aws

개발자 가이드

Amazon Elastic Transcoder

API 버전 2012-09-25

Amazon Elastic Transcoder: 개발자 가이드

Table of Contents

Amazon Elastic Transcoder란 무엇입니까? 1
Elastic Transcoder 액세스 2
지역 및 엔드포인트 2
Elastic Transcoder 리소스에 대한 리전 선택 3
파이프라인, 작업, 프리셋에 대한 제한 3
시작하기 4
버킷 만들기 4
파이프라인 만들기 5
프리셋 만들기 6
작업 만들기 7
진행률 모니터링 8
파이프라인 작업 9
파이프라인 생성 9
파이프라인 일시 중지 및 재활성화 10
파이프라인 알림 업데이트 10
파이프라인 나열 및 보기 11
파이프라인 삭제 12
파이프라인 설정 13
리전 13
일반 설정 13
트랜스코딩한 파일 및 재생 목록의 Amazon S3 버킷 구성 14
썸네일을 위한 Amazon S3 버킷 구성 16
알림 18
작업 사용 21
작업 만들기 21
작업 나열 및 보기 22
작업 취소 23
작업 설정 25
리전 25
일반 설정 26
입력 세부 정보, 1부 26
입력 세부 정보, 2부, 클립 및 캡션 설정 28
출력 세부 정보, 1부 32
출력 세부 정보, 2부 34

출력 세부 정보, 3부, 캡션 설정 36
(선택 사항) 출력 암호화 38
(비디오/썸네일 전용) 워터마크 41
(FLAC/MP3/MP4 전용) 앨범 아트 43
(선택 사항) 사용자 메타데이터 47
(Fragmented MP4/MPEG-TS 출력 전용) 재생 목록 48
(Fragmented MP4/MPEG-TS 출력 전용, 선택 사항) HLS 콘텐츠 보호 50
(HLSv3 및 Smooth 재생 목록 전용, 선택사항) 디지털 권한 관리 53
프리셋 작업 56
프리셋 만들기 56
프리셋 나열 및 보기 57
프리셋 수정 57
프리셋 삭제 58
프리셋 설정 58
일반 설정 59
비디오 설정 61
워터마크 설정 74
오디오 설정 79
썸네일 설정 85
Amazon S3 버킷 구성 89
콘텐츠 보안 유지 91
액세스 제어 91
Elastic Transcoder에 대한 액세스 제어 91
파이프라인 서비스 역할 98
데이터 암호화 옵션 101
암호화 옵션 101
KMS 사용 103
HLS 콘텐츠 보호 105
DRM 108
알림 113
클립 스티칭 116
캡션 117
워터마크 118
디지털 오디오 119
시스템 프리셋 120
크기 조정 정책 및 종횡비 126

종횡비 썸네일 126
Fit 127
Fill 128
Stretch 130
Keep 132
Shrink to Fit 133
Shrink to Fill 135
모니터링 138
모니터링 도구 139
자동 도구 139
수동 도구 139
Amazon CloudWatch를 사용한 모니터링 140
지표 및 차원 140
지표 사용 143
경보 만들기 143
를 사용하여 Elastic Transcoder API 호출 로깅 AWS CloudTrail 146
CloudTrail의 Elastic Transcoder 정보 146
Elastic Transcoder 로그 파일 항목 이해 147
AWS SDK 샘플 코드 152
AWS SDK 샘플 코드의 배경 153
HLS(HTTP Live Streaming) 153
알림 153
환경 설정 154
Java 154
Ruby 158
PHP 162
Python 165
Java 샘플 169
HLS 샘플 169
알림 샘플 170
Ruby 샘플 172
HLS 샘플 172
알림 샘플 173
PHP 샘플 175
HLS 샘플 175
알림 샘플 176

Python 샘플 177
HLS 샘플 178
알림 샘플 178
API 참조 181
AWS SDK 사용 181
HTTP 요청 만들기 181
HTTP 헤더 콘텐츠 182
HTTP 요청 본문 183
HTTP 응답 184
요청에 서명하기 185
오류 처리 186
API 오류 코드(클라이언트 및 서버 오류) 187
작업 처리 중 오류 발생 192
오류 발견 198
오류 재시도 횟수 및 지수 백오프 199
파이프라인 작업 201
파이프라인 생성 201
파이프라인 목록 조회 211
파이프라인 읽기 217
파이프라인 업데이트 224
파이프라인 상태 업데이트 234
파이프라인 알림 업데이트 237
파이프라인 삭제 242
테스트 역할 244
작업 248
작업 생성 248
파이프라인별 작업 나열 264
상태별 작업 나열 278
작업 읽기 291
작업 취소 304
프리셋 작업 307
프리셋 만들기 307
프리셋 나열 316
프리셋 읽기 323
프리셋 삭제 329
문서 기록 332

AWS 용어집 336
..... cccxxxvii

Amazon Elastic Transcoder란 무엇입니까?

주제

- Elastic Transcoder 액세스
- Elastic Transcoder 리소스에 대한 리전 선택
- Elastic Transcoder 파이프라인, 작업, 프리셋 개수 제한

Amazon Elastic Transcoder로 Amazon Simple Storage Service(S3)에 저장한 미디어 파일을 소비자 재생 디바이스에 필요한 형식의 미디어 파일로 변환할 수 있습니다. 예를 들어 대용량 고품질 미디어 파일을 사용자가 모바일 장치, 태블릿, 웹 브라우저 및 연결된 TV에서 재생할 수 있는 형식으로 변환할 수 있습니다.

Elastic Transcoder에는 네 가지 구성 요소가 있습니다.

- 작업은 트랜스코딩을 수행합니다. 각 작업은 하나의 파일을 최대 30가지 형식으로 변환합니다. 예를 들어 미디어 파일을 6가지 다른 형식으로 변환하려는 경우 하나의 작업을 생성하여 6가지 형식의 파일을 생성할 수 있습니다.

  작업을 생성할 때 트랜스코딩하려는 파일의 이름, Elastic Transcoder가 트랜스코딩된 파일에 지정하도록 할 이름 및 몇 가지 기타 설정을 지정해야 합니다. 또한 트랜스코딩하려는 각 형식에 대해 트랜스코딩된 파일에 사용하려는 오디오 설정과 비디오 설정을 포함하는 프리셋(아래 참조)이라고 하는 템플릿을 지정해야 합니다.
- 파이프라인은 트랜스코딩 작업을 관리하는 대기열입니다. 작업을 생성할 때 작업을 추가할 파이프라인을 지정합니다. Elastic Transcoder는 작업을 추가한 순서대로 파이프라인에서 작업을 처리하기 시작합니다. 한 작업을 두 가지 이상의 형식으로 트랜스코딩하도록 구성하는 경우, Elastic Transcoder는 작업에 형식을 지정한 순서대로 각 형식에 대해 파일을 생성합니다.

  일반적인 구성 방식은 파이프라인 2개(표준 우선 순위 작업용 파이프라인 1개와 높은 우선 순위 작업용 파이프라인 1개)를 생성하는 것입니다. 대부분의 작업은 표준 우선 순위 파이프라인에 속하며, 파일을 즉시 트랜스코딩해야 할 때만 높은 우선 순위 파이프라인을 사용하게 됩니다.

  작업을 새로 생성할 때 파이프라인에 작업이 이미 있으면, Elastic Transcoder는 최신 작업을 대기열에 추가하고 해당 파이프라인에 대한 리소스가 사용 가능해지는 즉시 작업을 처리합니다. 파이프라인이 이미 모든 리소스를 사용하고 있으면, Elastic Transcoder는 현재 처리 중인 작업 중 하나가 완료될 때 파이프라인의 다음 작업 처리를 시작합니다.

파이프라인은 작업 여러 개를 동시에 처리할 수 있으며, 작업 완료 시간은 변환하는 파일의 크기와 작업 사양에 따라 크게 다릅니다. 따라서 작업이 생성 순서대로 완료되지 않을 수 있습니다.

파이프라인을 일시 중지하여 작업 처리를 일시적으로 중지할 수 있습니다.

- 프리셋은 미디어 파일을 한 형식에서 다른 형식으로 트랜스코딩하기 위한 대부분의 설정이 들어 있는 템플릿입니다. Elastic Transcoder에는 일반적인 형식을 위한 몇 가지 기본 프리셋이 포함되어 있습니다(예: 여러 iPod 및 iPhone 버전). 기본 프리셋에 포함되지 않은 형식의 고유 프리셋을 생성할 수도 있습니다. 작업을 만들 때 사용할 프리셋을 지정합니다.
- 알림을 사용하면 원할 경우 작업의 상태를 알리도록 Elastic Transcoder와 Amazon Simple Notification Service를 구성할 수 있습니다. 즉, Elastic Transcoder가 작업을 처리하기 시작한 시간, Elastic Transcoder가 작업을 완료한 시간, Elastic Transcoder에서 처리 중 경고나 오류의 발생 여부 등을 알리도록 할 수 있습니다. 알림을 사용하면 작업이 완료된 시간을 확인하기 위해 폴링할 필요가 없습니다. 알림은 파이프라인을 만들 때 구성합니다.

Elastic Transcoder 액세스

Elastic Transcoder는 프로토콜로 HTTPS를 사용하고, 메시지 형식으로 JavaScript Object Notation(JSON)을 사용하는 RESTful 웹 서비스입니다. 애플리케이션 코드는 Elastic Transcoder API로 직접 요청할 수 있습니다. REST API를 직접 사용하는 경우 요청에 서명하고 이를 인증하기 위해 필요한 코드를 작성해야 합니다. API 및 요청 서명에 대한 자세한 내용은 API 참조 섹션을 참조하세요.

Elastic Transcoder는 관리 콘솔도 제공합니다. Elastic Transcoder API를 사용하여 수행할 수 있는 동일한 모든 작업을 콘솔을 사용하여 수행할 수 있습니다. 콘솔을 사용하여 파이프라인, 프리셋 및 작업을 만들고 관리하는 자세한 내용은 다음과 같은 애플리케이션 섹션을 참조하세요.

- 작업 사용
- 파이프라인 작업
- 프리셋 작업

지역 및 엔드포인트

파이프라인은 특정 AWS 리전에 만듭니다. Elastic Transcoder 요청은 항상 리전의 엔드포인트로 전송됩니다. 지원되는 AWS 리전 목록은 Amazon Web Services 일반 참조의 리전 및 엔드포인트 섹션을 참조하세요.

Elastic Transcoder 리소스에 대한 리전 선택

파이프라인과 작업은 특정 리전과 밀접하게 연결됩니다. 파이프라인과 작업을 만들면 현재 리전에 생성됩니다. 작업을 새로 만들 때 현재 리전에 파이프라인을 지정해야 합니다.

Elastic Transcoder 리소스와 다른 리전의 Amazon S3 버킷을 지정할 수 있지만, AWS 리전 간에 파일을 전송하면 추가 요금이 발생하므로 동일한 리전에 만드는 것이 좋습니다.

리전 간 요금에 대한 자세한 내용은 Amazon S3 요금의 데이터 전송 요금을 참조하세요.

Elastic Transcoder 파이프라인, 작업, 프리셋 개수 제한

Elastic Transcoder 파이프라인, 작업, 프리셋에는 다음 한도가 적용됩니다.

- 파이프라인: 각 리전에 대해 AWS 계정당 파이프라인 4개
- 대기열에 추가되는 최대 작업 수: 파이프라인당 1,000,000개
- 최대 출력 수 : 작업당 30개
- 각 파이프라인에서 동시에 처리되는 최대 작업 수: 파이프라인당 100개
- 프리셋: AWS 계정당 사용자 정의 프리셋 50개(Elastic Transcoder에는 한도에 포함되지 않는 사전 정의 프리셋도 있음)
- 작업 요청을 제출할 수 있는 최대 속도:
  - 작업 만들기: AWS 계정당 1초에 2개의 지속적인 속도로 `Create Job` 요청을 제출할 수 있습니다. 일시적 피크 제한은 초당 요청 100개입니다.
  - 작업 읽기: 지속적인 속도로 AWS 계정당 1초에 4개의 `Read Job` 요청을 제출할 수 있습니다. 일시적 피크 제한은 초당 요청 50개입니다.

https://console.aws.amazon.com/support/home#/case/create?issueType=service-limit-increase&limitType=service-code-elastic-transcoders에서 한도 상향 조정을 요청할 수 있습니다.

Elastic Transcoder 시작하기

이 섹션의 예제에서는 Elastic Transcoder를 사용하여 미디어 파일의 형식을 다른 형식으로 트랜스코딩하는 방법을 간략하게 알아봅니다. Elastic Transcoder를 사용하여 미디어 파일을 트랜스코딩하는 작업은 간단한 기본 단계를 통해 시작할 수 있습니다. 첫 단계는 AWS에 가입하는 것입니다. 그리고 Amazon S3 버킷을 만든 후 트랜스코딩하려는 미디어 파일을 업로드합니다. 다음에는, 작업을 처리할 파이프라인을 만들고 특정 파일을 특정 형식으로 트랜스코딩하는 작업을 만듭니다. 프리셋(템플릿)을 제공하지 않는 형식으로 트랜스코딩하려면, 프리셋을 사용자 지정한 후 작업을 만듭니다.

> **Note**
>
> Elastic Transcoder의 기본 개념인 작업, 파이프라인, 프리셋에 익숙하지 않은 경우 Amazon Elastic Transcoder란 무엇입니까? 섹션을 간단히 살펴 보세요.

주제

- 2단계: Amazon S3 버킷을 한두 개 만든 후 미디어 파일 업로드
- 파이프라인 만들기
- (선택 사항) 프리셋 생성
- 작업 만들기
- 작업의 진행 상황 모니터링

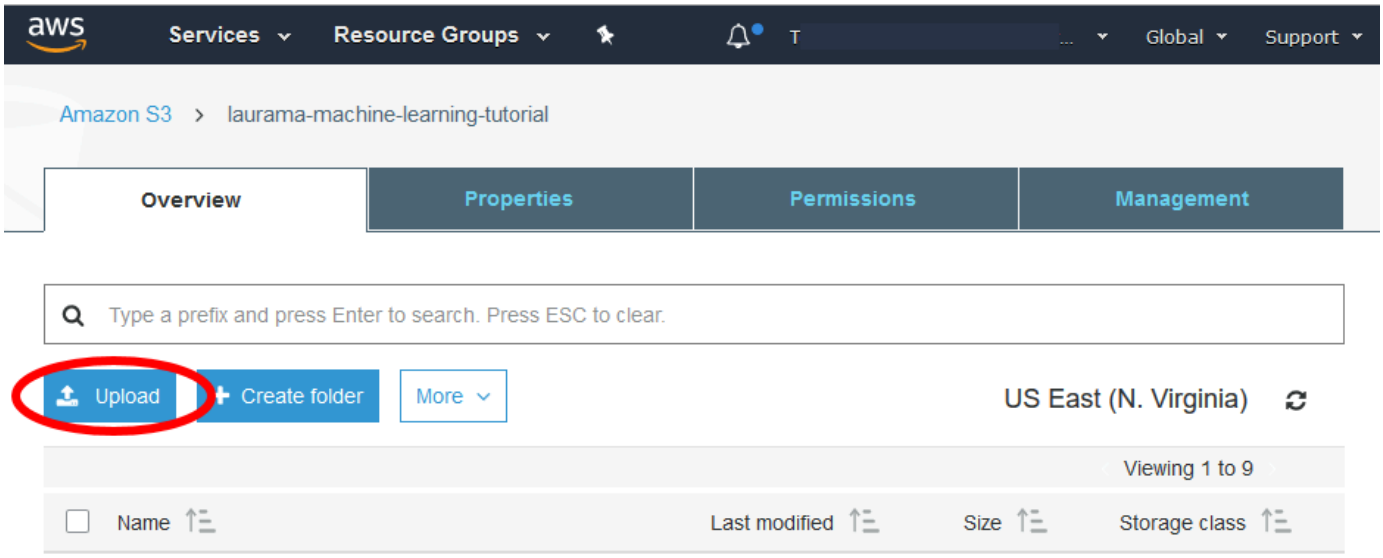
2단계: Amazon S3 버킷을 한두 개 만든 후 미디어 파일 업로드

트랜스코딩하려는 파일에 대한 Amazon S3 버킷(입력 버킷)과 트랜스코딩한 파일에 대한 버킷(출력 버킷)을 따로 만듭니다. 입력 버킷과 출력 버킷에 동일한 버킷을 사용할 수도 있습니다.

Amazon S3 버킷을 만든 후 미디어 파일을 업로드하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/s3/ Amazon S3 콘솔을 엽니다.
2. Amazon S3 콘솔에서 버킷 생성을 클릭합니다.
3. 버킷 생성 대화 상자에 버킷 이름을 입력합니다. 입력 버킷 및 출력 버킷을 별도로 만들려면, 버킷에 적절한 이름을 지정합니다.

4. 버킷에 해당하는 리전을 선택합니다. 기본적으로 Amazon S3는 미국 표준 리전에 버킷을 만듭니다. 지연 시간을 최적화하거나 비용을 최소화하거나 규제 요구 사항을 해소하려면 가까운 리전을 선택하는 것이 좋습니다. Elastic Transcoder가 트랜스코딩을 수행할 리전이기도 합니다.
5. 생성을 클릭합니다.
6. 트랜스코딩하려는 파일과 Elastic Transcoder가 트랜스코딩한 파일에 대한 버킷을 별도로 만들려면, 2단계에서 5단계까지 반복합니다.
7. 버킷 창에서 입력 버킷의 이름을 클릭합니다.
8. 작업을 클릭한 후 업로드를 클릭합니다.
9. 업로드 - 파일 선택 페이지에서 파일 추가를 클릭한 후 트랜스코딩할 미디어 파일을 업로드합니다.



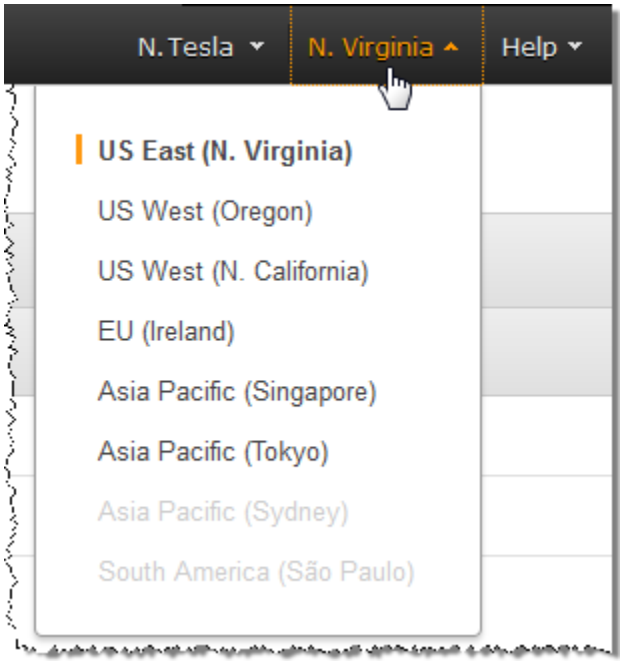
10. Start Upload를 클릭합니다.

파이프라인 만들기

파이프라인은 파일의 트랜스코딩 작업을 관리합니다. 일반적으로는 예를 들어 우선순위가 보통인 작업과 우선순위가 높은 작업을 관리하기 위해 파이프라인을 두 개 이상 만듭니다. 이 예에서는 파이프라인을 한 개만 만듭니다. 파이프라인에 대한 자세한 내용은 파이프라인 작업 섹션을 참조하세요.

Elastic Transcoder 콘솔을 사용하여 파이프라인을 만들려면

1. https://console.aws.amazon.com/elastictranscoder/에서 Elastic Transcoder 콘솔을 엽니다.
2. Elastic Transcoder 콘솔의 탐색 모음에서 파이프라인을 만들 리전을 선택합니다.



3. 콘솔의 왼쪽 창에서 파이프라인을 클릭합니다.
4. 파이프라인 페이지에서 새 파이프라인 생성을 클릭합니다.
5. 관련 값들을 입력합니다. 각 필드에 대한 자세한 내용은 Elastic Transcoder 파이프라인을 생성할 때 지정하는 설정 섹션을 참조하세요.

   > **Note**
   >
   > 작업의 진행 상황을 모니터링하려면 Elastic Transcoder가 작업 처리를 시작한 시간과 Elastic Transcoder가 작업을 완료한 시간을 알리도록 Amazon Simple Notification Service(SNS) 주제를 지정합니다.

6. 파이프라인 생성을 클릭합니다.

(선택 사항) 프리셋 생성

프리셋은 트랜스코딩 프로세스 중에 Elastic Transcoder가 적용할 설정(예: 트랜스코딩된 파일에서 원하는 해상도 및 코덱)을 포함하는 템플릿입니다. 작업을 만들 때 사용할 프리셋을 지정합니다. 여러 일반 장치에서 재생되는 미디어 파일을 생성하는 시스템 프리셋과 특정 디바이스를 대상으로 하는 시스템 프리셋을 제공합니다. 호환성을 극대화하려면 다양한 디바이스에서 재생되도록 출력하는 프리셋을 선택합니다. 품질과 파일 크기를 극대화하려면 특정 디바이스나 디바이스 클래스에 대해 출력하도록 프리셋을 선택합니다. 현재 시스템 프리셋의 목록은 Elastic Transcoder 세부 정보 페이지의 프리셋 목록을 참조하세요.

시스템 프리셋에 필요한 설정이 없다면, 원하는 대로 사용자 지정하여 프리셋을 만들 수 있습니다. 자세한 내용은 Elastic Transcoder에서 프리셋 생성 섹션을 참조하세요.

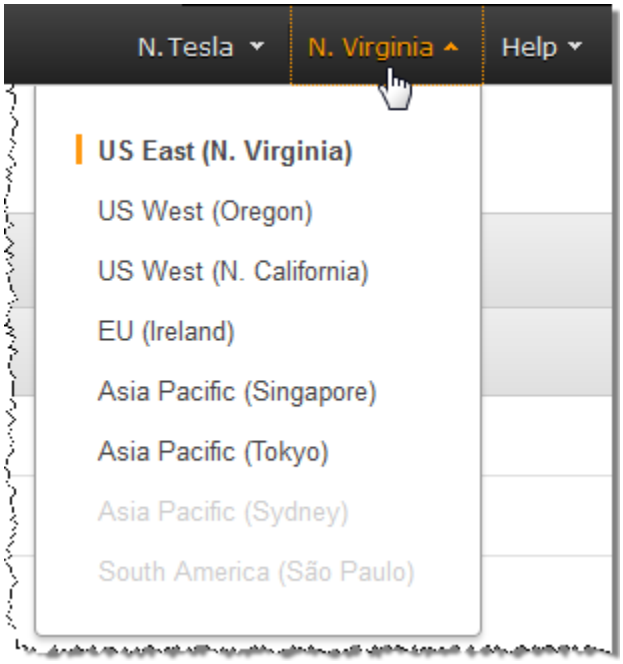
작업 만들기

작업은 트랜스코딩 작업을 수행합니다. 트랜스코딩하려는 파일의 이름(입력 파일), Elastic Transcoder가 트랜스코딩된 파일에 부여할 이름, Elastic Transcoder에서 사용할 프리셋 및 기타 몇 가지 설정을 지정합니다. Elastic Transcoder는 파이프라인에서 지정한 Amazon S3 입력 버킷에서 입력 파일을 가져와 파일을 트랜스코딩한 다음, 파이프라인에서 지정한 Amazon S3 출력 버킷에 트랜스코딩된 파일을 저장합니다.

작업에 대한 자세한 내용은 작업 사용 섹션을 참조하세요.

Elastic Transcoder 콘솔을 사용하여 작업을 생성하려면

1. https://console.aws.amazon.com/elastictranscoder/에서 Elastic Transcoder 콘솔을 엽니다.
2. Elastic Transcoder 콘솔의 탐색 모음에서 작업을 만들 리전을 선택합니다.



3. 콘솔의 왼쪽 창에서 파이프라인을 클릭합니다. (파일을 트랜스코딩하는 데 사용할 파이프라인(대기열)에 작업을 만듭니다.)
4. Pipelines 페이지에서 Create New Job을 클릭합니다.
5. 관련 값들을 입력합니다. 각 필드에 대한 자세한 내용은 Elastic Transcoder 작업을 만들 때 지정하는 설정 섹션을 참조하세요.
6. 작업 생성을 클릭합니다.

작업의 진행 상황 모니터링

파이프라인 만들기 섹션에서 파이프라인을 만들었다면, Elastic Transcoder가 작업 처리를 시작하고 완료할 때 Elastic Transcoder가 Amazon Simple Notification Service(SNS) 주제로 메시지를 보내도록 알림을 구성할 수 있습니다. 알림을 구성하고 해당하는 Amazon SNS 주제를 구독한 경우, 작업의 진행 상황을 모니터링할 수 있습니다.

파이프라인 작업

주제

- Elastic Transcoder에서 파이프라인 생성
- Elastic Transcoder에서 파이프라인 일시 중지 및 재활성화
- Elastic Transcoder에서 파이프라인 알림 업데이트
- Elastic Transcoder에서 파이프라인 목록 작성 및 보기
- Elastic Transcoder 파이프라인 삭제
- Elastic Transcoder 파이프라인을 생성할 때 지정하는 설정

파이프라인은 트랜스코딩 작업을 관리하는 대기열입니다. 작업을 생성할 때 작업을 추가할 파이프라인을 지정합니다. Elastic Transcoder는 작업을 추가한 순서대로 파이프라인에서 작업을 처리하기 시작합니다.

일반적인 구성 방식은 파이프라인 2개(표준 우선 순위 작업용 파이프라인 1개와 높은 우선 순위 작업용 파이프라인 1개)를 생성하는 것입니다. 대부분의 작업은 표준 우선 순위 파이프라인에 속하며, 파일을 즉시 트랜스코딩해야 할 때만 높은 우선 순위 파이프라인을 사용하게 됩니다.

작업을 생성하는 시점에서 파이프라인에 다른 작업이 있으면 Elastic Transcoder는 리소스를 사용할 수 있을 때 새 작업을 처리합니다. 파이프라인은 작업 여러 개를 동시에 처리할 수 있으며, 작업 완료 시간은 변환하는 파일의 크기와 작업 사양에 따라 크게 다릅니다. 따라서 작업이 생성 순서대로 완료되지 않을 수 있습니다.

파이프라인은 일시 중지해 작업 처리를 잠시 멈출 수 있습니다. 작업을 하나 이상 취소하고자 할 때 유용하며, 일시 중지를 해제하면 Elastic Transcoder가 작업 처리를 시작합니다.

Elastic Transcoder에서 파이프라인 생성

AWS Management Console 또는 Elastic Transcoder 파이프라인 생성 API 작업을 사용하여 파이프라인을 생성할 수 있습니다. 다음 절차에서는 콘솔을 사용하여 파이프라인을 생성하는 방법을 설명합니다. API를 사용해 파이프라인을 생성하는 방법에 대한 자세한 내용은 파이프라인 생성 섹션을 참조하세요.

Elastic Transcoder 콘솔을 사용하여 파이프라인을 만들려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/elastictranscoder/ Elastic Transcoder 콘솔을 엽니다.
2. Elastic Transcoder 콘솔의 탐색 모음에서 파이프라인을 만들 리전을 선택합니다. 이 리전에서 작업을 생성하게 됩니다.
3. 콘솔의 왼쪽 탐색 창에서 Pipelines를 클릭합니다.
4. Pipelines 페이지에서 Create New Pipeline을 클릭합니다.
5. 관련 값들을 입력합니다. 각 필드에 대한 자세한 내용은 Elastic Transcoder 파이프라인을 생성할 때 지정하는 설정 섹션을 참조하세요.
6. Create Pipeline을 클릭합니다.

Elastic Transcoder에서 파이프라인 일시 중지 및 재활성화

작업을 취소할 때는 Elastic Transcoder가 작업 처리를 시작하지 않도록 먼저 해당 파이프라인을 일시 중지하는 것이 좋습니다. 작업 상태가 제출됨에서 처리 중으로 변경되면 작업을 취소할 수 없습니다.

다음 절차에서는 콘솔을 사용하여 파이프라인을 일시 중지하고 재활성화하는 방법을 설명합니다. API를 사용해 파이프라인을 일시 중지하고 재활성화하는 방법에 대한 자세한 내용은 파이프라인 상태 업데이트 섹션을 참조하세요.

Elastic Transcoder 콘솔을 사용해 파이프라인을 일시 중지하거나 재활성화하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/elastictranscoder/ Elastic Transcoder 콘솔을 엽니다.
2. Elastic Transcoder 콘솔의 탐색 모음에서 파이프라인을 일시 중지하거나 재활성화할 리전을 선택합니다.
3. 왼쪽 탐색 창에서 Pipelines를 클릭합니다.
4. 일시 중지하거나 재활성화할 파이프라인 옆의 확인란을 선택합니다.
5. 원하는 작업에 따라 Pause 또는 Activate를 클릭합니다.

Elastic Transcoder에서 파이프라인 알림 업데이트

파이프라인을 생성할 때는 작업의 상태가 변경되는 경우 Amazon Simple Notification Service(SNS) 주제에 메시지를 보내도록 Elastic Transcoder를 구성할 수도 있습니다. 이러한 경우로는 Elastic

Transcoder가 작업 처리를 시작하거나 완료할 때, 작업 처리 중 Elastic Transcoder에 경고 또는 오류 상태가 발생할 때 등이 있습니다. 필요에 따라 Elastic Transcoder에서 메시지를 보내거나 보내지 않도록 변경할 수 있으며, 메시지를 보내도록 하는 경우 메시지를 보낼 SNS 주제 역시 변경할 수 있습니다.

Amazon SNS는 Amazon Simple Queue Service(Amazon SQS) 대기열에 Amazon SNS 메시지를 전송하는 기능을 포함하여 다양한 알림 옵션을 제공합니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서를 참조하세요.

다음 절차에서는 콘솔을 사용하여 알림을 업데이트하는 방법을 설명합니다. API를 사용해 알림을 업데이트하는 방법에 대한 자세한 내용은 파이프라인 알림 업데이트 섹션을 참조하세요.

Elastic Transcoder 콘솔을 사용해 파이프라인 알림을 업데이트하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/elastictranscoder/ Elastic Transcoder 콘솔을 엽니다.
2. Elastic Transcoder 콘솔의 탐색 모음에서 파이프라인을 일시 중지하거나 재활성화할 리전을 선택합니다.
3. 왼쪽 탐색 창에서 Pipelines를 클릭합니다.
4. 알림을 변경할 파이프라인 옆의 확인란을 선택합니다.
5. 편집을 클릭합니다.
6. 적용되는 값들을 변경합니다. 자세한 내용은 Elastic Transcoder 파이프라인을 생성할 때 지정하는 설정 단원을 참조하십시오.
7. Save를 클릭해 변경 사항을 저장합니다.

Elastic Transcoder에서 파이프라인 목록 작성 및 보기

현재 AWS 계정과 연결된 파이프라인을 나열하고, 특정한 파이프라인의 설정을 확인할 수 있습니다. 다음 절차에서는 콘솔을 사용해 파이프라인을 나열하는 방법과 파이프라인의 설정을 확인하는 방법을 설명합니다.

API를 사용하는 경우:

- 현재 AWS 계정의 파이프라인을 나열하는 방법은 파이프라인 목록 조회 섹션을 참조하세요.
- 특정 파이프라인의 설정을 확인하는 방법은 파이프라인 읽기 섹션을 확인하세요.

Elastic Transcoder 콘솔을 사용해 파이프라인을 나열하고 파이프라인 설정을 확인하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/elastictranscoder/ Elastic Transcoder 콘솔을 엽니다.
2. Elastic Transcoder 콘솔의 탐색 모음에서 작업을 나열할 리전을 선택합니다.
3. 콘솔의 왼쪽 탐색 창에서 Pipelines를 클릭합니다. 오른쪽 창에 현재 계정과 연결된 파이프라인이 나열됩니다.
4. 파이프라인에 대한 자세한 정보를 표시하려면 파이프라인 옆

   아이콘을 클릭합니다.

Elastic Transcoder 파이프라인 삭제

를 사용하거나 Elastic Transcoder Delete Pipeline API AWS Management Console 를 사용하여 파이프라인을 삭제할 수 있습니다. 다음 절차에서는 콘솔을 사용하여 파이프라인을 삭제하는 방법을 설명합니다. API를 사용해 파이프라인을 삭제하는 방법에 대한 자세한 내용은 파이프라인 삭제 섹션을 참조하세요.

> **Note**
>
> 처리하지 않은 작업이 포함되어 있는 파이프라인은 삭제할 수 없습니다.

Elastic Transcoder 콘솔을 사용하여 파이프라인을 삭제하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/elastictranscoder/ Elastic Transcoder 콘솔을 엽니다.
2. Elastic Transcoder 콘솔의 탐색 모음에서 삭제할 파이프라인이 포함되어 있는 리전을 선택합니다.
3. 콘솔의 왼쪽 탐색 창에서 Pipelines를 클릭합니다.
4. 삭제할 파이프라인의 확인란을 선택합니다.
5. Remove를 클릭합니다.

Elastic Transcoder 파이프라인을 생성할 때 지정하는 설정

Elastic Transcoder 콘솔을 사용해 파이프라인을 생성할 때 다음 값을 지정할 수 있습니다.

주제

- 리전
- 일반 설정
- 트랜스코딩한 파일 및 재생 목록의 Amazon S3 버킷 구성
- 썸네일을 위한 Amazon S3 버킷 구성
- 알림

리전

Elastic Transcoder는 사용자가 속한 리전에 파이프라인을 생성합니다. 작업을 실행하고자 하는 리전을 선택하세요.

일반 설정

파이프라인 이름

파이프라인의 이름. AWS 계정 내에서 고유한 이름을 사용하는 것이 좋지만 고유성이 적용되지는 않습니다. 파이프라인 이름의 최대 길이는 40자입니다.

입력 버킷

트랜스코딩할 미디어 파일과 워터마크로 사용할 그래픽 파일(있는 경우)을 저장한 Amazon S3 버킷입니다.

IAM 역할

Elastic Transcoder가 이 파이프라인의 작업을 트랜스코딩하는 데 사용할 IAM Amazon 리소스 이름(ARN)입니다.

AWS KMS 키 ARN

이 파이프라인에서 사용할 AWS Key Management Service(AWS KMS) 키.

s3 또는 s3-aws-kms를 Encryption:Mode로 사용하는 경우 AWS-KMS 키라고 하는 기본 키가 자동으로 생성되므로 작업에 키를 제공할 필요가 없습니다. 기본 AWS-KMS 키가 아닌 키를 사용하는 경우 또는 Encryption:Mode로 aes-pkcs7, aes-ctr 또는 aes-gcm을 사용하는 경우에만 AWS-KMS 키를 제공해야 합니다.

트랜스코딩한 파일 및 재생 목록의 Amazon S3 버킷 구성

이 섹션의 설정을 사용하면 Elastic Transcoder가 이 파이프라인에 제출된 작업의 트랜스코딩된 파일 및 재생 목록을 저장할 Amazon S3 버킷, 파일에 할당할 Amazon S3 스토리지 클래스, 파일에 액세스할 수 있는 사용자, 사용자에게 부여할 액세스 유형을 지정할 수 있습니다.

버킷

Elastic Transcoder가 이 파이프라인에 제출된 작업의 트랜스코딩된 파일 및 재생 목록(해당되는 경우)을 저장할 Amazon S3 버킷입니다.

스토리지 클래스

Elastic Transcoder가 Amazon S3 버킷에 저장하는 트랜스코딩된 파일과 재생 목록에 할당할 Amazon S3 스토리지 클래스(표준 또는 중복성 감소)입니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 Reduced Redundancy Storage를 참조하세요.

피부여자 유형

트랜스코딩된 파일 및 재생 목록의 액세스 권한을 부여할 대상 사용자 또는 그룹을 식별하는 방법을 지정할 수 있습니다. 피부여자 유형을 선택할 때 다음과 같은 Grantee Type 필드가 표시됩니다.

- Canonical: 피부여자의 값이 AWS 계정의 고유 사용자 ID 또는 Amazon CloudFront 배포의 원본 액세스 ID입니다. 표준 사용자 ID에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 액세스 제어 목록(ACL) 개요를 참조하세요. CloudFront 원본 액세스 자격 증명을 사용하여 사용자에게 Amazon S3 URL 대신 CloudFront URL을 사용하도록 요구하는 방법에 대한 자세한 내용은 Amazon CloudFront 개발자 안내서의 원본 액세스 ID를 사용하여 Amazon S3 콘텐츠에 대한 액세스 제한을 참조하세요.

  > **Important**
  > 고유 사용자 ID는 AWS 계정 번호와 다릅니다.

- Email: 피부여자의 값은 AWS 계정의 등록된 이메일 주소입니다.
- Group: 피부여자의 값이 사전 정의된 Amazon S3 그룹(AllUsers, AuthenticatedUsers 또는 LogDelivery) 중 하나입니다.

피부여자

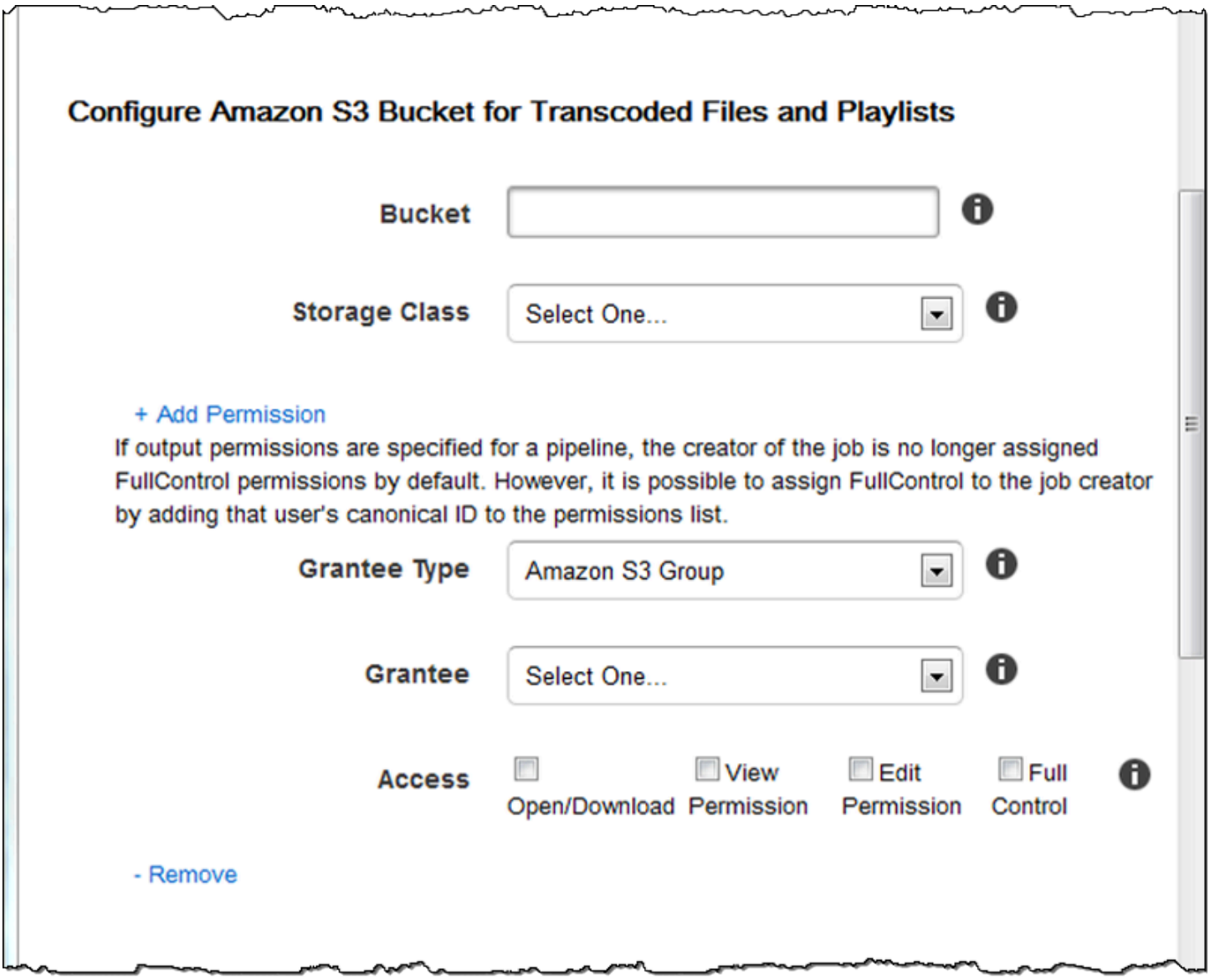
트랜스코딩된 파일 및 재생 목록에 대한 액세스 권리를 부여하려는 AWS 사용자 또는 그룹. 사용자 또는 그룹을 식별하기 위해 AWS 계정의 고유 사용자 ID, CloudFront 배포의 원본 액세스 ID, AWS

계정의 등록된 이메일 주소 또는 사전 정의된 Amazon S3 그룹을 지정할 수 있습니다. 자세한 내용은 Grantee Type 단원을 참조하십시오.

액세스

Grantee에 지정된 AWS 사용자에게 부여하려는 권한. 권한은 Elastic Transcoder가 버킷에 추가하는 트랜스코딩된 파일 및 재생 목록에 대해 부여됩니다. 유효한 값으로는 다음이 포함됩니다.

- Open/Download: 피부여자는 Elastic Transcoder가 Amazon S3 버킷에 추가하는 트랜스코딩된 파일 및 재생 목록의 객체와 메타데이터를 읽을 수 있습니다.
- View Permissions: 피부여자는 Elastic Transcoder가 Amazon S3 버킷에 추가하는 트랜스코딩된 파일 및 재생 목록의 객체 ACL을 읽을 수 있습니다.
- Edit Permissions: 피부여자는 Elastic Transcoder가 Amazon S3 버킷에 추가하는 트랜스코딩된 파일 및 재생 목록의 ACL을 쓸 수 있습니다.
- Full Control: 피부여자는 객체를 읽고, Elastic Transcoder가 Amazon S3 버킷에 추가하는 트랜스코딩된 파일 및 재생 목록의 ACL을 확인 및 편집할 권한이 있습니다.



썸네일을 위한 Amazon S3 버킷 구성

이 섹션의 설정을 사용하면 Elastic Transcoder가 이 파이프라인에 제출된 작업의 썸네일을 저장할 Amazon S3 버킷, 썸네일에 할당할 Amazon S3 스토리지 클래스, 썸네일에 액세스할 수 있는 사용자, 사용자에게 부여할 액세스 유형을 지정할 수 있습니다.

버킷

Elastic Transcoder가 이 파이프라인에 제출된 작업의 썸네일을 저장할 Amazon S3 버킷입니다.

스토리지 클래스

Elastic Transcoder가 Amazon S3 버킷에 저장하는 썸네일에 할당할 Amazon S3 스토리지 클래스(표준 또는 중복성 감소)입니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 Reduced Redundancy Storage를 참조하세요.

피부여자 유형

썸네일의 액세스 권한을 부여할 대상 사용자 또는 그룹을 식별하는 방법을 지정할 수 있습니다. 피부여자 유형을 선택할 때 다음과 같은 Grantee Type 필드가 표시됩니다.

- Canonical: 피부여자의 값이 AWS 계정의 고유 사용자 ID 또는 Amazon CloudFront 배포의 원본 액세스 ID입니다. 표준 사용자 ID에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 액세스 제어 목록(ACL) 개요를 참조하세요. CloudFront 원본 액세스 자격 증명을 사용하여 사용자에게 Amazon S3 URL 대신 CloudFront URL을 사용하도록 요구하는 방법에 대한 자세한 내용은 Amazon CloudFront 개발자 안내서의 원본 액세스 ID를 사용하여 Amazon S3 콘텐츠에 대한 액세스 제한을 참조하세요.

  > **Important**
  >
  > 고유 사용자 ID는 AWS 계정 번호와 다릅니다.

- Email: 피부여자의 값은 AWS 계정의 등록된 이메일 주소입니다.
- Group: 피부여자의 값이 사전 정의된 Amazon S3 그룹(AllUsers, AuthenticatedUsers 또는 LogDelivery) 중 하나입니다.

