000",
  "DurationMillis":"1003000"
}
]],
"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",
    "DurationMillis":"1003000",
    "Width":"1280",
    "Height":"720",
    "FrameRate":"30.00",
    "FileSize":"5872000",
    "Status":"Complete",
    "StatusDetail":"","
    "Captions":{
      "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",
  "DurationMillis": "1003000",
  "Width": "1136",
  "Height": "640",
  "FrameRate": "30.00",
  "FileSize": "4718600",
  "Status": "Complete",
  "StatusDetail": ""
},
],
"PipelineId": "111111111111-abcde1",
"Playlists": [
  {
    "Format": "HLSv3",
    "Name": "playlist-iPhone-lasagna.m3u8",
    "OutputKeys": [
      "iphone/lasagna-1024k",
      "iphone/lasagna-512k"
    ]
  }
],
"Timing": {
  "SubmitTime": "1427212800000",
  "StartTime": "1427212856000",
  "FinishTime": "1427212875000"
},
"Status": "Complete"
},
{
  "Id": "44444444444444-abcde4",
  "Input": {
```

```
    "Key": "cooking/spaghetti.mp4",
    "FrameRate": "auto",
    "Resolution": "auto",
    "AspectRatio": "auto",
    "Interlaced": "auto",
    "Container": "mp4",
    "DetectedProperties": {
      "Width": "1280",
      "Height": "720",
      "FrameRate": "30.00",
      "FileSize": "5872000",
      "DurationMillis": "1003000"
    }
  },
  "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",
      "DurationMillis": "1003000",
      "Width": "1136",
      "Height": "640",
      "FrameRate": "30.00",
      "FileSize": "5872000",
      "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",
"Timing":{
  "SubmitTime":"1427212800000",
  "StartTime":"1427212856000",
  "FinishTime":"1427212875000"
}
}
],
"NextPageToken":null
}
```

## ジョブの読み取り

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

### 説明

ジョブに関する詳細情報を取得するには、`/2012-09-25/jobs/jobId` リソースに GET リクエストを送信します。

### リクエスト

### 構文

```
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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
```

## リクエストパラメータ

このオペレーションでは、次のリクエストパラメーターを使用します。

### JobId

詳細情報を取得するジョブの識別子。

## リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

## リクエスト本文

この操作にリクエストボディはありません。

## レスポンス

### 構文

```
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",
    "Inputs":[{
      "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"
  },
  "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"
  },
  "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|aac|asf|avi|divx|flv|m4a|mkv|mov|mp2|mp3|
    mp4|mpeg|mpeg-ps|mpeg-ts|mxfl|ogg|vob|wav|webm",
  "DetectedProperties": {
    "Width": "video width in pixels",
    "Height": "video height in pixels",
    "FrameRate": "video frame rate in fps",
    "FileSize": "file size in bytes",
    "DurationMillis": "file duration in milliseconds"
  },
  "InputCaptions": {
    "MergePolicy": "MergeOverride|MergeRetain|Override",
    "CaptionSources": [
      {
        "Key": "name of the input caption file",
        "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"
        "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"
        },
      },
      { ... }
    ]
  }
},
{ ... }],

```