피부여자

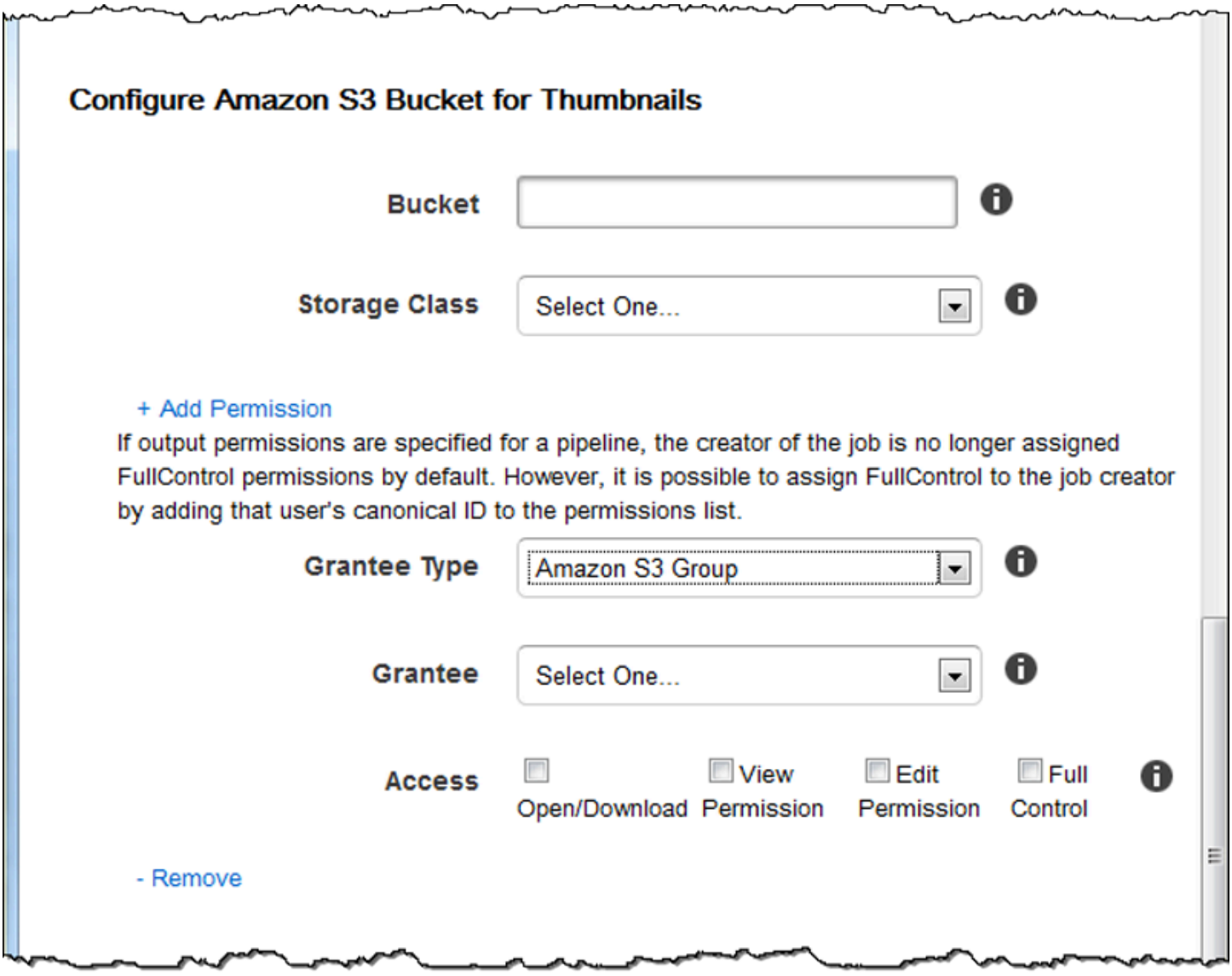
썸네일에 대한 액세스 권리를 부여하려는 AWS 사용자 또는 그룹. 사용자 또는 그룹을 식별하기 위해 AWS 계정의 고유 사용자 ID, CloudFront 배포의 원본 액세스 ID, AWS 계정의 등록된 이메일 주소 또는 사전 정의된 Amazon S3 그룹을 지정할 수 있습니다. 자세한 내용은 Grantee Type 단원을 참조하십시오.

액세스

Grantee에 지정된 AWS 사용자에게 부여하려는 권한. 권한은 Elastic Transcoder가 버킷에 추가하는 썸네일에 대해 부여됩니다. 유효한 값으로는 다음이 포함됩니다.

- Open/Download: 피부여자는 Elastic Transcoder가 Amazon S3 버킷에 추가하는 썸네일의 객체와 메타데이터를 읽을 수 있습니다.
- View Permissions: 피부여자는 Elastic Transcoder가 Amazon S3 버킷에 추가하는 썸네일의 객체 ACL을 읽을 수 있습니다.

- Edit Permissions: 피부여자는 Elastic Transcoder가 Amazon S3 버킷에 추가하는 썸네일의 ACL을 쓸 수 있습니다.
- Full Control: 피부여자는 객체를 읽고, Elastic Transcoder가 Amazon S3 버킷에 추가하는 썸네일의 ACL을 확인 및 편집할 권한이 있습니다.



알림

이 섹션의 설정을 사용하면 작업의 상태가 변경될 때 알림을 전송하도록 Elastic Transcoder를 구성할 수 있습니다.

On Progressing Event

Elastic Transcoder가 작업을 처리하기 시작했을 때 알리려는 Amazon Simple Notification Service(SNS) 주제입니다.

> **Important**
>
> 알림을 수신하려면 Amazon SNS 콘솔에서 새 주제를 구독해야 합니다.

Amazon SNS는 Amazon Simple Queue Service(Amazon SQS) 대기열에 Amazon SNS 메시지를 전송하는 기능을 포함하여 다양한 알림 옵션을 제공합니다. 자세한 내용은 Amazon Simple Notification Service 개발자 안내서를 참조하세요.

On Complete Event

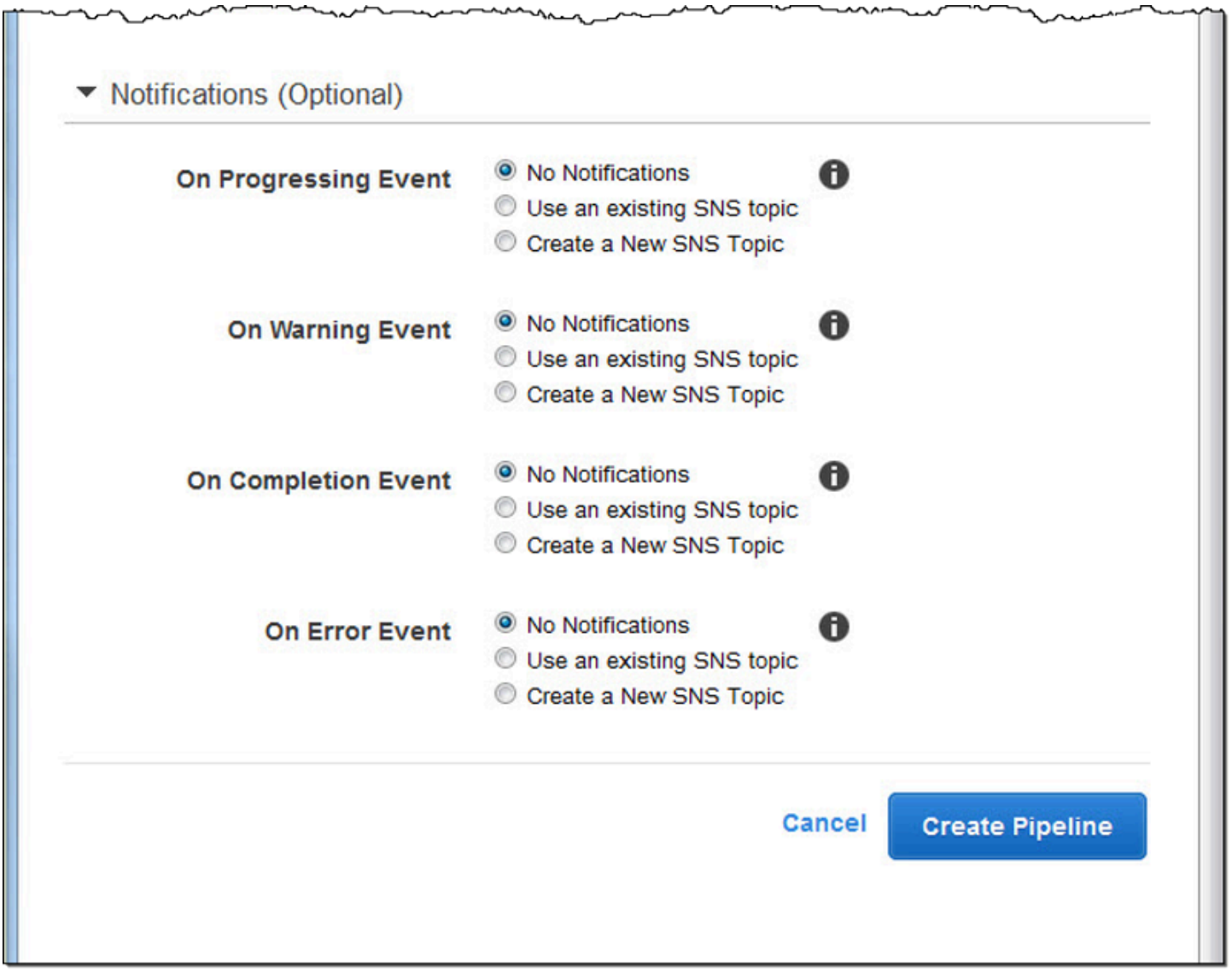
Elastic Transcoder가 작업 처리를 완료하면 알림을 전송하도록 할 Amazon SNS 주제입니다.

On Warning Event

Elastic Transcoder에 경고 상태가 발생하면 알림을 전송하도록 할 Amazon SNS 주제입니다.

On Error Event

Elastic Transcoder에 오류 상태가 발생하면 알림을 전송하도록 할 Amazon SNS 주제입니다.


Notifications (Optional)
On Progressing Event
No Notifications
Use an existing SNS topic
Create a New SNS Topic
On Warning Event
No Notifications
Use an existing SNS topic
Create a New SNS Topic
On Completion Event
No Notifications
Use an existing SNS topic
Create a New SNS Topic
On Error Event
No Notifications
Use an existing SNS topic
Create a New SNS Topic
Cancel
Create Pipeline

작업 사용

주제

- Elastic Transcoder에서 작업 생성
- Elastic Transcoder에서 작업 나열 및 작업 설정 보기
- Elastic Transcoder 작업 취소
- Elastic Transcoder 작업을 만들 때 지정하는 설정

작업은 미디어 파일을 한 형식에서 다른 형식으로 트랜스코딩하는 작업을 수행합니다. 작업을 만들 때 트랜스코딩을 수행하기 위해 Elastic Transcoder에 필요한 정보, 즉 트랜스코딩할 파일, 트랜스코딩된 파일의 이름, 사용할 프리셋(프리셋은 Elastic Transcoder가 적용할 설정이 포함된 템플릿) 등을 지정합니다.

Elastic Transcoder에서 작업 생성

AWS Management Console 또는 Elastic Transcoder 작업 생성 API 작업을 사용하여 작업을 생성할 수 있습니다. 다음 절차에서는 콘솔을 사용하여 작업을 만드는 방법을 설명합니다. API를 사용해 작업을 만드는 방법에 대한 자세한 내용은 작업 생성를 참조하세요.

작업의 상태가 변경되거나(Elastic Transcoder가 작업 처리를 시작 및 종료하는 경우 등) 경고 또는 오류 조건이 발생하면 이를 알리도록 Elastic Transcoder를 구성할 수 있습니다. 자세한 내용은 Elastic Transcoder에서 파이프라인 생성 단원을 참조하십시오.

작업을 만든 후에는 업데이트할 수 없습니다. 작업의 설정을 변경해야 하는 경우 이를 취소하고, 취소한 작업을 기반으로 새 작업을 만들고, 해당되는 값을 업데이트한 후, 새 작업을 만듭니다.

> **Note**
>
> 작업을 만들기 전에 작업을 관리할 파이프라인(대기열)을 생성해야 합니다. 파이프라인 생성에 대한 자세한 내용은 Elastic Transcoder에서 파이프라인 생성 섹션을 참조하세요. 또한 Elastic Transcoder 기본 프리셋에 제공한 설정 이외의 설정을 사용하여 파일을 트랜스코딩하려면 새 프리셋을 생성해야 합니다. 프리셋 만들기에 대한 자세한 내용은 Elastic Transcoder에서 프리셋 생성 섹션을 참조하세요.

Elastic Transcoder 콘솔을 사용하여 작업을 생성하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/elastictranscoder/ Elastic Transcoder 콘솔을 엽니다.
2. Elastic Transcoder 콘솔의 탐색 모음에서 트랜스코딩 작업에 사용할 파이프라인에 해당하는 리전을 선택합니다. 파이프라인은 생성된 리전에서만 사용할 수 있습니다.
3. 콘솔의 왼쪽 탐색 창에서 Pipelines를 클릭합니다. (파일을 트랜스코딩하는 데 사용할 파이프라인(대기열)에 작업을 만듭니다.)
4. Pipelines 페이지에서 Create New Job을 클릭합니다.
5. 관련 값들을 입력합니다. 각 필드에 대한 자세한 내용은 Elastic Transcoder 작업을 만들 때 지정하는 설정 섹션을 참조하세요.
6. Create Job을 클릭합니다. 작업은 만드는 즉시 시작됩니다.

> **Note**
>
> 작업이 `Access Denied` 오류로 인해 실패한 경우, `Test Role` API 작업을 실행하여 오류의 원인을 확인하는 것이 좋습니다. 자세한 내용은 테스트 역할 단원을 참조하십시오.

Elastic Transcoder에서 작업 나열 및 작업 설정 보기

Elastic Transcoder 콘솔을 사용하거나 해당되는 API 작업을 사용하여 지정된 파이프라인에 있거나 지정된 상태인 작업을 나열할 수 있습니다. 개별 작업의 설정을 볼 수도 있습니다. 다음 절차에서는 콘솔을 사용해 작업을 나열하는 방법과 작업의 설정을 확인하는 방법을 설명합니다.

> **Note**
>
> 파이프라인별로 작업 목록을 표시하면, Elastic Transcoder는 사용자가 해당 파이프라인에 대해 지난 6개월 동안 만든 모든 작업을 나열합니다. 상태별로 작업 목록을 표시하면, Elastic Transcoder는 사용자가 지난 6개월 동안 만든 현재 지정된 상태의 모든 작업을 나열합니다.

API를 사용하는 경우:

- 지정된 파이프라인의 작업 나열은 파이프라인별 작업 나열을 참조하세요.
- 지정된 상태의 작업 나열은 상태별 작업 나열을 참조하세요.

- 지정된 작업의 설정 확인은 작업 읽기를 참조하세요.

> **Note**
>
> 작업에 두 개 이상의 출력을 지정한 경우(예: Kindle Fire에 출력 한 개, Apple iPhone 4s에 출력 한 개), 현재 Elastic Transcoder API를 사용하여 작업을 나열해야 합니다.

Elastic Transcoder 콘솔을 사용하여 작업을 나열하고 작업 설정을 보려면

1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/elastictranscoder/에서 Elastic Transcoder 콘솔을 엽니다.
2. Elastic Transcoder 콘솔의 탐색 모음에서 작업을 나열할 리전을 선택합니다.
3. 콘솔의 왼쪽 탐색 창에서 Jobs를 클릭합니다.
4. Jobs 페이지에서 해당 값을 지정합니다. 필드에 대한 자세한 내용은 해당 필드 옆의

   아이콘을 클릭합니다.
5. Search를 클릭합니다.
6. 검색 결과에 나열된 작업에 대한 세부 정보를 표시하려면 해당 작업 옆의

   아이콘을 클릭합니다.

Elastic Transcoder 작업 취소

상태가 Submitted인 작업을 취소할 수 있습니다. 이 상태는 Elastic Transcoder가 파일 트랜스코딩을 시작하지 않았음을 뜻합니다. 다음 절차에서는 Elastic Transcoder 콘솔을 사용하여 작업을 취소하는 방법을 설명합니다.

API를 사용하여 작업을 취소하려면 Elastic Transcoder가 작업 처리를 시작하지 않도록 해당 파이프라인을 일시 중지하고, 상태가 Submitted인 작업을 나열하여 해당 작업 ID를 가져온 후, 작업 ID로 취소할 작업을 식별하여 작업을 취소합니다. 자세한 내용은 다음을 참조하세요.

- 파이프라인 상태 업데이트
- 상태별 작업 나열
- 작업 취소

Elastic Transcoder 콘솔을 사용하여 작업을 취소하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/elastictranscoder/ Elastic Transcoder 콘솔을 엽니다.
2. Elastic Transcoder 콘솔의 탐색 모음에서 작업을 취소할 리전을 선택합니다.
3. 권장 선택 사항: Elastic Transcoder가 작업 처리를 시작하지 않도록 작업을 제출한 파이프라인을 일시 중지합니다. Elastic Transcoder가 처리하기 시작한 작업은 취소할 수 없습니다.

   a. 왼쪽 탐색 창에서 Pipelines를 클릭합니다.

   b. 일시 중지할 파이프라인 옆의 확인란을 선택합니다.

   c. Pause를 클릭합니다.
4. 콘솔의 탐색 창에서 Jobs를 클릭합니다.
5. Jobs 페이지에서 다음 값을 지정합니다.

   검색 기준

   Status를 클릭합니다.

   작업 상태

   Submitted를 선택합니다.

   > **Note**
   > 상태가 Submitted인 작업만 취소할 수 있습니다.

   Order 및 Number of Jobs에 해당 값을 입력합니다.
6. Search를 클릭합니다.
7. 검색 결과에서 취소하려는 작업인지 여부를 확인하기 위해 작업에 대한 세부 정보를 봐야 하는 경우, 해당 작업 옆의

   아이콘을 클릭합니다.
8. 작업을 취소하려면 해당 작업 옆의 확인란을 선택한 후 Cancel을 클릭합니다.
9. 3단계에서 파이프라인을 일시 중지한 경우, 작업 처리를 재개하도록 이를 다시 활성화합니다.

   a. 탐색 창에서 Pipelines를 클릭합니다.

b. 다시 활성화할 파이프라인 옆의 확인란을 선택합니다.

c. Activate를 클릭합니다.

Elastic Transcoder 작업을 만들 때 지정하는 설정

Elastic Transcoder 콘솔을 사용해 작업을 만들 때 다음 값을 지정할 수 있습니다.

> **Note**
>
> 작업의 상태가 변경되거나(Elastic Transcoder가 작업 처리를 시작 및 종료하는 경우 등) Elastic Transcoder에 경고 또는 오류 조건이 발생하면 이를 알리도록 Elastic Transcoder를 구성할 수 있습니다. 자세한 내용은 작업 상태 알림 단원을 참조하십시오.

주제

- 리전
- 일반 설정
- 입력 세부 정보, 1부
- 입력 세부 정보, 2부, 클립 및 캡션 설정
- 출력 세부 정보, 1부
- 출력 세부 정보, 2부
- 출력 세부 정보, 3부, 캡션 설정
- (선택 사항) 출력 암호화
- (비디오/썸네일 전용) 워터마크
- (FLAC/MP3/MP4 전용) 앨범 아트
- (선택 사항) 사용자 메타데이터
- (Fragmented MP4/MPEG-TS 출력 전용) 재생 목록
- (Fragmented MP4/MPEG-TS 출력 전용, 선택 사항) HLS 콘텐츠 보호
- (HLSv3 및 Smooth 재생 목록 전용, 선택사항) 디지털 권한 관리

리전

Elastic Transcoder는 사용자가 속한 리전에 작업을 만듭니다.

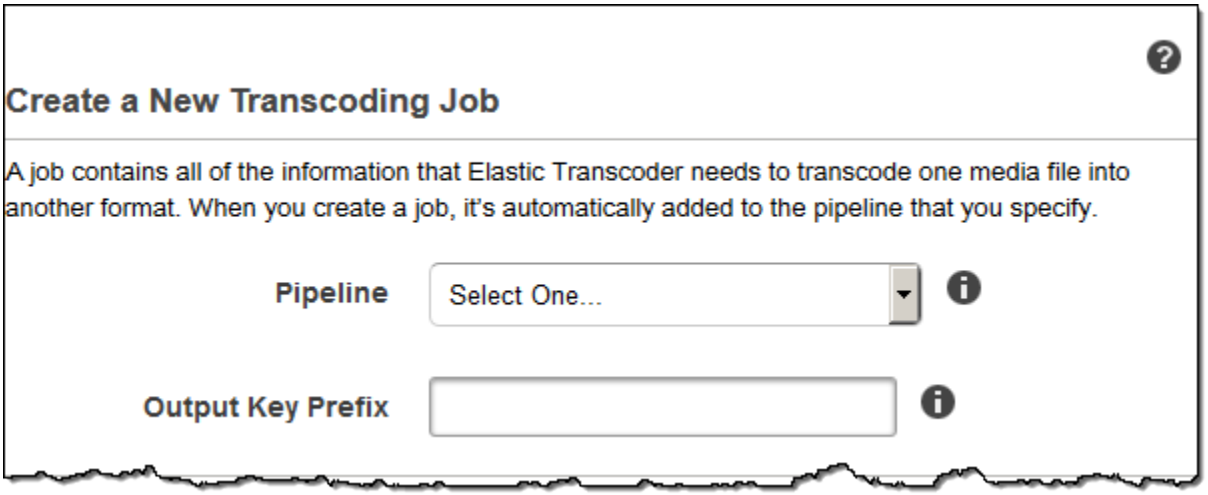
일반 설정

파이프라인

Elastic Transcoder가 트랜스코딩에 사용할 파이프라인의 이름입니다. 파이프라인은 Elastic Transcoder가 트랜스코딩할 파일을 가져올 Amazon S3 버킷, 가 Elastic Transcoder트랜스코딩된 파일을 내보낼 버킷을 포함하여 여러 설정을 결정합니다.

출력 키 접두사

Elastic Transcoder가 출력 파일, 썸네일, 재생 목록 등 이 작업이 생성하는 모든 파일의 이름 앞에 배치할 값입니다(있는 경우). 값을 지정할 경우 첫 번째 문자 뒤 어딘가에 /를 포함해야 합니다. 그러면 Amazon S3파일 관리가 간소화됩니다.



입력 세부 정보, 1부

이 섹션의 설정을 사용하여 입력 파일에 대한 정보를 지정합니다. 두 개 이상의 입력 파일을 제공하면, Elastic Transcoder는 이를 트랜스코딩한 후 연결하여 하나의 출력 파일로 만듭니다. 자세한 내용은 클립 스티칭 단원을 참조하십시오.

입력 키

트랜스코딩할 파일의 이름. 파일을 가져올 Amazon S3 버킷을 확인하기 위해 Elastic Transcoder는 사용자가 이 작업에 대해 지정한 파이프라인의 입력 버킷 필드를 참조합니다.

파일 이름에 접두사(예: `cooking/lasagna.mpg`)가 포함된 경우 키에 접두사를 포함합니다. 파일이 지정된 버킷에 없으면 Elastic Transcoder가 오류를 반환합니다.

입력 암호 해독

입력 파일을 해독하는 데 사용되는 암호화 설정입니다(있는 경우). 입력 파일을 암호화하는 경우 Elastic Transcoder가 파일을 해독하는 데 사용할 모드를 지정해야 합니다.

암호 해독 모드(암호 해독의 경우 필수)

파일을 해독할 때 Elastic Transcoder가 사용할 특정 암호화 모드입니다.

Elastic Transcoder는 다음 옵션을 지원합니다.

- Amazon S3 서버 측 암호화: Amazon S3가 파일 암호화 및 해독을 처리합니다. Elastic Transcoder가 Amazon S3 버킷에 대한 액세스 권한이 있으면 다른 조치가 필요하지 않습니다.

  자세한 내용은 Amazon Simple Storage Service 사용 설명서의 서버 측 암호화를 사용하여 데이터 보호를 참조하세요.
- 고객 제공 키를 사용한 클라이언트측 암호화: Elastic Transcoder가 고객 제공 키를 사용하는 세 가지 암호화 유형을 지원합니다.
  - aes-cbc-pkcs7: 패딩된 암호-블록 작업 모드입니다.
  - aes-ctr: AES Counter Mode입니다.
  - aes-gcm: AES Galois Counter Mode입니다. 인증된 암호화 형식의 작업 모드이므로 변조된 파일, 키 또는 초기화 벡터는 해독 프로세스에 실패합니다.

  AES 암호화 모드 중 하나를 선택한 경우 다음 세 값도 지정해야 합니다(모두 base64로 인코딩되어야 함).
  - 암호화 키
  - 암호화 키 MD5
  - 암호화 초기화 벡터

암호 해독 키(AES 암호 해독의 경우 필수)

파일을 암호화하는 데 사용되는 데이터 암호화 키. 이 키는 base64로 인코딩되어야 하며, base64 인코딩 전에 다음 비트 길이 중 하나여야 합니다.

96 (AES-GCM만 해당), 128, 192 또는 256.

키도를 사용하여 암호화해야 합니다 AWS KMS. 자세한 내용은 AWS Key Management Service 개발자 안내서의 데이터 암호화 및 해독을 참조하세요.

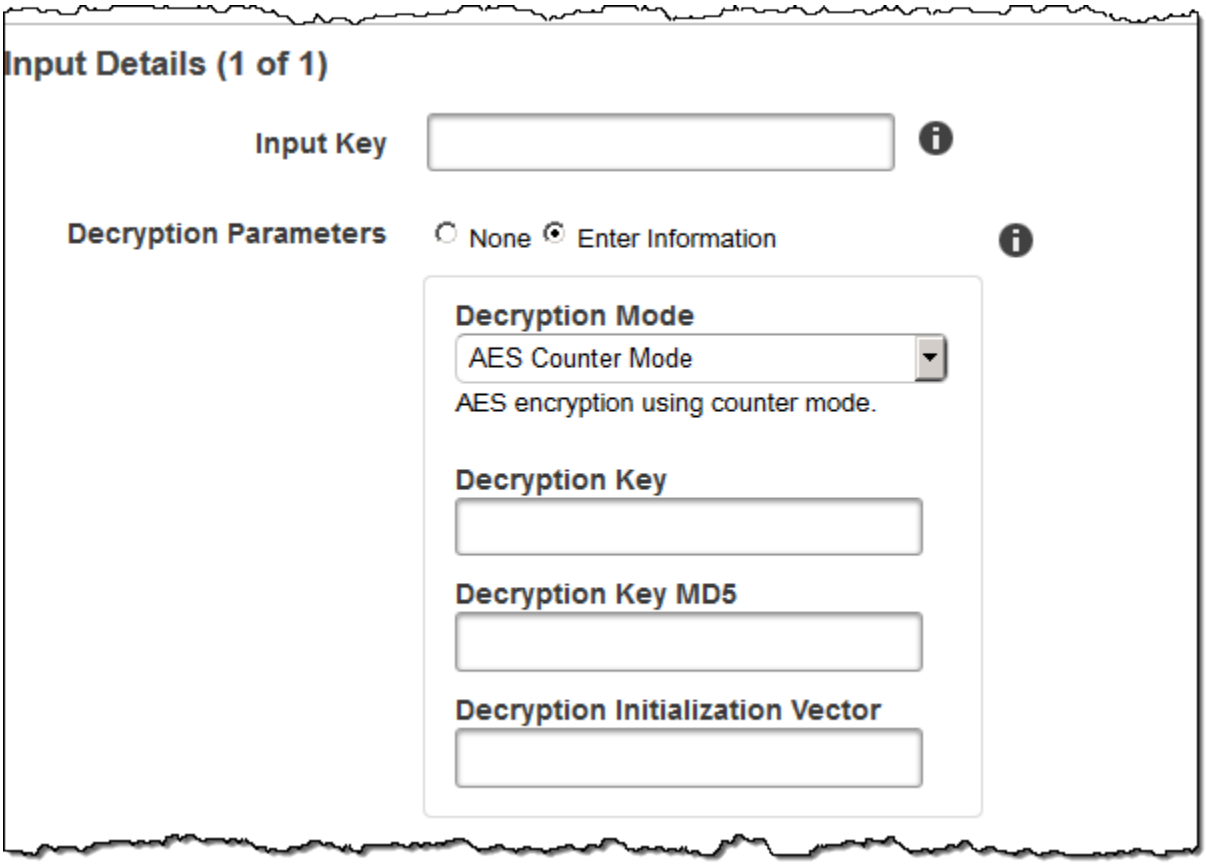
암호 해독 키 MD5(AES 암호 해독의 경우 필수)

입력 파일을 암호화하는 데 사용되며 Elastic Transcoder가 전송 도중 키가 손상되지 않았는지 확인하기 위한 체크섬으로 사용할 키의 MD5 다이제스트입니다. 키 MD5는 base64로 인코딩되어야 하며, base64 인코딩 전에 정확히 16바이트여야 합니다.

암호 해독 초기화 벡터(AES 암호 해독의 경우 필수)

무작위 비트 생성기에 의해 생성되는, 모든 암호화 작업에 고유한 일련의 무작위 비트로, 입력 파일을 암호화하는 데 사용합니다. 초기화 벡터는 base64로 인코딩되어야 하며, base64 인코딩 전에 정확히 16바이트여야 합니다.

자세한 내용은 Initialization Vector를 참조하세요.



입력 세부 정보, 2부, 클립 및 캡션 설정

이 섹션의 설정을 사용하여 입력 파일의 클립 및 캡션에 대한 정보를 지정합니다. 여러 입력을 제공한 경우에만 이러한 설정을 사용할 수 있습니다.

캡션에 대한 자세한 내용은 캡션를 참조하세요.

(선택 사항) 클립 시작 시간

입력 파일의 발췌가 포함된 출력 파일을 만들 수 있습니다. Clip Start Time은 입력 파일에서 클립을 시작하려는 위치를 나타냅니다. 형식은 HH:mm:ss.SSS(최대값: 23:59:59.999, SSS는 1/1,000초) 또는 sssss.SSS(최대값: 86399.999)입니다. 값을 지정하지 않을 경우 Elastic Transcoder는 입력 파일의 첫 부분에서 시작합니다.

(선택 사항) 클립 지속시간

발췌 클립의 지속시간. 형식은 HH:mm:ss.SSS(최대값: 23:59:59.999, SSS는 1/1,000초) 또는 sssss.SSS(최대값: 86399.999)입니다. 값을 지정하지 않을 경우 Elastic Transcoder는 파일의 클립 시작 시간에서 끝까지 잘라냅니다.

입력 파일의 지속시간보다 긴 값을 지정할 경우 Elastic Transcoder는 파일의 클립 시작 시간에서 끝까지 트랜스코딩하고 경고 메시지를 반환합니다.

(비디오 전용) 캡션 병합 정책

Elastic Transcoder가 다중 캡션을 처리하는 방법을 결정하는 정책.

- MergeOverride: Elastic Transcoder가 임베디드 캡션과 사이드카 캡션을 모두 출력에 트랜스코딩합니다. 특정 언어의 캡션이 입력 파일에 포함되고 사이드카 파일에도 나타나는 경우 Elastic Transcoder는 해당 언어에 사이드카 캡션을 사용하고 임베디드 캡션은 무시합니다.
- MergeRetain: Elastic Transcoder가 임베디드 캡션과 사이드카 캡션을 모두 출력에 트랜스코딩합니다. 특정 언어의 캡션이 입력 파일에 포함되고 사이드카 파일에도 나타나는 경우 Elastic Transcoder는 해당 언어에 임베디드 캡션을 사용하고 사이드카 캡션은 무시합니다. 캡션 소스가 비어 있을 경우 Elastic Transcoder는 출력 파일에서 모든 사이드카 캡션을 생략합니다.
- Override: Elastic Transcoder가 캡션 소스에서 지정된 사이드카 캡션만 트랜스코딩합니다.

(비디오 전용, 선택 사항) 입력 키

Elastic Transcoder가 트랜스코딩하고 출력에 포함할 사이드카 캡션 파일의 이름입니다.

(비디오 전용) 언어

캡션의 언어를 다음 형식 중 하나로 지정하는 문자열:

- 2자 ISO 639-1 코드(예: 영어의 경우 en)
- 3자 ISO 639-2 코드(예: 영어의 경우 eng)

ISO 언어 코드에 대한 자세한 내용은 List of ISO 639-1 codes를 참조하세요.

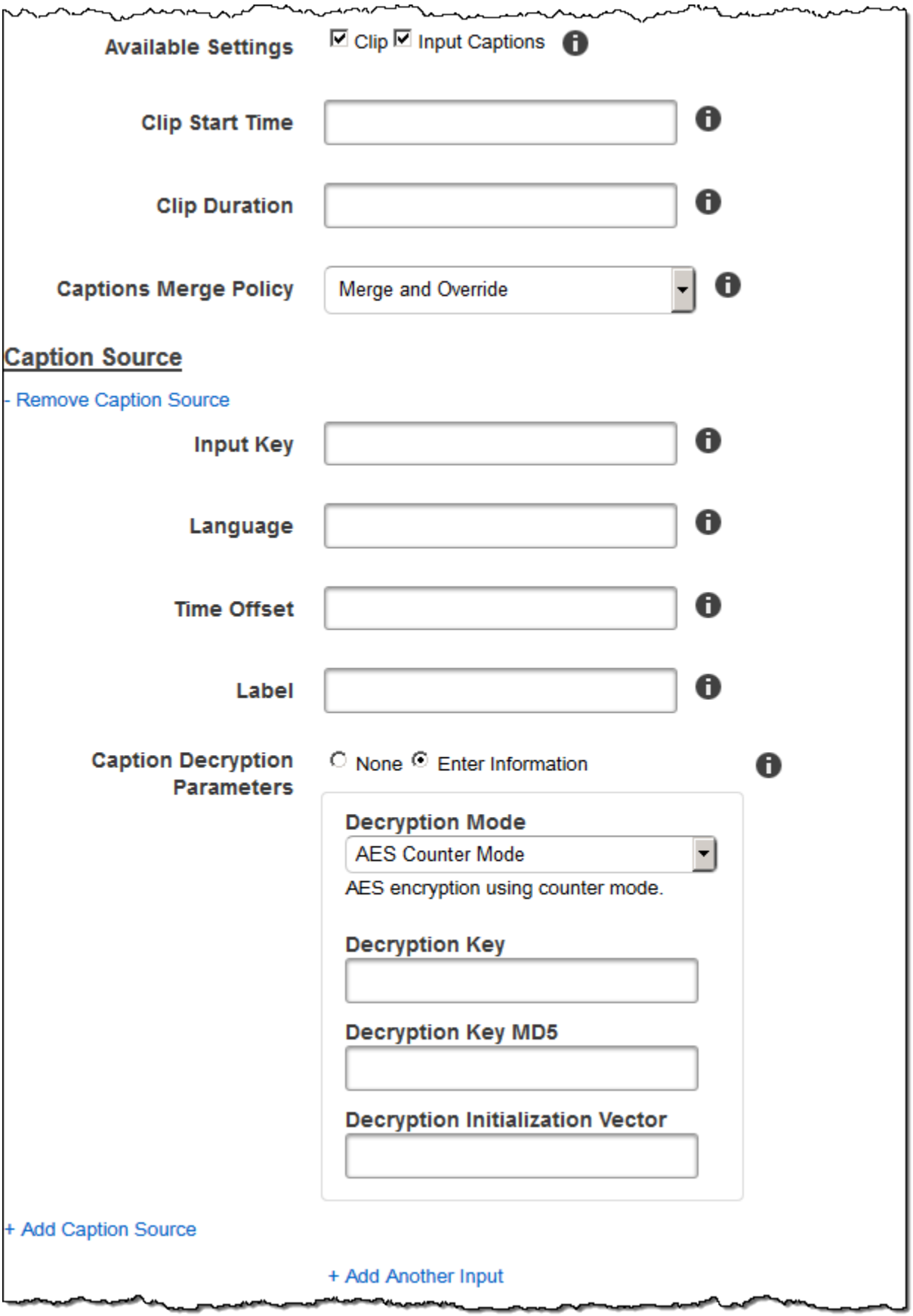
(비디오 전용, 선택 사항) TimeOffset

연결된 비디오 파일과 동시에 시작하지 않는 클립 생성 또는 캡션의 경우 TimeOffset은 Elastic Transcoder가 캡션을 포함하기 전에 인코딩할 비디오 길이를 지정합니다.

TimeOffset을 [+-]SS.sss 또는 [+-]HH:mm:SS.ss 형식으로 지정합니다.

(비디오 전용, 선택 사항) 레이블

언어를 선택할 때 플레이어에 표시되는 캡션의 레이블. 캡션 언어 이름은 여기에 캡션 언어로 지정하는 것이 좋습니다.


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 Counter Mode
AES encryption using counter mode.
Decryption Key
Decryption Key MD5
Decryption Initialization Vector
+ Add Caption Source
+ Add Another Input

출력 세부 정보, 1부

이 섹션의 설정을 사용하여 출력 파일에 대한 정보를 지정합니다.

프리셋

이 출력에 사용할 프리셋. 이 프리셋은 Elastic Transcoder가 트랜스코딩에 사용하는 오디오, 비디오 및 썸네일 설정을 결정합니다.

(Fragmented MP4/MPEG-TS 출력 전용) 세그먼트 지속시간

Container 값이 ts(MPEG-TS) 또는 fmp4(Fragmented MP4)인 현재 출력에 프리셋을 지정할 경우 Segment Duration은 각 세그먼트의 목표 최대 지속시간(초)입니다. HLSv3 형식 재생 목록인 경우 각 미디어 세그먼트가 별도의 `.ts` 파일에 저장됩니다. HLSv4, MPEG-DASH 및 Smooth 형식 재생 목록인 경우 특정 출력의 모든 미디어 세그먼트가 단일 파일에 저장됩니다. 각 세그먼트는 대략 Segment Duration의 길이입니다(개별 세그먼트는 이보다 짧거나 길 수 있음).

유효한 값 범위는 1~60초입니다. 비디오 지속시간이 Segment Duration에 의해 균등하게 분할되지 않는 경우 마지막 세그먼트의 지속시간은 다음 계산의 나머지입니다.

```
total length/Segment Duration
```

Elastic Transcoder는 출력 키에 지정된 각 HLS 출력에 대해 출력별 재생 목록을 생성합니다. 이 작업의 마스터 재생 목록에 출력을 추가하려면 Outputs in Master Playlist에서 추가합니다.

Elastic Transcoder는 이 세그먼트화를 출력 비디오와 연결된 모든 캡션에 적용합니다.

출력 키

Elastic Transcoder가 트랜스코딩된 파일 및 재생 목록에 할당할 이름입니다. Elastic Transcoder는 Pipeline에서 지정한 파이프라인의 버킷 필드에 지정된 Amazon S3 버킷에 파일을 하나 이상 저장합니다. 해당 버킷에 지정된 이름의 파일이 이미 있을 경우 출력이 실패합니다. 하지만 동일한 작업의 다른 출력은 성공할 수 있습니다.

파일 이름 형식은 컨테이너 유형과 세그먼트 지속시간이 설정되었는지 여부에 따라 다릅니다. 컨테이너 유형이 `ts`가 아니거나 세그먼트 지속시간이 지정되지 않은 경우 출력 파일의 이름은 Output Key Prefix 및 Output Key의 연결입니다.

컨테이너 유형이 `ts`이고 세그먼트 지속시간이 지정된 경우 Elastic Transcoder는 출력 키 값을 사용하여 출력의 재생 목록과 `.ts` 파일을 모두 명명합니다.

- 재생 목록:

- HLSv3: 파일 이름은 Output Key Prefix 및 Output Key의 연결에 파일 이름 확장명 .m3u8이 추가된 형식입니다.

  Output Key PrefixOutput Key.m3u8

- HLSv4: 파일 이름은 Output Key Prefix 및 Output Key 연결에 파일 이름 확장명 _v4.m3u8이 추가된 형식입니다. 비디오 출력은 Output Key Prefix 및 Output Key의 연결에 파일 이름 확장명 _iframe.m3u8이 추가된 파일 이름을 갖는 두 번째 파일을 생성합니다.

  Output Key PrefixOutput Key_v4.m3u8

  Output Key PrefixOutput Key_iframe.m3u8(비디오 전용)

- 세그먼트(.ts) 파일:
  - HLSv3: 파일 이름은 Output Key Prefix 및 Output Key의 연결에 00000부터 시작하는 5자리 순차 카운터 및 파일 이름 확장명 .ts가 추가된 형식입니다.

    Output Key PrefixOutput Key00000.ts

  - HLSv4: 파일 이름은 Output Key Prefix 및 Output Key의 연결에 파일 이름 확장명 .ts가 추가된 형식입니다.

    Output Key PrefixOutput Key.ts

세그먼트화된 `ts` 출력이 마스터 재생 목록에 포함되지 않는 경우 Elastic Transcoder는 출력을 HLSv3으로 취급합니다.

> **Note**
>
> Elastic Transcoder는 자동으로 HLSv3 또는 HLSv4 재생 목록의 출력에 관련 파일 확장명을 추가합니다. HLSv3 또는 HLSv4 재생 목록 출력에 대한 출력 키에 파일 확장자를 포함하는 경우 파일 이름에 두 개의 확장자가 들어갑니다.

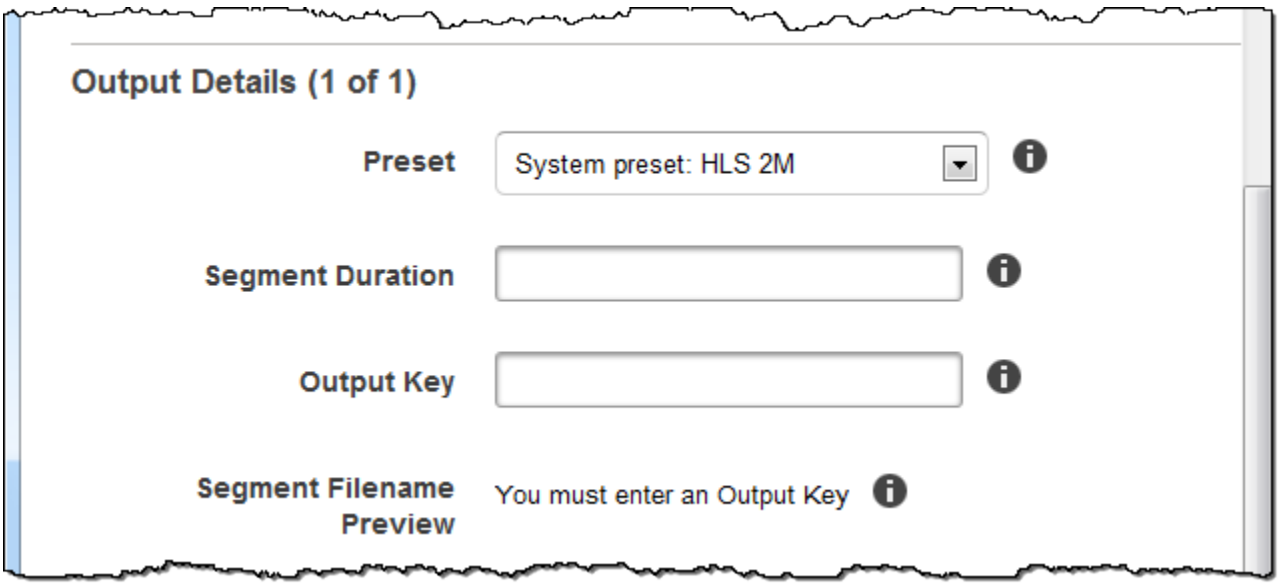
출력 키 접두사는 Amazon S3 버킷에서 단일 작업의 모든 파일을 그룹화합니다. 단일 작업 내 각 출력의 파일을 그룹화하려면 Output Key 값에 접두사를 포함할 수 있습니다. 예:

OutputKeyPrefixiPhone/Key00000.ts

OutputKeyPrefixKindleFireHD/Key00000.ts

세그먼트 파일 이름 미리 보기

출력 키에 입력한 값에 기반한 첫 번째 세그먼트 파일의 이름.



출력 세부 정보, 2부

이 섹션의 설정을 사용하여 출력 파일에 대한 정보를 지정합니다.

(비디오 전용) 썸네일 생성

Elastic Transcoder가 비디오의 썸네일을 생성하게 하려면 예를 선택한 후, 썸네일 파일 이름 패턴 필드에서 파일 이름의 형식을 지정합니다.