```

"OutputKeyPrefix": "prefix for file names in Amazon S3 bucket",
"Outputs": [{
  "Id": "sequential counter",
  "Key": "name of the transcoded file",
  "Encryption": {
    "Mode": "s3|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|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|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"
    },
    {...}
]
},
{...}],
"Duration": "duration in seconds",
"DurationMillis": "duration in milliseconds",
"Height": "height in pixels",
"Width": "width in pixels",
"FrameRate": "frame rate in fps",
"FileSize": "file size in bytes",
"Status": "Submitted|In Progress|Complete|Error",
"StatusDetail": "detail associated with Status",
"Captions": {
    "CaptionFormats": [
        {
            "Format": "cea-708|dfxp|mov-text|scc|srt|webvtt",
            "Pattern": "myCaption/file-language",
            "Encryption": {
                "Mode": "s3|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|MPEG-DASH|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 acquisition url",
      "KeyStoragePolicy":"NoStore|WithVariantPlaylists"
    },
    "PlayReadyDrm":{
      "Format":"microsoft|discretix-3.0",
      "Key":"encrypted and base64-encoded DRM key",
      "KeyId":"id of the DRM key",
      "KeyMd5":"base64-encoded key digest",
      "InitializationVector":"base64-encoded initialization vector",
      "LicenseAcquisitionUrl":"license acquisition url"
    }
  },
  {...}
],
"UserMetadata":
{
  "Key":"Value",
  "Second user metadata key":"Second user metadata value"
},
"PipelineId":"PipelineId for the job",
"Status":"Submitted|Progressing|Complete|Canceled|Error",
"Timing":{
  "SubmitTimeMillis":"job submitted time in epoch milliseconds",
  "StartTimeMillis":"job start time in epoch milliseconds",
  "FinishTimeMillis":"job finish time in epoch milliseconds"
}
}
```

```
}
```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

検索条件を満たすジョブごとに、レスポンス本文にはジョブの作成時に指定した値が含まれます。ジョブ値の詳細については、「[Elastic Transcoder ジョブの作成時に指定する設定](#)」を参照してください。

また、以下の値も Elastic Transcoder によって返されます。

### (自動) Id

Elastic Transcoder でジョブに割り当てられた識別子。この値を使用して、ジョブのプリセットの設定を取得したり、削除したりできます。

### (自動) Inputs:DetectedProperties

検出された入力ファイルプロパティ。Elastic Transcoder では、入力ファイルからこれらの値が識別されます。

### (自動) Inputs:Width

検出された入力ファイル幅 (ピクセル)。

### (自動) Inputs:Height

検出された入力ファイル高さ (ピクセル)。

### (自動) Inputs:FrameRate

検出された入力ファイルフレームレート (フレーム/秒)。

### (自動) Inputs: FileSize

検出された入力ファイルサイズ (バイト)。

### (自動) Inputs:DurationMillis

検出された入力ファイル継続時間 (ミリ秒)。

### (自動) Outputs:Id

1 から始まるシーケンシャルカウンター。現在のジョブから出力間の出力を識別します。Output 構文ではこの値は常に 1 になります。

### (自動) Outputs:Duration

出力ファイルの継続時間 (秒)。端数は切り上げられます。

### (自動) Outputs:DurationMillis

出力ファイルの継続時間 (ミリ秒)。

### (自動) Outputs:Width

出力ファイルの幅 (ピクセル単位)。

### (自動) Outputs:Height

出力ファイルの高さ (ピクセル単位)。

### (自動) Outputs:FrameRate

出力ファイルのフレームレート (フレーム/秒)。

### (自動) Outputs:FileSize

出力ファイルのファイルサイズ (バイト)。

### (自動) Outputs:Status

ジョブ内の 1 つの出力のステータス。ジョブに対して出力を 1 つだけ指定した場合、Outputs:Status は常に Job:Status と同一になります。複数の出力を指定した場合:

- Elastic Transcoder で最初の出力の処理が開始されるまで、すべての出力の Job:Status と Outputs:Status が Submitted になります。
- Elastic Transcoder で最初の出力の処理が開始されると、その出力の Outputs:Status と Job:Status の両方が Progressing に変わります。各出力については、Outputs:Status の値は、Elastic Transcoder で出力の処理が開始されるまで Submitted のままになります。
- Job:Status は、すべての出力が終了ステータス (Complete または Error) に達するまで Progressing のままになります。
- すべての出力が終了ステータスに達すると、すべての出力の Outputs:Status が Complete である場合のみ、Job:Status が Complete に変わります。1 つまたは複数の出力の Outputs:Status が Error になっている場合は、Job:Status の終了ステータスも Error になります。

Status の値は、Submitted、Progressing、Complete、Canceled、Error のいずれかになります。

#### (自動) Outputs:StatusDetail

Outputs:Status を詳しく説明した情報。

#### (自動) Outputs:AppliedColorSpaceConversion

Elastic Transcoder で ColorSpaceConversionMode とともにプリセットを使用して出力ファイルのトランスコードを行う場合、AppliedColorSpaceConversion パラメータは、使用される変換を示します。プリセットで定義されている ColorSpaceConversionMode がいない場合、このパラメータはジョブ応答に含まれません。

#### (自動) Status

ジョブに対して複数の出力を指定した場合のジョブ全体のステータス。Elastic Transcoder でジョブの処理が開始された時点で Job:Status の値が Progressing に変わり、Elastic Transcoder でがすべての出力の処理が完了するまでそれは変わりません。処理が完了すると、Job:Status は、Complete、または、いずれかの出力が失敗した場合は Error に変わります。

ジョブに対して出力を 1 つだけ指定した場合、Job:Status は Outputs:Status と同一になります。

Job:Status の値は、Submitted、Progressing、Complete、Canceled、Error のいずれかになります。

#### (自動) Timing

ジョブのタイミングに関する詳細。

#### (自動) Timing:SubmitTimeMillis

ジョブが Elastic Transcoder に送信された時間 (エポックミリ秒単位)。

#### (自動) Timing:StartTimeMillis

ジョブがトランスコーディングを開始した時間 (エポックミリ秒単位)。

#### (自動) Timing:FinishTimeMillis

ジョブがトランスコーディングを完了した時間 (エポックミリ秒単位)。

エポックタイムの詳細については、Wikipedia の「[Epoch \(computing\)](#)」のページを参照してください。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

### 例

#### リクエスト例

次のサンプルリクエストでは、ジョブ ID が 3333333333333-abcde3 であるジョブを取得します。