(비디오 전용) 썸네일 파일 이름 패턴

썸네일 파일 이름 패턴으로 예 를 선택한 경우 파일 이름의 형식을 지정합니다. 다음 값을 임의의 순서로 지정할 수 있습니다.

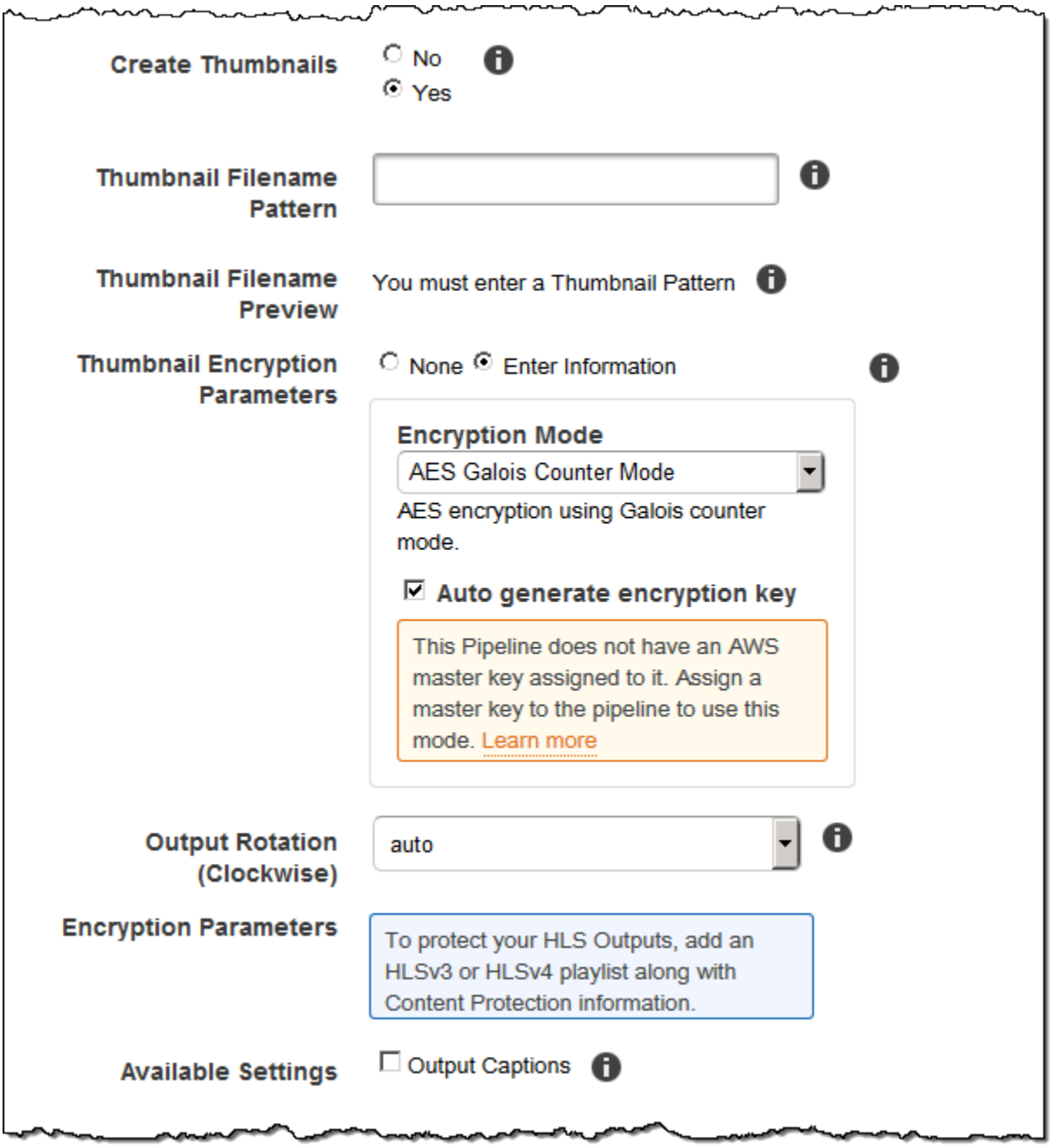
- {count}(필수): 해당 썸네일이 트랜스코딩된 파일의 썸네일 시퀀스에서 표시되는 위치를 나타내는 00001로 시작하는 5자리 숫자입니다. 필드의 어딘가에 {count}를 포함시켜야 합니다. 이를 생략하면 Elastic Transcoder가 파일 이름의 끝이자 파일 이름 확장명(.jpg 또는 .png) 바로 앞에 카운트를 자동으로 추가합니다.
- (선택 사항) 리터럴 값: 파일 이름 접두사로 또는 `{resolution}`과 `{count}` 사이의 구분 기호 등으로 필드 내 임의의 위치에 리터럴 값을 지정할 수 있습니다.
- (선택 사항) {resolution}: Elastic Transcoder가 파일 이름에 해상도를 포함하게 하려면 필드에 `{resolution}`을 포함합니다.

Thumbnail Filename Preview 필드에는 Thumbnail Filename Pattern에 입력한 값에 기반한 썸네일의 파일 이름 샘플이 표시됩니다.

썸네일을 생성할 때 Elastic Transcoder는 자동으로 파일을 Preset에 지정된 프리셋에 나타나는 형식(.jpg 또는 .png)으로 저장합니다. 또한 Elastic Transcoder는 해당하는 파일 이름 확장명을 추가합니다.

(비디오/썸네일 전용, 선택 사항) 출력 회전

Elastic Transcoder가 입력에 대해 출력을 시계 방향으로 회전할 각도입니다. Elastic Transcoder가 자동으로 입력 파일이 회전하는지 여부를 감지하게 하려면 auto를 선택합니다. 그러나 트랜스코딩하는 파일에 회전 메타데이터가 포함된 경우 Elastic Transcoder는 일반적으로 출력을 회전해야 하는지 여부만 감지할 수 있습니다.



출력 세부 정보, 3부, 캡션 설정

이 섹션의 설정을 사용하여 출력 파일의 캡션에 대한 정보를 지정합니다.

캡션에 대한 자세한 내용은 캡션를 참조하세요.

(비디오 전용) 캡션 형식

지정하는 형식은 Elastic Transcoder가 이 출력에 임베디드 캡션 또는 사이드카 캡션을 생성하는지 여부를 결정합니다. 이 값을 비워 두면 Elastic Transcoder가 오류를 반환합니다.

- 임베디드 캡션 형식: MP4 컨테이너의 경우 mov-text 및 CEA-708이 지원됩니다. MPEG-TS 컨테이너의 경우 CEA-708이 지원됩니다. 다른 컨테이너 형식의 경우 임베디드 캡션 형식이 지원되지 않습니다.

  CEA-708 캡션은 스트림의 H.264 SEI 사용자 데이터에 포함됩니다. Elastic Transcoder는 출력당 최대 1개의 임베디드 형식을 지원합니다.

- 사이드카 캡션 형식: Elastic Transcoder는 dfxp, scc, srt, webvtt를 지원합니다. Smooth 재생 목록을 갖는 Fmp4 컨테이너는 dfxp만 지원하며, Elastic Transcoder가 확장명 `.ismt`로 파일을 생성합니다. MPEG-DASH 재생 목록을 갖는 Fmp4 컨테이너는 webvtt만 지원하며, Elastic Transcoder가 확장명 `.vtt`로 파일을 생성합니다. ttml 또는 smpte-tt 호환 캡션을 사용하려는 경우 출력 형식으로 dfxp를 지정하세요.

(비디오 전용) 캡션 파일 이름 패턴

description-{language} 형식의 캡션 파일 이름의 접두사. 여기서,

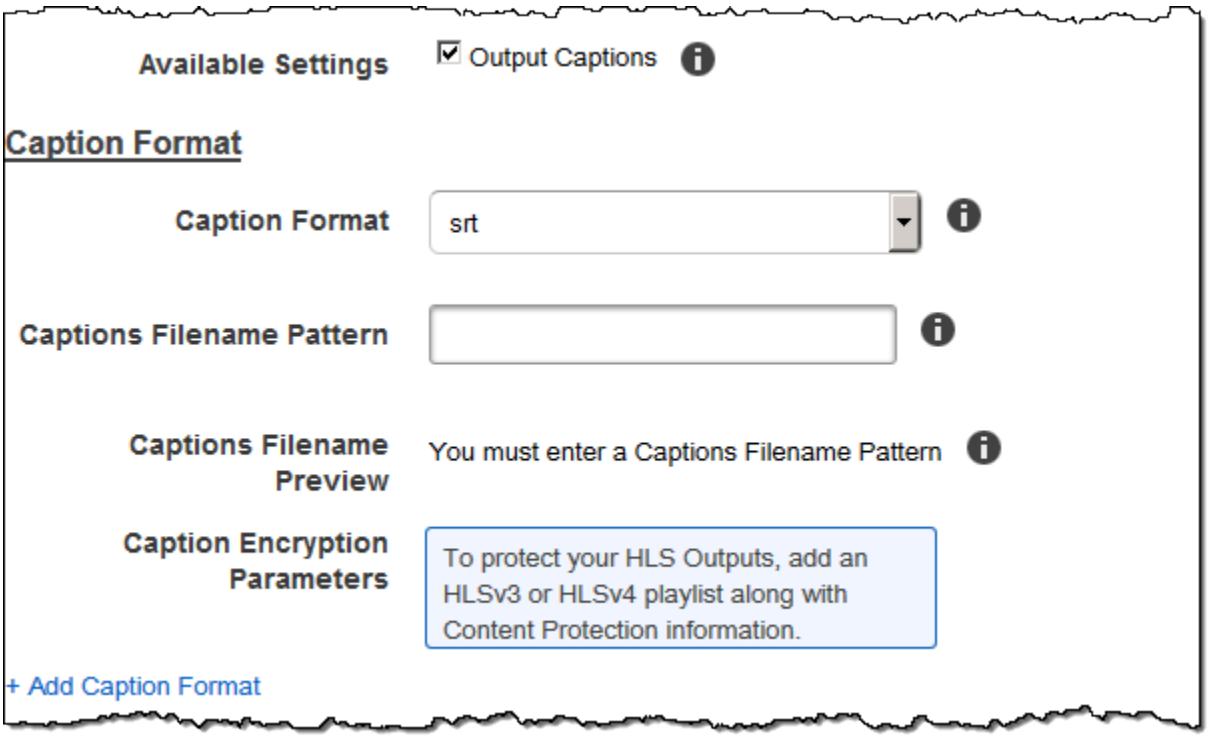
- *description*은 비디오에 대한 설명입니다.
- {language}는 Elastic Transcoder가 출력 파일 이름 내 캡션 언어의 2자 또는 3자 코드로 대체하는 리터럴 값입니다.

파일 이름 패턴에 {language}를 포함하지 않을 경우 Elastic Transcoder는 자동으로 "{language}"를 *##*에 지정하는 값에 추가합니다. 또한 Elastic Transcoder는 자동으로 세그먼트 파일의 끝에 카운터를 추가합니다.

예를 들어 srt 형식으로 트랜스코딩하는 것으로 가정해 보겠습니다. "Sydney-{language}-sunrise"를 입력하고 캡션 언어가 영어(en)일 경우 첫 번째 캡션 파일의 이름은 `Sydney-en-sunrise00000.srt`가 됩니다.

(비디오 전용) 캡션 파일 이름 미리 보기

Captions Filename Pattern에 입력한 값에 기반한 첫 번째 캡션 파일의 이름. 미리 보기는 영어 ISO 코드 "en"을 사용하여 {language}가 파일 이름에 어떻게 나타나는지 보여 줍니다.



(선택 사항) 출력 암호화

트랜스코딩 작업에 사용할 출력 파일의 암호화 설정을 지정할 수 있습니다. 여기에는 출력 파일 및 워터마크, 썸네일, 앨범 아트 또는 사용할 캡션 등이 포함됩니다. 각 파일에 개별적으로 암호화 설정을 지정해야 합니다.

출력 암호화

Elastic Transcoder가 출력 파일에 적용할 암호화 설정(있는 경우). 암호화를 사용하기로 했으면 사용할 모드를 지정해야 합니다. 암호화를 사용하지 않도록 설정하면 Elastic Transcoder가 Amazon S3 버킷에 암호화되지 않은 파일을 기록합니다.

(파일 수준 암호화의 경우 필수) 암호화 모드

출력 파일을 개별적으로 암호화할 때 Elastic Transcoder에서 사용하려는 특정 암호화 모드입니다. Elastic Transcoder는 다음과 같은 암호화 모드 옵션을 지원합니다.

- s3: 이 파일을 암호화하는 데 사용되는 키를 생성하고 관리합니다.

  자세한 내용은 Amazon Simple Storage Service 사용 설명서의 서버 측 암호화를 사용하여 데이터 보호를 참조하세요.

- s3-aws-kms: 파일 암호화에 사용되는 키를 AWS KMS생성하고 관리하는 Amazon S3 호출입니다. s3-aws-kms를 지정하되 기본 키를 사용하지 않으려면 사용할 AWS-KMS 키를 파이프라인에 추가합니다.

  자세한 내용은 Amazon Simple Storage Service 사용 설명서의 AWS KMS 키를 사용한 서버 측 암호화(SSE-KMS) 사용을 참조하세요.
- aes-cbc-pkcs7: 패딩된 암호-블록 작업 모드입니다.
- aes-ctr: AES Counter Mode입니다.
- aes-gcm: AES Galois Counter Mode입니다. 인증된 암호화 형식의 작업 모드이므로 변조된 파일, 키 또는 초기화 벡터는 해독 프로세스에 실패합니다.

AES 암호화 모드 중 하나를 선택한 경우 다음 세 값도 지정해야 합니다(모두 base64로 인코딩되어야 함).

- 암호화 키
- 암호화 키 MD5
- 암호화 초기화 벡터

AES 암호화 모드 중 하나를 선택한 경우 Elastic Transcoder가 사용자 대신 128비트 AES 암호화 키를 생성하게 하려면 암호화 키, 암호화 키 MD5 또는 암호화 초기화 벡터를 지정하지 마세요. Elastic Transcoder가 키를 생성하면 `ReadJob`을 호출하여 키를 가져올 수 있습니다. 이 키는 `CreateJobResponse` 객체에 포함되지 않습니다.

> **Important**
>
> AES 모드에서는 절대로 미디어별 프라이빗 암호화 키와 암호화되지 않은 데이터가 AWS에 저장되지 않습니다. 따라서 암호화 키를 안전하게 관리하는 것이 중요합니다. 암호화 키를 잃어버릴 경우 데이터의 암호를 해독할 수 없습니다.

(선택 사항) 암호화 키

Elastic Transcoder가 사용자 대신 키를 생성하게 하려면 이 필드를 비워 둡니다. Elastic Transcoder가 키를 생성하면 작업 읽기를 호출하여 키를 가져올 수 있습니다. 이 키는 Create Job Response(작업 응답 생성) 객체에 포함되지 않습니다.

자체 키를 제공하려면 AWS KMS를 사용해 키를 암호화해야 합니다. 이 키는 base64로 인코딩되어야 하며, base64 인코딩 전에 다음 비트 길이 중 하나여야 합니다.

96 (AES-GCM만 해당), 128, 192 또는 256.

Elastic Transcoder가 사용자 대신 키를 생성하도록 구성한 경우 Elastic Transcoder는 작업 생성 응답에서 이 필드를 비워 둡니다. 생성된 데이터 암호화 키를 가져오려면 Read Job(작업 읽기) 요청을 제출합니다.

를 사용한 키 암호화에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 데이터 암호화 및 해독을 AWS KMS참조하세요.

(암호화 키를 입력한 경우 필수) 암호화 키 MD5

Elastic Transcoder가 출력 파일을 암호화하는 데 사용하고 Elastic Transcoder가 전송 도중 키가 손상되지 않았는지 확인하기 위한 체크섬으로 사용할 키의 MD5 다이제스트입니다. 키 MD5는 base64로 인코딩되어야 하며, base64 인코딩 전에 정확히 16바이트여야 합니다.

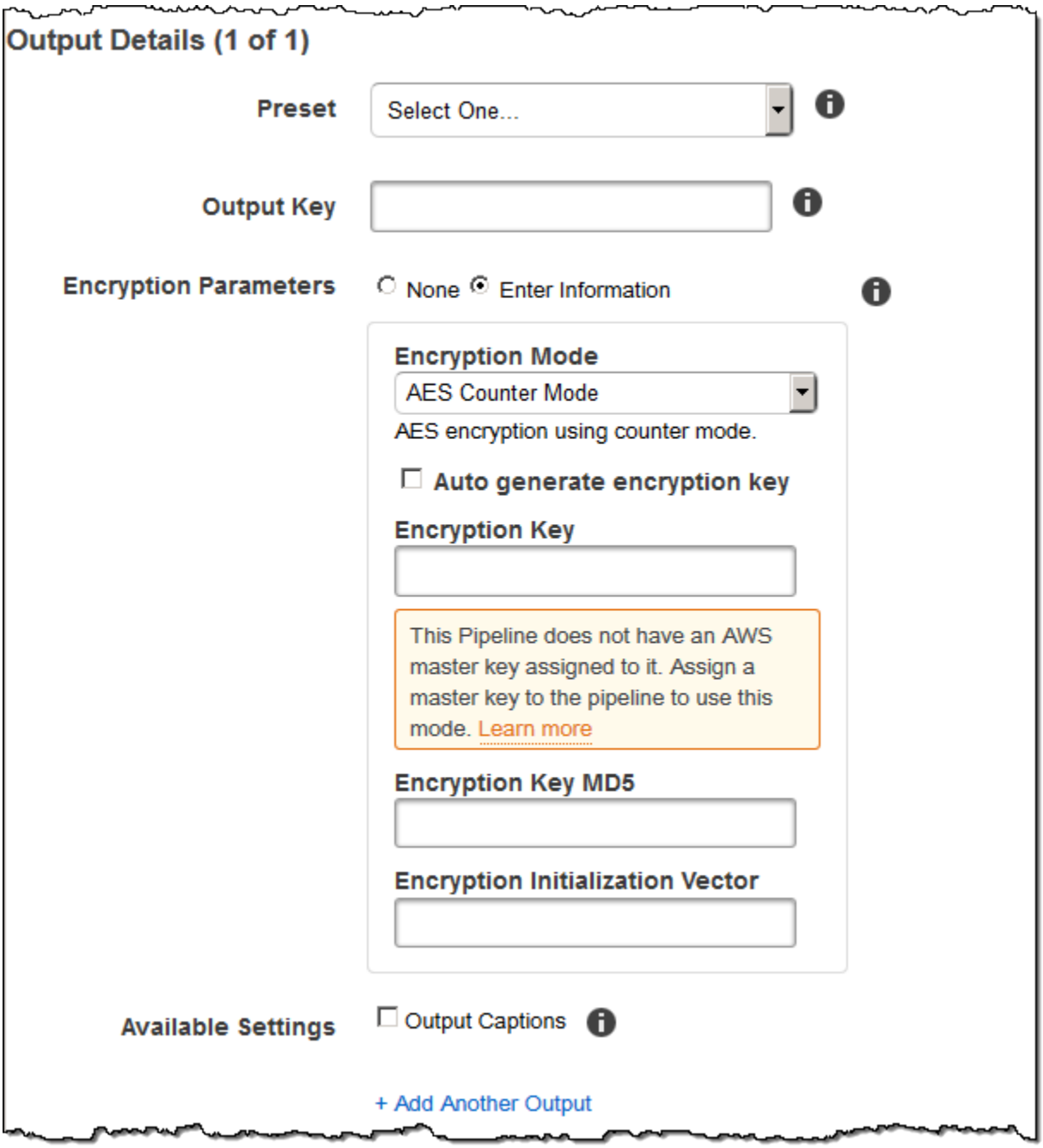
Elastic Transcoder가 사용자 대신 키를 생성하는 경우 이 필드를 비워 두어야 합니다.

(암호화 키를 입력한 경우 필수) 암호화 초기화 벡터

무작위 비트 생성기에 의해 생성되는, 모든 암호화 작업에 고유한 일련의 무작위 비트로, Elastic Transcoder가 출력 파일을 암호화하는 데 사용합니다. 초기화 벡터는 base64로 인코딩되어야 하며, base64 인코딩 전에 정확히 16바이트여야 합니다.

Elastic Transcoder가 사용자 대신 키를 생성하는 경우 이 필드를 비워 두어야 합니다.

자세한 내용은 Initialization Vector를 참조하세요.



(비디오/썸네일 전용) 워터마크

Elastic Transcoder가 트랜스코딩 도중 비디오에 추가할 워터마크에 대한 정보입니다. 각 출력에 최대 4개의 워터마크를 지정할 수 있습니다. 각 워터마크에 대한 설정은 현재 출력의 Preset에 지정된 프리셋에서 정의해야 합니다.

워터마크는 작업 출력에 나열한 순서대로 출력 비디오에 추가됩니다. 즉, 목록에서 첫 번째 워터마크가 출력 비디오에 맨 먼저 추가되고, 목록에서 두 번째 워터마크가 그 다음으로 추가되는 식입니다. 따라

서 프리셋 내 설정 때문에 Elastic Transcoder가 모든 워터마크를 동일한 위치에 배치할 경우 추가한 두 번째 워터마크가 첫 번째 워터마크를 덮고, 세 번째 워터마크가 두 번째 워터마크를, 네 번째 워터마크가 세 번째 워터마크를 덮습니다.

워터마크에 대한 자세한 내용은 워터마크 섹션을 참조하세요.

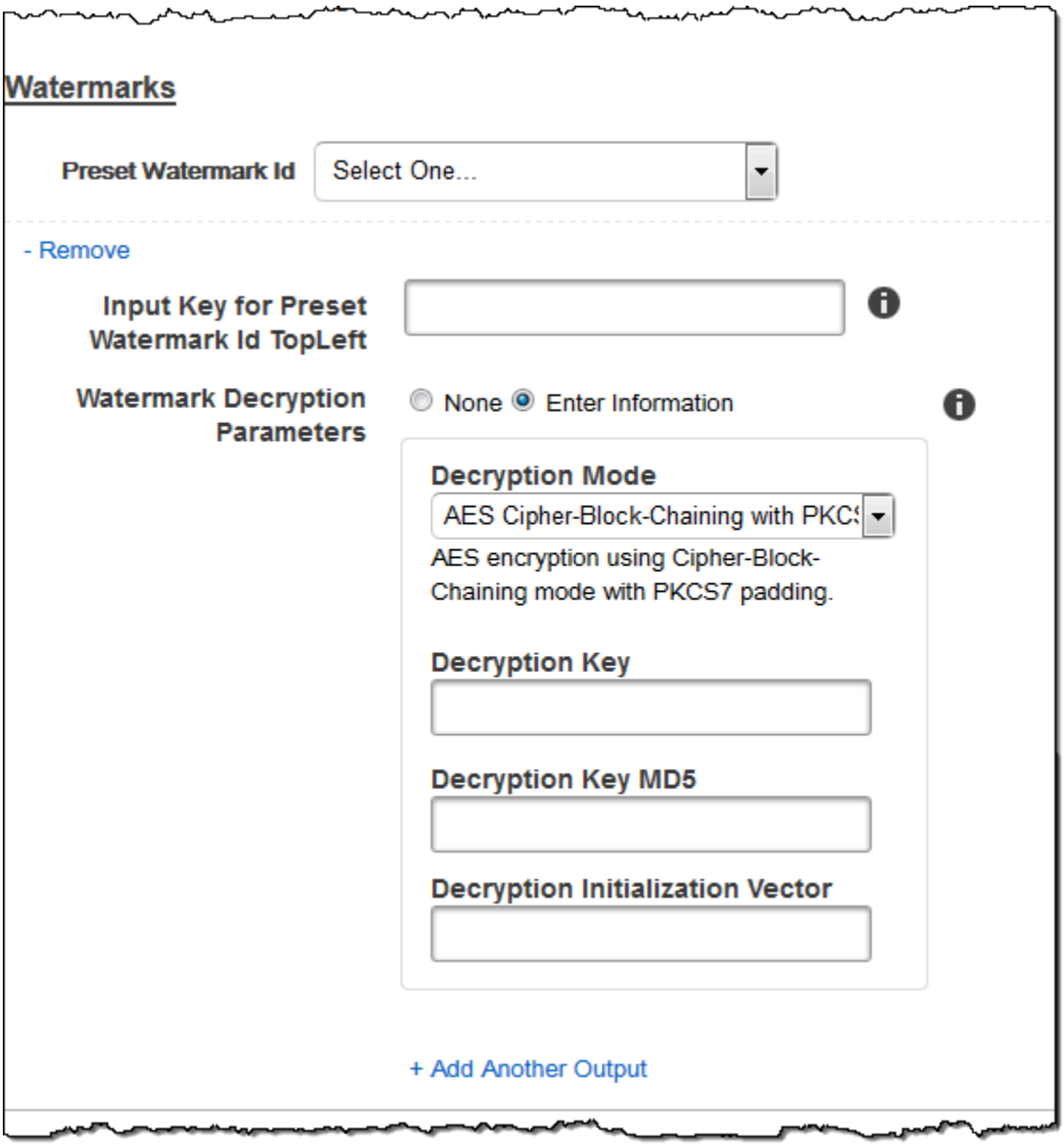
프리셋 워터마크 ID

Elastic Transcoder가 트랜스코딩 도중 비디오에 워터마크를 추가할 때 사용하는 워터마크 설정의 ID입니다. 이 설정은 현재 출력의 Preset로 지정된 프리셋에 저장됩니다. 해당 프리셋에서 워터마크 ID 값이 Elastic Transcoder가 사용할 설정을 지정합니다.

프리셋 워터마크 ID의 입력 키

워터마크에 사용할 .png 또는 .jpg 파일의 이름. 어느 Amazon S3 버킷에 지정된 파일이 저장되어 있는지 확인하기 위해 Elastic Transcoder는 Pipeline에 의해 지정된 파이프라인을 확인합니다. 해당 파이프라인의 Input Bucket 객체가 버킷을 식별합니다.

파일 이름에 접두사(예: logos/128x64.png)가 포함된 경우 키에 접두사를 포함합니다. 파일이 지정된 버킷에 없으면 Elastic Transcoder가 오류를 반환합니다.



(FLAC/MP3/MP4 전용) 앨범 아트

출력 파일과 연결할 앨범 아트(있는 경우).

아트워크를 제거하거나 아트워크를 비워 두려면 Artwork를 null로 설정하거나 Merge Policy를 "Replace"로 설정하고 빈 Artwork 어레이를 사용할 수 있습니다.

기존 아트워크를 그대로 전달하려면 Merge Policy를 "Prepend", "Append" 또는 "Fallback"으로 설정하고 빈 Artwork 어레이를 사용합니다.

> **Note**
>
> 앨범 아트는 `flac`, `mp3` 또는 `mp4` 컨테이너의 오디오 전용 출력에 사용할 수 있습니다.

앨범 아트 병합 정책

Elastic Transcoder가 복수의 앨범 아트워크 파일을 처리할 방법을 결정하는 정책입니다.

- Replace: 지정된 앨범 아트가 기존 앨범 아트를 대체합니다.
- Prepend: 지정된 앨범 아트가 기존 앨범 아트 앞에 배치됩니다.
- Append: 지정된 앨범 아트가 기존 앨범 아트 뒤에 배치됩니다.
- Fallback: 입력 파일에 아트워크가 포함된 경우 Elastic Transcoder는 출력에 해당 아트워크를 사용합니다. 입력 파일에 아트워크가 포함되지 않은 경우 Elastic Transcoder는 지정된 앨범 아트 파일을 사용합니다.

앨범 아트 아트워크

앨범 아트로 사용할 파일. 여러 아트워크를 한 오디오 파일과 연결할 수 있습니다(최대 20개).

앨범 아트 입력 키

앨범 아트로 사용할 파일의 이름. 어느 Amazon S3 버킷에 지정된 파일이 포함되어 있는지 확인하기 위해 Elastic Transcoder는 PipelineId에 의해 지정된 파이프라인을 확인합니다. 해당 파이프라인의 InputBucket 객체가 버킷을 식별합니다.

파일 이름에 접두사(예: `cooking/pie.jpg`)가 포함된 경우 키에 접두사를 포함합니다. 파일이 지정된 버킷에 없으면 Elastic Transcoder가 오류를 반환합니다.

앨범 아트 형식

앨범 아트(있는 경우)의 형식. 유효한 값은 `jpg` 및 `png`입니다.

앨범 아트 최대 너비

출력 앨범의 최대 너비(픽셀). `auto`를 지정할 경우 Elastic Transcoder는 600을 기본값으로 사용합니다. 숫자 값을 지정하려면 32~4096 범위의 정수를 입력합니다.

앨범 아트 최대 높이

출력 앨범의 최대 높이(픽셀). `auto`를 지정할 경우 Elastic Transcoder는 600을 기본값으로 사용합니다. 숫자 값을 지정하려면 32~3072 범위의 정수를 입력합니다.

앨범 아트 크기 조정 정책

출력 앨범 아트의 확대/축소를 제어하는 값.

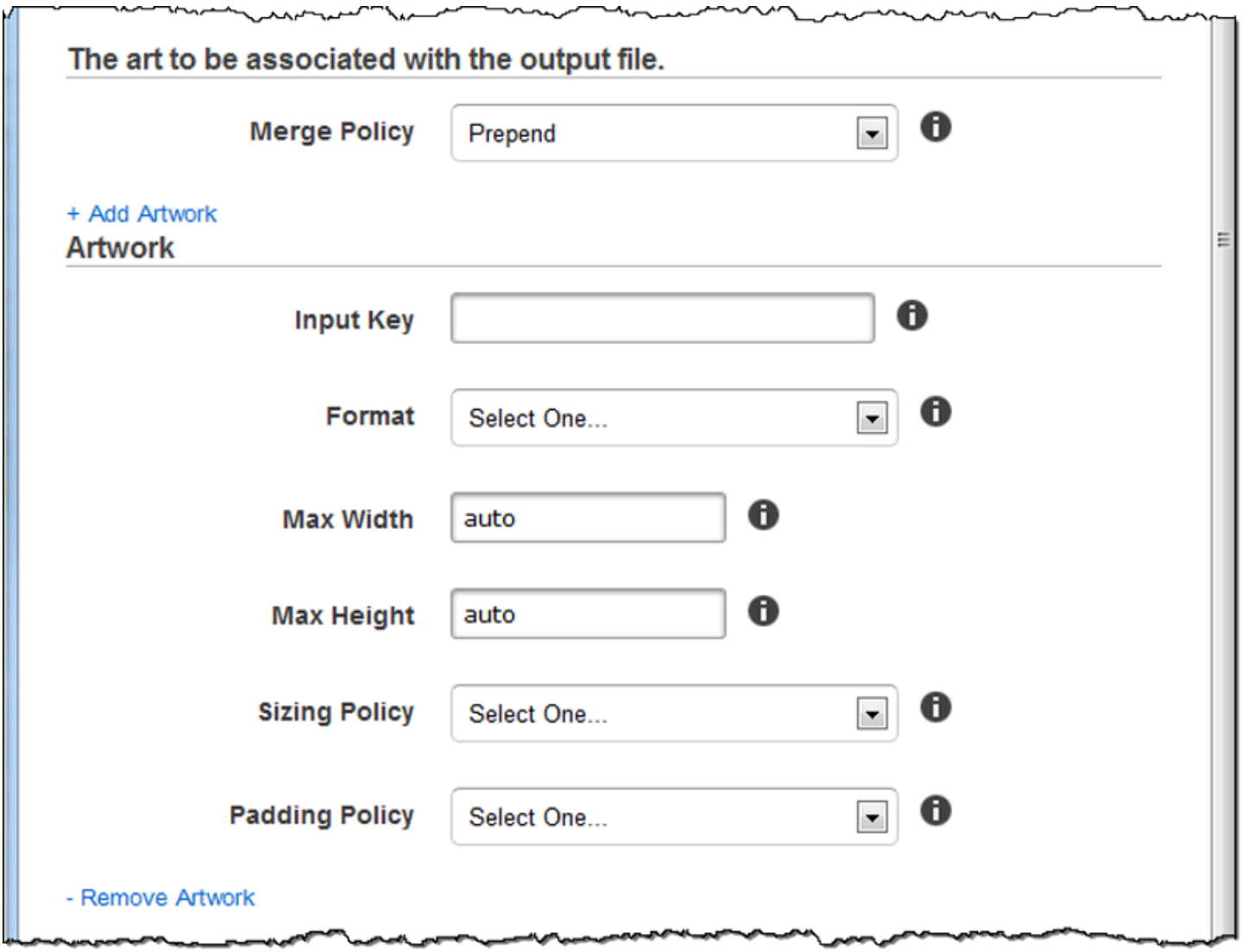
- Fit: Elastic Transcoder가 MaxWidth 또는 MaxHeight에 지정된 값과 일치하되 다른 값은 초과하지 않도록 출력 아트를 확대 또는 축소합니다.
- Fill: Elastic Transcoder가 MaxWidth 또는 MaxHeight에 지정된 값과 일치하고 다른 값과 일치하거나 초과하도록 출력 아트를 확대 또는 축소합니다. Elastic Transcoder는 출력 아트를 중앙에 배치한 다음 최대값을 초과하는 크기(있는 경우)에서 잘라냅니다.
- Stretch: Elastic Transcoder가 MaxWidth 및 MaxHeight에 지정된 값과 일치하도록 출력 아트를 늘립니다. 입력 아트의 상대 비율과 출력 아트의 상대 비율이 다를 경우 출력 아트가 변형됩니다.
- Keep: Elastic Transcoder는 출력 아트 크기를 조정하지 않습니다. 입력 아트의 치수가 MaxWidth 및 MaxHeight에 지정된 값을 초과하는 경우 Elastic Transcoder는 출력 아트를 자릅니다.
- ShrinkToFit: Elastic Transcoder가 출력 아트의 치수가 MaxWidth 및 MaxHeight 중 하나 이상에 지정된 값과 일치하되 어떤 값도 초과하지 않도록 출력 아트를 축소합니다. 이 옵션을 지정할 경우 Elastic Transcoder는 아트를 확대하지 않습니다.
- ShrinkToFill: Elastic Transcoder가 출력 아트의 치수가 MaxWidth 및 MaxHeight 중 하나 이상에 지정된 값과 일치하되 어떤 값도 하회하지 않도록 출력 아트를 축소합니다. 이 옵션을 지정할 경우 Elastic Transcoder는 아트를 확대하지 않습니다.

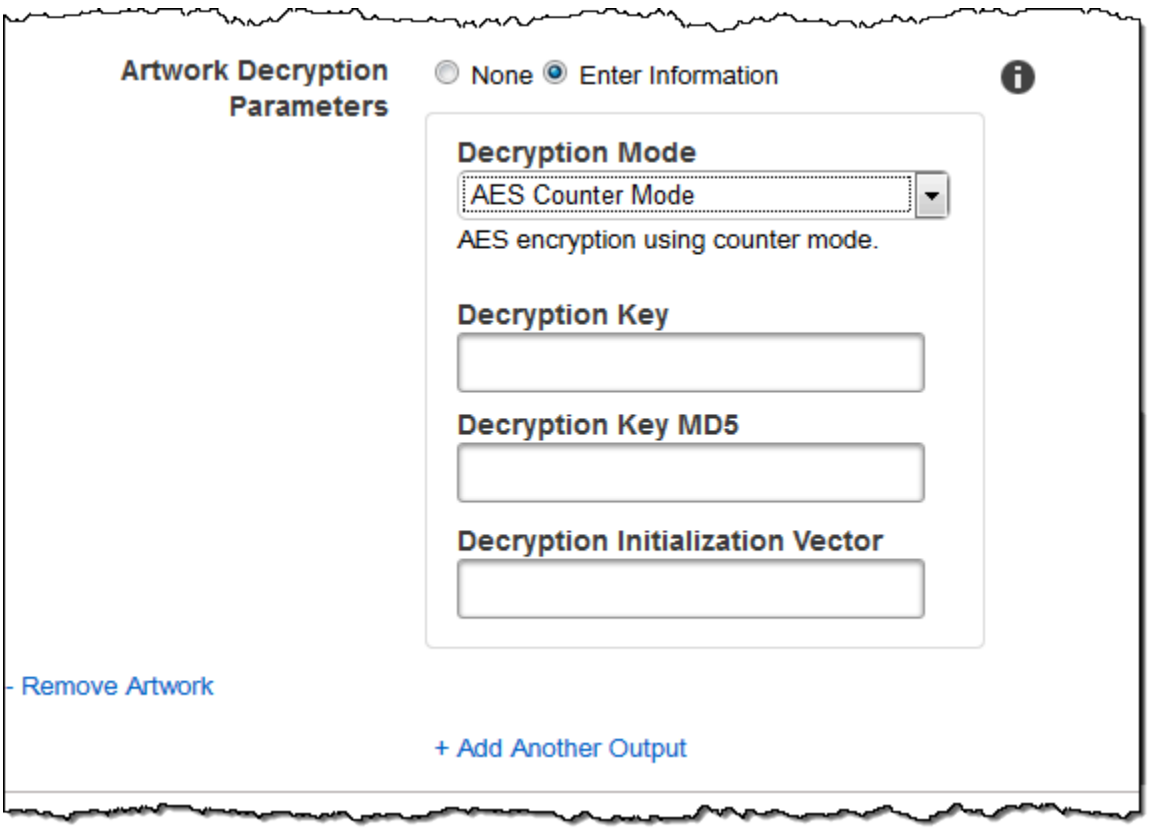
다음 표는 출력 앨범 아트에 대한 SizingPolicy 설정의 효과를 보여줍니다.

SizingPolicy	출력 앨범 아트가 확대될 수 있음	패딩 정책이 "Pad"일 경우 출력 앨범 아트가 패딩될 수 있음	출력 앨범 아트의 픽셀 종횡비가 입력 앨범 아트와 다를 수 있음	출력 앨범 아트가 잘릴 수 있음
Fit	예	예		
Fill	예			예
Stretch	예		예	
Keep	예	예		예
ShrinkToFit		예		
ShrinkToFill		예		예

앨범 아트 패딩 정책

PaddingPolicy를 Pad로 설정하면 Elastic Transcoder가 출력 아트의 전체 크기를 MaxWidth 및 MaxHeight에 지정된 값과 일치시키기 위해 출력 앨범 아트의 상하 및/또는 좌우에 화이트 바를 추가할 수 있습니다. 자세한 내용은 `AlbumArt:Art:SizingPolicy`의 표를 참조하세요.





(선택 사항) 사용자 메타데이터

Elastic Transcoder 작업과 연결할 사용자 정의 메타데이터입니다. 메타데이터를 key/value 페어로 지정합니다. key/value 페어를 사용하여 예를 들어 Season 1: Episode 3 같은 파일 세부 정보를 추적할 수 있습니다.

각 작업에 최대 10개의 키/값 쌍을 추가할 수 있습니다. Elastic Transcoder는 key/value 쌍이 지정된 순서와 동일한 순서로 반환된다고 보장하지 않습니다.

메타데이터 키

출력 파일과 함께 반환할 메타데이터 key/value 페어의 키. 각 키는 1-128자 길이의 고유한 문자열이어야 하고 다음 목록에 포함된 문자만 사용해야 합니다.

- 0-9
- A-Z 및 a-z
- Space
- 다음 기호: _.:/=+-%@

메타데이터를 정리하고, 메타데이터의 추가 128자를 저장하고, 값에 저장된 메타데이터에 레이블을 지정하기 위한 번호 지정 시스템으로 이들 키를 사용할 수 있습니다. 값 메타데이터만 사용하려는 경우 키에 key1과 같은 일회성 문자열을 배치하고 Elastic Transcoder에서 메타데이터를 가져올 때 키를 무시할 수 있습니다.

> **Important**
>
> 작업의 모든 키에 고유한 문자열을 지정해야 합니다. 단일 작업에서 2개 이상의 키에 동일한 문자열이 사용될 경우 Elastic Transcoder는 해당 키를 사용하는 키/값 페어 중 하나만 반환합니다. 어느 값이 반환될지는 알 수 없습니다.

메타데이터 값

작업과 함께 반환할 메타데이터 key/value 페어의 값. 각 값은 0-256자 길이의 문자열이어야 하고 다음 목록에 포함된 문자만 사용해야 합니다.

- 0-9
- A-Z 및 a-z
- Space
- 다음 기호: _.:/=+-%@

(Fragmented MP4/MPEG-TS 출력 전용) 재생 목록

프리셋 목록에서 컨테이너의 값이 ts(MPEG-TS) 또는 fmp4(Fragmented MP4)인 프리셋을 선택하는 경우, 이 섹션의 설정을 사용하여 Elastic Transcoder에서 만들려는 마스터 재생 목록에 대한 정보를 지정하세요. 재생 목록 형식당 최대 1개의 마스터 재생 목록을 생성하는 것이 좋습니다.

마스터 재생 목록 이름

Elastic Transcoder가 마스터 재생 목록에 할당할 이름입니다. 이름에 / 문자가 포함된 경우, 마지막 / 앞의 이름 부분은 모든 재생 목록 이름에서 동일해야 합니다. 마스터 재생 목록을 2개 이상 생성하는 경우 각 이름이 고유해야 합니다.

> **Note**
>
> Elastic Transcoder는 관련 파일 확장자를 파일 이름에 자동으로 추가합니다(HLSv3 및 HLSv4 재생 목록의 .m3u8, MPEG-DASH 재생 목록의 .mpd, Smooth 재생 목록의 .ism 및 .ismc). Master Playlist Name에 파일 확장명을 포함하는 경우 파일 이름에 2개의 확장명이 들어갑니다.

모든 세그먼트 지속시간 설정, 클립 설정 또는 캡션 설정은 재생 목록의 모든 출력에 대해 동일해야 합니다. Smooth 재생 목록의 경우 Audio:Profile, Video:Profile, Frame Rate는 Maximum Number of Frames Between Keyframes에 대해 모든 출력에서 동일해야 합니다.

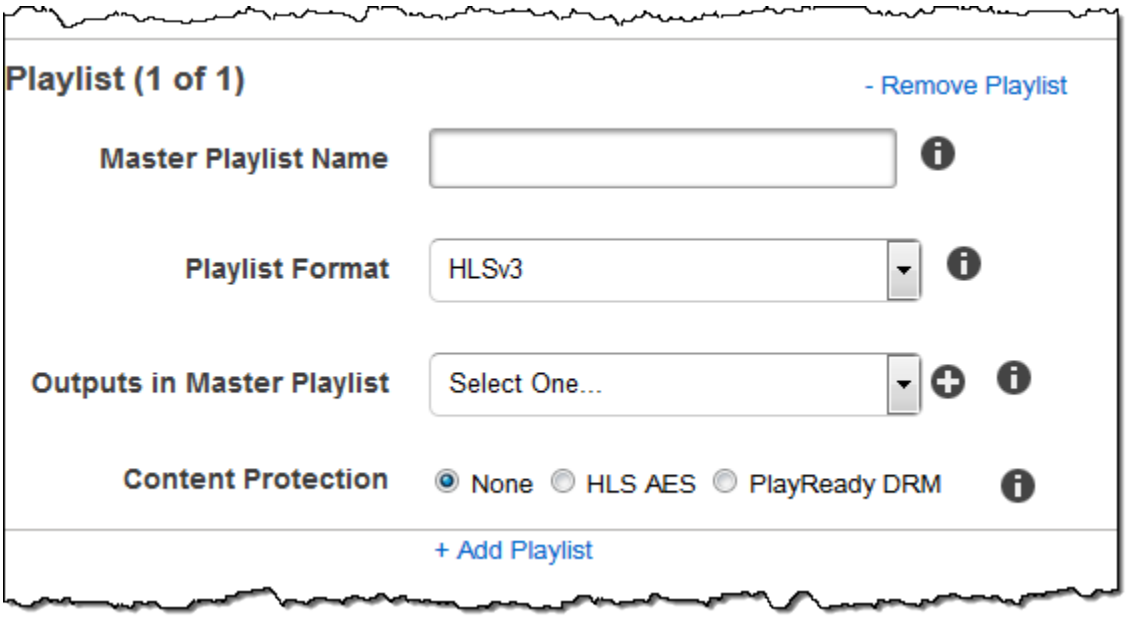
재생 목록 형식

재생 목록의 형식. 유효한 형식은 HLSv3, HLSv4, MPEG-DASH, Smooth 등입니다.

마스터 재생 목록의 출력

마스터 재생 목록에 포함하려는 이 작업의 각 출력에 대해 Output Key 필드의 값. 재생 목록에 2개 이상의 출력을 포함하는 경우 모든 출력의 Segment Duration 값이 동일해야 합니다.

HLSv4 마스터 재생 목록의 경우 Elastic Transcoder는 출력 재생 목록에서 어느 오디오 및 비디오 입력의 조합을 링크할지 선택합니다. 첫 번째 오디오 및 비디오 입력이 기본 재생 환경으로 링크 및 렌더링되어 선호하는 재생 기본값을 선택할 수 있게 합니다. 마스터 재생 목록 내 다른 개별 재생 목록에 대해서는 Elastic Transcoder가 어느 오디오 및 비디오 비트 전송률 조합이 최상의 재생을 제공할지 선택합니다.



(Fragmented MP4/MPEG-TS 출력 전용, 선택 사항) HLS 콘텐츠 보호

> **Note**
>
> HLS 콘텐츠 보호는 파일의 각 개별 세그먼트를 암호화하며, Elastic Transcoder는 파일 수준 암호화와 결합된 HLS 콘텐츠 보호를 지원하지 않습니다.

프리셋 목록에서 컨테이너 값이 ts(MPEG-TS) 또는 fmp4(Fragmented MP4)인 프리셋을 Smooth 재생 목록과 함께 선택하면 파일 스트리밍을 위한 암호화 설정을 지정할 수 있습니다. 데이터 스트림 암호화를 파일 또는 캡션 암호화와 결합할 수 없습니다. 데이터 스트림을 암호화할 경우, 이 섹션의 설정을 사용하여 데이터 스트림 암호화에 대한 정보를 지정합니다.

HLS 콘텐츠 보호를 사용하려면 세션을 인증할 수 있는 웹 서버(예: Amazon Elastic Compute Cloud), 스트리밍 미디어 파일을 배포할 방법(예: Amazon CloudFront), 암호화된 스트리밍 미디어를 재생할 방법(예: 플레이어 지원 브라우저)이 있어야 합니다.

방법

출력에 대한 콘텐츠 보호 방법. 유일하게 유효한 값은

`aes-128`.

이 값은 출력 재생 목록 내 `method` 메타데이터 태그의 `EXT-X-KEY` 속성에 기록됩니다.

Key(키)

Elastic Transcoder가 사용자 대신 키를 생성하게 하려면 이 필드를 비워 둡니다. Elastic Transcoder가 키를 생성하면 `ReadJob`을 호출하여 키를 가져올 수 있습니다. 이 키는 `CreateJobResponse` 객체에 포함되지 않습니다.

자체 키를 제공하려면 AWS KMS를 사용해 키를 암호화해야 합니다. 이 키는 base64로 인코딩되어야 하며, base64 인코딩 전에 다음 비트 길이 중 하나여야 합니다.

96 (AES-GCM만 해당), 128, 192 또는 256.

Elastic Transcoder가 사용자 대신 키를 생성하도록 구성한 경우 Elastic Transcoder는 `CreateJob` 응답에서 이 필드를 비워 둡니다. 생성된 데이터 암호화 키를 가져오려면 `ReadJob` 요청을 제출합니다.

를 사용한 키 암호화에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 데이터 암호화 및 해독을 AWS KMS참조하세요.

HLS 콘텐츠 보호 방법으로 `aes-128`를 선택할 경우 키가 128비트여야 합니다. `relative` KeyStoragePolicy가 설정된 경우 Elastic Transcoder는 Amazon S3 서버 측 암호화를 사용하여 Amazon S3 버킷에 키를 기록합니다.

키 MD5(암호화 키를 입력한 경우 필수)

Elastic Transcoder가 출력 파일을 암호화하는 데 사용하고 Elastic Transcoder가 전송 도중 키가 손상되지 않았는지 확인하기 위한 체크섬으로 사용할 키의 MD5 다이제스트입니다. 키 MD5는 base64로 인코딩되어야 하며, base64 인코딩 전에 정확히 16바이트여야 합니다.

Elastic Transcoder가 사용자 대신 키를 생성하는 경우 이 필드를 비워 두어야 합니다.

초기화 벡터(암호화 키를 입력한 경우 필수)

무작위 비트 생성기에 의해 생성되는, 모든 암호화 작업에 고유한 일련의 무작위 비트로, Elastic Transcoder가 출력 파일을 암호화하는 데 사용합니다. 초기화 벡터는 base64로 인코딩되어야 하며, base64 인코딩 전에 정확히 16바이트여야 합니다.

Elastic Transcoder가 사용자 대신 키를 생성하는 경우 이 필드를 비워 두어야 합니다.

자세한 내용은 Initialization Vector를 참조하세요.

라이선스 취득 URL

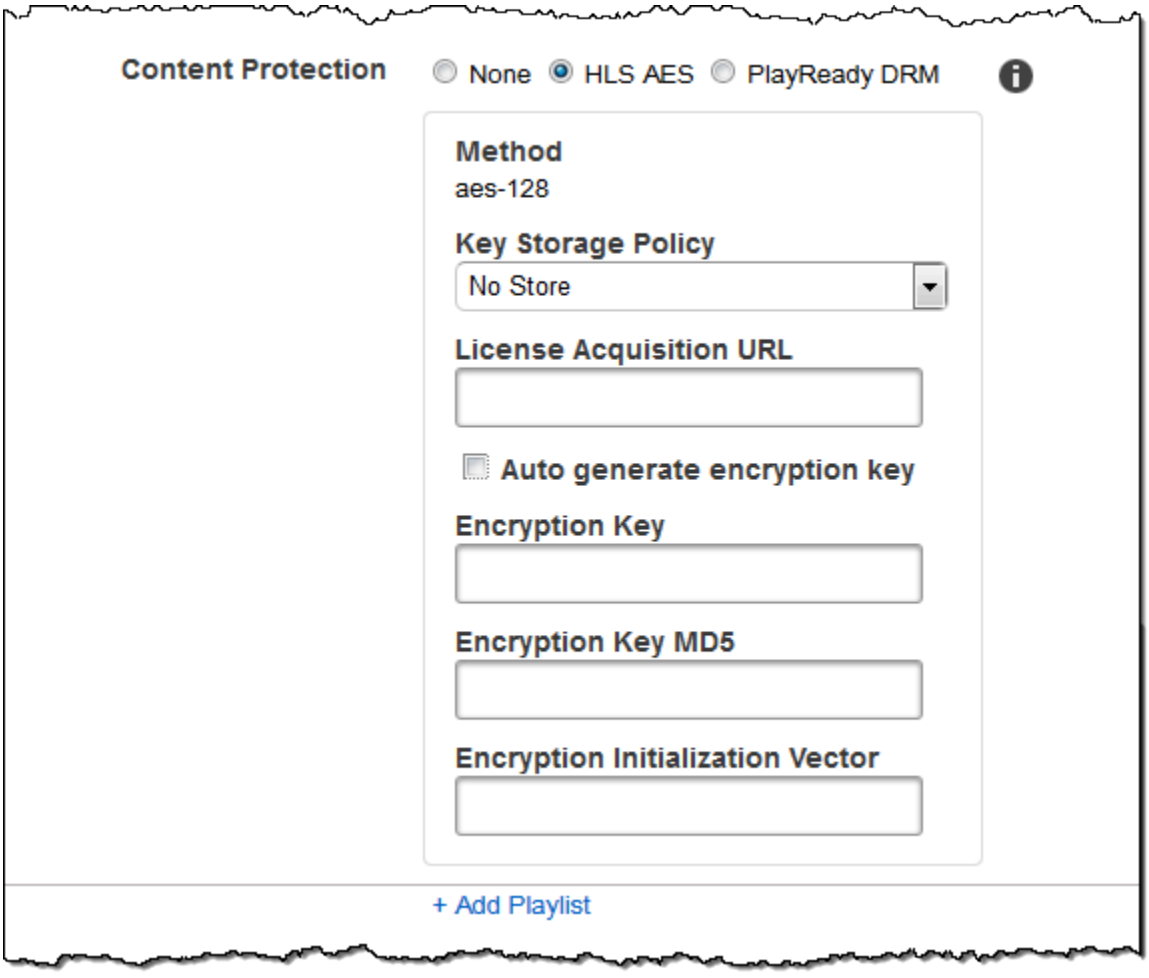
HLS 재생 목록을 해독하는 데 필요한 라이선스 키의 위치. 이 URL은 절대 경로여야 하며 재생 목록 파일 내 EXT-X-KEY 메타데이터의 URI 속성에서 참조됩니다. 예시:

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

키 스토리지 정책

Elastic Transcoder가 HLS 라이선스 키를 Amazon S3 버킷에 기록하게 할지 여부를 지정합니다. `WithVariantPlaylists`를 선택할 경우 Elastic Transcoder는 연결된 재생 목록과 동일한 Amazon S3 버킷에 암호화된 키를 기록합니다.

> **Important**
>
> `NoStore`를 선택할 경우 Elastic Transcoder는 키를 저장하지 않습니다. 키를 저장하고 최종 사용자가 키를 사용할 수 있도록 키를 저장하는 License Acquisition URL(라이선스 획득 URL)을 제공하는 것은 사용자의 책임입니다.



(HLSv3 및 Smooth 재생 목록 전용, 선택사항) 디지털 권한 관리

프리셋 목록에서 컨테이너의 값이 fmp4(Fragmented MP4) 또는 ts(MPEG-TS)인 프리셋을 선택하고 재생 목록의 유형이 HLSv3 또는 Smooth인 경우 파일에 DRM 설정을 지정할 수 있습니다. DRM을 사용하여 파일을 보호할 경우, 이 섹션의 설정을 사용하여 DRM 설정에 대한 정보를 지정합니다.

DRM을 사용하려면 라이선스를 추적 및 부여할 수 있는 라이선스 공급자 서버, 사용자를 인증할 수 있는 웹 서버, DRM으로 파일을 암호화하는 패키지 서버(예: Elastic Transcoder), 미디어 파일을 배포하는 방법(예: Amazon CloudFront), DRM 보호 미디어를 재생하는 방법(예: DRM 지원 플레이어)이 있어야 합니다.

> **Note**
>
> PlayReady DRM, HLS 콘텐츠 보호 및 출력 암호화는 함께 사용할 수 없는 옵션입니다.

(선택 사항) PlayReady DRM

파일을 볼 수 있는 사용자를 제한하는 데 사용되는 DRM 설정. 이를 위해 출력 재생 목록에 PlayReady DRM 헤더를 포함합니다. 아트워크, 캡션, 썸네일 또는 워터마크에는 사용할 수 없습니다. PlayReady DRM은 `aes-ctr` 암호화를 사용하여 미디어 파일을 암호화합니다.

HLSv3 재생 목록에 DRM을 사용할 경우 출력에 마스터 재생 목록이 있어야 합니다.

자세한 내용은 디지털 권한 관리 단원을 참조하십시오.

(DRM 보호의 경우 필수) DRM 형식

출력 재생 목록의 DRM 형식. 유효한 값은 `discretix-3.0` 및 `microsoft`입니다.

Smooth 유형의 재생 목록인 경우 `microsoft`를 지정합니다. HLSv3 유형의 재생 목록인 경우 `discretix-3.0`을 지정합니다.

(DRM 보호의 경우 필수) DRM 키

DRM 라이선스 공급자가 제공하는, 파일의 DRM 키. 이 키는 base64로 인코딩되어야 하며, base64 인코딩 전에 다음 비트 길이 중 하나여야 합니다.

128, 192, 또는 256.

키도를 사용하여 암호화해야 합니다 AWS KMS. 자세한 내용은 AWS Key Management Service 개발자 안내서의 데이터 암호화 및 해독을 참조하세요.

(DRM 보호의 경우 필수) DRM 키 Id

DRM 키의 ID. 따라서 DRM 라이선스 공급자가 어느 키를 공급해야 할지 알 수 있습니다.

키 ID는 빅 엔디안으로 제공되어야 하며, Elastic Transcoder가 이를 리틀 엔디안으로 변환한 후 PlayReady DRM 헤더에 삽입합니다. 라이선스 서버가 키 ID를 빅 엔디안 또는 리틀 엔디안으로 공급하는지 확실치 않은 경우 DRM 공급자에게 확인합니다.

(DRM 보호의 경우 필수) DRM 키 MD5

파일에 대한 DRM에 사용되고 Elastic Transcoder가 전송 도중 키가 손상되지 않았는지 확인하기 위한 체크섬으로 사용할 키의 MD5 다이제스트입니다. 키 MD5는 base64로 인코딩되어야 하며, base64 인코딩 전에 정확히 16바이트여야 합니다.

(DRM 보호의 경우 필수) DRM 초기화 벡터

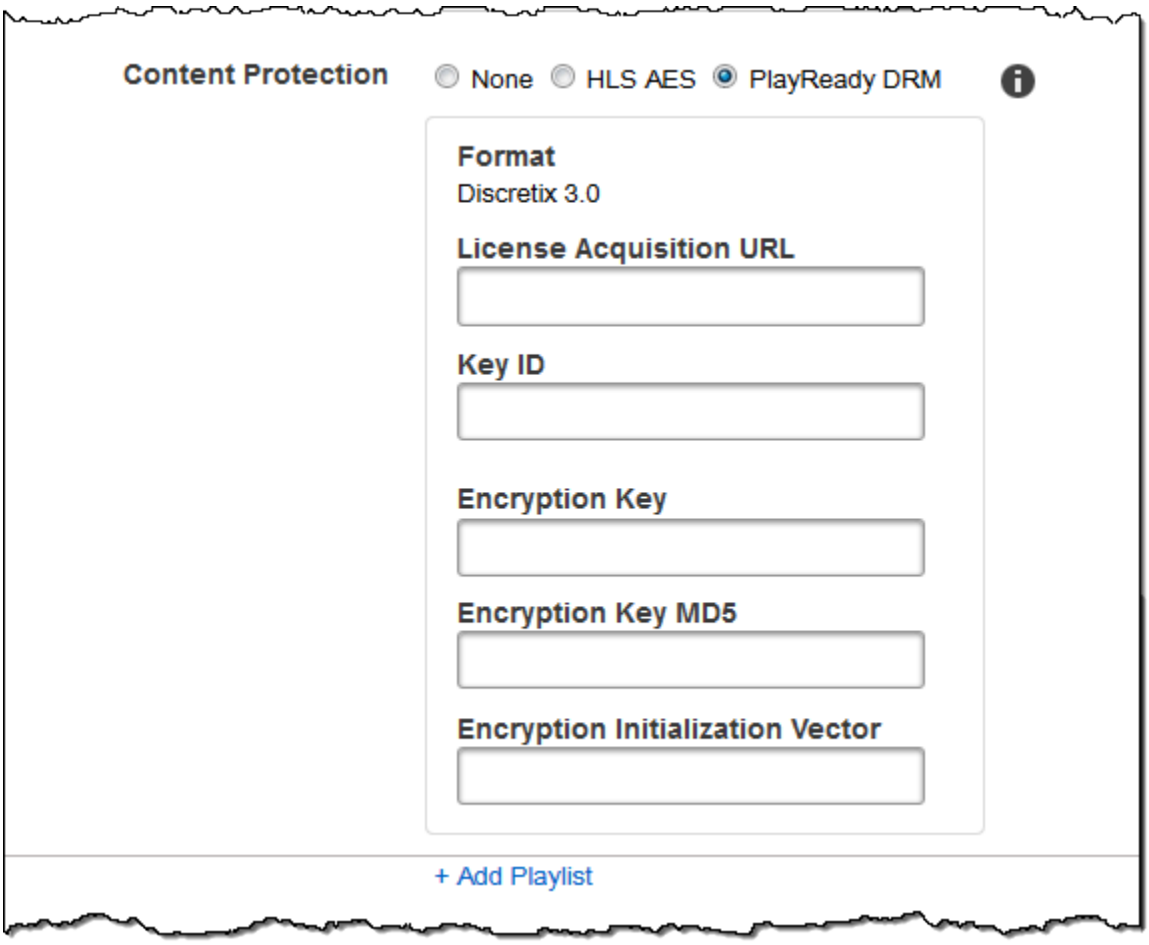
무작위 비트 생성기에 의해 생성되는, 모든 암호화 작업에 고유한 일련의 무작위 비트로, Elastic Transcoder가 파일을 암호화하는 데 사용합니다. 초기화 벡터는 base64로 인코딩되어야 하며, base64 인코딩 전에 정확히 8바이트여야 합니다. 초기화 벡터가 제공되지 않을 경우 Elastic Transcoder가 사용자 대신 생성합니다.

자세한 내용은 Initialization Vector를 참조하세요.