```
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 endpoint/
    elastictranscoder/aws4_request,
    SignedHeaders=host;x-amz-date;x-amz-target,
    Signature=calculated-signature
```

#### レスポンス例

```
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",
    "Inputs": [ {
      "Key": "cooking/lasagna.mp4",
      "FrameRate": "auto",
      "Resolution": "auto",
      "AspectRatio": "auto",
      "Interlaced": "auto",
      "Container": "mp4",
      "InputCaptions": {
        "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"
    }
  ],
}
"DetectedProperties": {
  "Width": "1280",
  "Height": "720",
  "FrameRate": "30.00",
  "FileSize": "5872000",
  "DurationMillis": "1003000"
}
}],
"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",
    "DurationMillis": "1003000",
    "Width": "1280",
    "Height": "720",
    "FrameRate": "30.00",
    "FileSize": "5872000",
    "Status": "Progressing",
    "StatusDetail": "",
    "Captions": {
```

```
    "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",
    "DurationMillis":"1003000",
    "Width":"1136",
    "Height":"640",
    "FrameRate":"30.00",
    "FileSize":"4718600",
    "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",
    "DurationMillis":"1003000",
    "Width":"1136",
    "Height":"640",
    "FrameRate":"30.00",
```

```
        "FileSize": "3508900",
        "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": "111111111111-abcde1",
"Status": "Progressing",
"Timing": {
    "SubmitTime": "1427212800000",
    "StartTime": "1427212856000",
    "FinishTime": "1427212875000"
}
}
```

## ジョブをキャンセルする

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

## 説明

Elastic Transcoder による処理がまだ開始されていないジョブをキャンセルするには、DELETE リクエストを `/2012-09-25/jobs/jobId` リソースに送信します。Read Job、List Job by Pipeline、List Job by Status のいずれかのリクエストを送信した場合、Elastic Transcoder によりそれでもジョブが返される可能性はありますが、Elastic Transcoder により入力ファイルのトランスコードが行われないため、そのジョブに対して課金されることはありません。

### Note

キャンセルできるのは、ステータスが [Submitted] (送信済み) になっているジョブのみです。ジョブの識別子の取得中にパイプラインでジョブの処理が開始されないようにするには、「[パイプラインのステータスを更新する](#)」の手順を使用してパイプラインを一時停止します。

## リクエスト

### 構文