(DRM 보호의 경우 필수) DRM 라이선스 획득 Url

DRM 콘텐츠를 재생하는 데 필요한 라이선스 키의 위치. 이 URL은 절대 경로여야 하며 PlayReady 헤더가 참조해야 합니다. PlayReady 헤더는 Smooth Streaming 출력용 클라이언트 매니페스트의 보호 헤더에서 참조되고 HLS 재생 목록 출력의 EXT-X-DXDRM 및 EXT-XDXDRMINFO 메타데이터 태그에서 참조됩니다. 다음은 URL의 예입니다.

```
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

프리셋 작업

주제

- Elastic Transcoder에서 프리셋 생성
- Elastic Transcoder에서 프리셋 나열 및 프리셋 설정 보기
- Elastic Transcoder에서 프리셋 수정하기
- Elastic Transcoder 프리셋 삭제
- Elastic Transcoder 프리셋을 생성할 때 지정하는 설정

프리셋은 트랜스코딩 프로세스 중에 Elastic Transcoder가 적용할 설정(예: 트랜스코딩된 파일에서 원하는 비디오 해상도 및 오디오 채널 수)이 포함된 템플릿입니다. 작업을 만들 때 사용할 프리셋을 지정합니다. Elastic Transcoder에는 여러 가지 일반적인 출력 형식에 대한 기본 프리셋이 포함되어 있습니다. 다른 출력 형식의 프리셋을 추가할 수도 있습니다. 시스템 프리셋 목록은 시스템 프리셋를 참조하세요.

Elastic Transcoder에서 프리셋 생성

AWS Management Console 또는 Elastic Transcoder 사전 설정 생성 API 작업을 사용하여 사전 설정을 생성할 수 있습니다. 다음 절차에서는 콘솔을 사용하여 프리셋을 생성하는 방법을 설명합니다. API를 사용해 프리셋을 생성하는 방법에 대한 자세한 내용은 프리셋 만들기 섹션을 참조하세요.

> **Note**
>
> 기존 프리셋은 업데이트할 수 없습니다. 프리셋의 설정을 변경해야 하는 경우 변경할 프리셋을 기반으로 새 프리셋을 생성하고, 해당 값을 업데이트한 후, 새 프리셋을 저장합니다.

Elastic Transcoder 콘솔을 사용하여 프리셋을 만들려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/elastictranscoder/ Elastic Transcoder 콘솔을 엽니다.
2. Elastic Transcoder 콘솔의 탐색 모음에서 프리셋을 생성할 리전을 선택합니다.
3. 콘솔의 왼쪽 창에서 Presets를 클릭합니다.
4. Presets 페이지에서 Create New Preset을 클릭합니다.

5. 관련 값들을 입력합니다. 각 필드에 대한 자세한 내용은 Elastic Transcoder 프리셋을 생성할 때 지정하는 설정 섹션을 참조하세요.
6. Create Preset을 클릭합니다.

Elastic Transcoder에서 프리셋 나열 및 프리셋 설정 보기

를 사용하거나 Elastic Transcoder List Presets API를 사용하여 Elastic Transcoder에 포함된 기본 프리셋과 AWS 리전에 추가한 프리셋을 나열 AWS Management Console 할 수 있습니다. 개별 프리셋의 설정을 볼 수도 있습니다. 다음 절차에서는 콘솔을 사용해 프리셋을 나열하는 방법과 프리셋의 설정을 확인하는 방법을 설명합니다.

API를 사용해 프리셋을 나열하는 방법에 대한 자세한 내용은 프리셋 나열을 참조하세요. API를 사용해 프리셋의 설정을 보는 방법에 대한 자세한 내용은 프리셋 읽기를 참조하세요. 시스템 프리셋 목록은 시스템 프리셋를 참조하세요.

Elastic Transcoder 콘솔을 사용하여 프리셋을 나열하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/elastictranscoder/ Elastic Transcoder 콘솔을 엽니다.
2. Elastic Transcoder 콘솔의 탐색 모음에서 프리셋을 나열할 리전을 선택합니다.
3. 콘솔의 왼쪽 창에서 Presets를 클릭합니다.
4. 개별 프리셋의 설정을 표시하려면 해당 프리셋 옆의

   아이콘을 클릭합니다.

Elastic Transcoder에서 프리셋 수정하기

Elastic Transcoder는 기존 프리셋의 설정을 변경하는 것을 허용하지 않습니다. 이는 Elastic Transcoder에 포함된 기본 프리셋과 추가한 프리셋에 모두 적용됩니다. 그러나 다음 절차의 설명대로 변경할 프리셋을 복사하고, 해당 설정을 변경하고, 새 프리셋을 저장한 후, 이전 프리셋을 삭제하여 동일한 결과를 손쉽게 얻을 수 있습니다.

Elastic Transcoder 콘솔을 사용하여 프리셋을 수정하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/elastictranscoder/ Elastic Transcoder 콘솔을 엽니다.

2. Elastic Transcoder 콘솔의 탐색 모음에서 변경할 프리셋이 포함되어 있는 리전을 선택합니다.
3. 콘솔의 왼쪽 창에서 Presets를 클릭합니다.
4. 변경할 프리셋의 확인란을 선택합니다.
5. Copy를 클릭합니다.
6. 변경할 프리셋의 사본에서 해당 값을 변경합니다. 각 필드에 대한 자세한 내용은 Elastic Transcoder 프리셋을 생성할 때 지정하는 설정 섹션을 참조하세요.
7. Create Preset을 클릭합니다.
8. Presets 페이지로 돌아가서 이전 버전의 프리셋의 확인란을 선택합니다.
9. Remove를 클릭합니다.

Elastic Transcoder 프리셋 삭제

AWS Management Console 또는 Elastic Transcoder Delete Preset API를 사용하여 AWS 리전에 추가한 프리셋을 삭제할 수 있습니다. 다음 절차에서는 콘솔을 사용하여 프리셋을 삭제하는 방법을 설명합니다. API를 사용해 프리셋을 삭제하는 방법에 대한 자세한 내용은 프리셋 삭제를 참조하세요.

> **Note**
>
> Elastic Transcoder에 포함된 시스템 프리셋은 삭제할 수 없습니다.

Elastic Transcoder 콘솔을 사용하여 프리셋을 삭제하려면

1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/elastictranscoder/에서 Elastic Transcoder 콘솔을 엽니다.
2. Elastic Transcoder콘솔의 탐색 모음에서 삭제할 프리셋이 포함되어 있는 리전을 선택합니다.
3. 콘솔의 왼쪽 창에서 Presets를 클릭합니다.
4. 삭제할 프리셋의 확인란을 선택합니다.
5. Remove를 클릭합니다.

Elastic Transcoder 프리셋을 생성할 때 지정하는 설정

프리셋을 만들 때 다음 값을 지정합니다.

주제

- 일반 설정
- 비디오 설정
- 워터마크 설정
- 오디오 설정
- 썸네일 설정

일반 설정

프리셋 시작

새 프리셋의 시작점으로 사용할 프리셋입니다. Elastic Transcoder는 선택한 프리셋의 값을 새 프리셋의 필드에 복사합니다. 기본 Elastic Transcoder 프리셋 또는 생성한 프리셋 중 하나를 선택할 수 있습니다.

이름

프리셋의 이름. AWS 계정 내에서 고유한 이름을 사용하는 것이 좋지만 고유성이 적용되지는 않습니다. 최대 길이는 40자입니다.

설명

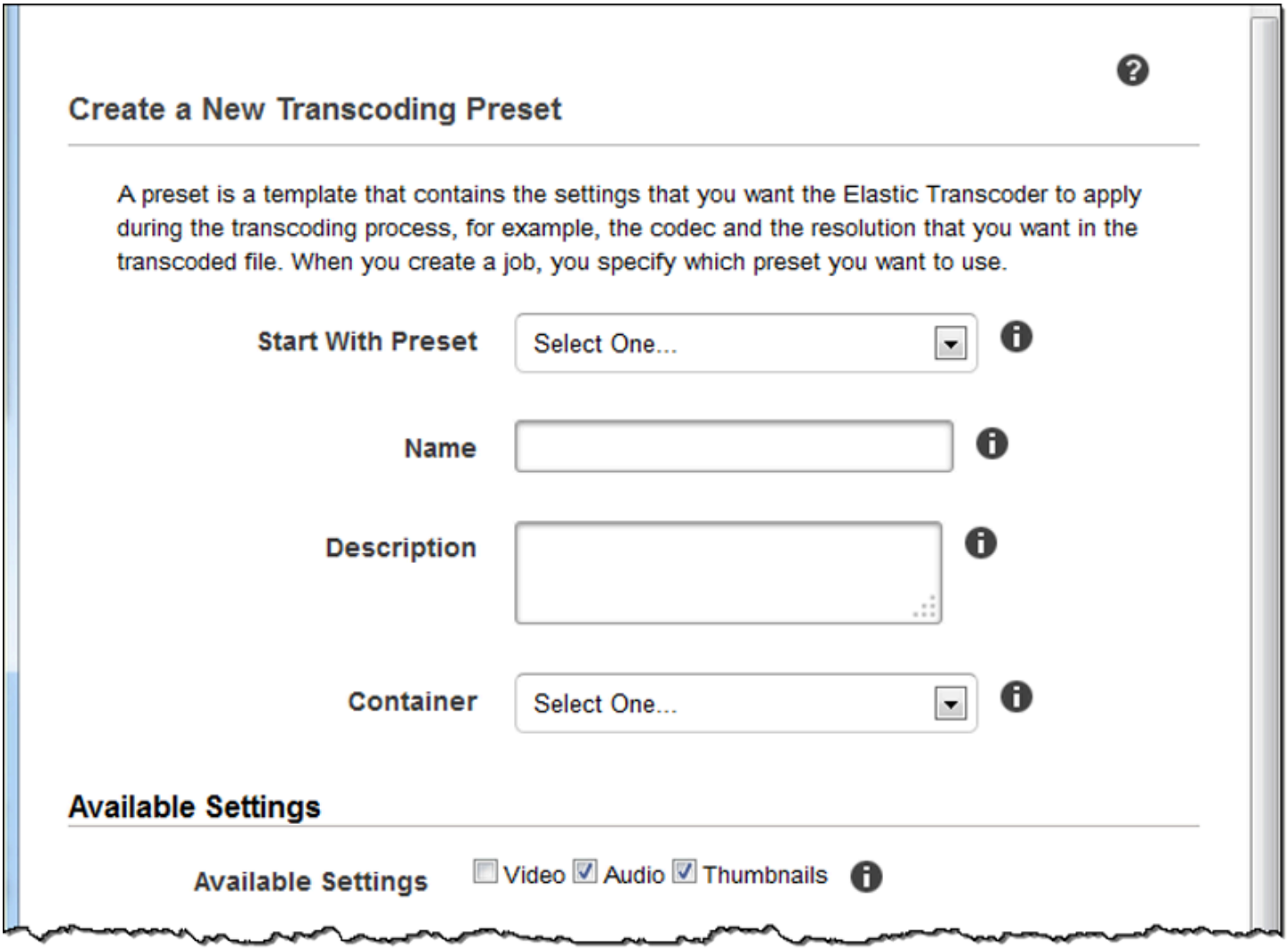
프리셋에 대한 설명. 최대 길이는 255자입니다.

컨테이너

출력 파일의 컨테이너 유형. 유효한 값은 flac, flv, fmp4, gif, mp2, mp3, mp4, mpg, mxf, oga, ogg, ts, wav 및 webm입니다. 다음 표는 컨테이너에서 지원되는 코덱을 보여줍니다.

컨테이너	오디오 코덱	비디오 코덱
FLAC	flac	지원되지 않음
FLV	AAC, mp3	H.264
FMP4	AAC	H.264
GIF	지원되지 않음	gif
MP3	mp3	지원되지 않음
MP4	AAC, mp3	H.264

컨테이너	오디오 코덱	비디오 코덱
MPG	mp2	mpeg2
MXF	pcm	mpeg2
OGA	flac, vorbis	지원되지 않음
OGG	flac, vorbis	지원되지 않음
TS	AAC, mp3	H.264
WAV	pcm	지원되지 않음
WEBM	vorbis	vp8, vp9



비디오 설정

코덱

출력 파일용 비디오 코덱. 유효한 값은 gif, H.264, mpeg2, vp8 및 vp9입니다. 다음 표는 사용 가능한 컨테이너 및 비디오 코덱 조합을 보여줍니다.

컨테이너	GIF	H.264	MPEG2	VP8	VP9
FLAC					
FLV		x			
GIF	x				

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

H.264 비디오 압축 형식에 대한 자세한 내용은 Wikipedia 페이지 H.264/MPEG-4 AVC를 참조하세요.

VP8에 대한 자세한 내용은 VP8을 참조하세요. VP9에 대한 자세한 내용은 VP9를 참조하세요.

(H.264/VP8 전용) 프로필

출력 비디오에 사용할 프로필. 비디오 코덱이 H.264인 경우 Elastic Transcoder는 다음 프로필을 지원합니다.

- baseline: 화상 회의 및 모바일 애플리케이션용으로 가장 자주 사용되는 프로필
- main: 표준 화질 디지털 TV 방송에 사용되는 프로필
- high: 고화질 디지털 TV 방송 및 블루레이 디스크에 사용되는 프로필

H.264 프로필에 대한 자세한 내용은 Wikipedia 항목 "H.264/MPEG-4 AVC"에서 Profiles를 참조하세요.

비디오 코덱이 VP8인 경우 Elastic Transcoder는 값 0, 1, 2, 3을 지원합니다.

(H.264) 레벨

출력 비디오에 사용할 H.264 레벨. 해당 값을 선택합니다.

레벨에 대한 자세한 내용은 Wikipedia 항목 "H.264/MPEG-4 AVC"에서 Levels를 참조하세요.

(H.264 전용) 최대 참조 프레임 수

이후 프레임 디코딩을 위한 참조로 사용할 이전에 디코딩된 프레임의 최대 수. Max Height, Max Height, Level에 지정된 값에 기반한 권장 값보다 큰 값을 입력하면, Elastic Transcoder에 권장 값이 포함된 메시지가 표시됩니다. Elastic Transcoder가 수행하는 계산 및 자세한 정보는 프리셋 만들기 주제의 MaxReferenceFrames를 참조하세요.

(선택 사항, H.264/MPEG2/VP8/VP9 전용) 최대 비트 전송률

출력 비디오의 초당 최대 킬로비트 수. 16~62,500 범위의 값을 지정합니다.

비트 전송률로 auto를 지정하는 경우 Elastic Transcoder는 입력 비디오의 비트 전송률을 출력 비디오의 평균 비트 전송 속도로 사용합니다. Maximum Bit Rate를 통해 출력 비디오의 비트 전송률 상한을 설정할 수 있습니다. 이는 대상 디바이스가 지원하는 최대 비트 전송률가 입력 비디오의 비트 전송률보다 낮을 경우 유용합니다. 최대 비트 전송률를 낮추면 비디오 품질이 저하될 수 있습니다.

(선택 사항, H.264/MPEG2/VP8/VP9 전용) 버퍼 크기

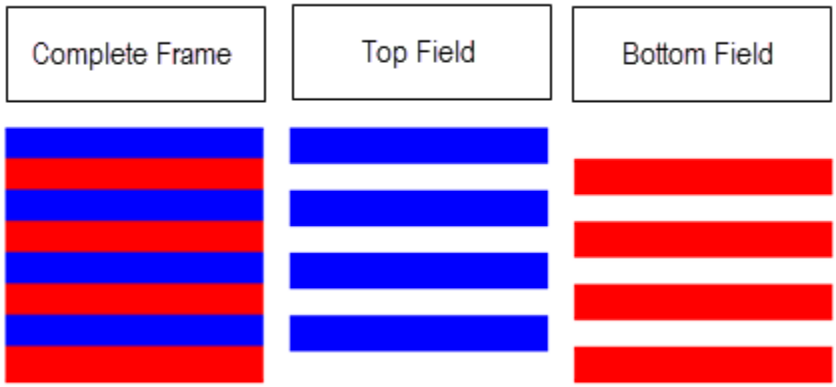
출력 비디오 중 임의의 x초에서의 최대 킬로비트 수. 이 기간은 일반적으로 10초입니다. 이는 출력 비디오의 컨테이너 유형으로 MPEG-TS를 사용하는 경우 표준적 세그먼트 지속시간입니다. 0보다 큰 정수를 지정합니다. 최대 비트 전송률을 지정하고 버퍼 크기를 생략하면, Elastic Transcoder는 버퍼 크기를 최대 비트 전송률 값의 10배로 설정합니다.

(선택 사항, H.264/MPEG2 전용) 인터레이스 모드

출력 비디오용 인터레이스 모드.

인터레이스 방식은 사람의 눈이 프레임당 복수의 화상을 인식하도록 두 필드(홀수 선과 짝수 선에 필드 하나씩)를 인터레이스하여 지각된 비디오 프레임 속도를 두 배로 높이는 데 사용됩니다. 인터레이싱은 비디오를 전송하는 데 필요한 대역폭을 줄여주지만 흐릿한 이미지와 깜박임이 발생할 수 있습니다.

두 개의 선 집합을 필드라고 하며, 인터레이스 방식 프레임은 필드에 따라 두 개의 이미지로 분할됩니다.



유효한 값은 Progressive(프로그레시브)(비 인터레이싱, 상위에서 하위로), Top First(상위 필드 우선), Bottom First(하위 필드 우선) 및 Auto(자동)입니다.

InterlaceMode를 지정하지 않을 경우 Elastic Transcoder는 출력에 Progressive를 사용합니다. Auto를 지정할 경우 Elastic Transcoder는 출력을 인터레이스합니다.

자세한 내용은 Wikipedia 페이지 Interlaced video를 참조하세요.

(선택 사항, H.264/MPEG2 전용) 컬러 스페이스 변환 모드

색상 공간 변환 Elastic Transcoder가 출력 비디오에 적용됩니다. 색상 공간은 컴퓨터가 색상 렌더링 방법에 대한 정보를 저장하는 데 사용하는 알고리즘입니다. Bt.601은 표준 화질 비디오용 표준이고, Bt.709는 고화질 비디오용 표준입니다.

유효한 값은 없음, Bt709toBt601, Bt601toBt709 및 자동입니다.

ColorSpaceConversionMode에 대해 Auto를 선택하고, 출력이 인터레이스 방식이고, 프레임 속도가 23.97, 24, 25, 29.97, 50, 60,중 하나이고, SegmentDuration이 null이고, 아래 그래프의 해상도 변경 중 하나를 사용하는 경우, Elastic Transcoder는 다음 색상 공간 변환을 적용합니다.

입력 파일	입력 해상도	출력 해상도	적용된 색상 공간 변환
표준에서 HD로	720x480	1920x1080	Bt601ToBt709
표준에서 HD로	720x576	1920x1080	Bt601ToBt709
HD에서 표준으로	1920x1080	720x576	Bt709ToBt601

입력 파일	입력 해상도	출력 해상도	적용된 색상 공간 변환
HD에서 표준으로	1920 x 1080	720x480	Bt709ToBt601

> **Note**
>
> Elastic Transcoder가 이후의 ColorspaceConversionMode Auto 모드의 동작을 변경할 수 있습니다. 재생 목록의 모든 출력은 동일한 ColorSpaceConversionMode를 사용해야 합니다.

ColorspaceConversionMode를 지정하지 않을 경우 Elastic Transcoder는 파일의 색상 공간을 변경하지 않습니다.

출력 파일에 어떤 ColorSpaceConversionMode가 적용되었는지 확실치 않은 경우 작업 응답에 포함된 AppliedColorSpaceConversion 파라미터를 확인할 수 있습니다. 작업 응답에 AppliedColorSpaceConversion이 포함되지 않은 경우 어떤 ColorSpaceConversionMode도 적용되지 않은 것입니다.

색상 공간에 대한 자세한 내용은 Wikipedia 페이지 Color space를 참조하세요. Bt.601 및 Bt.709에 대한 자세한 내용은 Wikipedia 페이지 Rec. 601과 Rec. 709.

(MPEG2 전용) 크로마 서브샘플링

출력 비디오의 크로마(색상) 채널에 대한 샘플링 패턴. 유효한 값은 yuv420p와 yuv422p입니다.

yuv420p는 한 줄 건너 수평선과 한 줄 건너 수직선에 대한 크로마 정보를 샘플링하고, yuv422p는 모든 수평선과 한 줄 건너 수직선의 색상 정보를 샘플링합니다.

크로마 하위 샘플링에 대해 자세히 알아보려면 Wikipedia 페이지 Chroma subsampling을 참조하세요.

(Gif 전용) 루프 횟수

출력 gif가 루프할 횟수.

유효한 값은 Infinite(무제한) 및 0~100 사이의 정수입니다.

(H.264/MPEG2/VP8 전용) 키프레임 간 최대 프레임 수

키 프레임 간의 최대 프레임 수. 컨테이너 유형 gif에는 해당되지 않습니다. 키 프레임은 완전히 인코딩된 프레임이며, 키 프레임 간 프레임은 부분적으로 키 프레임의 콘텐츠에 기반하여 인코딩됩니다. 이 값은 문자열 형식 정수이며, 유효한 값은 1(모든 프레임이 키 프레임)~100000 사이입니다. 이보다 값이 높으면 압축률이 증가하지만 비디오 품질이 눈에 띄게 저하될 수 있습니다.

Smooth 출력의 경우 Frame Rate는 Maximum Number of Frames Between Keyframes에 대해 일정한 비율을 유지해야 합니다. 그러면 Smooth 재생 목록이 파일 재생 도중 서로 다른 품질 수준 사이를 전환할 수 있습니다.

예를 들어 입력 파일의 Frame Rate가 30이고 Maximum Number of Frames Between Keyframes가 90일 수 있습니다. 그러면 출력 파일은 1:3의 비율을 가져야 합니다. 유효한 출력은 Frame Rates가 30, 25, 10이고, Maximum Number of Frames Between Keyframes가 각각 90, 75, 30일 것입니다.

또는 Frame Rate를 auto로 설정하고 Video Max Frame Rate 및 Maximum Number of Frames Between Keyframes에 동일한 값을 지정해 이를 구현할 수도 있습니다.

키 프레임에 대한 자세한 내용은 Wikipedia 항목 Video compression picture types를 참조하세요.

(H.264/MPEG2/VP8 전용) 키프레임 간 고정 프레임 수

Fixed Number of Frames Between Keyframes에 고정 값을 사용할지 여부:

- Yes: Elastic Transcoder는 키 프레임 간격(화상 그룹(GOP) 내 프레임 수)으로 Maximum Number of Frames Between Keyframes 값을 사용합니다.
- No: 프레임 간격이 변동할 수 있습니다.

> **Note**
>
> fmp4 컨테이너의 경우 키프레임 간 고정 프레임 수를 `true`로 설정해야 합니다.

비트 전송률

출력 비디오 내 비디오 스트림의 비트 전송률(KB/s). 가변 비트 전송률 또는 고정 비트 전송률 인코딩을 구성할 수 있습니다.

- 가변 비트 전송률 인코딩: auto를 지정합니다. Elastic Transcoder는 비트 전송률을 최적화하고 출력의 각 프레임에 대해 일관된 품질을 유지합니다.
- Constant bit rate encoding(고정 비트 레이트 인코딩): 비트 레이트를 지정합니다.

비디오 비트 전송률의 유효한 값은 Codec에 선택한 값에 따라 달라집니다.

- H.264: 유효한 값은 Level 및 Profile의 값에 따라 달라집니다. 레벨 및 프로필에 다음 표에 나열된 최대 H.264 호환 값보다 작거나 같은 값을 지정하는 것이 좋습니다.
- VP8: 다음 표를 사용하지 마세요. Level은 비디오 코덱이 H.264인 경우에만 적용됩니다.

Level	최대 비디오 비트 전송률(KB/s): Baseline Profile 및 Main Profile	최대 비디오 비트 전송률(KB/s): High Profile
1	64	80
1b	128	160
1.1	192	240
1.2	384	480
1.3	768	960
2	2000	2500
2.1	4000	5000
2.2	4000	5000
3	10000	12500
3.1	14000	17500
3.2	20000	25000
4	20000	25000
4.1	50000	62500

프레임 속도

출력 비디오 내 비디오 스트림의 초당 프레임. 해당 값을 선택합니다.

auto를 지정하면 Elastic Transcoder가 입력 소스의 감지된 프레임 속도를 사용합니다. 프레임 속도를 지정하는 경우 다음 계산을 수행하는 것이 좋습니다.

프레임 속도 = 최대 권장 디코딩 속도(luma 샘플/초) / (비디오 최대 너비) * (비디오 최대 높이)

여기서 최대 권장 디코딩 속도(luma 샘플/초)는 Level에 지정된 값을 기준으로 다음 표에 나열된 최대값보다 작거나 같습니다.

> **Note**
>
> Max Width, Max Height, Level에 지정된 값에 기반한 권장 값보다 큰 값을 입력하면, Elastic Transcoder에 권장 값이 포함된 메시지가 표시됩니다.

Smooth 출력의 경우 Frame Rate는 Maximum Number of Frames Between Keyframes에 대해 일정한 비율을 유지해야 합니다. 예를 들어 Frame Rate가 30이고 Maximum Number of Frames Between Keyframes가 90인 입력 파일을 지정하면, 출력의 Frame Rate와 Maximum Number of Frames Between Keyframes의 비율도 1:3이어야 합니다.

Level	최대 권장 디코딩 속도(Luma 샘플/초)
1	380160
1b	380160
1.1	768000
1.2	1536000
1.3	3041280
2	3041280
2.1	5068800
2.2	5184000
3	10368000
3.1	27648000
3.2	55296000
4	62914560

Level	최대 권장 디코딩 속도(Luma 샘플/초)
4.1	62914560

비디오 최대 프레임 속도

프레임 속도를 auto로 지정할 경우 Elastic Transcoder는 입력 비디오의 프레임 속도를 출력 비디오의 프레임 속도로 사용합니다(최대 프레임 속도까지). 비디오 최대 프레임 속도를 지정하지 않을 경우 Elastic Transcoder는 기본값 30을 사용합니다.

입력 비디오의 프레임 속도가 출력 비디오의 원하는 프레임 속도 또는 기본 최대 프레임 속도보다 클 경우 Elastic Transcoder가 사용할 최대 프레임 속도를 지정합니다.

Elastic Transcoder가 다음 두 기준을 충족하는, 지원되는 가장 높은 프레임 속도를 사용합니다.

- 프레임 속도가 최대 프레임 속도보다 낮거나 같습니다.
- 프레임 속도가 입력 프레임 속도로 나누어 떨어집니다.

예를 들어 프레임 속도가 50인 입력 파일이 있고 비디오 최대 프레임 속도 값을 30으로 지정한 경우 Elastic Transcoder는 프레임 속도가 25fps인 출력 비디오를 생성합니다. 25가 30보다 작고 50은 25로 나누어 떨어지기 때문입니다.

최대 너비

출력 비디오의 최대 너비(픽셀). auto를 지정할 경우 Elastic Transcoder는 1920(Full HD)을 기본값으로 사용합니다. 숫자 값을 지정하려면 128~4096 범위의 정수를 입력합니다.

자세한 내용은 Max Height 단원을 참조하십시오.

최대 높이

출력 비디오의 최대 높이(픽셀). auto를 지정할 경우 Elastic Transcoder는 1080(Full HD)을 기본값으로 사용합니다. 숫자 값을 지정하려면 96~3072 범위의 정수를 입력합니다.

Codec에 H.264를 지정한 경우 최대 너비 및 최대 높이 값의 곱이 다음 표의 해당 값보다 작거나 같도록 두 값을 지정하는 것이 좋습니다.

> **Note**
>
> Codec에 VP8 을 지정한 경우 다음 표를 사용하지 마세요. Level은 비디오 코덱이 H.264일 경우에만 적용됩니다.

Level 값	최대 Max Width x Max Height값
1	25344
1b	25344
1.1	101376
1.2	101376
1.3	101376
2	101376
2.1	202752
2.2	404720
3	404720
3.1	921600
3.2	1310720
4	2097152
4.1	2097152

크기 조정 정책

다음 값 중 하나를 지정하여 출력 비디오의 조정을 제어합니다.

- Fit: Elastic Transcoder가 최대 너비 또는 최대 높이에 지정된 값과 일치하되 다른 값은 초과하지 않도록 출력 비디오를 확대 또는 축소합니다.
- Fill: Elastic Transcoder가 최대 너비 또는 최대 높이에 지정된 값과 일치하되 다른 값과 일치하거나 초과하도록 출력 비디오의 크기를 조정합니다. Elastic Transcoder는 출력 비디오를 중앙에 배치한 다음 최대값을 초과하는 크기(있는 경우)에서 잘라냅니다.
- Stretch: Elastic Transcoder가 최대 너비 및 최대 높이에 지정된 값과 일치하도록 출력 비디오를 늘립니다. 입력 비디오의 상대 비율과 출력 비디오의 상대 비율이 다를 경우 출력 비디오가 변형됩니다.

- Keep: Elastic Transcoder가 출력 비디오를 확대 또는 축소하지 않습니다. 입력 비디오의 치수가 최대 너비 및 최대 높이에 지정된 값을 초과하는 경우 Elastic Transcoder는 출력 비디오를 자릅니다.
- ShrinkToFit: Elastic Transcoder가 출력 비디오의 치수가 최대 너비 및 최대 높이 중 하나 이상에 지정된 값과 일치하되 어떤 값도 초과하지 않도록 출력 비디오를 축소합니다. 이 옵션을 지정할 경우 Elastic Transcoder는 비디오를 확대하지 않습니다.
- ShrinkToFill: Elastic Transcoder가 출력 비디오의 치수가 최대 너비 및 최대 높이 중 하나 이상에 지정된 값과 일치하되 어떤 값도 하회하지 않도록 출력 비디오를 축소합니다. 이 옵션을 지정할 경우 Elastic Transcoder는 비디오를 확대하지 않습니다.

다음 표는 출력 비디오에 대한 Sizing Policy 설정의 효과를 보여줍니다.

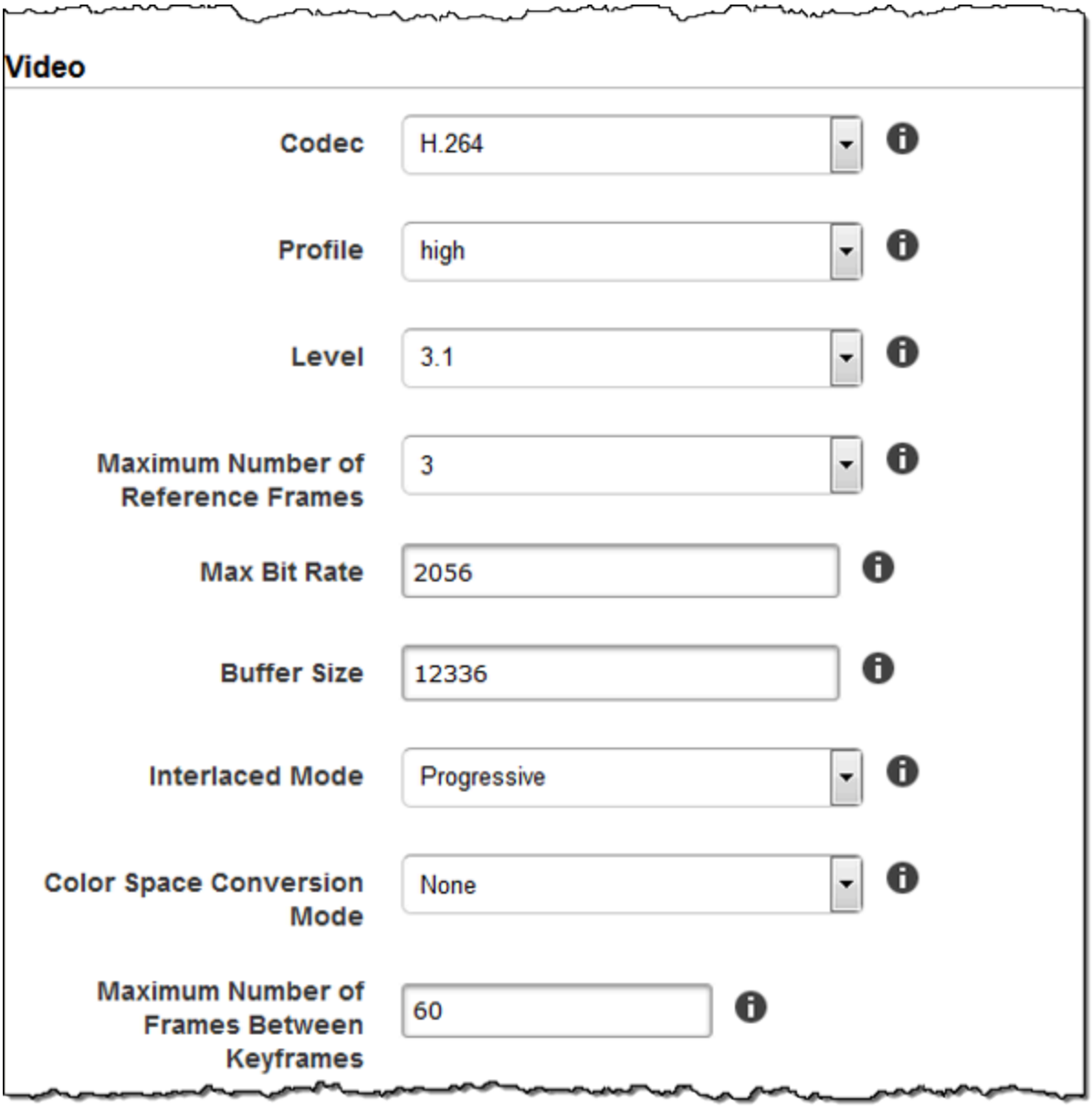
SizingPolicy	출력 비디오가 확대될 수 있음	패딩 정책이 "Pad"일 경우 출력 비디오가 패딩될 수 있음	출력 비디오의 픽셀 종횡비가 입력 비디오와 다를 수 있음	출력 비디오가 잘릴 수 있음
Fit	예	예		
Fill	예			예
Stretch	예		예	
Keep	예	예		예
ShrinkToFit		예		
ShrinkToFill		예		예

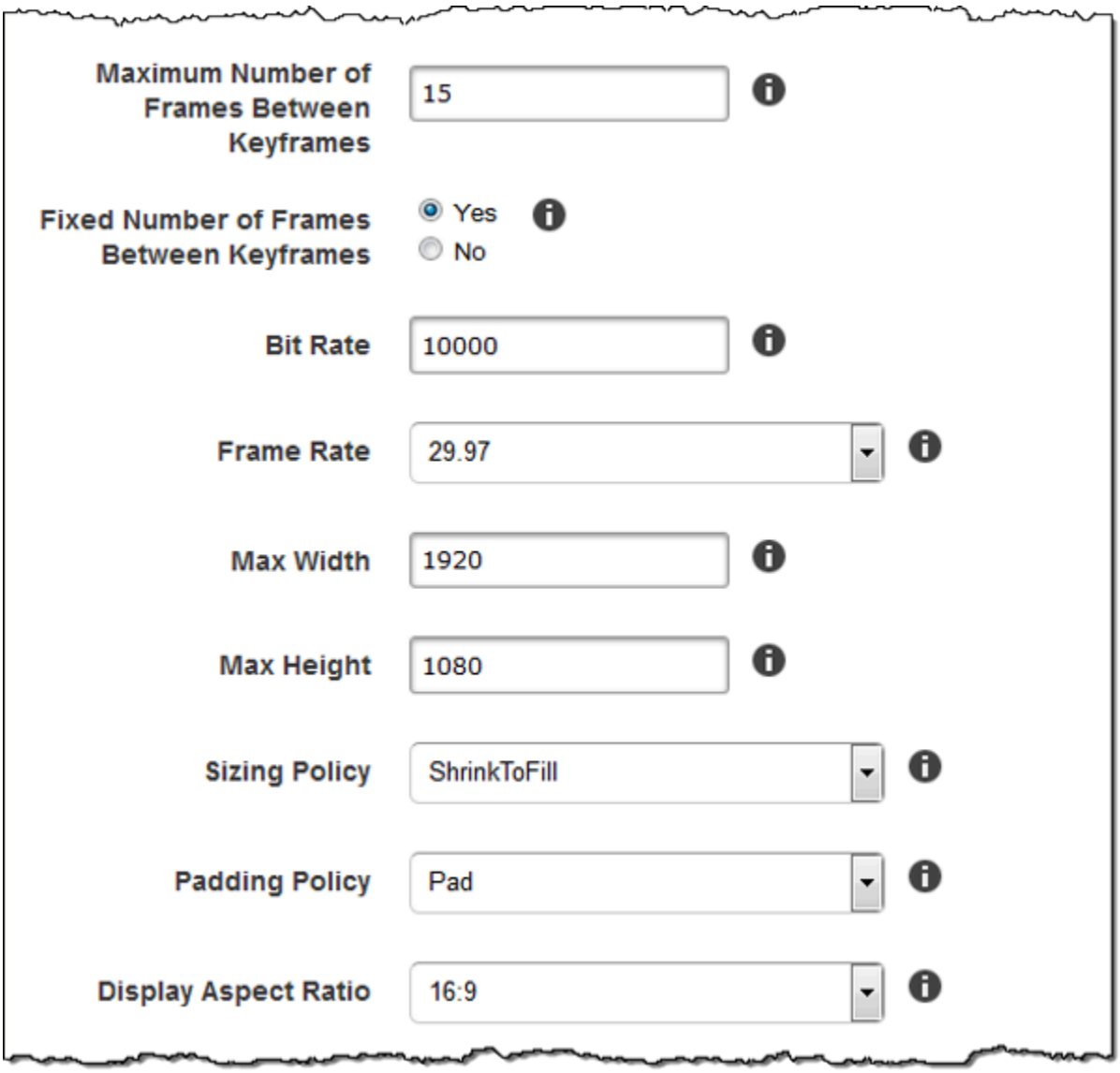
패딩 정책

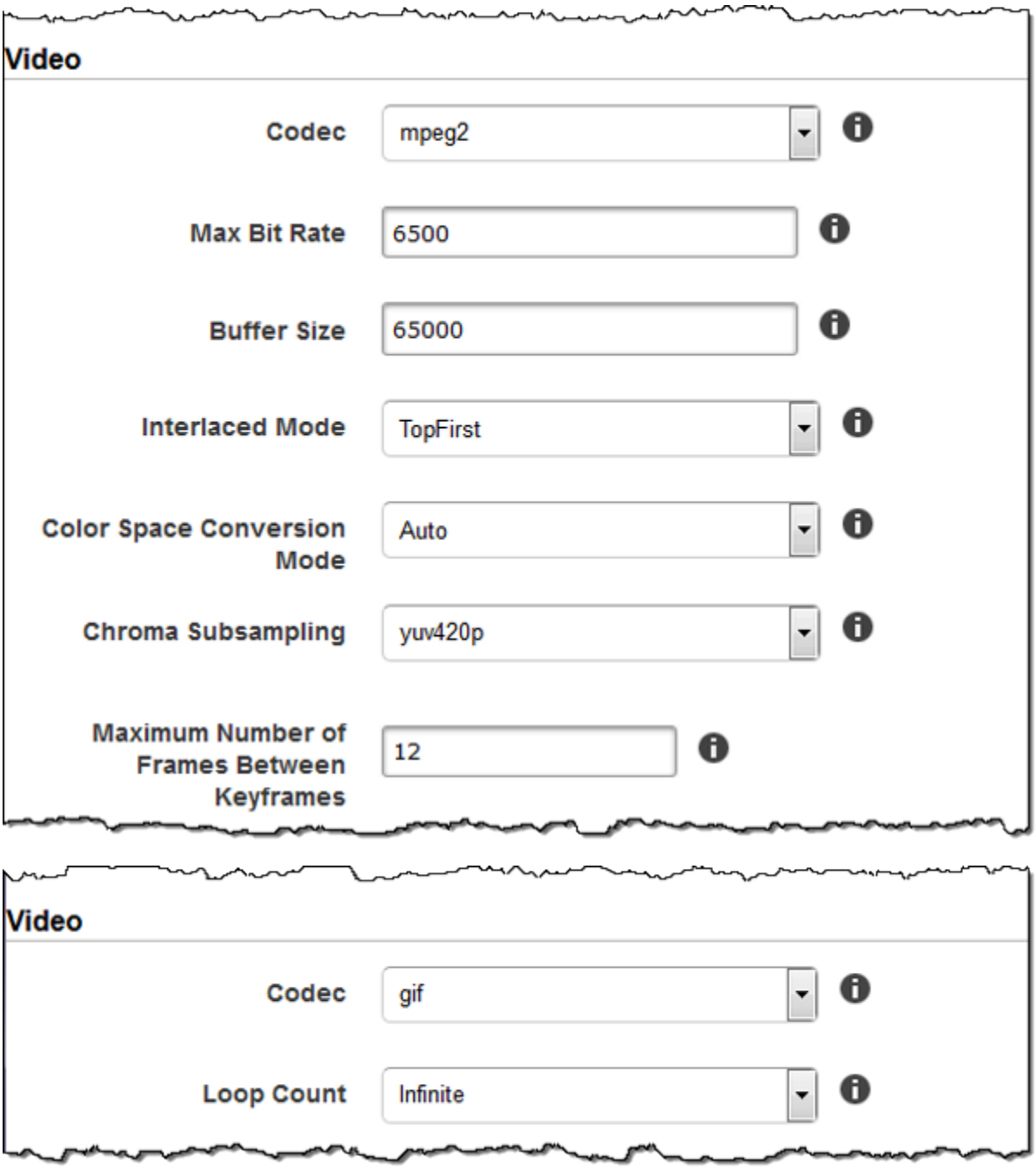
PaddingPolicy를 Pad로 설정하면 Elastic Transcoder가 출력 비디오의 전체 크기를 최대 너비 및 최대 높이에 지정된 값과 일치시키기 위해 출력 비디오의 상하 및/또는 좌우에 블랙 바를 추가할 수 있습니다. 자세한 내용은 Sizing Policy의 표를 참조하세요.

디스플레이 종횡비

Elastic Transcoder가 출력 비디오의 메타데이터에 추가하는 값입니다. 디스플레이 종횡비를 auto로 설정할 경우 Elastic Transcoder는 정사각형 픽셀을 보장하는 종횡비를 선택합니다. 다른 옵션을 지정할 경우 Elastic Transcoder가 출력 비디오에서 해당 값을 설정합니다.




Maximum Number of Frames Between Keyframes
15
Fixed Number of Frames Between Keyframes
Yes
No
Bit Rate
10000
Frame Rate
29.97
Max Width
1920
Max Height
1080
Sizing Policy
ShrinkToFill
Padding Policy
Pad
Display Aspect Ratio
16:9



워터마크 설정

이 섹션의 설정을 사용하면 Elastic Transcoder가 이 프리셋을 사용하여 트랜스코딩된 비디오에 오버레이할 그래픽의 크기, 위치, 축척 및 불투명도를 지정할 수 있습니다. 최대 4개의 워터마크에 대한 설정을 지정할 수 있습니다. 워터마크는 트랜스코딩된 비디오의 지속시간 동안 표시됩니다.

워터마크는 .png 또는 .jpg 형식일 수 있습니다. 장방형이 아닌 워터마크를 표시하려면 투명도를 지원하는 .jpg 형식을 사용합니다.

이 프리셋을 사용하는 작업을 생성할 때는 Elastic Transcoder가 트랜스코딩된 비디오에 포함할 .png 또는 .jpg 그래픽을 지정합니다. Elastic Transcoder는 해당 프리셋에서 지정한 만큼 각 작업 출력에 워터마크를 많이 지정할 필요가 없습니다. 예를 들어 프리셋에서는 4개의 워터마크에 대한 설정을 지정하고 작업 출력에서는 하나의 워터마크만 지정할 수 있습니다.

그래픽이 변형되지 않도록 워터마크 설정을 구성하려면 Sizing Policy 값을 Shrink to Fit로 설정하고, Maximum Width 및 Maximum Height 값을 동일한 백분율로 설정합니다. 그래픽을 원본과 동일한 크기로 표시하려는 경우 최대 너비 및 최대 높이를 100%로 설정합니다.

자세한 내용은 워터마크 단원을 참조하십시오.

id

단일 워터마크에 대한 설정의 고유한 식별자. Id 값은 최대 40자입니다.

최대 너비

다음 중 한 형식의 워터마크 최대 너비:

- *number of pixels* px: 최솟값은 16픽셀이며, 최댓값은 MaxHeight의 값입니다.
- *integer percentage* %: 유효한 값의 범위는 0~100입니다. 대상 값을 사용하여 Elastic Transcoder가 계산 시 Elastic Transcoder에 의해 추가된 블랙 바(있을 경우)를 포함할지 여부를 지정합니다.

값을 픽셀로 지정할 경우 MaxHeight의 값보다 작거나 같아야 합니다.

최대 높이

다음 중 한 형식의 워터마크 최대 높이:

- *number of pixels* px: 최솟값은 16픽셀이며, 최댓값은 MaxHeight의 값입니다.
- *integer percentage* %: 유효한 값의 범위는 0~100입니다. 대상 값을 사용하여 Elastic Transcoder가 계산 시 Elastic Transcoder에 의해 추가된 블랙 바(있을 경우)를 포함할지 여부를 지정합니다.

값을 픽셀로 지정할 경우 MaxHeight의 값보다 작거나 같아야 합니다.

크기 조정 정책

워터마크의 확대/축소를 제어하는 값:

- Fit: Elastic Transcoder가 최대 너비 또는 최대 높이에 지정된 값과 일치하되 다른 값은 초과하지 않도록 워터마크를 확대 또는 축소합니다.

- Stretch: Elastic Transcoder가 최대 너비 및 최대 높이에 지정된 값과 일치하도록 워터마크를 늘립니다. 워터마크의 상대 비율과 Maximum Width 및 Maximum Height의 값이 다를 경우 워터마크가 변형됩니다.
- Shrink to Fit: Elastic Transcoder가 워터마크의 치수가 최대 너비 및 최대 높이 중 하나 이상에 지정된 값과 일치하되 어떤 값도 초과하지 않도록 워터마크를 축소합니다. 이 옵션을 지정할 경우 Elastic Transcoder는 워터마크를 확대하지 않습니다.

Horizontal Alignment(가로 맞춤)

워터마크의 수평 위치. 좌측 또는 우측 경계를 기준으로 워터마크를 배치하려면 또한 Horizontal Offset에 0이 아닌 값을 지정합니다.

- Left: 워터마크의 좌측 가장자리가 비디오의 좌측 경계와 정렬됩니다.
- Right: 워터마크의 우측 가장자리가 비디오의 우측 경계와 정렬됩니다.
- Center: 워터마크가 좌측 경계와 우측 경계 사이의 중앙에 배치됩니다.

수평 오프셋

워터마크의 수평 위치를 Horizontal Alignment에 지정된 위치로부터 오프셋할 거리:

- *number of pixels* px: 최솟값은 0픽셀이며, 최댓값은 MaxHeight의 값입니다.
- *integer percentage* %: 유효한 값의 범위는 0~100입니다.

예를 들어 가로 맞춤에 Left를 설정하고 수평 오프셋에 5px를 설정할 경우 워터마크의 좌측이 출력 비디오의 좌측 경계로부터 5픽셀에 표시됩니다.

HorizontalOffset은 가로 맞춤 값이 Left 또는 Right인 경우에만 유효합니다.

워터마크가 오프셋 설정 때문에 좌측 또는 우측 경계를 넘어서지만 Elastic Transcoder가 블랙 바를 추가하지 않은 경우 워터마크는 잘립니다. Elastic Transcoder가 블랙 바를 추가했다면 워터마크는 블랙 바까지 확장됩니다. 워터마크가 블랙 바를 넘어설 경우 워터마크는 잘립니다.

대상 값을 사용하여 Elastic Transcoder가 오프셋 계산 시 Elastic Transcoder에 의해 추가된 블랙 바(있을 경우)를 포함할지 여부를 지정합니다.

세로 맞춤

워터마크의 수직 위치. 상단 또는 하단 경계를 기준으로 워터마크를 배치하려면 또한 Vertical Offset에 0이 아닌 값을 지정합니다.

- Top: 워터마크의 상단 가장자리가 비디오의 상단 경계와 정렬됩니다.
- Bottom: 워터마크의 하단 가장자리가 비디오의 하단 경계와 정렬됩니다.
- Center: 워터마크가 상단 경계와 하단 경계 사이의 중앙에 배치됩니다.

수직 오프셋

워터마크의 수직 위치를 Vertical Alignment에 지정된 위치로부터 오프셋할 거리:

- *number of pixels* px: 최소값은 0픽셀이며, 최대값은 Maximum Height의 값입니다.
- *integer percentage* %: 유효한 값의 범위는 0~100입니다.

예를 들어 세로 맞춤에 Top을 설정하고 수직 오프셋에 5px를 설정할 경우 워터마크의 상단이 출력 비디오의 상단 경계로부터 5픽셀에 표시됩니다.

수직 오프셋은 세로 맞춤 값이 Top 또는 Bottom인 경우에만 유효합니다.

워터마크가 오프셋 설정 때문에 상단 또는 하단 경계를 넘어서지만 Elastic Transcoder가 블랙 바를 추가하지 않은 경우 워터마크는 잘립니다. Elastic Transcoder가 블랙 바를 추가했다면 워터마크는 블랙 바까지 확장됩니다. 워터마크가 블랙 바를 넘어설 경우 워터마크는 잘립니다.

대상 값을 사용하여 Elastic Transcoder가 오프셋 계산 시 Elastic Transcoder에 의해 추가된 블랙 바(있을 경우)를 포함할지 여부를 지정합니다.

Opacity(불투명도)

워터마크가 표시 위치에서 비디오를 가리는 비율(%). 유효한 값은 0(워터마크가 보이지 않음)~100(워터마크가 지정된 위치에서 비디오를 완전히 가림) 사이입니다.

Elastic Transcoder는 투명한 `.png` 그래픽을 지원합니다. 투명 `.png`를 사용하는 경우 비디오의 투명한 부분이 Opacity 값을 0으로 설정한 것처럼 나타납니다. `.jpg` 파일 형식은 투명도를 지원하지 않습니다.

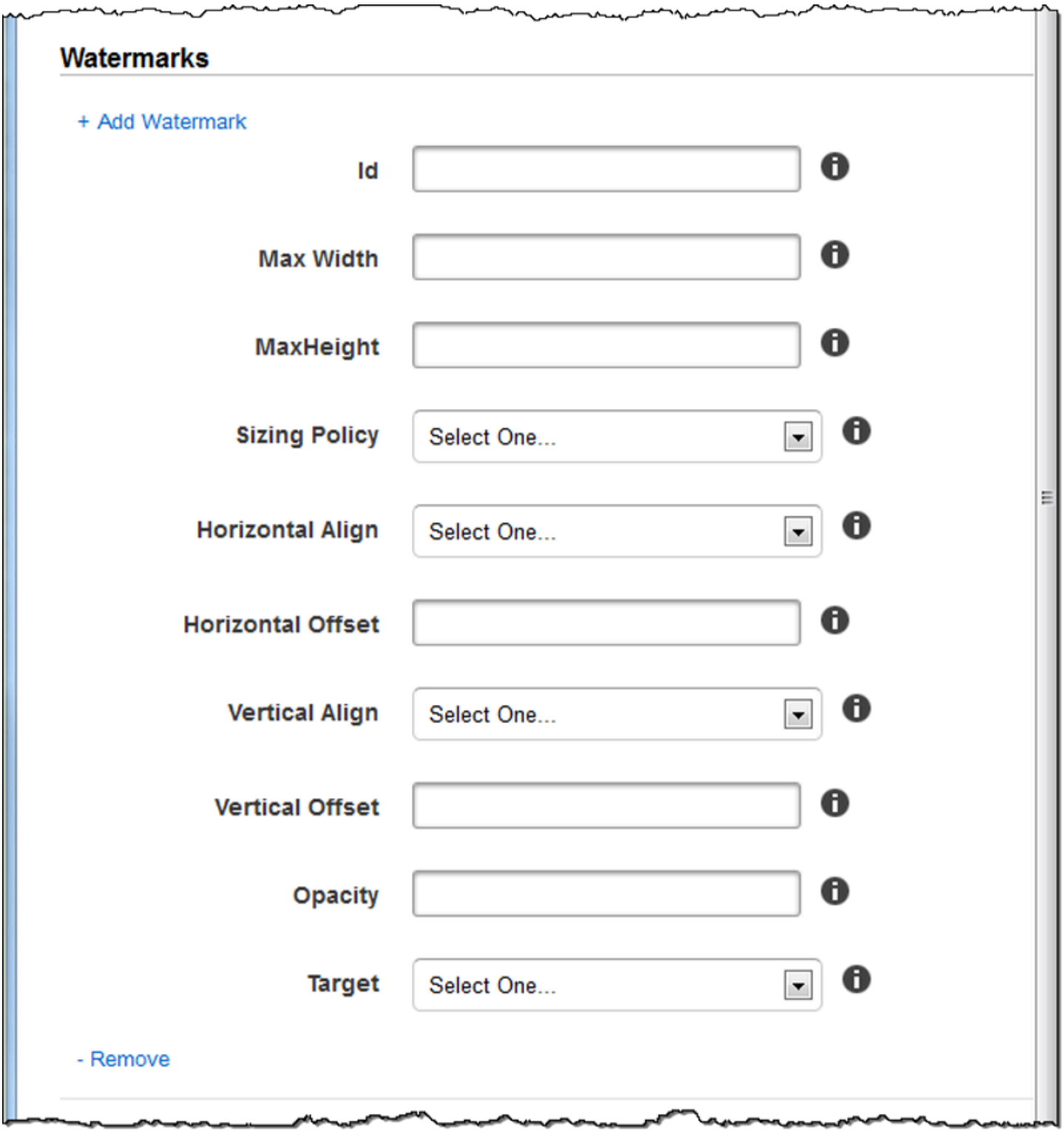
대상

Elastic Transcoder가 워터마크 설정 수평 오프셋, 수직 오프셋, 최대 너비, 최대 높이에 지정된 값을 해석하는 방법을 결정하는 값입니다.

- Content: 수평 오프셋 및 수직 오프셋 값은 Elastic Transcoder에서 추가한 블랙 바(있을 경우)를 제외한 비디오의 테두리를 기준으로 계산됩니다.

  또한 최대 너비 및 최대 높이가 백분율로 지정된 경우 Elastic Transcoder에서 추가한 블랙 바(있을 경우)를 제외한 비디오의 테두리를 기준으로 계산됩니다.
- Frame: 수평 오프셋 및 수직 오프셋 값은 Elastic Transcoder에서 추가한 블랙 바(있을 경우)를 포함하여 비디오의 테두리를 기준으로 계산됩니다.

  또한 최대 너비 및 최대 높이가 백분율로 지정된 경우 Elastic Transcoder에서 추가한 블랙 바(있을 경우)를 포함하여 비디오의 테두리를 기준으로 계산됩니다.


Watermarks
+ Add Watermark
Id
Max Width
MaxHeight
Sizing Policy
Select One...
Horizontal Align
Select One...
Horizontal Offset
Vertical Align
Select One...
Vertical Offset
Opacity
Target
Select One...
- Remove

오디오 설정

코덱

출력 파일용 오디오 코덱. 유효한 값은 `AAC`, `flac`, `mp2`, `mp3`, `pcm` 및 `vorbis`입니다. 다음 표는 사용 가능한 컨테이너 및 오디오 코덱 조합을 보여줍니다.

컨테이너	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 전용) 프로필

Audio:Codec에 AAC를 지정한 경우 출력 파일의 AAC 프로파일을 선택합니다. Elastic Transcoder는 다음과 같은 프로필을 지원합니다.

- auto: auto를 지정할 경우 Elastic Transcoder는 출력 파일에 선택된 비트 전송률을 기준으로 프로필을 선택합니다.
- AAC-LC: 가장 일반적인 AAC 프로필. 64kbps 미만의 비트 전송률에 사용합니다. 자세한 내용은 Advanced Audio Coding을 참조하세요.
- HE-AAC: 일부 구형 플레이어 및 디바이스에서는 지원되지 않습니다. 40~80kbps 범위의 비트 전송률용입니다. 자세한 내용은 High-Efficiency Advanced Audio Coding을 참조하세요.
- HE-AACv2: 일부 플레이어 및 디바이스에서는 지원되지 않습니다. 48kbps 미만의 비트 전송률에 사용합니다. 자세한 내용은 High-Efficiency Advanced Audio Coding을 참조하세요.

Smooth 재생 목록의 모든 출력은 Profile의 값이 동일해야 합니다.

> **Note**
>
> AAC 프로필이 추가되기 전에 프리셋을 생성한 경우 Elastic Transcoder는 해당 프리셋에 AAC-LC 프로필을 사용합니다.

AAC에 대한 자세한 내용은 Wikipedia 항목 "MPEG-4 Part 3"에서 Audio Profiles를 참조하세요.

(선택 사항, FLAC/PCM 전용) 비트 깊이

샘플의 비트 깊이는 오디오 샘플에 포함되는 정보의 비트 수입니다. 비트 깊이가 높을수록 오디오 품질이 좋지만 파일도 커집니다.

FLAC 코덱의 유효한 값은 16 및 24입니다.

PCM 코덱의 유효한 값은 8, 16, 24 및 32입니다.

(선택 사항, PCM 전용) PCM 비트 서명

오디오 샘플이 음수 및 양수를 사용하여(signed) 또는 양수만 사용하여(unsigned) 표현되는지 여부.

유효한 값은 Signed(서명됨) 및 Unsigned(서명되지 않음)입니다.

가장 일반적인 값은 Signed(서명됨)입니다.

(선택 사항, PCM 전용) PCM 비트 순서

PCM 샘플의 비트가 저장되는 순서.

지원되는 값은 LittleEndian입니다.

샘플 속도

출력 파일 내 오디오 스트림의 샘플 레이트(Hz). Elastic Transcoder가 자동으로 입력 파일의 샘플 레이트를 감지하여 출력 파일에 해당 값을 사용하게 하려면 auto를 선택합니다. 샘플 레이트를 지정하려면 해당 값을 선택합니다.

비트 전송률

출력 파일 내 오디오 스트림의 비트 전송률(KB/s). 64~320 범위의 정수를 입력합니다.

채널

출력 파일의 오디오 채널 수. 유효한 값은 다음과 같습니다.

`auto`, `0`, `1`, `2`

1개 채널이 단일 스피커를 통해 재생되는 정보를 운반합니다. 예를 들어 2채널 스테레오 트랙은 한 채널을 좌측 스피커, 다른 채널을 우측 스피커로 보냅니다. 출력 채널을 트랙으로 구성됩니다. Elastic Transcoder가 자동으로 입력 파일 오디오 채널 수를 감지하여 출력 파일에 해당 값을 사용하게 하려면 `auto`를 선택합니다.

지정된 채널 값	입력 트랙 및 채널	출력 사운드 유형
auto	모든 입력	최대 8개 입력 채널을 전달합니다.
0	모든 입력	출력에서 생략된 오디오.
1	1개 채널 이상	모노 사운드.
2	모든 입력	동일한 모노 채널 2개 또는 스테레오. 트랙에 대한 자세한 내용은 Audio Packing Mode 섹션을 참조하세요.

디지털 오디오 작동 원리에 대한 자세한 내용은 디지털 오디오 섹션을 참조하세요. Elastic Transcoder가 채널 및 트랙을 구성하는 방식에 대한 자세한 내용은 Audio Packing Mode 섹션을 참조하세요.

(PCM 방식 MXF 전용) 오디오 패킹 모드

오디오 채널 및 트랙을 구성하는 방법. 채널을 사용하여 출력 내 채널 수를 지정하고, Audio Packing Mode(오디오 패킹 모드)를 사용하여 트랙 수 및 채널과의 관계를 지정합니다. 오디오 패킹 모드를 지정하지 않을 경우 Elastic Transcoder는 `SingleTrack`을 사용합니다.

유효한 값은 다음과 같습니다.

`SingleTrack`, `OneChannelPerTrack` 및 `OneChannelPerTrackWithMosTo8Tracks`

단일 트랙

Elastic Transcoder가 출력으로 단일 트랙을 생성합니다. 트랙은 최대 8개 채널을 가질 수 있습니다. 모든 비 `SingleTrack` 컨테이너에는 `mxf`을 사용합니다.

지정된 채널 값	입력 트랙 및 채널	출력 트랙 및 채널
0	모든 입력	출력에서 생략된 오디오.
1, 2 또는 auto	오디오 입력 없음	출력에서 생략된 오디오
1	모든 오디오 포함 입력	1채널 트랙 1개(필요한 경우 다운믹싱)
2	1채널 트랙 1개	동일 채널이 2개인 트랙 1개
2 또는 auto	1채널 트랙 2개	2채널 트랙 1개
2 또는 auto	2채널 트랙 1개	2채널 트랙 1개
2	다중 채널 트랙 1개	2채널 트랙 1개
auto	1채널 트랙 1개	1채널 트랙 1개
auto	다중 채널 트랙 1개	다중 채널 트랙 1개

(MXF 전용) 트랙당 1개 채널

Elastic Transcoder가 출력 내 각 채널에 대해 새 트랙을 생성합니다. 출력은 1채널 트랙을 최대 8개 포함할 수 있습니다.

지정된 채널 값	입력 트랙 및 채널	출력 트랙 및 채널
0	모든 입력	출력에서 생략된 오디오
1, 2 또는 auto	오디오 입력 없음	출력에서 생략된 오디오
1	모든 오디오 포함 입력	1채널 트랙 1개(필요한 경우 다운믹싱)
2	1채널 트랙 1개	각각 동일 채널 1개를 갖는 트랙 2개
2 또는 auto	1채널 트랙 2개	1채널 트랙 2개
2 또는 auto	2채널 트랙 1개	1채널 트랙 2개
2	다중 채널 트랙 1개	1채널 트랙 2개
auto	1채널 트랙 1개	1채널 트랙 1개
auto	다중 채널 트랙 1개	1채널 트랙 최대 8개

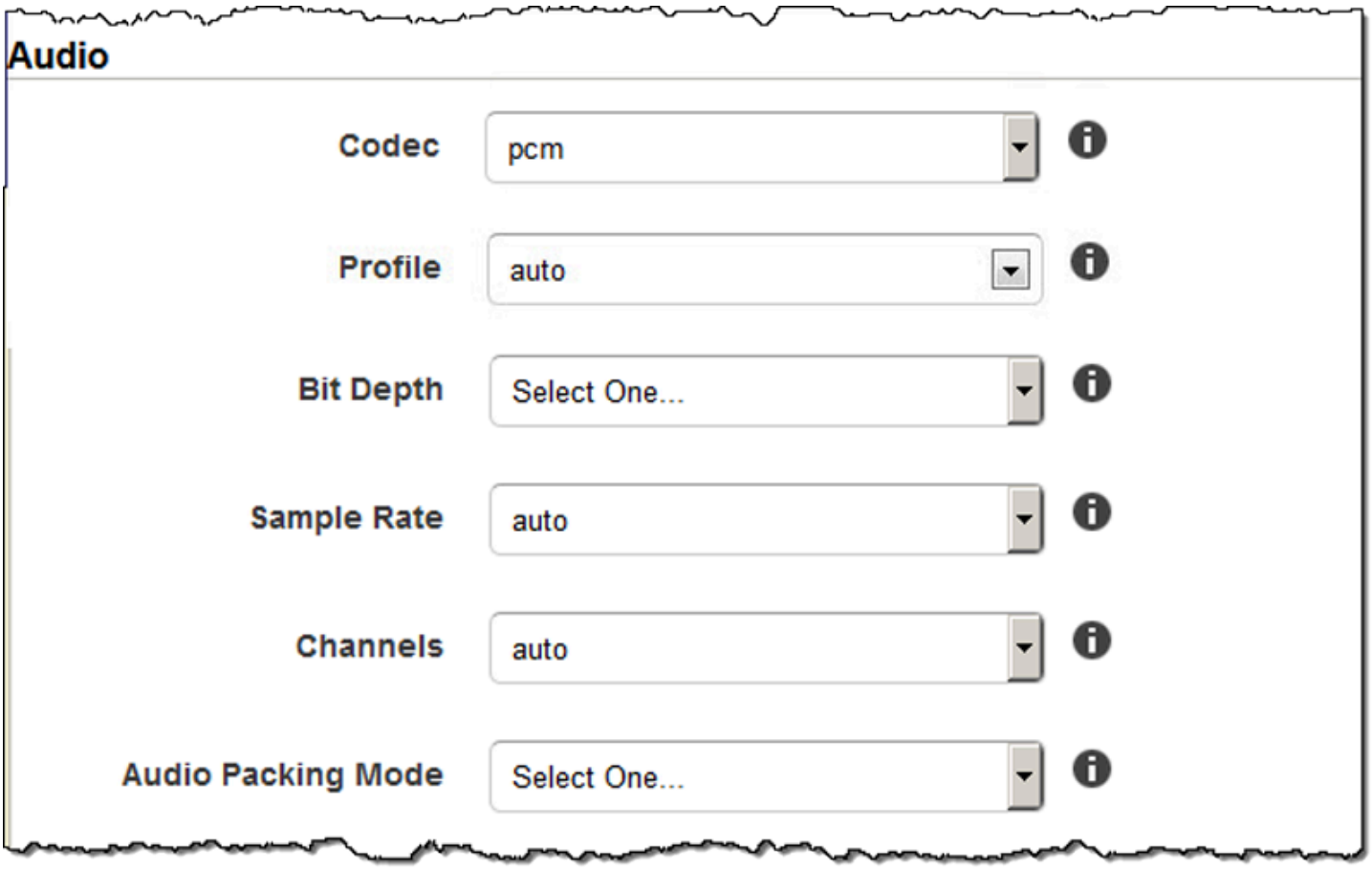
(MXF 전용) 8개 트랙까지 MOS가 있는 트랙당 1개 채널

Elastic Transcoder가 출력으로 8개의 1채널 트랙을 생성합니다. 입력 채널의 오디오 데이터를 포함하지 않는 트랙은 모두 MOS(Mit Out Sound) 트랙입니다.

지정된 채널 값	입력 트랙 및 채널	출력 트랙 및 채널
0	모든 입력	출력에서 생략된 오디오
1, 2 또는 auto	오디오 입력 없음	출력에서 생략된 오디오
1	모든 오디오 포함 입력	1채널 트랙 1개(필요한 경우 다운믹싱) 및 MOS 트랙 7개
2	1채널 트랙 1개	각각 동일 채널 1개를 갖는 트랙 2개 및 MOS 트랙 6개

지정된 채널 값	입력 트랙 및 채널	출력 트랙 및 채널
2 또는 auto	1채널 트랙 2개	1채널 트랙 2개 및 MOS 트랙 6개
2 또는 auto	2채널 트랙 1개	1채널 트랙 2개 및 MOS 트랙 6개
2	다중 채널 트랙 1개	1채널 트랙 2개 및 MOS 트랙 6개
auto	1채널 트랙 1개	1채널 트랙 1개 및 MOS 트랙 7개
auto	다중 채널 트랙 1개	총 8개 트랙이 될 때까지 1채널 트랙 최대 8개 및 MOS 트랙

채널 및 트랙에 대한 자세한 내용은 디지털 오디오 섹션을 참조하세요.



썸네일 설정

형식

썸네일(있는 경우)의 형식. 유효한 값은 jpg 및 png입니다.

작업을 생성할 때 Elastic Transcoder가 썸네일을 생성하도록 할지 여부를 지정합니다. 자세한 내용은 Thumbnail Filename Pattern 주제에서 Elastic Transcoder 작업을 만들 때 지정하는 설정 섹션을 참조하세요.

간격

썸네일 간의 대략적 시간(초). 정수 값을 지정합니다. 썸네일 간의 실제 간격은 몇 초 정도 변동할 수 있습니다.

최대 높이

썸네일의 최대 높이(픽셀). auto를 지정할 경우 Elastic Transcoder는 1080(Full HD)을 기본값으로 사용합니다. 숫자 값을 지정하려면 32~3072 범위의 정수를 입력합니다.

최대 너비

썸네일의 최대 너비(픽셀). auto를 지정할 경우 Elastic Transcoder는 1920(Full HD)을 기본값으로 사용합니다. 숫자 값을 지정하려면 32~4096 범위의 정수를 입력합니다.

크기 조정 정책

다음 값 중 하나를 지정하여 썸네일의 조정을 제어합니다.

- Fit: Elastic Transcoder가 썸네일 최대 너비 또는 최대 높이 설정에 지정된 값과 일치하되 다른 값은 초과하지 않도록 썸네일을 확대 또는 축소합니다.
- Fill: Elastic Transcoder가 최대 너비 또는 최대 높이에 지정된 값과 일치하되 다른 값과 일치하거나 초과하도록 썸네일의 크기를 조정합니다. Elastic Transcoder는 썸네일의 중앙에 이미지를 배치한 다음 최대값을 초과하는 크기(있는 경우)에서 잘라냅니다.
- Stretch: Elastic Transcoder가 썸네일 최대 너비 및 최대 높이 설정에 지정된 값과 일치하도록 썸네일을 늘립니다. 입력 비디오의 상대 비율과 썸네일의 상대 비율이 다를 경우 썸네일이 변형됩니다.
- Keep: Elastic Transcoder는 썸네일의 크기를 조정하지 않습니다. 입력 비디오의 치수가 썸네일 최대 너비 및 최대 높이 설정에 지정된 값을 초과하는 경우 Elastic Transcoder는 썸네일을 자릅니다.
- ShrinkToFit: Elastic Transcoder가 썸네일의 치수가 썸네일 최대 너비 및 최대 높이 중 하나 이상에 지정된 값과 일치하되 어떤 값도 초과하지 않도록 썸네일을 축소합니다. 이 옵션을 지정할 경우 Elastic Transcoder는 썸네일을 확대하지 않습니다.
- ShrinkToFill: Elastic Transcoder가 썸네일의 치수가 최대 너비 및 최대 높이 중 하나 이상에 지정된 값과 일치하되 어떤 값도 하회하지 않도록 썸네일을 축소합니다. 이 옵션을 지정할 경우 Elastic Transcoder는 썸네일을 확대하지 않습니다.

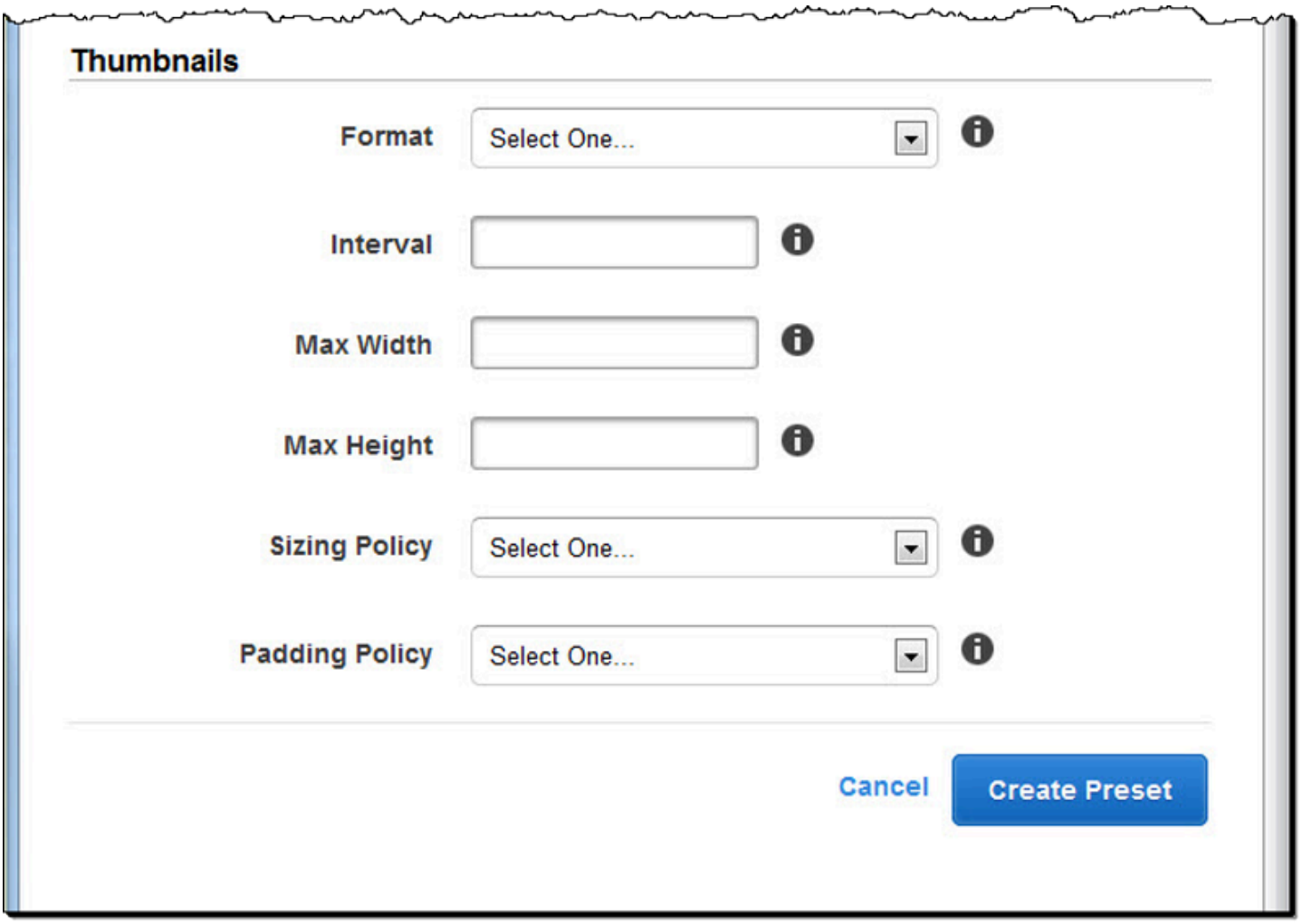
다음 표는 썸네일에 대한 Sizing Policy 설정의 효과를 보여줍니다.

SizingPolicy	썸네일이 확장될 수 있음	패딩 정책이 "Pad"일 경우 썸네일이 패딩될 수 있음	썸네일의 픽셀 종횡비가 입력 비디오와 다를 수 있음	썸네일이 잘릴 수 있음
Fit	예	예		
Fill	예			예

SizingPolicy	썸네일이 확장될 수 있음	패딩 정책이 "Pad"일 경우 썸네일이 패딩될 수 있음	썸네일의 픽셀 종횡비가 입력 비디오와 다를 수 있음	썸네일이 잘릴 수 있음
Stretch	예		예	
Keep	예	예		예
ShrinkToFit		예		
ShrinkToFill		예		예

패딩 정책

PaddingPolicy를 Pad로 설정하면 Elastic Transcoder가 썸네일의 전체 크기를 썸네일 최대 너비 및 최대 높이 설정에 지정된 값과 일치시키기 위해 썸네일의 상하 및/또는 좌우에 블랙 바를 추가할 수 있습니다. 자세한 내용은 Sizing Policy의 표를 참조하세요.


Thumbnails
Format
Select One...
Interval
Max Width
Max Height
Sizing Policy
Select One...
Padding Policy
Select One...
Cancel
Create Preset

Amazon S3 버킷에서 파일 정리

Amazon Elastic Transcoder는 Amazon S3와 통합되어 입력 및 출력 파일을 저장합니다. Amazon S3의 버킷 스토리지 한도가 100개로 제한되어 있기 때문에 Amazon S3 파일을 구성하고 관리하는 방식에 영향을 줍니다.

Amazon S3 를 사용할 경우 일반적인 파일 시스템을 모방하는 것이 좋습니다. Elastic Transcoder에서 OutputKeyPrefix를 사용하여 출력 파일에 파일 경로를 추가하면 됩니다. 파일 시스템을 모방하면 Amazon S3 API의 접두사 및 구분 기호 파라미터를 사용하여 파일을 찾을 수 있습니다.

예를 들어 다수의 고객을 위한 비디오를 가져와 일반적인 여러 디바이스에서 사용할 수 있도록 트랜스코딩하는 사용자 생성 콘텐츠(UGC) 사이트를 만든다고 가정해 보겠습니다. 비디오를 저장하려면 다음과 같이 입력 버킷과 출력 버킷을 구성합니다.

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

이렇게 하면 별도로 형식을 저장하여 쉽게 찾을 수 있습니다. 예를 들어, 다양한 영화를 여러 형식으로 트랜스코딩하는 서비스를 사용하는 고객 두 명 있을 경우 버킷을 다음과 같이 구성할 수 있습니다.

```
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` 및`OutputKey`의 마지막 부분이 함께 실행되지 않도록 OutputKeyPrefix 뒤에 '/'를 붙이세요. 이렇게 하지 않으면 파일이 다음과 같이 됩니다.

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

또한 리전당 버킷 하나에 입력 미디어를 보관하고 동일한 리전에서 별도의 버킷을 만들어 트랜스코딩된 미디어와 썸네일을 보관하세요. 이렇게 하면 Amazon S3 버킷과 Elastic Transcoder 간의 교차 리전 전송 요금이 부과되는 것을 피할 수 있습니다.

> **Note**
>
> Amazon S3 버킷에 액세스하는 초당 100개 이상의 요청이 있을 것으로 예상되는 경우 시스템을 설계할 때 Amazon S3 성능 고려 사항을 따르는 것이 좋습니다.

콘텐츠 보안 유지

이 단원에서는 Elastic Transcoder 리소스에 대한 액세스를 제어하는 방법, 파일이 저장되어 있는 동안 파일을 암호화하는 방법, 파일에 DRM(Digital Rights Management)을 적용하는 방법을 설명합니다. 리소스에 대한 액세스를 제어하면 작업 만들기와 같은 동작을 IAM 역할 중 하나에 지정하는 동시에, 파이프라인 및 프리셋 리소스를 업데이트하거나 삭제하는 능력을 보유할 수 있습니다. 저장 파일을 암호화하면 특히 민감하거나 엄격하게 제어되는 파일에 대해 추가 수준의 콘텐츠 보호가 제공되는 반면, DRM을 사용하면 리소스에 대한 AWS 권한에서 제공하는 수준 이상으로 파일을 재생할 수 있는 사용자를 추가로 제어할 수 있습니다.

IAM 액세스 제어는 파일에 대해 액세스 권한을 가진 사용자 및 파이프라인과 프리셋 등의 리소스에 영향을 줄 수 있는 사용자를 통제하려는 경우에 유용합니다. 파일 암호화(저장 시 암호화)는 파일을 암호화된 상태로 저장하려는 경우를 위한 기능이고, HLS 및 DRM은 파일을 재생할 수 있는 권한을 가진 사용자를 제어하려는 경우에 사용합니다.

보안 모범 사례에 대한 자세한 내용은 IAM 모범 사례 가이드를 참조하세요.

주제

- Elastic Transcoder에 대한 액세스 제어
- 데이터 암호화 옵션

Elastic Transcoder에 대한 액세스 제어

Amazon Elastic Transcoder를 사용하면 AWS Identity and Access Management (IAM)를 사용하여 사용자가 Elastic Transcoder로 수행할 수 있는 작업을 제어하고 Elastic Transcoder에서 필요한 다른 AWS 서비스에 대한 Elastic Transcoder의 액세스를 제어할 수 있습니다. IAM 사용자, IAM 그룹 또는 역할과 연결할 수 있는 권한 모음인 IAM 정책을 사용하여 액세스를 제어합니다.

주제

- Elastic Transcoder에 대한 액세스 제어
- Elastic Transcoder 파이프라인의 서비스 역할

Elastic Transcoder에 대한 액세스 제어

액세스 권한을 제공하려면 사용자, 그룹 또는 역할에 권한을 추가하세요:

- 의 사용자 및 그룹 AWS IAM Identity Center:

  권한 세트를 생성합니다. AWS IAM Identity Center 사용 설명서의 권한 세트 생성의 지침을 따릅니다.

- 보안 인증 공급자를 통해 IAM에서 관리되는 사용자:

  ID 페더레이션을 위한 역할을 생성합니다. IAM 사용 설명서의 Create a role for a third-party identity provider (federation)의 지침을 따릅니다.

- IAM 사용자:
  - 사용자가 맡을 수 있는 역할을 생성합니다. IAM 사용 설명서에서 Create a role for an IAM user의 지침을 따릅니다.
  - (권장되지 않음)정책을 사용자에게 직접 연결하거나 사용자를 사용자 그룹에 추가합니다. IAM 사용 설명서에서 사용자(콘솔)에 권한 추가의 지침을 따르세요.

다른 AWS 서비스에 대한 Elastic Transcoder의 액세스를 제어하려면 서비스 역할을 생성할 수 있습니다. 이는 파이프라인을 만들 때 할당하고, 트랜스코딩과 연결된 작업을 수행할 권한을 Elastic Transcoder 자체에 부여하는 IAM 역할입니다.

에 대한 역할을 생성하려면 AWS 서비스 (IAM 콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/iam/ IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 역할을 선택하고 역할 생성을 선택합니다.
3. 신뢰할 수 있는 엔터티 유형에 AWS 서비스를 선택합니다.
4. 서비스 또는 사용 사례의 경우 서비스를 선택한 다음, 사용 사례를 선택합니다. 사용 사례는 서비스에 필요한 신뢰 정책을 포함하기 위해 서비스에서 정합니다.
5. 다음을 선택합니다.
6. 권한 정책의 경우 선택한 사용 사례에 따라 옵션이 달라집니다.
   - 서비스가 역할에 대한 권한을 정의하는 경우 권한 정책을 선택할 수 없습니다.
   - 제한된 권한 정책 세트에서 선택합니다.
   - 모든 권한 정책에서 선택합니다.
   - 권한 정책 없음을 선택하고 역할이 생성된 후 정책을 생성한 다음, 정책을 역할에 연결합니다.
7. (선택 사항) 권한 경계를 선택합니다. 이는 서비스 역할에서 가능한 고급 기능이며 서비스 링크된 역할은 아닙니다.

a. 권한 경계 설정 섹션을 열고 최대 역할 권한을 관리하기 위한 권한 경계 사용을 선택합니다.

   IAM에는 계정의 AWS 관리형 및 고객 관리형 정책 목록이 포함됩니다.

b. 정책을 선택하여 권한 경계를 사용하세요.

8. 다음을 선택합니다.
9. 역할 이름의 경우 옵션은 서비스에 따라 달라집니다.

   - 서비스에서 역할 이름을 정의하는 경우 이 역할 이름을 편집할 수 없습니다.
   - 서비스에서 역할 이름에 대한 접두사를 정의하는 경우 사용자가 선택적 접미사를 입력할 수 있습니다.
   - 서비스에서 역할 이름을 정의하지 않는 경우 역할 이름을 지정할 수 있습니다.

   > **Important**
   >
   > 역할 이름을 지정할 때는 다음 사항에 유의하세요.
   >
   > - 역할 이름은 내에서 고유해야 하며 대/소문자를 구분할 AWS 계정수 없습니다.
   >
   >   예를 들어, 이름이 **`PRODROLE`**과 **`prodrole`**, 두 가지로 지정된 역할을 만들지 마세요. 역할 이름이 정책 또는 ARN의 일부로 사용되는 경우 역할 이름은 대소문자를 구분합니다. 그러나 로그인 프로세스와 같이 콘솔에서 역할 이름이 고객에게 표시되는 경우에는 역할 이름이 대소문자를 구분하지 않습니다.
   > - 다른 엔터티가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다.

10. (선택 사항)설명에 역할에 대한 설명을 입력합니다.
11. (선택 사항) 역할에 대한 사용 사례와 권한을 편집하려면 1단계: 신뢰할 수 있는 엔터티 선택 또는 2단계: 권한 추가 섹션에서 편집을 선택합니다.
12. (선택 사항) 태그를 키-값 페어로 연결하여 역할을 식별, 구성 또는 검색합니다. IAM에서 태그를 사용하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS Identity and Access Management 리소스에 대한 태그를 참조하세요.
13. 역할을 검토한 다음 역할 생성을 선택합니다.

트랜스코딩 프로세스 중에 사용자 역할과 서비스 역할은 모두 중요하며, 한 예로 Elastic Transcoder는 Amazon S3 버킷에서 파일을 가져오고 다른 Amazon S3 버킷에 트랜스코딩된 파일을 저장하기 위해

서비스 역할이 필요하고, 사용자는 Elastic Transcoder에서 작업을 만들 수 있도록 허용하는 IAM 역할이 필요합니다.

IAM에 대한 자세한 내용은 IAM 사용 설명서를 참조하세요. 서비스 역할에 대한 자세한 내용은 AWS 서비스에 대한 역할 생성을 참조하세요.

Elastic Transcoder에 대한 예제 정책

사용자가 파이프라인 만들기 및 작업 실행과 같은 Elastic Transcoder 관리 기능을 수행할 수 있게 하려면 해당 사용자와 연결할 수 있는 정책이 있어야 합니다. 이 섹션에서는 정책을 생성하는 방법을 보여 주고 Elastic Transcoder 작업과 Elastic Transcoder가 의존하는 관련 서비스 작업에 대한 액세스를 제어하기 위한 세 가지 정책도 보여줍니다. AWS 계정 사용자에게 모든 Elastic Transcoder 작업 또는 그 하위 집합에만 대한 액세스 권한을 부여할 수 있습니다.

정책 관리에 대한 자세한 내용은 IAM 사용 설명서의 IAM 정책 관리를 참조하세요.

JSON 정책 편집기를 사용하여 정책을 생성하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/iam/ IAM 콘솔을 엽니다.
2. 왼쪽의 탐색 창에서 정책을 선택합니다.

   정책을 처음으로 선택하는 경우 관리형 정책 소개 페이지가 나타납니다. 시작을 선택합니다.
3. 페이지 상단에서 정책 생성을 선택합니다.
4. 정책 편집기 섹션에서 JSON 옵션을 선택합니다.
5. JSON 정책 문서를 입력하거나 붙여 넣습니다. IAM 정책 언어에 대한 자세한 내용은 IAM JSON 정책 참조를 참조하세요.
6. 정책 검증 중에 생성되는 모든 보안 경고, 오류 또는 일반 경고를 해결하고 다음을 선택합니다.

   > **Note**
   >
   > 언제든지 시각적 편집기 옵션과 JSON 편집기 옵션을 서로 전환할 수 있습니다. 그러나 변경을 적용하거나 시각적 편집기에서 다음을 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 정책 재구성을 참조하세요.
7. (선택 사항)에서 정책을 생성하거나 편집할 때 템플릿에 사용할 AWS Management Console수 있는 JSON 또는 YAML 정책 CloudFormation 템플릿을 생성할 수 있습니다.

이렇게 하려면 정책 편집기에서 작업을 선택한 다음, CloudFormation 템플릿 생성을 선택합니다. 자세한 내용은 AWS CloudFormation 사용 설명서의 AWS Identity and Access Management 리소스 유형 참조를 CloudFormation참조하세요.

8. 정책에 권한 추가를 완료했으면 다음을 선택합니다.
9. 검토 및 생성 페이지에서 생성하는 정책의 정책 이름과 설명(선택 사항)을 입력합니다. 이 정책에 정의된 권한을 검토하여 정책이 부여한 권한을 확인합니다.
10. (선택 사항) 태그를 키 값 페어로 연결하여 메타데이터를 정책에 추가합니다. IAM에서 태그를 사용하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS Identity and Access Management 리소스에 대한 태그를 참조하세요.
11. 정책 생성을 선택하고 새로운 정책을 저장합니다.

Elastic Transcoder 및 Amazon S3에 대한 읽기 전용 액세스 권한 부여

다음 정책은 Elastic Transcoder 리소스에 대한 읽기 전용 액세스 및 Amazon S3의 나열 작업에 대한 액세스를 부여합니다. 이 정책은 트랜스코딩된 파일을 찾고 볼 권한과 IAM 계정에 사용할 수 있는 버킷을 볼 권한이 필요하지만, 리소스 또는 파일을 업데이트, 생성 또는 삭제할 능력은 필요하지 않은 경우 유용합니다. 또한 이 정책은 IAM 계정의 사용 가능한 모든 파이프라인, 프리셋 및 작업을 나열할 수 있도록 허용합니다. 특정 버킷에 대한 액세스를 제한하려면 VPC의 리소스에 대한 액세스 제한 섹션을 참조하세요.

JSON

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

작업을 만들 권한 부여

다음 정책은 계정과 연결된 모든 Elastic Transcoder 리소스를 나열 및 가져오고, 작업 및 프리셋을 생성 또는 수정하며, Amazon S3 및 Amazon SNS의 나열 작업을 사용할 권한을 부여합니다.

이 정책은 트랜스코딩 설정을 수정하고 프리셋 또는 작업을 생성 또는 삭제하는 기능을 수정하는 데 유용합니다. 파이프라인, Amazon S3 버킷 또는 Amazon SNS 알림을 생성, 업데이트 또는 삭제하는 것은 허용하지 않습니다.

JSON

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

제어 가능한 액세스와 Elastic Transcoder 작업

다음은 Elastic Transcoder 작업의 전체 목록입니다.

```
    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
```

VPC의 리소스에 대한 액세스 제한

작업에 대한 액세스를 제한하는 것 외에도 특정 작업, 파이프라인 및 프리셋에 대한 액세스를 추가로 제한할 수 있으며, 이를 리소스 수준 권한 부여라고 합니다.

Elastic Transcoder 리소스의 하위 집합에 대한 액세스를 제한하거나 부여하려면 리소스의 ARN을 정책의 리소스 요소에 넣습니다. Elastic Transcoder ARN은 다음과 같은 일반적 형식을 취합니다.

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

region, *account*, *resource*, *ID* 변수를 유효한 값으로 바꿉니다. 유효한 값은 다음과 같습니다.

- *region*: 리전의 이름입니다. 리전 목록은 여기에서 확인할 수 있습니다. 모든 리전을 나타내려면 와일드카드(*)를 사용합니다. 값을 지정해야 합니다.
- *##*: AWS 계정의 ID입니다. 값을 지정해야 합니다.
- *resource*: Elastic Transcoder 리소스의 유형으로, `preset`, `pipeline` 또는 `job`입니다.
- *ID*: 특정 프리셋, 파이프라인 또는 작업의 ID 또는 현재 AWS 계정과 연결된 지정된 유형의 모든 리소스를 나타내는 *입니다.

예를 들어 다음 ARN은 계정 `111122223333`의 `us-east-2` 리전에 있는 모든 프리셋 리소스를 지정합니다.

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

파이프라인, 프리셋 또는 작업 콘솔 페이지에서 리소스 이름 옆의 돋보기 아이콘
( )
을 클릭하여 리소스의 ARN을 찾을 수 있습니다.

자세한 내용은 IAM 사용 설명서의 리소스를 참조하세요.

리소스를 제한하기 위한 정책 예제

다음 정책은 Amazon S3의 `amzn-s3-demo-bucket`이라는 버킷에 대한 권한, Elastic Transcoder의 모든 항목에 대한 나열 및 읽기 권한, `example_pipeline`이라는 파이프라인에서 작업을 만들 권한을 부여합니다.

이 정책은 사용 가능한 파일과 리소스를 볼 수 있고, 그러한 리소스를 사용하여 자신의 트랜스코딩 작업을 만들 수 있어야 하는 SDK 및 CLI 사용자에게 유용합니다. 리소스 업데이트 또는 삭제, 작업 이외의 리소스 생성 또는 여기에 지정된 리소스 이외의 리소스 작업은 허용하지 않으며, 콘솔 사용자에게는 적용되지 않습니다.

Elastic Transcoder 파이프라인의 서비스 역할

트랜스코딩 작업을 관리하는 파이프라인을 생성할 때 IAM 서비스 역할을 지정해야 합니다. IAM 서비스 역할에는 트랜스코딩을 위해 해당 파이프라인에서 사용하는 권한을 지정하는 정책이 있습니다.

파이프라인의 역할을 지정할 때 두 가지 옵션이 있습니다.

- 트랜스코딩을 위해 Elastic Transcoder에 필요한 권한만 포함된 기본 역할을 사용합니다. Elastic Transcoder 콘솔을 사용하여 파이프라인을 생성하는 경우, 첫 번째 파이프라인을 생성할 때 콘솔이 기본 역할을 만들 수 있는 옵션을 자동으로 부여합니다. 기본 역할을 포함하여 IAM 서비스 역할을 만들 관리 권한이 있어야 합니다.
- 기존 역할을 선택합니다. 이 경우 IAM에서 이전에 역할을 만들었고, 파일을 트랜스코딩할 충분한 권한을 Elastic Transcoder에 부여하는 정책을 해당 역할에 연결했어야 합니다. 이는 다른 AWS 서비스의 역할도 사용하려는 경우에 유용합니다.

파이프라인의 기본 IAM 역할

Elastic Transcoder는 Elastic Transcoder에서 만든 기본 역할을 사용하여 다음 작업을 수행할 수 있습니다.

- 트랜스코딩을 위해 Amazon S3 버킷에서 파일 가져오기
- Amazon S3 버킷의 콘텐츠 나열
- Amazon S3 버킷에 트랜스코딩된 파일 저장
- Amazon S3 멀티파트 업로드 생성

- SNS 주제에 알림 게시

이 정책은 Elastic Transcoder가 다음 작업을 수행하지 못하게 합니다.

- Amazon SNS 삭제 작업 수행 또는 주제에서 정책 설명 추가 또는 제거
- Amazon S3 버킷 또는 항목 삭제 작업 수행 또는 버킷 정책 추가, 제거 또는 수정

기본 역할의 액세스(권한) 정책 정의는 다음과 같습니다.

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 서비스 연결 역할을 지원하는 리전

Elastic Transcoder는 다음과 같은 리전에서 서비스 연결 역할 사용을 지원합니다.

리전 이름	리전 자격 증명	Elastic Transcoder의 지원
미국 동부(버지니아 북부)	us-east-1	예
미국 동부(오하이오)	us-east-2	아니요
미국 서부(캘리포니아 북부)	us-west-1	예
미국 서부(오리건)	us-west-2	예
아시아 태평양(뭄바이)	ap-south-1	예
아시아 태평양(오사카)	ap-northeast-3	아니요
아시아 태평양(서울)	ap-northeast-2	아니요
아시아 태평양(싱가포르)	ap-southeast-1	예
아시아 태평양(시드니)	ap-southeast-2	예
아시아 태평양(도쿄)	ap-northeast-1	예
캐나다(중부)	ca-central-1	아니요
유럽(프랑크푸르트)	eu-central-1	아니요
유럽(아일랜드)	eu-west-1	예
유럽(런던)	eu-west-2	아니요
유럽(파리)	eu-west-3	아니요

리전 이름	리전 자격 증명	Elastic Transcoder의 지원
남아메리카(상파울루)	sa-east-1	아니요

데이터 암호화 옵션

파일을 Amazon S3에 저장하는 동안 트랜스코딩 작업에 사용하려는 입력 파일과 출력 파일을 암호화함으로써 Elastic Transcoder 데이터를 보호할 수 있습니다. 이러한 데이터에는 입력 파일, 출력 파일, 및 기타 썸네일, 캡션, 입력 워터마크 또는 입력 앨범 아트 등이 포함됩니다. 재생 목록과 메타데이터는 암호화되지 않습니다.

파이프라인, Amazon S3 버킷, AWS Key Management Service 키를 포함한 작업의 모든 리소스는 동일한 AWS 리전에 있어야 합니다.

주제

- 암호화 옵션
- Elastic Transcoder AWS KMS 와 함께 사용
- HLS 콘텐츠 보호
- 디지털 권한 관리

암호화 옵션

Elastic Transcoder는 두 가지 기본 암호화 옵션을 지원합니다.

- Amazon S3 서버 측 암호화: AWS가 암호화 프로세스를 관리합니다. 예를 들어 Elastic Transcoder는 Amazon S3를 호출하고, Amazon S3는 데이터를 암호화하여 데이터 센터의 디스크에 저장한 후 데이터가 다운로드될 때 데이터를 해독합니다.

  기본적으로 Amazon S3 버킷은 암호화된 파일과 암호화되지 않은 파일을 모두 수락하지만, 암호화된 파일만 수락하도록 Amazon S3 버킷을 설정할 수 있습니다. Elastic Transcoder가 Amazon S3 버킷에 대해 액세스 권한이 있으면 권한을 변경할 필요가 없습니다.

  Amazon S3의 서버 측 암호화에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 서버 측 암호화를 사용하여 데이터 보호를 참조하세요. AWS KMS 키에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS Key Management Service란 무엇입니까?를 참조하세요.

> **Note**
>
> AWS-KMS 키를 사용하면 추가 요금이 부과됩니다. 자세한 내용은 AWS Key Management Service 요금을 참조하세요.

- 고객측 제공 키를 사용하여 클라이언트측 암호화: Elastic Transcoder는 클라이언트 제공 암호화 키를 사용하여 사용자가 이미 암호화한 입력 파일을 해독하거나 출력 파일을 Amazon S3에 저장하기 전에 암호화할 수도 있습니다. 이 경우에는 사용자가 암호화 키와 관련 도구를 관리합니다.

  Elastic Transcoder가 클라이언트 제공 키를 사용하여 파일을 트랜스코딩하도록 하려면 작업 요청에 파일을 암호화하는 데 사용한 AWS KMS암호화 키, 체크섬으로 사용할 키의 MD5, 출력 파일을 암호화할 때 Elastic Transcoder가 사용할 초기화 벡터(또는 무작위 비트 생성기가 생성한 일련의 무작위 비트)가 포함되어야 합니다.

  Elastic Transcoder는 AWS KMS KMS 키로 암호화된 고객 제공 키만 사용할 수 있으며, KMS 키를 사용할 수 있는 권한이 Elastic Transcoder에 부여되어야 합니다. 키를 암호화하려면 다음 정보가 포함된 암호화 호출을 사용하여 AWS KMS 프로그래밍 방식으로를 호출해야 합니다.

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

> **Important**
>
> 프라이빗 암호화 키와 암호화되지 않은 데이터는 AWS;에서 절대로 저장하지 않으므로 암호화 키를 안전하게 관리하는 것이 중요합니다. 암호화 키를 잃어버릴 경우 데이터의 암호를 해독할 수 없습니다.

  Elastic Transcoder에 키를 사용할 수 있는 권한을 부여하려면 Elastic Transcoder AWS KMS 와 함께 사용 섹션을 참조하세요.

데이터 암호화에 대한 자세한 내용은 AWS KMS API 참조 및 데이터 암호화 및 해독을 참조하세요. 컨텍스트에 대한 자세한 내용은 AWS Key Management Service 개발자 가이드에서 암호화 컨텍스트를 참조하세요.

고객 제공 키에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 고객 제공 키(SSE-C)로 서버 측 암호화 사용을 참조하세요.

Elastic Transcoder 콘솔을 사용하여 파일을 해독하고 암호화할 때 필요한 설정에 대한 정보는 (선택 사항) 출력 암호화 섹션을 참조하세요. Elastic Transcoder API를 사용하여 파일을 복호화하고 암호화할 때 필요한 설정에 대한 자세한 내용은 암호화 요소로 시작하는 작업 생성 API 작업을 참조하세요.

Elastic Transcoder AWS KMS 와 함께 사용

AWS Key Management Service (AWS KMS)를 Elastic Transcoder와 함께 사용하여 데이터를 암호화하는 데 사용되는 암호화 키를 생성하고 관리할 수 있습니다. 사용할 Elastic Transcoder를 설정하려면 먼저 다음이 있어야 AWS KMS합니다.

- Elastic Transcoder 파이프라인
- Elastic Transcoder 파이프라인과 연결된 IAM 역할
- AWS KMS 키
- AWS KMS 키의 ARN

다음 절차는 기존 리소스를 식별하는 방법 또는 새로운 리소스를 만드는 방법을 보여줍니다.

AWS KMS Elastic Transcoder 사용 준비

파이프라인을 만들려면

- Elastic Transcoder에서 파이프라인 생성 섹션의 단계를 따르세요.

파이프라인과 연결된 IAM 역할을 식별하려면

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/elastictranscoder/ Elastic Transcoder 콘솔을 엽니다.
2. 탐색 창에서 Pipelines를 클릭합니다.
3. 파이프라인 이름 옆에 있는 돋보기 아이콘을 클릭합니다.

4. Permissions 섹션을 클릭하여 확장합니다.
5. IAM 역할을 메모해 둡니다. Elastic Transcoder가 생성한 기본 역할을 사용하는 경우 이 역할은 Elastic_Transcoder_Default_Role입니다.

AWS KMS 키를 생성하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 키 생성 단계를 따릅니다.

AWS KMS 키의 ARN을 식별하려면

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 Encryption Keys를 클릭합니다.
3. 리전 드롭다운 목록에서 해당 키와 파이프라인이 있는 리전을 선택합니다.
4. 사용하려는 키를 클릭합니다.
5. ARN을 메모해 둡니다.

콘솔을 사용하여 AWS KMS 키를 생성할 수 있지만 암호화 및 복호화 APIs를 사용하여 AWS KMS 키로 데이터를 암호화하거나 복호화해야 합니다. 자세한 내용은 데이터 암호화 및 해독을 참조하세요.

Elastic Transcoder 및 연결 AWS KMS

파이프라인, IAM 역할 및 AWS KMS 키가 있으면 파이프라인에 사용할 키를 알려주고 키에 사용할 수 있는 IAM 역할을 알려주어야 합니다.

파이프라인에 AWS KMS 키를 추가하려면

1. https://console.aws.amazon.com/elastictranscoder/에서 Elastic Transcoder 콘솔을 엽니다.
2. AWS KMS 키를 사용할 파이프라인을 선택하고 편집을 클릭합니다.
3. Encryption 섹션을 클릭하여 확장하고 AWS KMS Key ARN 섹션에서 Custom을 선택합니다.
4. AWS KMS 키의 ARN을 입력하고 저장을 클릭합니다.

AWS KMS 키에 IAM 역할을 추가하려면

파이프라인과 연결된 IAM 역할로 AWS KMS 키를 생성하지 않은 경우 다음 절차에 따라 키를 추가할 수 있습니다.

1. https://console.aws.amazon.com/kms AWS KMS 콘솔을 엽니다.
2. 리전 드롭다운 목록에서 키와 파이프라인 생성할 때 선택했던 리전을 선택합니다.
3. 탐색 창에서 고객 관리형 키를 선택합니다.
4. 오른쪽의 고객 관리형 키 섹션에서 사용하려는 키를 선택합니다.
5. 키 사용자 섹션에서 추가를 선택합니다.
6. 키 사용자 추가 페이지에서 해당 파이프라인과 연결된 역할을 검색하여 결과 페이지에서 선택하고 추가를 클릭합니다.

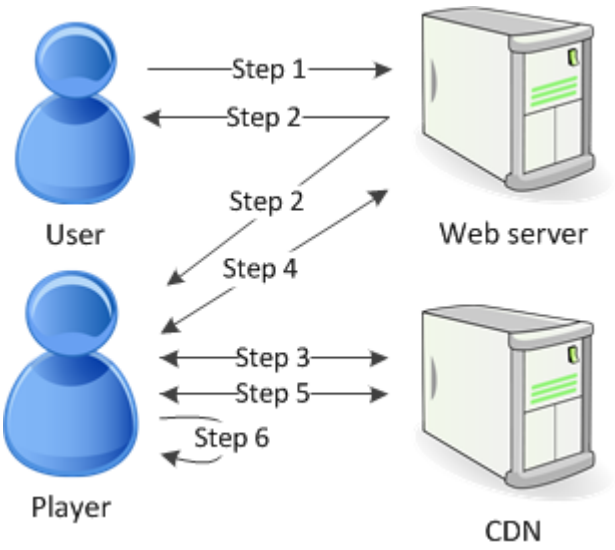
이제 Elastic Transcoder 파이프라인에서 AWS KMS 키를 사용할 수 있습니다.

HLS 콘텐츠 보호

HLS(HTTP Live Streaming)은 스트리밍 도중 최적화를 위해 미디어 파일을 분할하는 프로토콜입니다. HLS는 미디어 플레이어에서 세그먼트를 재생할 때 해당 네트워크 연결에서 지원하는 가장 높은 품질의 해상도로 세그먼트를 재생하도록 합니다.

Elastic Transcoder를 사용하면 스트리밍된 미디어 파일의 세그먼트를 암호화하고, 인터넷으로 암호화된 세그먼트를 보내어 재생 시 해독할 수 있습니다. 따라서 미디어 콘텐츠를 보호하고 권한이 있는 사용자만 미디어 파일의 암호화된 세그먼트를 볼 수 있습니다.

다음은 HLS 콘텐츠 보호를 지원하는 미디어 파일의 재생 프로세스를 요약한 것입니다.



1. 웹 페이지를 방문하는 사용자가 웹 서버에서 인증을 받고, 사용자 브라우저에 세션 쿠키를 설정합니다.
2. 사용자가 웹 서버에서 플레이어를 로드합니다.
3. 플레이어가 CDN(Content Delivery Network)에서 마스터 재생 목록을 가져옵니다. 마스터 재생 목록에서 미디어 파일에 적합한 비트 전송 속도와 해상도를 제공합니다.
4. 플레이어에서 웹 서버를 호출하여 세션 쿠키를 확인하고, 사용자에게 콘텐츠를 볼 수 있는 권한이 있는지 확인한 후 데이터 암호 해독 키를 반환합니다.
5. 플레이어가 해당 재생 목록을 선택한 후 CDN에서 해당 미디어 세그먼트를 가져옵니다.
6. 플레이어가 데이터 키를 사용하여 세그먼트의 암호를 해독한 후 미디어를 재생합니다.

> **Note**
>
> HLS 콘텐츠 보호를 사용하여 스트리밍된 파일의 세그먼트를 암호화하거나 전체 파일을 암호화할 수 있습니다. 둘 다 지원하지 않으며 HLS 콘텐츠 보호와 개별 파일 보호 중 하나만 선택할 수 있습니다.

HLS 콘텐츠 보호에 사용되는 키

Elastic Transcoder에서 HLS 콘텐츠 보호를 사용하려면 두 가지 유형의 키가 필요합니다.

- AWS KMS key - Elastic Transcoder 파이프라인과 관련된 키
- 데이터 키 - Elastic Transcoder 작업과 연결된 키

HLS 콘텐츠 보호를 사용하려면 AWS KMS key 가 있어야 합니다. KMS 키를 사용하여 데이터 키를 인터넷으로 보내기 전 암호화합니다. 모든 트랜스코딩 작업에 사용할 KMS 키를 하나 만드세요. KMS 키를 만들고 설정하는 방법은 Elastic Transcoder AWS KMS 와 함께 사용 섹션을 참조하세요.

데이터 키는 미디어 파일을 암호화하는 데 사용됩니다. 동일한 콘텐츠의 모든 변형 버전과 세그먼트가 동일한 데이터 키를 이용해 암호화됩니다. 데이터 키를 지정하지 않는 경우 Elastic Transcoder가 데이터 키를 생성합니다.

HLS 보호 콘텐츠 스트리밍

HLS 보호 콘텐츠를 전송하려면 다음이 필요합니다.

- 암호화된 미디어 파일과 데이터 키의 저장 위치. Amazon S3에 파일을 저장하고 DynamoDB 등과 같은 데이터베이스에 키를 보호하세요. DynamoDB에 대한 자세한 내용은 Amazon DynamoDB 개발자 안내서의 What is Amazon DynamoDB?를 참조하세요.
- (선택 사항) 파일을 스트리밍하는 CDN(Content Distribution Network). CDN에 대한 자세한 내용은 Amazon CloudFront 개발자 안내서의 Amazon CloudFront 시작하기를 참조하세요.
- 사용자를 인증하고 권한을 부여하며, 데이터 암호화 키의 보안을 지원하는 애플리케이션. Amazon EC2를 사용하여 이 애플리케이션을 실행할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2 설정(Windows 사용자의 경우) 또는 Amazon EC2 사용 설명서의 Amazon EC2 설정(Linux 사용자의 경우)을 참조하세요. Amazon EC2
- 암호화된 HLS 파일의 암호를 해독하는 플레이어. 자세한 내용은 Http 라이브 스트리밍 섹션을 참조하세요.

암호화된 스트리밍 콘텐츠 만들기

HLS 콘텐츠 보호를 사용할 파일을 준비하려면, KMS 키를 새로운 또는 기존 파이프라인에 연결해야 합니다.

KMS 키를 사용하여 지정한 파이프라인을 설정하려면 Elastic Transcoder AWS KMS 와 함께 사용 섹션을 참조하세요.

다음 단계에는 Elastic Transcoder 콘솔을 사용하여 HLS 콘텐츠 보호를 사용할 파일을 암호화하는 방법이 나와 있습니다.

파일에 HLS 콘텐츠 보호를 사용하려면

1. https://console.aws.amazon.com/elastictranscoder/에서 Elastic Transcoder 콘솔을 엽니다.
2. 탐색 창에서 Jobs를 클릭한 후 새 작업을 만듭니다. 자세한 내용은 Elastic Transcoder에서 작업 생성 섹션을 참조하세요.
3. Output Details의 Preset 드롭다운 목록에서 HLS 프리셋을 선택합니다.
4. Encryption Parameters를 None으로 설정된 상태로 둡니다.
5. Playlists에서 Add Playlist를 클릭한 후 HLSv3 또는 HLSv4를 재생 목록 유형으로 선택합니다.
6. Content Protection에서 Enter Information을 선택합니다.

   a. 본인 키를 관리하려면 키 스토리지 정책에서 No Store를 선택합니다. License Acquisition Url에서 절대 경로에 데이터 키를 저장할 위치를 입력합니다. 예시:

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

No Store를 선택하고 안전한 Amazon S3 버킷이나 DynamoDB 등의 데이터베이스에 키를 저장하세요.

b. 공개 Amazon S3 버킷에 키를 저장하려면 Key Storage Policy에서 With Variant Playlists를 선택합니다. Elastic Transcoder는 재생 목록 파일과 동일한 버킷에 데이터 키를 기록합니다.

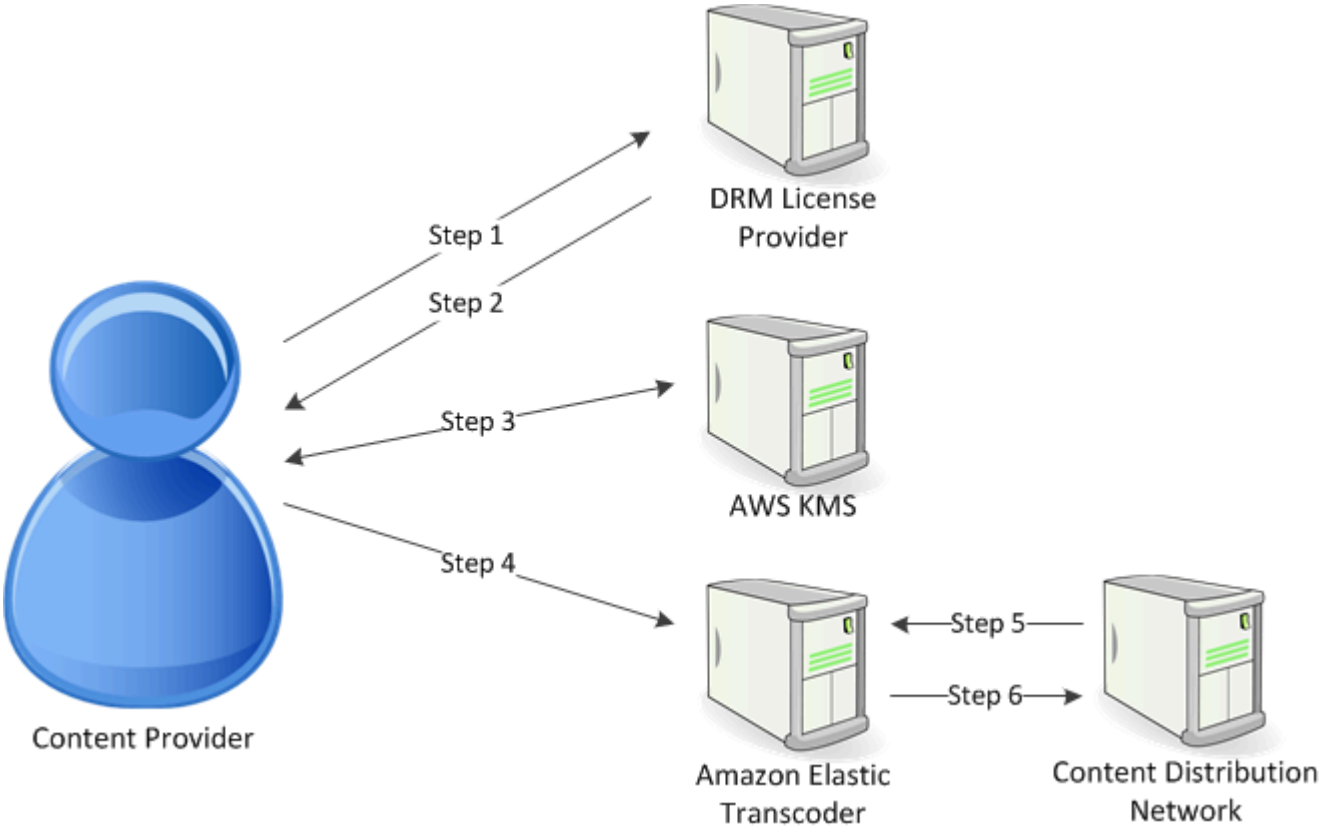
> **Important**
>
> With Variant Playlists를 사용하여 저장된 키는 퍼블릭 버킷에 기록됩니다. 실제 키에 No Store를 사용합니다.

> **Note**
>
> No Store를 선택하면 Elastic Transcoder는 작업 객체의 일부로 데이터 키를 반환하고 이 키를 보관하지는 않습니다. 데이터 키는 직접 보관해야 합니다.

디지털 권한 관리

DRM(Digital Rights Management)을 사용하여 재생 목록에 대한 액세스를 제한함으로써 승인된 사용자만 콘텐츠를 보도록 할 수 있습니다.

DRM은 타사 DRM 라이선스 서버에서 승인한 사용자만 파일 재생 목록을 액세스할 수 있도록 합니다. 이를 위해 DRM은 파일의 세그먼트에 DRM 헤더를 포함시키거나 패키징합니다. DRM 패키징에는 라이선스 서버와 연결하는 데 필요한 정보 및 파일을 재생하는 데 필요한 암호화 정보가 들어 있습니다. 파일이 패키징되면 Amazon S3 및 CloudFront와 같은 서비스 또는 Akamai와 같은 비AWS 서비스가 될 수 있는 콘텐츠 배포 네트워크로 전송됩니다. 파일이 배포되면 라이선스를 위해 라이선스 서버에 접속해야 파일을 재생할 수 있습니다. DRM은 콘텐츠 보호를 위한 통합 프로토콜을 지정함으로써 표준 암호화 이상의 기능을 제공합니다. 다음 그림은 DRM의 기본 작동 방식을 보여줍니다.



Elastic Transcoder가 DRM 패키징을 처리합니다(이 프로세스의 마지막 세 단계).

1. 콘텐츠 제공자가 콘텐츠 키 ID로 DRM 라이선스 제공자를 호출하여 콘텐츠 키를 생성합니다.
2. 콘텐츠 제공자가 콘텐츠 키 ID로 URL 라이선스 제공자를 호출하여 콘텐츠 키를 생성합니다.
3. 콘텐츠 공급자는를 호출 AWS KMS 하여 콘텐츠 키를 암호화하고 암호화된 콘텐츠 키를 AWS KMS 반환합니다.
4. 콘텐츠 제공자가 암호화된 콘텐츠 키, 키 ID, 라이선스 취득 URL로 패키지 서버(Elastic Transcoder)를 호출합니다.
5. 패키지 서버가 보호할 파일을가져와 DRM 정보를 포함시킵니다.
6. 패키지 서버가 배포 서버로 DRM 보호 파일을 보내고, 서버는 파일을 배포합니다.

주제

- PlayReady DRM

PlayReady DRM

Elastic Transcoder는 PlayReady DRM을 사용하여 DRM을 지원합니다. 이를 통해 미디어 콘텐츠를 보호하고 권한이 있는 사용자만 미디어 파일을 볼 수 있습니다. PlayReady 보호 파일의 인증과 재생 흐름은 특정 플레이어 및 라이선스 서버 설정에 따라 다릅니다.

DRM(Digital Rights Management)용 키

Elastic Transcoder에서 DRM을 사용하려면 두 가지 유형의 키와 하나의 키 ID가 필요합니다.