```
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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
```

### リクエストパラメータ

このオペレーションでは、次のリクエストパラメーターを使用します。

#### JobId

キャンセルするジョブの識別子。

ステータスが Submitted であるジョブのリストを (jobId も含めて) 取得するには、[ステータスに基づくジョブのリスト](#) API アクションを使用します。

## リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

## リクエスト本文

この操作にリクエストボディはありません。

## レスポンス

### 構文

```
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"
}
```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

応答の本文には、以下の JSON オブジェクトが含まれます。

### Success (成功)

ジョブが正常にキャンセルされた場合、Success の値は true です。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

## 例

以下のリクエスト例では、ID が 333333333333-abcde3 であるジョブをキャンセルしています。

### リクエスト例

```
DELETE /2012-09-25/jobs/333333333333-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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
```

### レスポンス例

```
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"
}
```

## プリセットのオプション

### トピック

- [プリセットを作成する](#)
- [プリセットのリスト](#)
- [プリセットを読み取る](#)
- [プリセットを削除する](#)

プリセットは、メディアファイルに対してある形式から別の形式へのトランスコードを行うための設定の大半が含まれているテンプレートです。Elastic Transcoder には、一般的な形式 (iPod と iPhone

の複数のバージョンなど)用にデフォルトのプリセットがいくつか用意されています。デフォルトのプリセットに含まれていない形式用に独自のプリセットを作成することもできます。使用するプリセットはジョブの作成時に指定します。

このセクションでは、Elastic Transcoder API を使用してプリセットに対して実行できるオペレーションについて説明します。Elastic Transcoder コンソールを使用して同じオペレーションを実行する方法など、プリセットの詳細については、「[プリセットの使用](#)」を参照してください。

## プリセットを作成する

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

### 説明

プリセットを作成するには、/2012-09-25/presets リソースに POST リクエストを送信します。

#### Important

指定した設定は Elastic Transcoder によってチェックされ、Elastic Transcoder の要件を満たしているかどうか、該当するスタンダードに準拠しているかどうかを確認されます。設定が Elastic Transcoder に対して有効でない場合、Elastic Transcoder によって HTTP 400 応答 (ValidationException) が返され、プリセットは作成されません。設定が Elastic Transcoder に対して有効であっても、スタンダードに厳密に準拠していない場合、Elastic Transcoder によってプリセットは作成されますが、応答で警告メッセージが返されます。これは、設定がスタンダードに準拠しているかどうかを判断しつつ、Elastic Transcoder によって生成されるファイルをより柔軟に受け入れるためにも役立ちます。

**Note**

Elastic Transcoder には、一般的な出力形式用にシステムプリセットがいくつか用意されています。詳細については、トピック「」の「PresetId [ジョブを作成する](#)」を参照してください。

## リクエスト

### 構文

```
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 endpoint/
elastictranscoder/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": "flac|flv|fmp4|gif|mp2|mp3|mp4|mpg|mx|oga|ogg|ts|wav|webm",
  "Audio": {
    "Codec": "AAC|flac|mp2|mp3|pcm|vorbis",
    "CodecOptions": {
      "Profile": "auto|AAC-LC|HE-AAC|HE-AACv2",
      "BitDepth": "8|16|24|32",
      "Signed": "Signed|Unsigned",
      "BitOrder": "LittleEndian"
    },
    "SampleRate": "auto|22050|32000|44100|48000|96000",
    "BitRate": "audio bit rate of output file in kilobits/second",
    "Channels": "auto|0|1|2",
    "AudioPackingMode": "SingleTrack|OneChannelPerTrack|
                        OneChannelPerTrackWithMosTo8Tracks"
  },
  "Video": {
    "Codec": "gif|H.264|mpeg2|vp8|vp9",
```