- 콘텐츠 키 - DRM 라이선스 서버에서 제공하는 키이며 해당 Elastic Transcoder 작업과 연결되어 있습니다.
- 키 ID - DRM 라이선스 서버에서 제공한 키의 ID이며, 라이선스 서버는 이 키 ID를 사용하여 암호를 해독하는 데 필요한 콘텐츠 키를 식별합니다.
- AWS KMS 키 - Elastic Transcoder 파이프라인과 연결된 AWS KMS 키

DRM을 사용하려면 콘텐츠 키가 있어야 합니다. 콘텐츠 키는 미디어 파일을 암호화하는 데 사용됩니다. 동일한 콘텐츠의 모든 변형 버전과 세그먼트가 동일한 콘텐츠 키를 이용해 암호화됩니다. Elastic Transcoder는 이 키를 자동으로 생성하지 않으므로 DRM 라이선스 서버에서 가져와야 합니다. 또한 키 ID가 있어야 Elastic Transcoder가 보호된 재생 목록에 이 키를 패키징하여 플레이어가 라이선스 서버에서 콘텐츠 키를 받아올 수 있습니다.

DRM을 사용하려면 AWS KMS 키가 있어야 합니다. AWS KMS 키는 인터넷을 통해 전송되기 전에 콘텐츠 키를 암호화하는 데 사용됩니다. 모든 트랜스코딩 작업에 사용할 하나의 AWS KMS 키를 생성하는 것이 좋습니다. AWS KMS 키 생성 및 설정에 대한 자세한 내용은 섹션을 참조하세요Elastic Transcoder AWS KMS 와 함께 사용.

DRM 보호 콘텐츠 스트리밍

DRM 콘텐츠를 전송하려면 다음이 필요합니다.

- DRM 콘텐츠 키를 생성하고 보관하는 DRM 라이선스 공급자.
- 암호화된 미디어 파일을 저장할 위치. Amazon S3에 파일을 저장하는 것이 좋습니다.
- (선택 사항) 파일을 스트리밍하는 CDN(Content Distribution Network). CDN에 대한 자세한 내용은 Amazon CloudFront 개발자 안내서의 Amazon CloudFront 시작하기를 참조하세요.
- 사용자를 인증하고 승인할 수 있는 애플리케이션. Amazon EC2를 사용하여 이 애플리케이션을 실행할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2 설정(Windows 사용자

의 경우) 또는 Amazon EC2 사용 설명서의 Amazon EC2 설정(Linux 사용자의 경우)을 참조하세요. Amazon EC2

- DRM 보호 및 암호화 파일을 재생할 수 있는 플레이어.

> **Note**
>
> HLS에서 PlayReady DRM을 사용하려면 사용자 지정 플레이어가 있어야 합니다.

DRM으로 보호되는 스트리밍 콘텐츠 만들기

DRM용 파일을 준비하려면를 새 파이프라인 또는 기존 파이프라인 AWS KMS 과 연결해야 합니다.

지정한 로 파이프라인 AWS KMS 을 설정하려면 섹션을 참조하세요Elastic Transcoder AWS KMS 와 함께 사용.

다음 단계에는 Elastic Transcoder 콘솔을 사용하여 DRM 파일을 암호화하는 방법이 나와 있습니다.

파일에 DRM을 사용하려면

1. https://console.aws.amazon.com/elastictranscoder/에서 Elastic Transcoder 콘솔을 엽니다.
2. 탐색 창에서 Jobs를 클릭한 후 새 작업을 만듭니다. 자세한 내용은 Elastic Transcoder에서 작업 생성 섹션을 참조하세요.
3. Encryption Parameters를 None으로 설정된 상태로 둡니다.
4. Playlists에서 Add Playlist를 클릭한 후 Smooth 또는 HLSv3를 재생 목록 유형으로 선택합니다.
5. PlayReady DRM에서 `Enter Information`을 선택합니다.

   a. 라이선스 획득 Url에 DRM 서버의 절대 경로를 입력합니다.

   b. 키 ID에 콘텐츠 키 ID를 입력합니다.

   c. 암호화 키에 base64 인코딩 형식의 콘텐츠 키를 입력합니다.

   d. 암호화 키 MD5에 해당 콘텐츠 키의 base64 인코딩 형식 MD5 해시를 입력합니다.

   e. 암호화 초기화 벡터에 base64 인코딩 형식 초기화 벡터를 입력합니다.

> **Note**
>
> Elastic Transcoder는 작업 객체의 일부로 콘텐츠 키를 반환하며, 이를 저장하지는 않습니다. 콘텐츠 키를 DRM 서버에 보관하는 작업은 사용자의 책임입니다.

작업 상태 알림

Elastic Transcoder는 작업 상태가 변경되면 알림을 보냅니다. 다음 상태 변화의 조합(또는 없음)에 대해 알림을 보내도록 Elastic Transcoder를 구성할 수 있습니다.

- 처리 중: Elastic Transcoder가 파이프라인에서 작업 처리를 시작했습니다.
- 완료됨: 가 파이프라인에서 작업 처리를 완료했습니다.
- 경고: 파이프라인에서 작업을 처리하는 도중 Elastic Transcoder에 경고 상태가 발생했습니다.
- 오류: 파이프라인에서 작업을 처리하는 도중 Elastic Transcoder에 오류 상태가 발생했습니다.

Elastic Transcoder는 Amazon Simple Notification Service(SNS)를 사용하여 알림을 보냅니다. Amazon SNS는 HTTP 엔드포인트, 이메일 주소 및 Amazon Simple Queue Service(Amazon SQS) 대기열에 메시지를 보내는 기능을 포함하여 다양한 알림 옵션을 제공합니다. 이러한 옵션 및 기타 옵션에 대한 자세한 내용은 Amazon Simple Notification Service 개발자 가이드를 참조하세요.

알림은 이벤트 중심 애플리케이션을 설계하는 데 유용합니다. 알림을 사용하여 작업이 완료된 시간을 확인하는 경우 폴링할 필요가 없으며 폴링으로 인해 가끔 발생하는 `Limit Exceeded` 예외(HTTP 상태 코드 429)가 발생하지 않습니다.

알림은 파이프라인을 만들거나 업데이트할 때 구성합니다. Elastic Transcoder는 해당 파이프라인에 제출하는 모든 작업에 관련 알림을 보냅니다.

> **Important**
>
> 알림을 변경하면 변경 사항이 즉시 적용됩니다. 알림을 변경한 후 제출한 작업 외에도 이미 제출했으나 Elastic Transcoder가 처리를 시작하지 않은 작업도 영향을 받습니다.

Elastic Transcoder API를 사용할 때 작업 상태의 변경 사항을 알리도록 Elastic Transcoder를 구성하려면

1. Amazon SNS 주제를 하나 이상 만들어 각 주제를 구독합니다. 자세한 내용은 Amazon Simple Notification Service 설명서를 참조하세요.

   사용할 Amazon SNS 주제가 이미 있으면 이 단계를 건너뛸 수 있습니다.

2. 파이프라인을 만들거나 기존 파이프라인을 업데이트합니다. 각 상태 변경에 대한 알림(처리 중, 완료, 경고, 오류)을 보내려면 1단계에서 만든 해당 Amazon SNS 주제를 지정합니다. 자세한 내용은 파이프라인 생성, 파이프라인 업데이트 또는 파이프라인 알림 업데이트 섹션을 참조하세요.
3. Elastic Transcoder로 작업을 제출하여 `Progressing` 및 `Complete` 상태에 대한 알림을 테스트합니다.

Elastic Transcoder 콘솔을 사용하여 알림을 구성할 수도 있습니다. 콘솔을 사용하는 경우 파이프라인을 만들거나 업데이트할 때 Amazon SNS 주제를 만들 수 있습니다. 그러나 새로운 주제를 구독하려면 Amazon SNS 콘솔, API 또는 CLI를 사용해야 합니다.

Elastic Transcoder가 상태 변화에 대한 알림을 보낼 때, 알림에 반환된 메시지는 다음과 같은 JSON 형식입니다. 메시지에는 각 출력의 상태와 해당 작업의 상태가 나와 있습니다.

```
{
   "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"
   }
```

```
}
```

클립 스티칭

여러 입력 파일의 부분 또는 클립을 연결하여 하나의 출력을 생성하도록 Elastic Transcoder를 구성할 수 있습니다. 이를 통해 개별적인 레코딩 세션 또는 게임의 하이라이트와 같은 서로 다른 입력에서 하나의 출력을 생성할 수 있습니다.

여러 입력의 클립을 결합하려면 작업을 만들 때 여러 입력을 지정하세요. Elastic Transcoder는 작업 요청에 지정한 순서대로 클립을 하나로 연결합니다. 여러 입력을 연결하지 않고 트랜스코딩하려면 각 입력에 대해 개별 작업을 만듭니다.

지정한 각 입력에 대해 시작 시간과 지속시간을 지정할 수 있습니다. 이를 통해 출력에 포함시킬 각 입력의 부분만 연결할 수 있습니다. 출력의 시작과 끝을 따로 떼어내려면 첫 번째 입력의 클립 설정과 마지막 입력의 클립 설정을 지정합니다. 예를 들어 출력의 시작에 있는 트레일러와 출력의 끝에 있는 크레딧을 제거하려면, 트레일러를 제거하는 첫 번째 입력의 시작 시간과 지속시간, 크레딧을 제거하는 마지막 입력의 시작 시간과 지속시간을 지정합니다.

Elastic Transcoder가 연결할 수 있도록 파일은 일관된 오디오 및 비디오 설정을 공유해야 합니다. 예를 들어 입력의 비디오 종횡비가 서로 다른 경우, 프리셋은 Elastic Transcoder가 출력의 일관된 종횡비를 생성할 수 있도록 패딩 정책 또는 크기 조정 정책을 사용해야 합니다. 이러한 정책이 종횡비에 주는 영향에 대한 자세한 내용은 크기 조정 정책 및 종횡비를 참조하세요. 또한 모든 입력에 비디오가 있거나 어떤 입력에도 비디오가 없어야 하며, 모든 입력에 오디오가 있거나 어떤 입력에도 오디오가 없어야 합니다. 마찬가지로 캡션 설정이 일관돼야 합니다. 입력 전체에서 동일한 언어의 캡션은 언어에 대해 동일한 값을 지정해야 하며, 그렇지 않을 경우 동일한 캡션 트랙의 부분으로 간주되지 않습니다.

워터마크의 배치는 프리셋에 콘텐츠 대상이 있는지, 프레임 대상이 있는지에 따라 결정됩니다. 콘텐츠 대상은 워터마크가 원래 입력의 콘텐츠에 따라 배치되고 패딩 정책에서 추가한 패딩은 무시함을 뜻합니다. 예를 들어 한 클립에 패딩이 필요하지 않고 워터마크가 왼쪽 상단에 있는 경우, 해당 클립의 워터마크는 비디오 가장자리에서 지정된 거리만큼 떨어진 곳에 배치됩니다. 클립에 패딩이 필요한 경우, 워터마크는 패딩의 가장자리에서 지정된 거리만큼 떨어진 곳에 배치됩니다. 프레임 대상은 워터마크가 출력의 콘텐츠에 따라 배치됨을 뜻합니다. 이는 항상 비디오 가장자리에서 지정된 거리만큼 떨어진 곳에 배치됩니다. 워터마크에 대한 자세한 내용은 워터마크 설정 섹션을 참조하세요.

Elastic Transcoder는 모든 입력을 연결한 후 썸네일을 생성합니다.

캡션

Elastic Transcoder가 캡션(자막)을 한 형식에서 다른 형식으로 트랜스코딩하도록 구성할 수 있습니다. Elastic Transcoder는 두 종류의 캡션을 지원합니다.

- 임베디드: 임베디드 캡션은 오디오 및 비디오와 동일한 파일에 포함됩니다. Elastic Transcoder는 언어당 하나의 임베디드 캡션만 지원합니다.

  유효한 입력 형식은 CEA-608(EIA-608, 비어 있지 않은 첫 번째 채널만 해당), CEA-708(EIA-708, 비어 있지 않은 첫 번째 채널만 해당), mov-text입니다.

  유효한 출력 값은 mov-text(MP4만 해당)와 CEA-708(MPEG-TS 및 MP4, 초당 29.97 및 30프레임만 해당)입니다. CEA-708 캡션은 스트림의 H.264 SEI 사용자 데이터에 포함됩니다.

  Elastic Transcoder는 출력당 최대 1개의 임베디드 형식을 지원합니다.
- Sidecar: 사이드카 캡션은 오디오 및 비디오 데이터와 별도의 메타데이터 파일로 유지됩니다. 사이드카 캡션은 비디오 파일과 사이드카 파일 간 관계를 이해할 수 있는 플레이어가 있어야 합니다. Elastic Transcoder는 언어당 하나의 사이드카 캡션만 지원하며 파일당 최대 20개의 사이드카 캡션을 지원합니다.

  유효한 입력 형식은 DFXP(첫 번째 div 요소만 해당), EBU-TT, SCC, SMPT, SRT, TTML(첫 번째 div 요소만 해당), WebVTT입니다.

  유효한 출력 형식은 DFXP(첫 번째 div 요소만 해당), SCC, SRT, WebVTT입니다.

TTML 또는 SMPTE-TT 호환 캡션을 사용하려는 경우 출력 형식으로 DFXP를 지정하세요.

Elastic Transcoder는 OCR(광학식 문자 인식)을 지원하지 않고, 화상을 유효한 캡션 입력으로 허용하지 않으며, 오디오 전용 트랜스코딩에 사용할 수 없습니다. Elastic Transcoder는 트랜스코딩 프로세스 도중 텍스트 서식 지정(예: 기울임꼴)을 보존하지 않습니다.

Elastic Transcoder 콘솔을 사용할 경우 캡션 트랜스코딩에 대한 자세한 내용은 출력 세부 정보, 3부, 캡션 설정 섹션을 참조하세요. Elastic Transcoder API를 사용할 때 캡션을 트랜스코딩하는 방법에 대한 자세한 내용은 Captions 요소로 시작하는 API 작업에 대한 작업 생성 설명서를 참조하세요.

임베디드 파일에 대한 자세한 내용은 Wikipedia의 자막(캡션) 페이지를 참조하세요.

사이드카 파일에 대한 자세한 내용은 Wikipedia의 Metadata Platform 페이지와 Sidecar file페이지를 참조하세요.

워터마크

트랜스코딩 중에 비디오에 워터마크라고 하는 최대 네 개의 그래픽을 오버레이하도록 Elastic Transcoder를 구성할 수 있습니다. 예를 들어 텔레비전 채널이 가끔 방송에 표시하는 로고가 워터마크입니다.

워터마크는 지정한 크기, 위치, 축척, 불투명도로 트랜스코딩된 비디오의 지속시간 동안 표시됩니다. 불투명도는 워터마크의 밀도를 완전 투명(워터마크가 보이지 않음)에서 완전 불투명(워터마크 아래의 비디오가 보이지 않음)까지 제어합니다.

워터마크에 사용하는 그래픽은 .png 또는 .jpg 형식일 수 있습니다. 장방형이 아닌 워터마크를 표시하려면 투명도를 지원하는 .png 형식을 사용합니다. 트랜스코딩할 비디오를 저장한 Amazon S3 버킷에 그래픽을 저장합니다. 파이프라인 설정에서 입력 버킷인 이 버킷을 지정합니다.

워터마크를 추가하려면 프리셋 및 작업 모두에서 설정을 지정합니다.

- 프리셋에는 최대 4개의 워터마크의 크기, 위치 및 불투명도 설정이 포함되어 있습니다. 각 설정 그룹에 식별자를 할당합니다. 작업을 만들 때 이 식별자를 사용하여 어떤 프리셋 설정을 어떤 그래픽에 사용할지 제어할 수 있습니다.

  Elastic Transcoder 콘솔을 사용하여 프리셋에서 워터마크 설정을 지정하는 방법에 대한 자세한 내용은 Elastic Transcoder에서 프리셋 생성 섹션을 참조하세요. Elastic Transcoder API를 사용하여 설정을 지정하는 방법에 대한 자세한 내용은 프리셋 만들기 섹션을 참조하세요.
- 작업은 워터마크로 사용할 .png 또는 .jpg 이미지 및 각 워터마크에 사용할 프리셋의 설정 그룹을 식별합니다. 작업을 만들고 워터마크 설정이 포함된 프리셋을 지정할 때, 프리셋에서 정의한 워터마크 설정 중 일부 또는 모두에 그래픽을 추가하거나 하나도 추가하지 않을 수 있습니다.

  Elastic Transcoder 콘솔을 사용하여 작업에서 워터마크 설정을 지정하는 방법에 대한 자세한 내용은 Elastic Transcoder에서 작업 생성 섹션을 참조하세요. Elastic Transcoder API를 사용하여 설정을 지정하는 방법에 대한 자세한 내용은 작업 생성 섹션을 참조하세요.

디지털 오디오

디지털 오디오는 채널 형태로 저장, 전송 및 재생되며 사용자는 여러 채널을 트랙(스트림)으로 저장, 전송 및 재생할 수 있습니다. 채널 및 트랙 여러 개를 사용해 서라운드 사운드, 스테레오, 다중 언어 등 추가 정보를 파일에 포함시킬 수 있습니다. 예를 들어 채널이 6개인 서라운드 사운드 트랙과 채널이 2개인 스테레오 트랙이 있는 파일을 만들 수 있습니다. Elastic Transcoder를 사용하면 파일의 트랙 및 채널 수를 변경할 수 있습니다. 서라운드 사운드 6채널 트랙만 있는 파일이 있다면 Elastic Transcoder를 사용해 스테레오 2채널 트랙이 있는 파일을 생성할 수 있습니다.

1개 채널이 단일 스피커를 통해 재생되는 정보를 운반합니다. 예를 들어 2채널 스테레오 트랙은 한 채널을 좌측 스피커, 다른 채널을 우측 스피커로 보냅니다. 트랙을 사용하면 여러 채널 세트 중 선택이 가능하도록 할 수 있습니다. 예를 들어 한 트랙은 채널이 2개, 다른 트랙은 채널이 6개로 트랙이 총 2개인 파일을 만들 수 있습니다. 플레이어에서 이 파일을 재생하면 스테레오 스피커 시스템의 경우 2채널 트랙을, 서라운드 사운드 스피커 시스템의 경우 6채널 트랙을 사용합니다.

트랙 중에는 오디오 정보가 없는 것도 있습니다. 오디오가 없는 트랙은 MOS(Mit Out Sound) 트랙이라고 합니다. MOS 트랙과 음향 편집 프로그램을 사용하면 파일에 음향 효과 및 음악을 추가할 수 있습니다. 출력 파일이 MXF 컨테이너를 사용하는 경우 Elastic Transcoder를 사용해 파일에 MOS 트랙을 추가할 수 있습니다.

Elastic Transcoder의 기본값은 단일 트랙이지만 사용자가 트랙의 채널 수를 선택할 수 있습니다. 선택 가능한 트랙 수는 최대 2개이며, 입력의 채널 수를 전달할 수도 있습니다. Elastic Transcoder를 사용해 MXF 컨테이너가 있는 파일을 트랜스코딩할 때는 출력에 트랙 여러 개를 생성할 수 있습니다.

Elastic Transcoder 프리셋을 사용하여 다음을 수행할 수 있습니다.

- 출력에서 오디오 제거
- 모노 채널을 다중 채널로 복제
- 스테레오를 모노로 결합(다운믹싱)
- 서라운드 사운드를 스테레오 또는 모노로 다운믹싱
- 기존 채널 전달
- (MXF 전용) 출력에 MOS 트랙 추가

Elastic Transcoder를 사용한 오디오 트랜스코딩에 대한 자세한 내용은 Channels 섹션을 참조하세요.

시스템 프리셋

작업에 사용할 프리셋에 대한 `Id` 객체의 값. 이 프리셋은 Elastic Transcoder가 트랜스코딩에 사용하는 오디오, 비디오 및 썸네일 설정을 결정합니다. 생성한 프리셋을 사용하려면 프리셋을 생성했을 때 Elastic Transcoder가 응답으로 반환한 프리셋 ID를 지정합니다.

> **Note**
>
> AAC 프로필이 추가되기 전에 프리셋을 생성한 경우 Elastic Transcoder는 해당 프리셋에 AAC-LC 프로필을 사용합니다.

Elastic Transcoder에 포함된 시스템 프리셋을 사용하려면 다음 ID를 사용하세요. (이 ID는 프리셋 나열를 사용해 확인할 수도 있습니다.)

설명	PresetId
오디오 AAC - 256k	1351620000001-100110
오디오 AAC - 160k	1351620000001-100120
오디오 AAC - 128k	1351620000001-100130
오디오 AAC - 64k(`Audio:CodecOptions:Profile` 의 경우 auto 사용)	1351620000001-100141
오디오 MP3 - 320k	1351620000001-300010
오디오 MP3 - 192k	1351620000001-300020
오디오 MP3 - 160k	1351620000001-300030
오디오 MP3 - 128k	1351620000001-300040
오디오 WAV 44100Hz, 16비트	1351620000001-300300
오디오 WAV 44100Hz, 8비트	1351620000001-300200
Amazon Kindle Fire HDX	1351620000001-100150

설명	PresetId
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
Full HD 1080i50	1351620000001-100180
Full HD 1080i60	1351620000001-100190
Full HD 1080i50 - XDCAM422	1351620000001-100230
Full HD 1080i60 - XDCAM422	1351620000001-100220
일반 1080p	1351620000001-000001
일반 720p	1351620000001-000010
일반 480p 16:9	1351620000001-000020
일반 480p 4:3	1351620000001-000030
일반 360p 16:9	1351620000001-000040
일반 360p 4:3	1351620000001-000050
일반 320x240(`Audio:CodecOptions:Profile` 의 경우 auto 사용)	1351620000001-000061
Gif(애니메이션)	1351620000001-100200
HLS v3(Apple HTTP Live Streaming), 2Mbps	1351620000001-200010
HLS v3 및 v4(Apple HTTP Live Streaming), 2Mbps, 비디오 전용	1351620000001-200015

설명	PresetId
HLS v3(Apple HTTP Live Streaming), 1.5Mbps	1351620000001-200020
HLS v3 및 v4(Apple HTTP Live Streaming), 1.5Mbps, 비디오 전용	1351620000001-200025
HLS v3(Apple HTTP Live Streaming), 1Mbps	1351620000001-200030
HLS v3 및 v4(Apple HTTP Live Streaming), 1Mbps, 비디오 전용	1351620000001-200035
HLS v3(Apple HTTP Live Streaming), 600Kbps	1351620000001-200040
HLS v3 및 v4(Apple HTTP Live Streaming), 600Kbps, 비디오 전용	1351620000001-200045
HLS v3(Apple HTTP Live Streaming), 400Kbps	1351620000001-200050
HLS v3 및 v4(Apple HTTP Live Streaming), 400Kbps, 비디오 전용	1351620000001-200055
HLS v3 및 v4 오디오, 160k	1351620000001-200060
HLS v3 및 v4 오디오, 64k(`Audio:CodecOptions:Profile` 의 경우 auto 사용)	1351620000001-200071
iPhone 5, iPhone 4S, iPad 4G 및 3G, iPad mini, Samsung Galaxy S2/S3/Tab 2	1351620000001-100020
iPod touch, iPhone 3 및 1, iPod classic	1351620000001-100040
MPEG-DASH 오디오 128k	1351620000001-500060
MPEG-DASH 비디오 600k	1351620000001-500050
MPEG-DASH 비디오 1200k	1351620000001-500040
MPEG-DASH 비디오 2400k	1351620000001-500030
MPEG-DASH 비디오 4800k	1351620000001-500020

설명	PresetId
NTSC	1351620000001-100160
PAL	1351620000001-100170
Smooth Streaming 3Mbps	1351620000001-400010
Smooth Streaming 2Mbps	1351620000001-400020
Smooth Streaming 1.5Mbps	1351620000001-400030
Smooth Streaming 1Mbps	1351620000001-400040
Smooth Streaming 800Kbps	1351620000001-400050
Smooth Streaming 600Kbps	1351620000001-400060
Smooth Streaming 500Kbps	1351620000001-400070
Smooth Streaming 400Kbps	1351620000001-400080
웹: Facebook, SmugMug, Vimeo, YouTube	1351620000001-100070
웹: 플래시 비디오	1351620000001-100210
Webm 720p	1351620000001-100240
Webm VP9 720p	1351620000001-100250
Webm VP9 360p	1351620000001-100260

다음 시스템 프리셋은 더 이상 지원되지 않으며 사용하지 않는 것이 좋습니다.

설명	지원 중단 PresetId	권장 PresetId
오디오 AAC - 64k(`Audio:CodecOptions:Profile` 의 경우 AAC-LC 사용)	1351620000001-100140	1351620000001-100141

설명	지원 중단 PresetId	권장 PresetId
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
일반 1080p	1351620000000-000001	1351620000001-000001
일반 720p	1351620000000-000010	1351620000001-000010
일반 480p 16:9	1351620000000-000020	1351620000001-000020
일반 480p 4:3	1351620000000-000030	1351620000001-000030
일반 360p 16:9	1351620000000-000040	1351620000001-000040
일반 360p 4:3	1351620000000-000050	1351620000001-000050
일반 320x240(`Audio:CodecOptions:Profile` 의 경우 AAC-LC 사용)	1351620000001-000060	1351620000001-000061
일반 320x240	1351620000000-000060	1351620000001-000060
HLS 오디오, 64k(`Audio:CodecOptions:Profile` 의 경우 AAC-LC 사용)	1351620000001-200070	1351620000001-200071
iPhone 4, iPod touch 5G 및 4G, iPad 2G 및 1G	1351620000001-100010	1351620000001-100020
iPhone4S	1351620000000-100020	1351620000001-100020

설명	지원 중단 PresetId	권장 PresetId
iPhone4	1351620000000-100010	1351620000001-100020
iPhone 3GS	1351620000001-100030	1351620000001-100020
iPhone3GS	1351620000000-100030	1351620000001-100020
iPod Touch	1351620000000-100040	1351620000001-100040
웹	1351620000000-100070	1351620000001-100070

크기 조정 정책 및 종횡비

다음 표에 나와 있는 것처럼 선택한 Sizing Policy는 Elastic Transcoder가 출력 이미지에 적용하는 조정에 영향을 미칩니다.

크기 조정 정책	출력 이미지가 확대될 수 있음	패딩 정책이 "Pad"일 경우 출력 이미지가 패딩될 수 있음	출력 이미지의 픽셀 종횡비가 입력 이미지와 다를 수 있음	출력 이미지가 잘릴 수 있음
Fit	예	예		
Fill	예			예
Stretch	예		예	
Keep		예		예
ShrinkToFit		예		
ShrinkToFill		예		예

종횡비 썸네일

다음 표에서는 Sizing Policy, Padding Policy, Max Height, Max Width가 상호 작용하여 출력 이미지를 변경하는 방법을 보여 줍니다.

주제

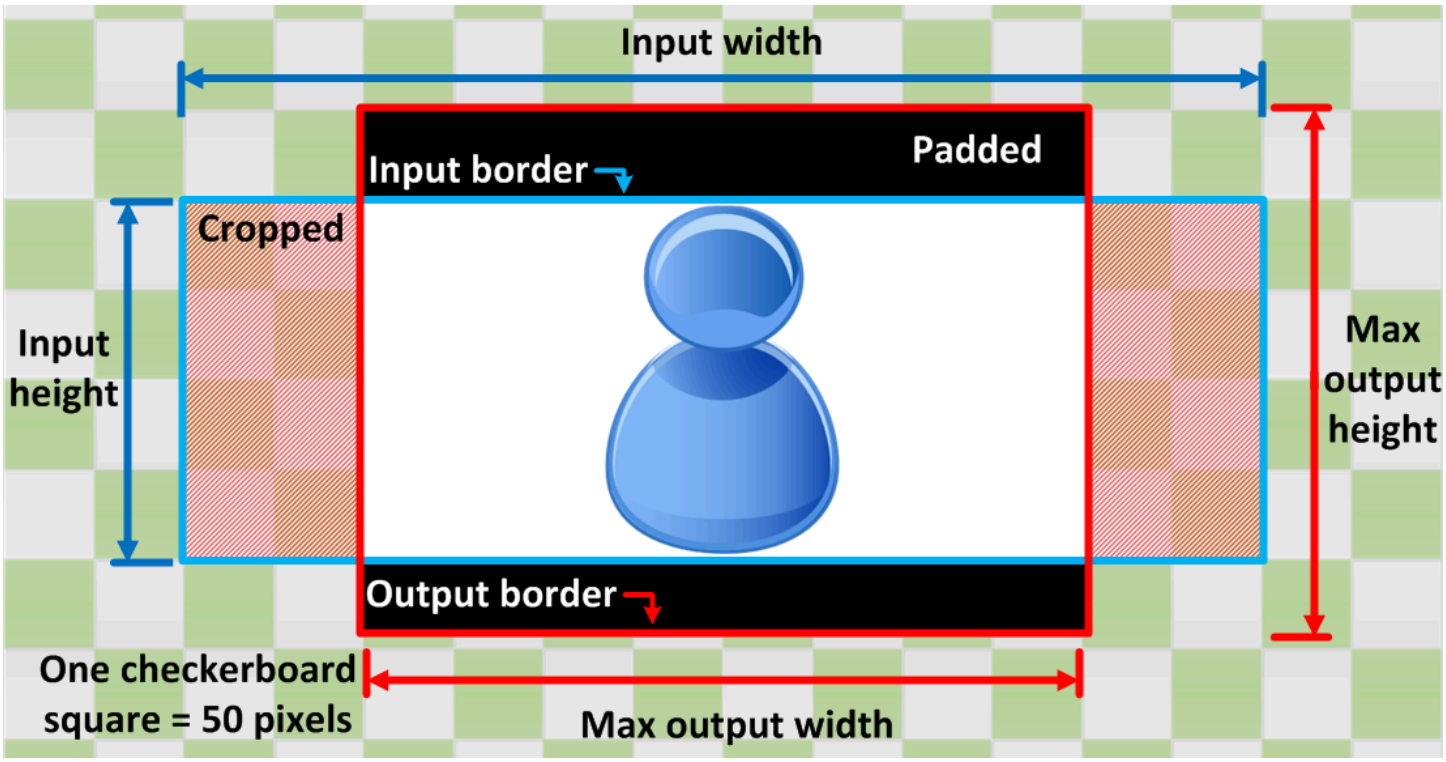
- Fit
- Fill
- Stretch
- Keep
- Shrink to Fit
- Shrink to Fill

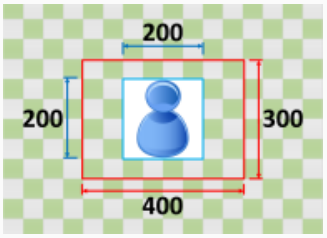
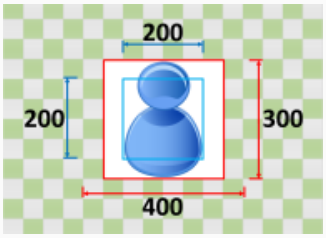
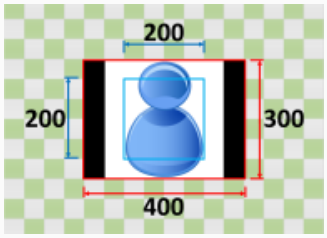
Fit

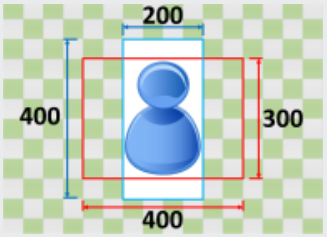
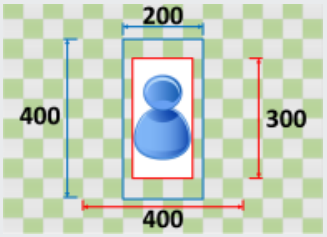
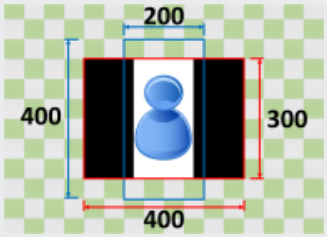
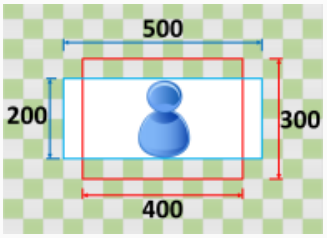
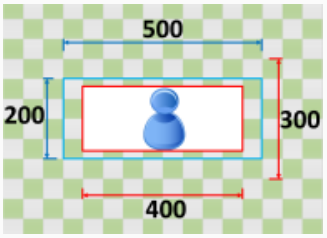
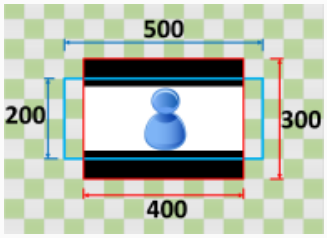
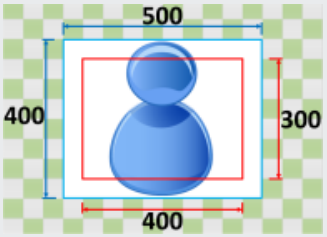
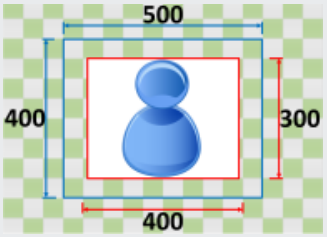
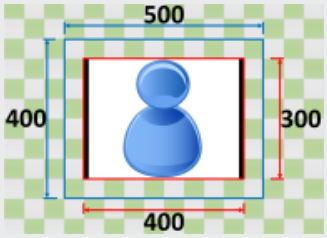
Sizing Policy로 Fit을 선택하면 Elastic Transcoder는 출력 이미지의 크기를 초과하지 않고 출력 이미지의 크기에 맞을 때까지 입력 파일의 크기를 조정합니다.

예를 들어 입력 파일이 200픽셀 x 200픽셀이고 출력 이미지를 300픽셀 x 400픽셀로 하려는 경우, Elastic Transcoder는 파일의 크기를 300픽셀 x 300픽셀로 늘리고 파일 양쪽에 패딩 정책을 적용합니다. Padding Policy로 Unpadded를 선택하면 Elastic Transcoder는 300픽셀 x 300픽셀 파일을 출력으로 반환합니다. Padded를 선택하면 Elastic Transcoder는 출력의 양쪽에 50픽셀의 패딩을 추가하고, 300픽셀 x 400픽셀 파일을 반환합니다.

Key(키)



Condition	입력	출력: NoPad	출력: Pad
입력 너비 < 최대 출력 너비 입력 높이 < 최대 출력 높이			

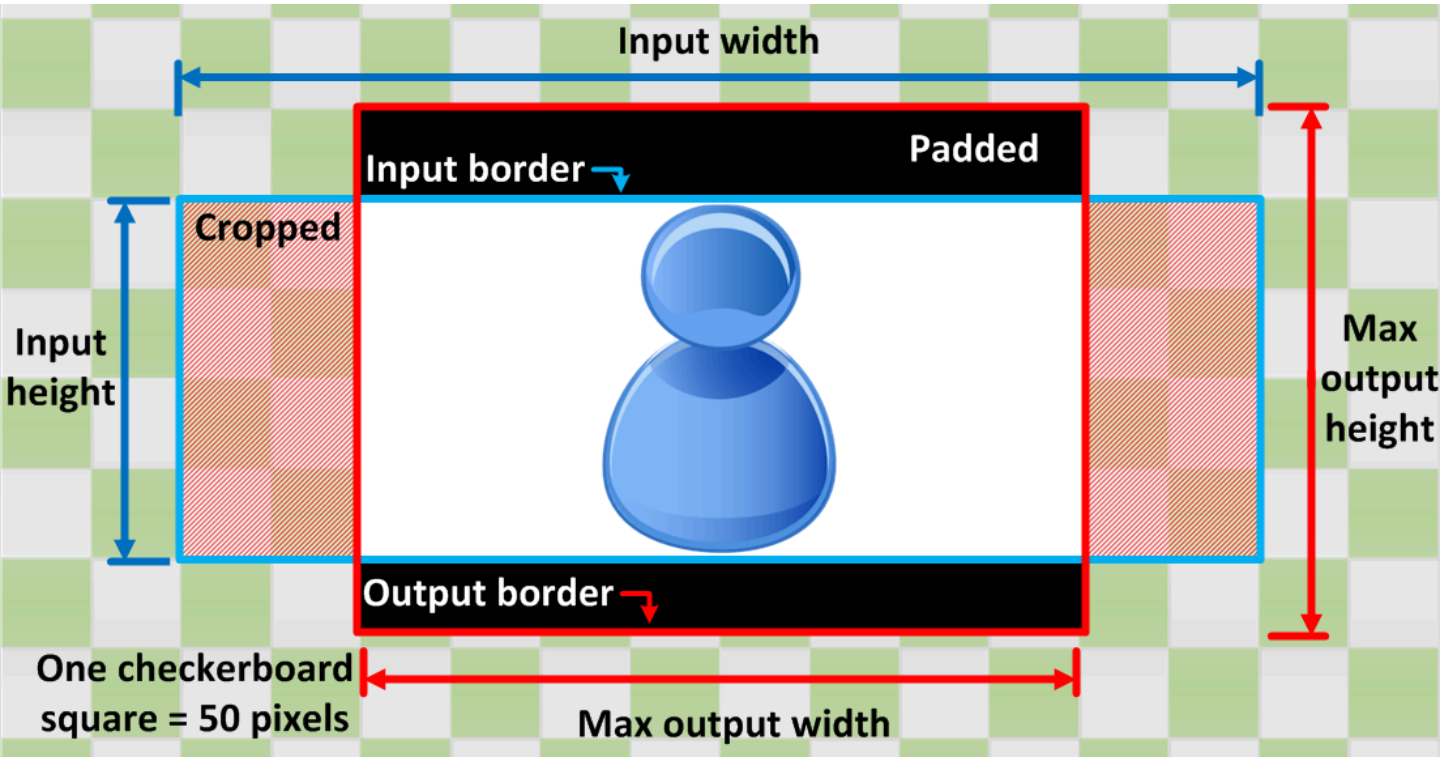
Condition	입력	출력: NoPad	출력: Pad
입력 너비 < 최대 출력 너비 입력 높이 > 최대 출력 높이			
입력 너비 > 최대 출력 너비 입력 높이 < 최대 출력 높이			
입력 너비 > 최대 출력 너비 입력 높이 > 최대 출력 높이			

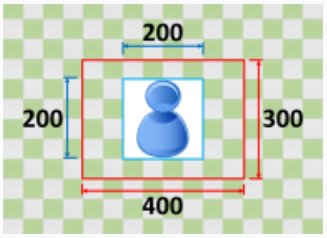
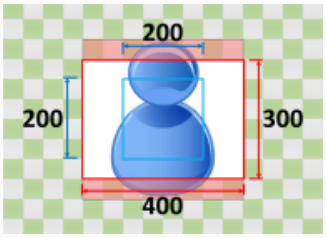
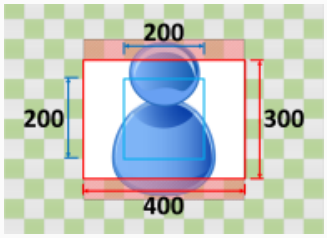
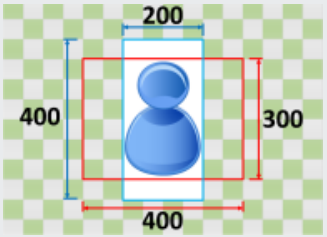
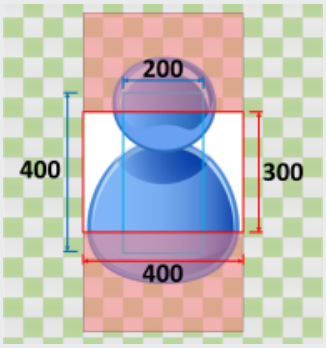
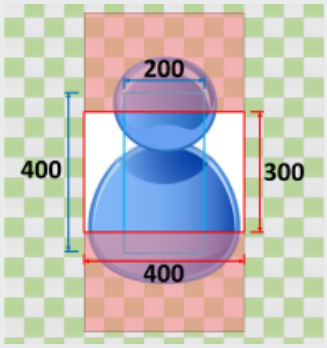
Fill

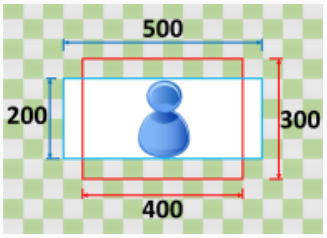
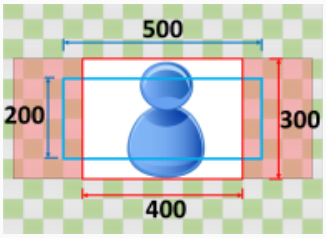
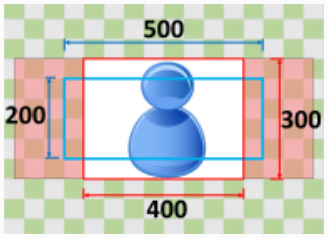
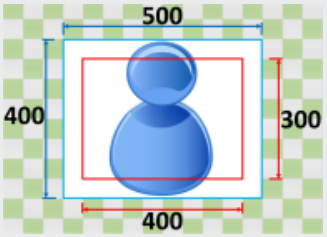
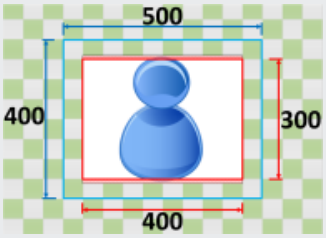
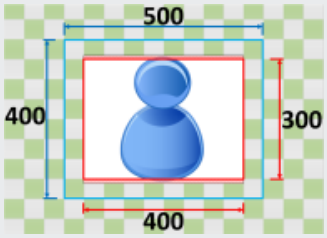
Sizing Policy로 Fill을 선택하면 Elastic Transcoder는 출력 이미지의 크기를 채울 때까지 입력 파일의 크기를 조정하고 출력 이미지 크기를 초과하는 모든 것을 잘라냅니다.

예를 들어 입력 파일이 200픽셀 x 200픽셀이고 출력 이미지를 300픽셀 x 400픽셀로 하려는 경우, Elastic Transcoder는 입력의 크기를 400픽셀 x 400픽셀로 늘리고, 상단 및 하단을 50픽셀씩 잘라낸 후, 300픽셀 x 400픽셀 파일을 반환합니다. Elastic Transcoder는 Fill 정책에 패딩을 사용하지 않습니다.