```

"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 pixels height in pixels" <not recommended>,
"AspectRatio": "auto|1:1|4:3|3:2|16:9" <not recommended>
}
}
```

## リクエストパラメータ

このオペレーションはリクエストパラメータを使用しません。

## リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

## リクエスト本文

リクエストボディの JSON 文字列には、CreatePreset オペレーションの入力オブジェクトが含まれます。入力オブジェクトの詳細については、「[Elastic Transcoder プリセットの作成時に指定する設定](#)」を参照してください。

## レスポンス

### 構文

```
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": "flac|flv|fmp4|gif|mp2|mp3|mp4|mpg|mpeg|mx4|oga|ogg|ts|wav|webm",
    "Audio": {
      "Codec": "AAC|flac|mp2|mp3|pcm|vorbis",
      "CodecOptions": {
        "Profile": "auto|AAC-LC|HE-AAC|HE-AACv2",
        "BitDepth": "8|16|24|32",
        "Signed": "Signed|Unsigned",

```

```

    "BitOrder":"LittleEndian"
  },
  "SampleRate":"auto|22050|32000|44100|48000|96000",
  "BitRate":"audio bit rate of output file in kilobits/second",
  "Channels":"auto|0|1|2",
  "AudioPackingMode":"SingleTrack|OneChannelPerTrack|
    OneChannelPerTrackWithMosTo8Tracks"
},
"Video":{
  "Codec":"gif|H.264|mpeg2|vp8|vp9",
  "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 pixels x height in pixels",
    "AspectRatio": "auto|1:1|4:3|3:2|16:9"
}
},
"Warning": "message about codec compatibility"
}
```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

ジョブを作成すると、リクエストで指定した値が Elastic Transcoder によって返されます。詳細については、「[リクエスト本文](#)」を参照してください。

また、以下の値も Elastic Transcoder によって返されます。

### (自動) Id

新しいプリセットの識別子。この値を使用して、プリセットの設定の取得や削除を行います。

### (自動) Type

そのプリセットが、Elastic Transcoder によって提供されるデフォルトのプリセット (System) であるか、または定義したプリセット (Custom) であるかが示されます。

### (自動) Warning

プリセットの設定がコーデックの基準に準拠していないものの、その設定によって許容可能な出力が生成される可能性がある場合、Elastic Transcoder によりプリセットが作成され、プリセット設定が準拠していないことを示す警告が表示されます。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

### 例

以下のリクエスト例では、DefaultPreset という名前のプリセットを作成しています。

#### リクエスト例

```
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 endpoint/
    elastictranscoder/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"
}
}
```

## レスポンス例

```
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":"555555555555-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": ""
}
```

## プリセットのリスト

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

### 説明

現在の AWS アカウントに関連付けられたすべてのプリセットのリストを取得するには、`/2012-09-25/presets` リソースに GET リクエストを送信します。

## リクエスト

### 構文

```
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 endpoint/
    elastictranscoder/aws4_request,
    SignedHeaders=host;x-amz-date;x-amz-target,
    Signature=calculated-signature
```

### リクエストパラメータ

このオペレーションでは、次のリクエストパラメータを使用します。Elastic Transcoder により、すべての利用可能なプリセットが返されます。

### 昇順

プリセットを、送信された日時に基づいて古い順に一覧表示するには、true を入力します。プリセットを新しい順に一覧表示するには、false を入力します。

### PageToken

Elastic Transcoder から返される結果のページが複数にわたる場合、後続の GET リクエストで PageToken を使用することで、それぞれの結果のページを連続したページとして取得できます。

### リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

### リクエスト本文

リクエストボディの JSON 文字列には、次のオブジェクトが含まれます。

# レスポンス

## 構文

```
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": "flac|flv|fmp4|gif|mp2|mp3|mp4|mpg|mpeg|mxv|oga|ogg|ts|wav|webm",
      "Audio":{
        "Codec": "AAC|flac|mp2|mp3|pcm|vorbis",
        "CodecOptions":{
          "Profile": "auto|AAC-LC|HE-AAC|HE-AACv2",
          "BitDepth": "8|16|24|32",
          "Signed": "Signed|Unsigned",
          "BitOrder": "LittleEndian"
        },
        "SampleRate": "auto|22050|32000|44100|48000|96000",
        "BitRate": "audio bit rate of output file in kilobits/second",
        "Channels": "auto|0|1|2",
        "AudioPackingMode": "SingleTrack|OneChannelPerTrack|
          OneChannelPerTrackWithMosTo8Tracks"
      },
      "Video":{
        "Codec": "gif|H.264|mpeg2|vp8|vp9",
        "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"
  },
  },
  {...},
],
"NextPageToken":value for accessing the next page of results|null
}

```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

レスポンス本文の JSON 文字列は、プリセットの作成時に指定した値を返します。各オブジェクトの詳細については、「[Elastic Transcoder プリセットの作成時に指定する設定](#)」を参照してください。

また、以下の値も Elastic Transcoder によって返されます。

### (自動) Id

プリセットの識別子。この値を使用して、プリセットの設定を取得したり、削除したりできます。

### (自動) Type

そのプリセットが、Elastic Transcoder によって提供されるデフォルトのプリセット (System) であるか、または定義したプリセット (Custom) であるかが示されます。

### (自動) NextPageToken

結果の 2 ページ目以降のページ (存在する場合) にアクセスするために使用する値。プリセットが 1 ページに収まる場合、または結果の最後のページに達したとき、NextPageToken の値は null になります。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

## 例

次のサンプルリクエストでは、DefaultPreset という名前のプリセットを作成します。

### リクエスト例

```
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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
```

## レスポンス例

```
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": "55555555555555-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"
},
},
{...}
]
```

## プリセットを読み取る

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

## 説明

プリセットの詳細情報を取得するには、`/2012-09-25/presets/presetId` リソースに GET リクエストを送信します。

## リクエスト

### 構文

```
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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
Content-Length: number of characters in the JSON string
```

## リクエストパラメータ

このオペレーションでは、次のリクエストパラメーターを使用します。

### `presetId`

詳細情報を取得するプリセットの識別子。

## リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

## リクエスト本文

この操作にリクエストボディはありません。

## レスポンス

### 構文

```
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 endpoint/
elastictranscoder/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": "flac|flv|fmp4|gif|mp2|mp3|mp4|mpg|mpeg|mxv|oga|ogg|ts|wav|webm",
    "Audio":{
      "Codec": "AAC|flac|mp2|mp3|pcm|vorbis",
      "CodecOptions":{
        "Profile": "auto|AAC-LC|HE-AAC|HE-AACv2",
        "BitDepth": "8|16|24|32",
        "Signed": "Signed|Unsigned",
        "BitOrder": "LittleEndian"
      },
      "SampleRate": "auto|22050|32000|44100|48000|96000",
      "BitRate": audio bit rate of output file in kilobits/second,
      "Channels": "auto|0|1|2",
      "AudioPackingMode": "SingleTrack|OneChannelPerTrack|
        OneChannelPerTrackWithMosTo8Tracks"
    },
    "Video":{
      "Codec": "gif|H.264|mpeg2|vp8|vp9",
      "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]%x|[0,Video:MaxWidth]px",
      "VerticalAlign": "Top|Bottom|Center",
      "VerticalOffset": "[0,100]%x|[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"  
  }  
}
```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

プリセットを取得すると、プリセットの作成時に指定した値が Elastic Transcoder によって返されます。詳細については、「[Elastic Transcoder プリセットの作成時に指定する設定](#)」を参照してください。

また、以下の値も Elastic Transcoder によって返されます。

### (自動) Id

プリセットの識別子。この値を使用して、プリセットの設定を取得したり、削除したりできます。

### (自動) Type

そのプリセットが、Elastic Transcoder によって提供されるデフォルトのプリセット (System) であるか、または定義したプリセット (Custom) であるかが示されます。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

## 例

以下のリクエスト例では、ID が 5555555555555-abcde5 であるプリセットを取得しています。

### リクエスト例

```
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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
Content-Length: number of characters in the JSON string
```

## レスポンス例

```
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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature

{
  "Preset":{
    "Id":"55555555555555-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": ""
}
```

## プリセットを削除する

### トピック

- [説明](#)
- [リクエスト](#)
- [レスポンス](#)
- [エラー](#)
- [例](#)

### 説明

プリセットを削除するには、`/2012-09-25/presets/presetId` リソースに DELETE リクエストを送信します。

#### Note

使用されているプリセットは削除できません。

### リクエスト

#### 構文

```
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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
```

### リクエストパラメータ

このオペレーションでは、次のリクエストパラメーターを使用します。

## presetId

詳細情報を取得するプリセットの識別子。

## リクエストヘッダー

このオペレーションは、すべてのオペレーションに共通のリクエストヘッダーのみを使用します。共通のリクエストヘッダーの詳細については、「[HTTP ヘッダーの内容](#)」を参照してください。

## リクエスト本文

この操作にリクエストボディはありません。

## レスポンス

### 構文

```
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"
}
```

## レスポンスヘッダー

この操作はほとんどのレスポンスに共通のレスポンスヘッダーのみを使用します。共通のレスポンスヘッダーの詳細については、「[HTTP レスポンス](#)」を参照してください。

## レスポンス本文

応答の本文には、以下の JSON オブジェクトが含まれます。

### Success (成功)

プリセットが正常に削除された場合、Success の値は true です。

## エラー

Elastic Transcoder の例外とエラーメッセージについては、「[Elastic Transcoder でのエラー処理](#)」を参照してください。

### 例

以下のリクエスト例では、ID が 5555555555555-abcde5 であるプリセットを削除しています。

#### リクエスト例

```
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 endpoint/
elastictranscoder/aws4_request,
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
```

#### レスポンス例

```
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"
}
```

## ドキュメント履歴

以下の表に、Amazon Elastic Transcoder の今回のリリースでドキュメントに加えられた重要な変更を示します。

- API バージョン: 2012-09-25
- 前回のドキュメントの更新: 2016 年 11 月 17 日

変更	説明	日付
<a href="#">サポート終了通知</a>	サポート終了通知: 2025 年 11 月 13 日、AWS は Amazon Elastic Transcoder のサポートを終了します。2025 年 11 月 13 日以降、Elastic Transcoder コンソールまたは Elastic Transcoder リソースにアクセスできなくなります。への移行の詳細については AWS Elemental MediaConvert、この <a href="#">ブログ記事</a> を参照してください。	2024 年 11 月 12 日
<a href="#">新機能</a>	リージョン間の警告: リージョン間の設定を使用してパイプラインの作成、更新または取得を行うと、Elastic Transcoder により「警告されるようになりました。	2019 年 2 月 19 日
<a href="#">新機能</a>	クリップの結合: Elastic Transcoder を使用して、複数のファイルの部分を結合して 1 つの出力ファイルにすることができるようになりました。	2017 年 11 月 17 日

<a href="#">新機能</a>	AWS アジアパシフィック (ムンバイ) リージョン: AWS リージョン ap-south-1 で Elastic Transcoder が利用可能になりました。	2016 年 8 月 30 日
<a href="#">新機能</a>	CloudWatch を使用したモニタリング: アカウントを通じて Elastic Transcoder リソースと API コールのモニタリングを設定できるようになりました。	2016 年 8 月 20 日
<a href="#">新機能</a>	音声サポートの拡張 (WAV): Elastic Transcoder で、音声用の WAV コンテナがサポートされるようになりました。	2016 年 7 月 19 日
<a href="#">新機能</a>	プレイリストサポートの拡張 (MPEG-DASH): Elastic Transcoder で、FMP4 コンテナ用の MPEG-DASH プレイリストがサポートされるようになりました。	2016 年 5 月 24 日
<a href="#">新機能</a>	ビデオサポートの拡張 (VP9): Elastic Transcoder で、WEBM コンテナ用の VP9 ビデオコーデックがサポートされるようになりました。	2016 年 4 月 20 日
<a href="#">新機能</a>	キャプションサポートの拡張 (CEA-708): Elastic Transcoder で MP4 および MPEG-TS コンテナの CEA-708 埋め込みキャプション出力がサポートされるようになりました。	2015 年 10 月 21 日

<a href="#">新機能</a>	形式サポートの拡張 (MXF、FLAC など) : Elastic Transcoder で、MXF、FLAC や OGA などの追加フォーマットと、出力音声の高度な柔軟性がサポートされるようになりました。	2015 年 5 月 20 日
<a href="#">新機能</a>	PlayReady DRM: Elastic Transcoder でファイルの PlayReady DRM がサポートされるようになりました。	2015 年 3 月 31 日
<a href="#">新機能</a>	検出されたパラメータとジョブのタイミング: Elastic Transcoder で、検出された入力ファイルプロパティ、出力ファイル、および実際のジョブ処理時間 (エポックミリ秒) が記録されるようになりました。	2015 年 3 月 24 日
<a href="#">新機能</a>	形式サポートの拡張 (PAL、NTSC など): Elastic Transcoder で、追加の形式、インターレースシグナリング、クロマサブサンプリング、カラー変換がサポートされるようになりました。	2015 年 3 月 17 日
<a href="#">新機能</a>	HLS のコンテンツ保護: ストリーミングメディアのコンテンツ保護を設定できるようになりました。	2015 年 1 月 13 日

<a href="#">新機能</a>	ユーザー定義メタデータ: ファイルのカスタムメタデータを作成できるようになりました。Elastic Transcoder では、このメタデータに変更を加えることなく、メタデータのトランスコード処理が行われます。	2014 年 12 月 10 日
<a href="#">新機能</a>	暗号化ファイル: 出力ファイルを暗号化し、暗号化されたファイルをトランスコードできます。	2014 年 11 月 24 日
<a href="#">新機能</a>	CloudTrail を使用したログ記録: ご使用のアカウントを通じて Elastic Transcoder API コールのログ記録を設定できるようになりました。	2014 年 10 月 27 日
<a href="#">新機能</a>	HTTP Live Streaming v4: HTTP Live Streaming v4 ストリーミングファイルを生成できるようになりました。	2014 年 10 月 13 日
<a href="#">新機能</a>	スムーズストリーミング: Microsoft スムーズストリーミングファイルを生成できるようになりました。	2014 年 10 月 1 日
<a href="#">新機能</a>	クローズドキャプション: トランスコードした出力にクローズドキャプションを追加できるようになりました。	2014 年 6 月 18 日

<a href="#">新機能</a>	オーディオプロファイル: AAC-LC、HE-AAC、HE-A ACv2 など、さまざまなオー ディオ圧縮方式を使用するよ うに Elastic Transcoder を設 定できるようになりました。	2013 年 12 月 18 日
<a href="#">新機能</a>	クリップの生成: Elastic Transcoder を使用してソース メディアからコンテンツの抜 粋または「クリップ」を生成 できるようになりました。こ の機能を使用すると、ソース メディアの一部のみをトラン スコードできます。	2013 年 11 月 8 日
<a href="#">新機能</a>	新しい音声パラメータ	2013 年 10 月 9 日
<a href="#">新機能</a>	新しいジョブフレームレート パラメータ	2013 年 7 月 8 日
<a href="#">新機能</a>	新しいジョブパラメータ。	2013 年 5 月 16 日
<a href="#">初回リリース</a>	これは「Amazon Elastic Transcoder デベロッパーガイ ド」の初回リリースです。	2013 年 1 月 28 日

# AWS の用語集

---

## 空白

---

プレースホルダー

このページは、AWS 全般のリファレンスの「AWS 用語集」に移動します。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。