Key(키)



Condition	입력	출력: NoPad	출력: Pad
입력 너비 < 최대 출력 너비 입력 높이 < 최대 출력 높이			
입력 너비 < 최대 출력 너비 입력 높이 > 최대 출력 높이			

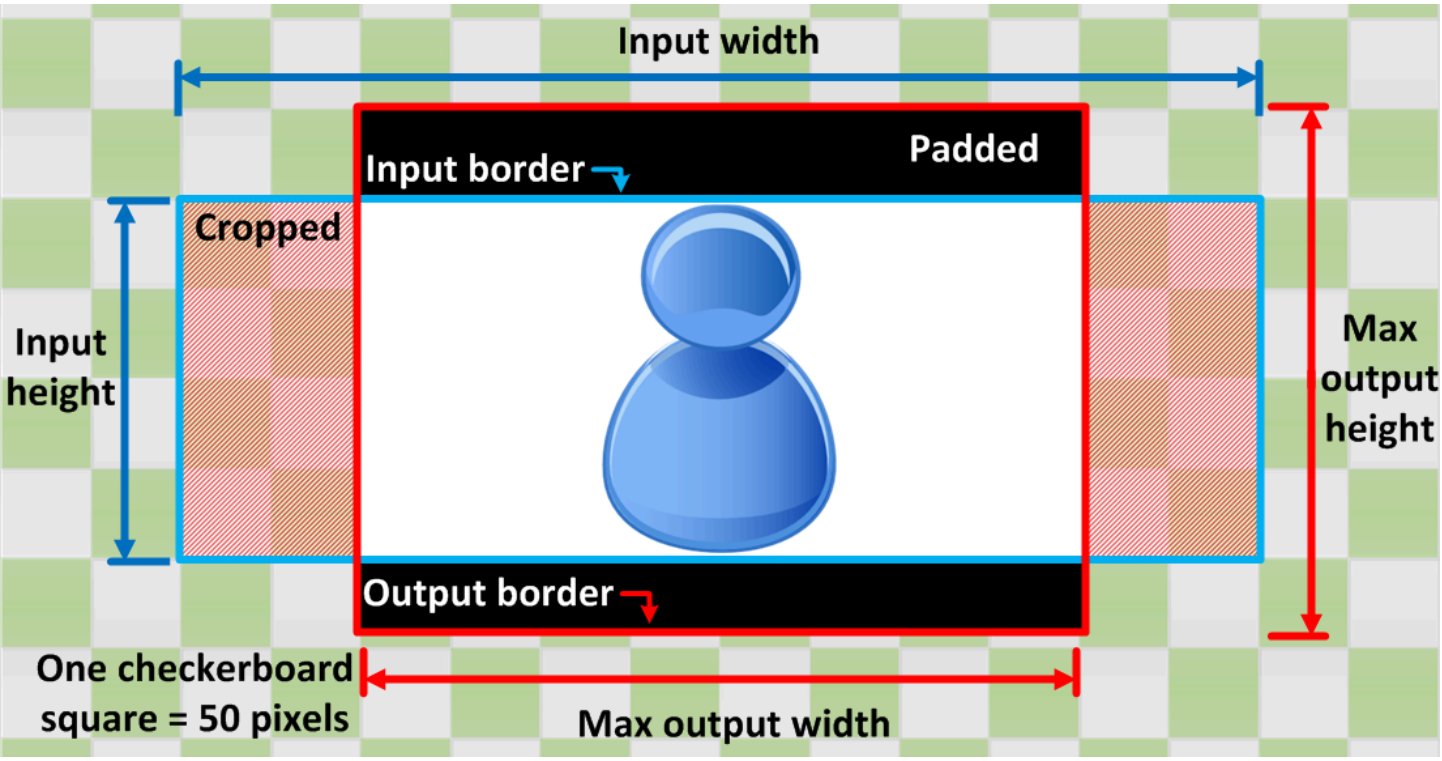
Condition	입력	출력: NoPad	출력: Pad
입력 너비 > 최대 출력 너비 입력 높이 < 최대 출력 높이			
입력 너비 > 최대 출력 너비 입력 높이 > 최대 출력 높이			

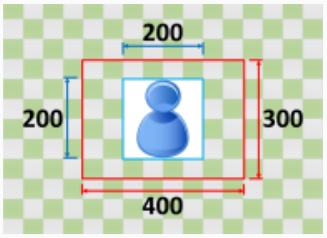
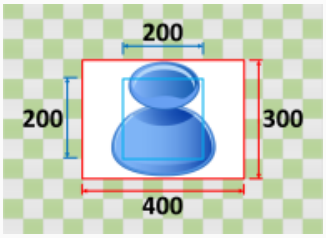
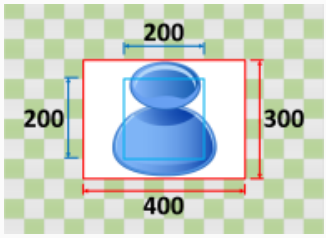
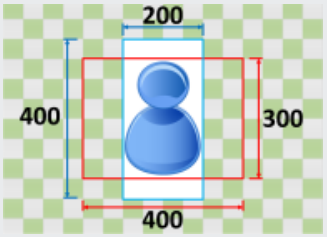
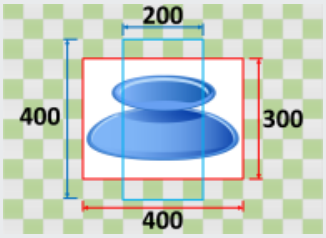
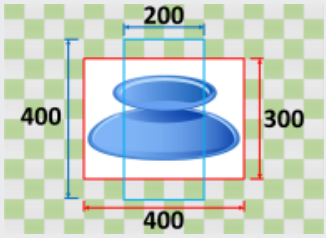
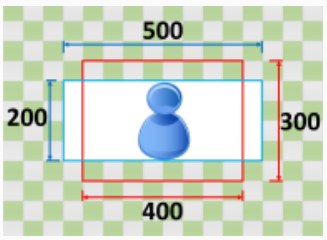
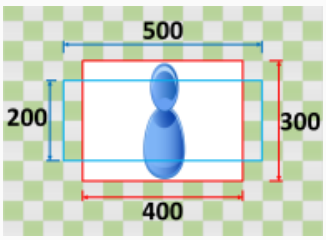
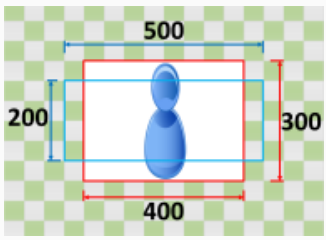
Stretch

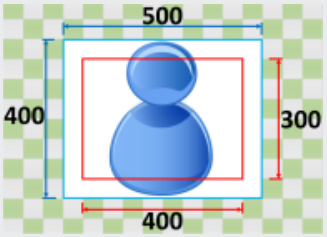
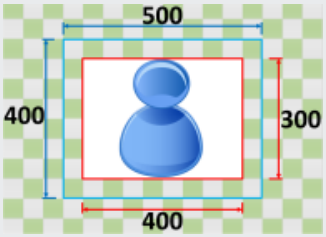
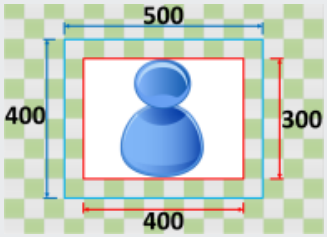
Sizing Policy로 Stretch를 선택하면 Elastic Transcoder는 출력 파일의 크기와 일치할 때까지 입력 파일을 늘리거나 줄입니다.

예를 들어 입력 파일이 200픽셀 x 200픽셀이고 출력 이미지를 300픽셀 x 400픽셀로 하려는 경우 Elastic Transcoder는 입력의 크기를 300픽셀 x 400픽셀로 늘려 출력 이미지의 비율을 바꿉니다. Elastic Transcoder는 Stretch 정책에 패딩이나 크롭을 사용하지 않습니다.

Key(키)



Condition	입력	출력: NoPad	출력: Pad
입력 너비 < 최대 출력 너비 입력 높이 < 최대 출력 높이			
입력 너비 < 최대 출력 너비 입력 높이 > 최대 출력 높이			
입력 너비 > 최대 출력 너비 입력 높이 < 최대 출력 높이			

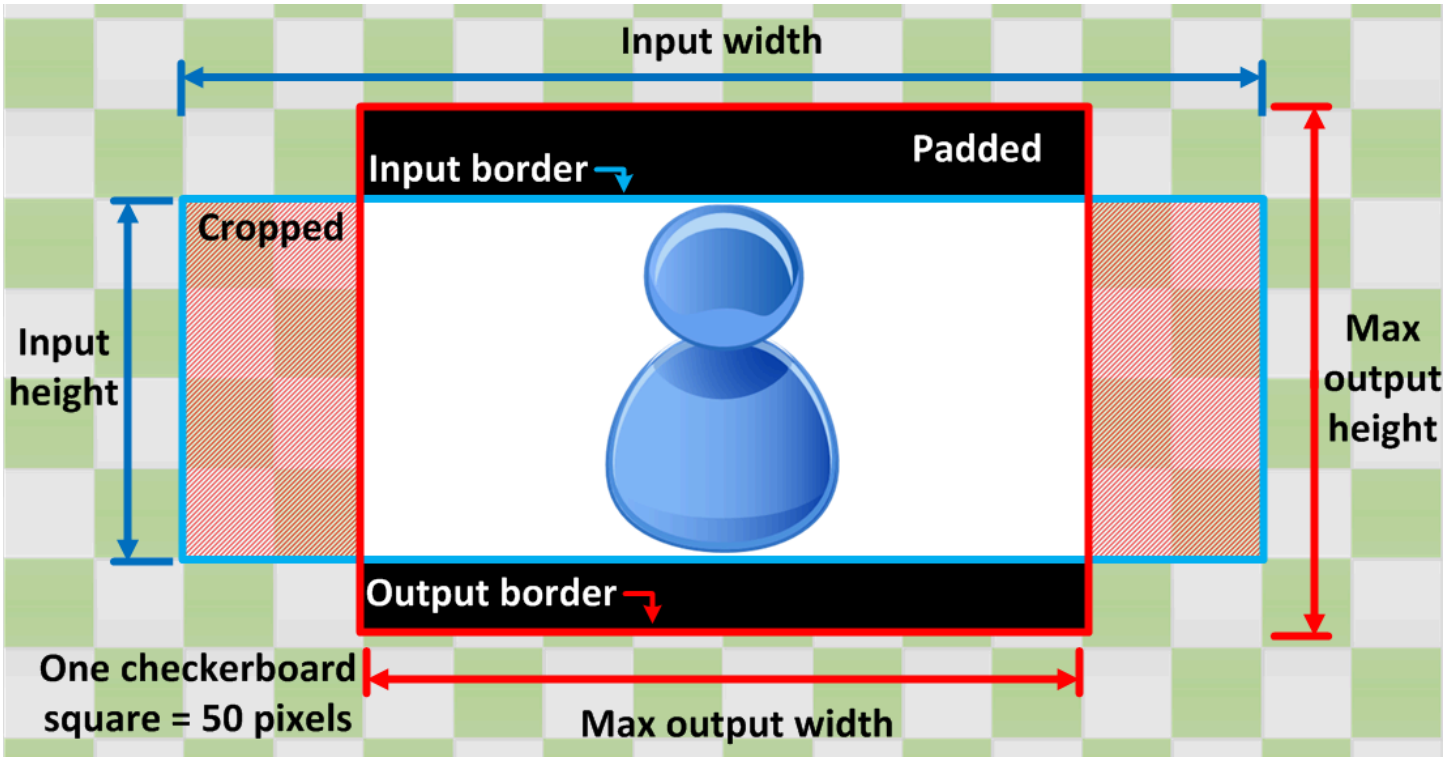
Condition	입력	출력: NoPad	출력: Pad
입력 너비 > 최대 출력 너비 입력 높이 > 최대 출력 높이			

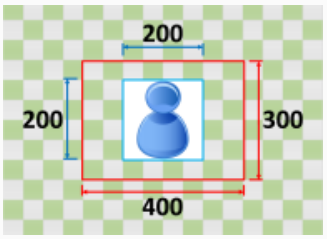
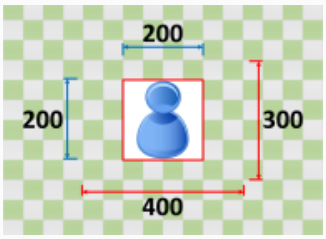
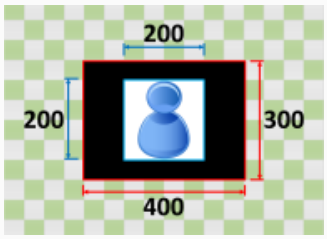
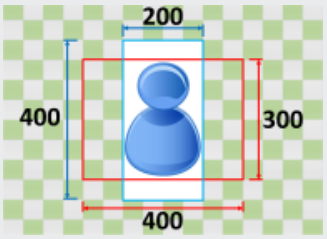
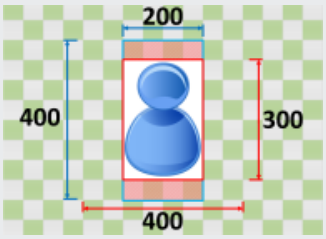
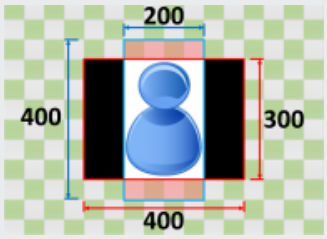
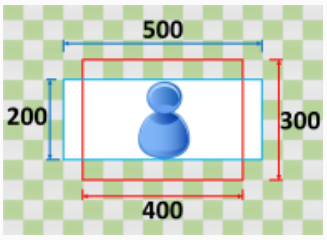
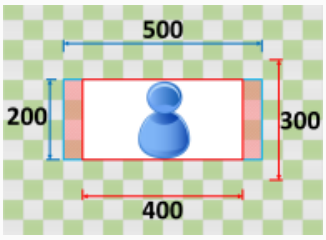
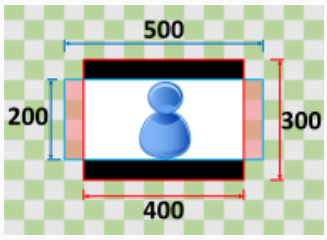
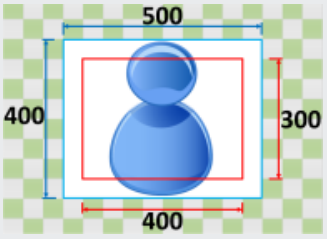
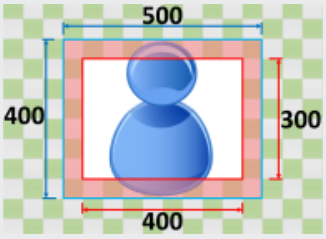
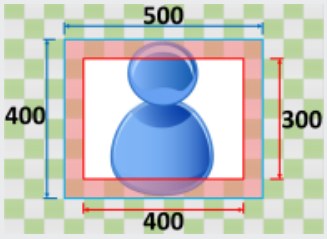
Keep

Sizing Policy로 Keep을 선택하면 Elastic Transcoder는 입력 파일의 크기를 조정하지 않습니다. Elastic Transcoder는 출력 이미지의 크기와 일치할 때까지 입력 파일을 자르거나 패딩합니다.

예를 들어 입력 파일이 400픽셀 x 200픽셀이고 출력 이미지를 300픽셀 x 300픽셀로 하려는 경우, Elastic Transcoder는 상단 및 하단에서 100픽셀씩 잘라낸 후 양쪽에 패딩 정책을 적용합니다. Padding Policy로 Unpadded를 선택하면 Elastic Transcoder는 300픽셀 x 200픽셀의 출력 파일을 반환합니다. Padded를 선택하면 Elastic Transcoder는 300픽셀 x 300픽셀 파일을 반환합니다.

Key(키)



Condition	입력	출력: NoPad	출력: Pad
입력 너비 < 최대 출력 너비 입력 높이 < 최대 출력 높이			
입력 너비 < 최대 출력 너비 입력 높이 > 최대 출력 높이			
입력 너비 > 최대 출력 너비 입력 높이 < 최대 출력 높이			
입력 너비 > 최대 출력 너비 입력 높이 > 최대 출력 높이			

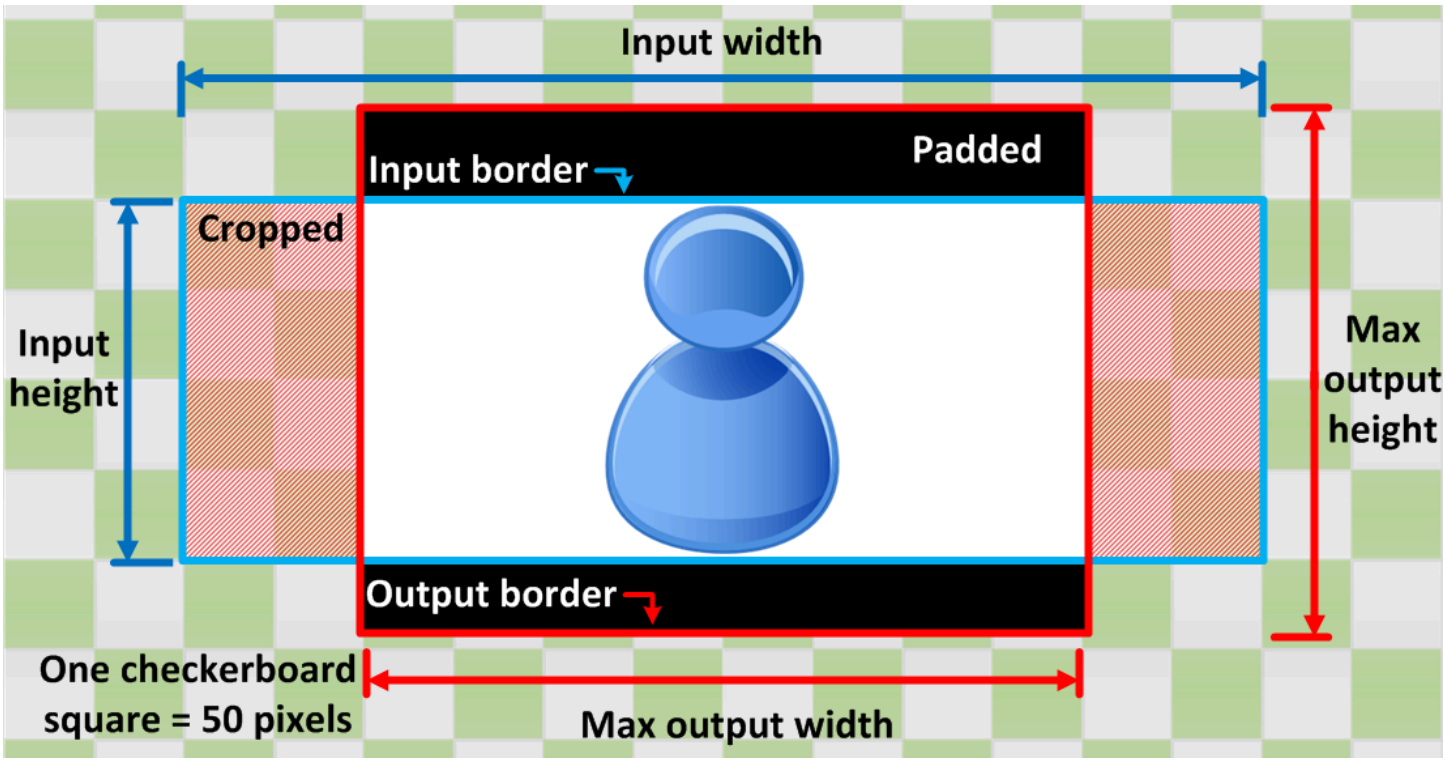
Shrink to Fit

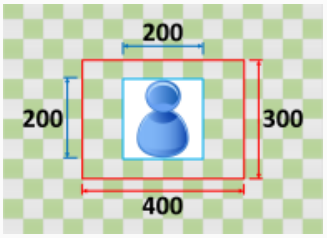
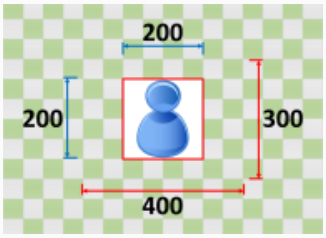
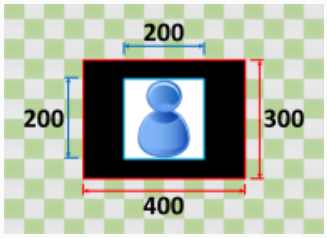
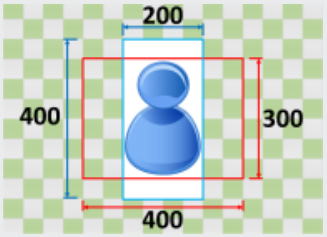
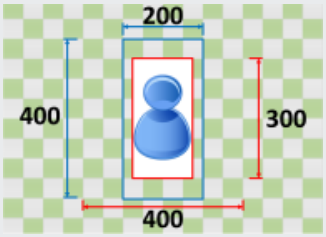
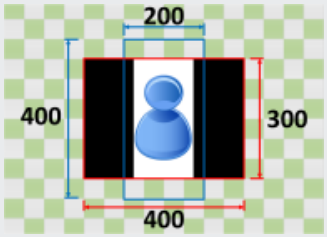
Sizing Policy로 Shrink to Fit을 선택하면 Elastic Transcoder는 출력 이미지의 크기를 초과하지 않고 출력 파일의 크기에 맞을 때까지 입력 파일의 크기를 줄입니다. 입력 파일이 출력 이미지보다 작은 경우 Elastic Transcoder는 파일의 크기를 늘리지 않습니다.

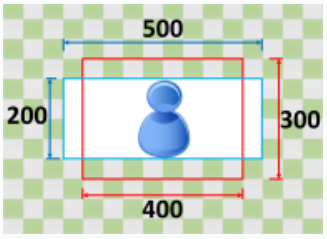
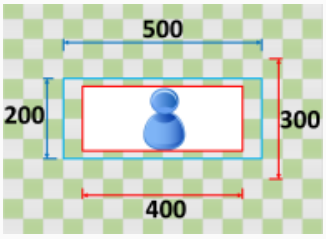
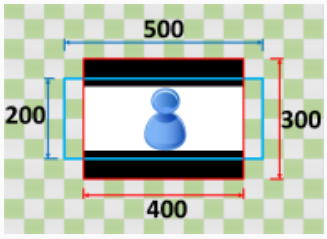
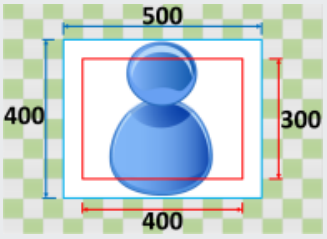
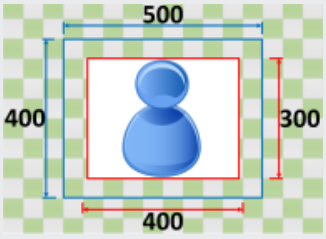
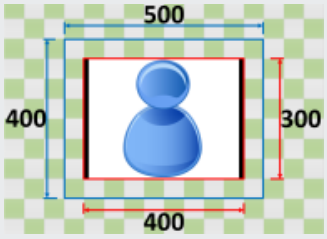
예를 들어 입력 파일이 400픽셀 x 400픽셀이고 출력 이미지를 200픽셀 x 300픽셀로 하려는 경우, Elastic Transcoder는 입력을 200픽셀 x 200픽셀로 축소하고 패딩 정책을 적용합니다. Padding Policy로 Unpadded를 선택하면 Elastic Transcoder는 200픽셀 x 200픽셀 파일을 출력으로 반환합니

다. Padded를 선택하면 Elastic Transcoder는 출력의 양쪽에 50픽셀의 패딩을 추가하고, 300픽셀 x 300픽셀 파일을 반환합니다.

Key(키)



Condition	입력	출력: NoPad	출력: Pad
입력 너비 < 최대 출력 너비 입력 높이 < 최대 출력 높이			
입력 너비 < 최대 출력 너비 입력 높이 > 최대 출력 높이			

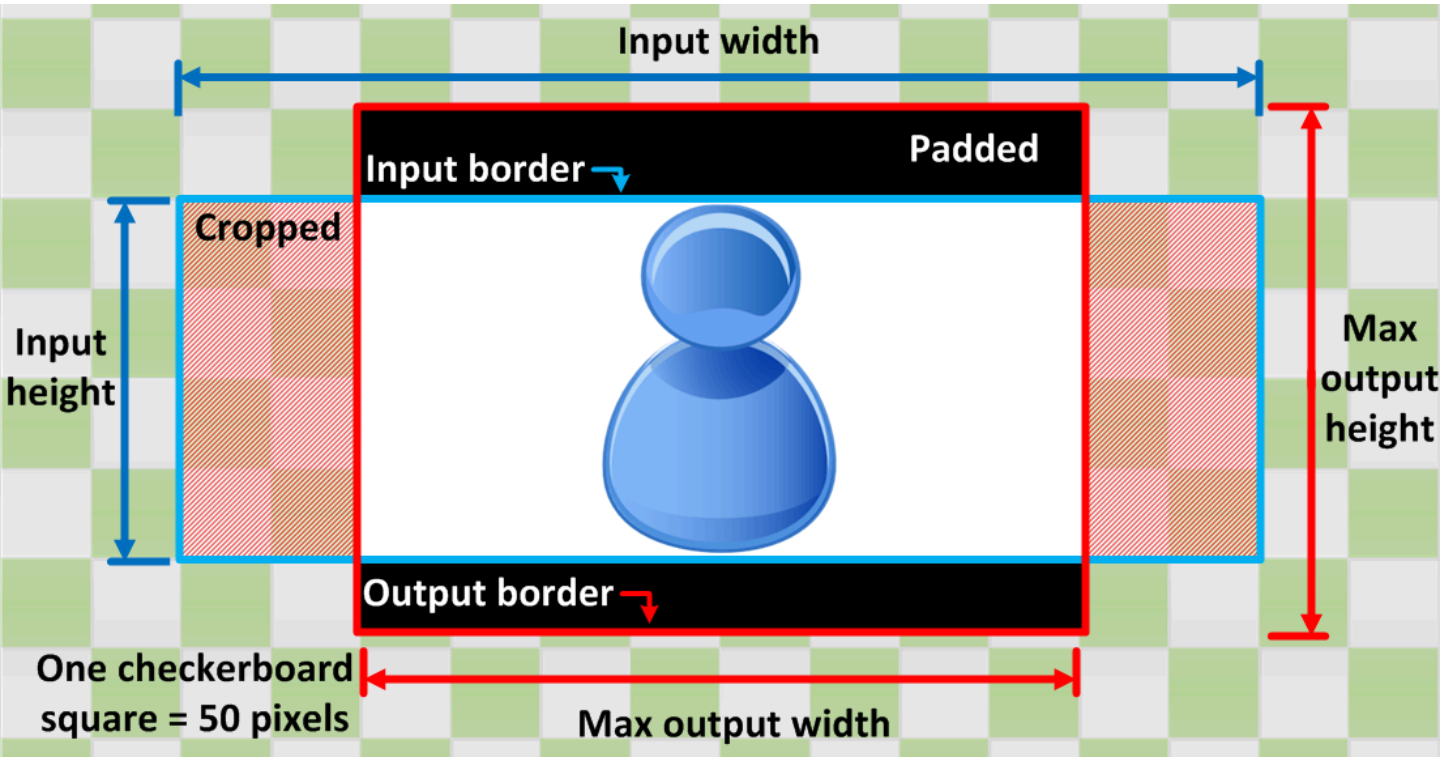
Condition	입력	출력: NoPad	출력: Pad
입력 너비 > 최대 출력 너비 입력 높이 < 최대 출력 높이			
입력 너비 > 최대 출력 너비 입력 높이 > 최대 출력 높이			

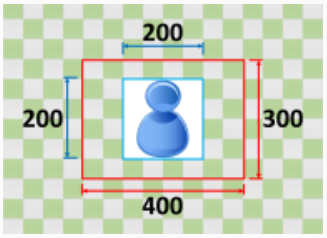
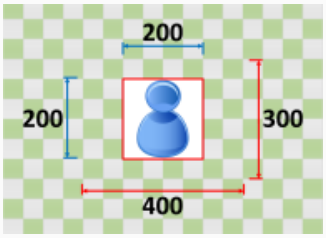
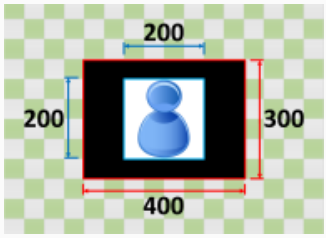
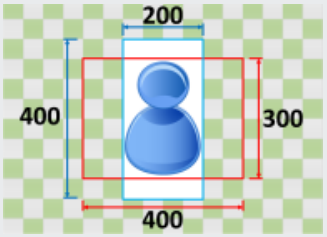
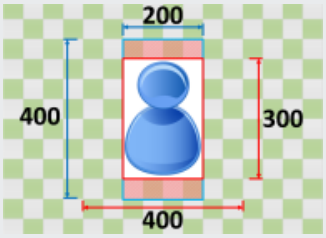
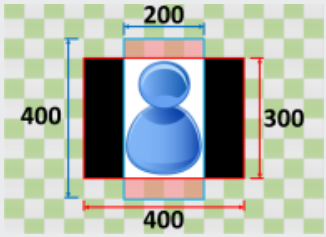
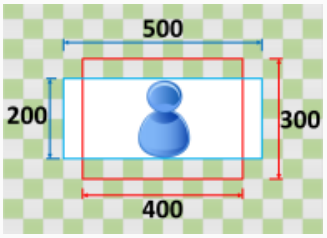
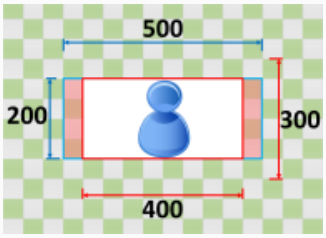
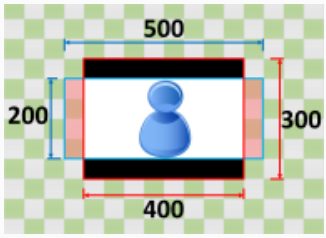
Shrink to Fill

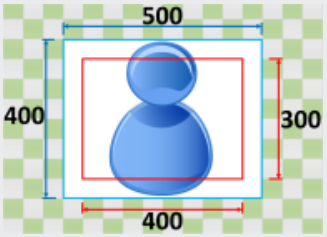
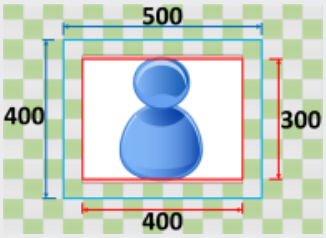
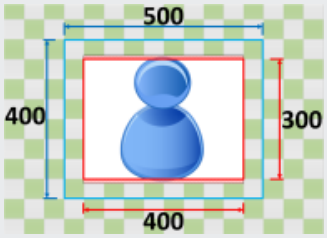
Sizing Policy로 Shrink to Fill을 선택하면 Elastic Transcoder는 출력 이미지의 크기를 채울 때까지 입력 파일의 크기를 줄이고, 출력 이미지에 맞지 않는 것은 잘라내고, 패딩 정책을 적용합니다. 출력 이미지가 입력 파일보다 큰 경우 Elastic Transcoder는 파일의 크기를 늘리지 않습니다.

예를 들어 입력 파일이 400픽셀 x 200픽셀이고 출력 이미지를 200픽셀 x 300픽셀로 하려는 경우, Elastic Transcoder 는 양쪽에서 100픽셀을 자른 후, 파일의 상단 및 하단에 패딩 정책을 적용합니다. Padding Policy로 Unpadded를 선택하면 Elastic Transcoder는 200픽셀 x 200픽셀의 출력 파일을 반환합니다. Padded를 선택하면 Elastic Transcoder는 200픽셀 x 300픽셀 파일을 반환합니다.

Key(키)



Condition	입력	출력: NoPad	출력: Pad
입력 너비 < 최대 출력 너비 입력 높이 < 최대 출력 높이			
입력 너비 < 최대 출력 너비 입력 높이 > 최대 출력 높이			
입력 너비 > 최대 출력 너비 입력 높이 < 최대 출력 높이			

Condition	입력	출력: NoPad	출력: Pad
입력 너비 > 최대 출력 너비 입력 높이 > 최대 출력 높이			

Elastic Transcoder 모니터링

모니터링은 Elastic Transcoder 및 AWS 솔루션의 신뢰성, 가용성 및 성능을 유지하는 데 중요한 부분입니다. 발생하는 다중 지점 실패를 보다 쉽게 디버깅할 수 있도록 AWS 솔루션의 모든 부분으로부터 모니터링 데이터를 수집해야 합니다. 하지만 Elastic Transcoder 모니터링을 시작하기 전에 다음 질문에 대한 답변을 포함하는 모니터링 계획을 작성해야 합니다.

- 모니터링의 목표
- 모니터링할 리소스
- 이러한 리소스를 모니터링하는 빈도
- 사용할 모니터링 도구
- 모니터링 작업을 수행할 사람
- 문제 발생 시 알려야 할 대상

다음 단계에서는 다양한 시간과 다양한 부하 조건에서 성능을 측정하여 환경에서 일반 Elastic Transcoder 성능의 기준선을 설정합니다. Elastic Transcoder가 과거 모니터링 데이터를 저장하는 것을 모니터링하면서 현재 성능 데이터를 이 과거 데이터와 비교하면 일반적인 성능 패턴과 성능 이상을 식별하고 이를 해결할 방법을 고안할 수 있습니다.

예를 들어 Amazon EC2를 사용하는 경우 인스턴스에 대해 CPU 사용률, 디스크 I/O 및 네트워크 사용률을 모니터링할 수 있습니다. 설정한 기준 이하로 성능이 떨어지면 인스턴스를 재구성하거나 최적화하여 CPU 사용률을 줄이거나 디스크 I/O를 개선하거나 네트워크 트래픽을 줄일 수 있습니다.

기준선을 설정하려면 최소한 다음 항목을 모니터링해야 합니다.

- 작업 완료됨
- 작업에 오류 발생
- StandbyTime
- 오류
- 제한

주제

- 모니터링 도구
- Amazon CloudWatch를 사용한 모니터링

모니터링 도구

AWS는 Elastic Transcoder를 모니터링하는 데 사용할 수 있는 다양한 도구를 제공합니다. 이들 도구 중에는 모니터링을 자동으로 수행하도록 구성할 수 있는 도구도 있지만, 수동 작업이 필요한 도구도 있습니다. 모니터링 작업은 최대한 자동화하는 것이 좋습니다.

자동 모니터링 도구

다음과 같은 자동 모니터링 도구를 사용하여 Elastic Transcoder를 관찰하고 문제 발생 시 보고할 수 있습니다.

- Amazon CloudWatch 경보 – 지정한 기간 동안 단일 지표를 감시하고, 여러 기간에 대해 지정된 임계값과 관련하여 지표 값을 기준으로 하나 이상의 작업을 수행합니다. 이 작업은 Amazon Simple Notification Service(Amazon SNS) 주제 또는 Amazon EC2 Auto Scaling 정책에 전송되는 알림입니다. CloudWatch 경보는 특정 상태에 있다는 이유만으로는 작업을 호출하지 않습니다. 상태가 변경되고 지정한 기간 동안 유지되어야 합니다. 자세한 내용은 Amazon CloudWatch를 사용한 모니터링을 참조하십시오.
- Amazon CloudWatch Logs – AWS CloudTrail 또는 기타 소스의 로그 파일을 모니터링, 저장 및 액세스합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 로그 파일 모니터링을 참조하세요.
- AWS CloudTrail 로그 모니터링 - 계정 간에 로그 파일을 공유하고, CloudWatch Logs로 전송하여 CloudTrail 로그 파일을 실시간으로 모니터링하고, Java에서 로그 처리 애플리케이션을 작성하고, CloudTrail에서 전송한 후 로그 파일이 변경되지 않았는지 확인합니다. CloudWatch 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 로그 파일 작업을 참조하세요.

수동 모니터링 도구

Elastic Transcoder 모니터링의 또 한 가지 중요한 부분은 CloudWatch 경보에 포함되지 않는 항목을 수동으로 모니터링해야 한다는 점입니다. Elastic Transcoder, CloudWatch 및 기타 AWS 콘솔 대시보드에서 AWS 환경의 상태를 한눈에 볼 수 있습니다. CloudTrail 로그 파일도 확인하는 것이 좋습니다.

- Elastic Transcoder 대시보드에는 다음이 표시됩니다.
  - 파이프라인과 그 상태
- CloudWatch 홈 페이지에 표시되는 항목은 다음과 같습니다.
  - 현재 경보 및 상태
  - 경보 및 리소스 그래프
  - 서비스 상태

또한 CloudWatch를 사용하여 다음을 수행할 수 있습니다.

- 맞춤 대시보드를 생성하여 관심 있는 서비스 모니터링
- 지표 데이터를 그래프로 작성하여 문제를 해결하고 추세 파악
- 모든 AWS 리소스 지표 검색 및 찾아보기
- 문제에 대해 알려주는 경보 생성 및 편집

Amazon CloudWatch를 사용한 모니터링

Elastic Transcoder에서 원시 데이터를 수집하여 읽기 가능하며 실시간에 가까운 지표로 처리하는 CloudWatch를 통해 파이프라인과 작업을 모니터링할 수 있습니다. 이러한 통계는 2주간 기록되므로 기록 정보를 보고 웹 애플리케이션이나 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다. 기본적으로 Elastic Transcoder 지표 데이터는 1분 간격으로 CloudWatch로 자동 전송됩니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch, Amazon CloudWatch Events 및 Amazon CloudWatch Logs란 무엇입니까?를 참조하세요.

주제

- Elastic Transcoder 지표와 차원
- Elastic Transcoder 지표를 사용하는 방법
- Elastic Transcoder 모니터링을 위한 CloudWatch 경보 생성

Elastic Transcoder 지표와 차원

작업을 만들면 Elastic Transcoder는 매분 다음 지표와 차원을 CloudWatch로 전송합니다. 다음 절차에 따라 Elastic Transcoder에 대한 지표를 볼 수 있습니다.

CloudWatch 콘솔을 사용하여 지표를 보려면

지표는 먼저 서비스 네임스페이스별로 그룹화된 다음 각 네임스페이스 내에서 다양한 차원 조합별로 그룹화됩니다.

1. https://console.aws.amazon.com/cloudwatch/에서 CloudWatch 콘솔을 엽니다.
2. 필요한 경우 리전을 변경합니다. 탐색 모음에서 AWS 리소스가 상주하는 리전을 선택합니다. 자세한 설명은 리전 및 엔드포인트를 참조하십시오.
3. 탐색 창에서 지표(Metrics)를 선택합니다.

4. 범주별 CloudWatch 지표 창의 Elastic Transcoder에 대한 지표 범주에서 지표 범주를 선택하고 위쪽 창에서 아래로 스크롤하여 모든 지표 목록을 봅니다.

AWS CLI를 사용하여 지표를 보려면

- 명령 프롬프트에서 다음 명령을 사용합니다.

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

CloudWatch에 Elastic Transcoder에 대한 다음 지표가 표시됩니다.

Elastic Transcoder 차원과 지표

Elastic Transcoder에서 Amazon CloudWatch로 전송하는 지표와 차원은 아래에 나와 있습니다.

Elastic Transcoder 지표

`AWS/ElasticTranscoder` 네임스페이스에는 다음과 같은 지표가 포함됩니다.

지표	설명
`Billed HD Output`	파이프라인에서 요금이 청구될 수 있는 HD 출력 시간(초) 유효한 차원: PipelineId 단위: 초
`Billed SD Output`	파이프라인에서 요금이 청구될 수 있는 SD 출력 시간(초) 유효한 차원: PipelineId 단위: 초
`Billed Audio Output`	파이프라인에서 요금이 청구될 수 있는 오디오 출력 시간(초) 유효한 차원: PipelineId

지표	설명
	단위: 초
Jobs Completed	파이프라인에서 완료된 작업 수 유효한 차원: PipelineId 단위: 수
Jobs Errored	지정한 입력 버킷에 없는 파일에 대한 트랜스코딩 요청 등 잘못된 입력으로 인해 실패한 작업 수 유효한 차원: PipelineId 단위: 수
Outputs per Job	Elastic Transcoder가 작업을 위해 생성한 출력 수입니다. 유효한 차원: PipelineId 단위: 수
Standby Time	Elastic Transcoder가 작업 트랜스코딩을 시작할 때까지 걸린 시간(초)입니다. 유효한 차원: PipelineId 단위: 초
Errors	작업 ID가 없는 작업 상태 요청 등 잘못된 작업 파라미터로 인해 발생한 오류 수 유효한 차원: Operation 단위: 수

지표	설명
Throttles	Elastic Transcoder가 작업을 자동으로 제한한 횟수입니다. 유효한 차원: Operation 단위: 수

Elastic Transcoder 지표의 차원

Elastic Transcoder 지표는 Elastic Transcoder 네임스페이스를 사용하며 다음 차원에게 지표를 제공합니다.

차원	설명
PipelineId	파이프라인 ID. 이 차원은 Elastic Transcoder 파이프라인에 대해 요청하는 데이터를 필터링합니다.
Operation	이 차원은 Elastic Transcoder가 제공하는 API에 대해 요청한 데이터를 필터링합니다.

Elastic Transcoder 지표를 사용하는 방법

Elastic Transcoder에서 보고하는 지표는 다양한 방법으로 분석이 기능한 정보를 제공합니다. 아래 목록은 몇 가지 일반적인 지표 사용 사례입니다. 모든 사용 사례를 망라한 것은 아니지만 시작하는 데 참고가 될 것입니다.

- 제 작업이 시작되기 전 대기 시간을 어떻게 추적할 수 있습니까?
- 오류로 인해 작업에 실패한 경우 어떻게 알림을 받을 수 있습니까?
- 작업을 완료했을 때 어떻게 알림을 받을 수 있습니까?

Elastic Transcoder 모니터링을 위한 CloudWatch 경보 생성

경보 상태가 변경되면 Amazon SNS 메시지를 전송하는 CloudWatch 경보를 만들 수 있습니다. 경보는 지정한 기간에 단일 지표를 감시하고 여러 기간에 지정된 임계값에 대한 지표 값을 기준으로 작업을 하

나 이상 수행합니다. 이 작업은 Amazon SNS 주제 또는 Auto Scaling 정책으로 전송되는 알림입니다. 경보는 지속적인 상태 변경에 대해서만 작업을 호출합니다. CloudWatch 경보는 특정 상태에 있다는 이유만으로는 작업을 호출하지 않습니다. 상태가 변경되고 지정한 기간 동안 유지되어야 합니다.

제 작업이 시작되기 전 대기 시간을 어떻게 추적할 수 있습니까?

`Standyby Time` 지표를 사용하여 파이프라인이 트랜스코딩 작업을 시작하는 데 걸린 시간을 추적할 수 있습니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudwatch/ CloudWatch 콘솔을 엽니다.
2. 경보 생성을 선택합니다. 그러면 Create Alarm Wizard가 시작됩니다.
3. AWS/ElasticTranscoder 지표를 선택하고 Elastic Transcoder 지표를 스크롤하여 Standby Time 지표를 찾습니다. Standby Time 지표를 선택하여 경보를 생성한 다음 Next를 선택합니다.
4. 지표에 대한 Name, Description, Whenever 값을 입력합니다.
5. 작업에서 기존 Amazon SNS 주제를 선택합니다.

   Amazon SNS 알림에 대한 자세한 정보는 Amazon Simple Notification Service 개발자 안내서에서 주제 생성 섹션을 참조하세요.
6. 경보 생성을 선택합니다.

오류로 인해 작업에 실패한 경우 어떻게 알림을 받을 수 있습니까?

`Job Errored` 지표를 사용하여 잘못된 입력 파라미터로 인해 파이프라인에서 실패한 작업의 수를 추적할 수 있습니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudwatch/ CloudWatch 콘솔을 엽니다.
2. 경보 생성을 선택합니다. 그러면 Create Alarm Wizard가 시작됩니다.
3. AWS/ElasticTranscoder 지표를 선택하고 Elastic Transcoder 지표를 스크롤하여 Job Errored 지표를 찾습니다. Job Errored 지표를 선택하여 경보를 생성한 다음 Next를 선택합니다.
4. 지표에 대한 Name, Description, Whenever 값을 입력합니다.
5. 작업에서 기존 Amazon SNS 주제를 선택합니다.

   Amazon SNS 알림에 대한 자세한 정보는 Amazon Simple Notification Service 개발자 안내서에서 주제 생성 섹션을 참조하세요.
6. 경보 생성을 선택합니다.

작업을 완료했을 때 어떻게 알림을 받을 수 있습니까?

`Jobs Completed` 지표를 사용하여 파이프라인이 트랜스코딩한 작업의 수를 추적할 수 있습니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/cloudwatch/ CloudWatch 콘솔을 엽니다.
2. 경보 생성을 선택합니다. 그러면 Create Alarm Wizard가 시작됩니다.
3. AWS/ElasticTranscoder 지표를 선택하고 Elastic Transcoder 지표를 스크롤하여 Jobs Completed 지표를 찾습니다. Jobs Completed 지표를 선택하여 경보를 생성한 다음 Next를 선택합니다.
4. 지표에 대한 Name, Description, Whenever 값을 입력합니다.
5. 작업에서 기존 Amazon SNS 주제를 선택합니다.

   Amazon SNS 알림에 대한 자세한 정보는 Amazon Simple Notification Service 개발자 안내서에서 주제 생성 섹션을 참조하세요.
6. 경보 생성을 선택합니다.

를 사용하여 Elastic Transcoder API 호출 로깅 AWS CloudTrail

Elastic Transcoder는 Elastic Transcoder에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다. CloudTrail은 Elastic Transcoder 콘솔의 호출 및 Elastic Transcoder API에 대한 코드 호출을 포함하여 Elastic Transcoder에 대한 모든 API 직접 호출을 이벤트로 캡처합니다. 추적을 생성하면 Elastic Transcoder 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 트레일을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 Elastic Transcoder에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간, 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

CloudTrail의 Elastic Transcoder 정보

AWS 계정을 생성할 때 계정에서 CloudTrail이 활성화됩니다. Elastic Transcoder에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다. 자세한 정보는 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

Elastic Transcoder에 대한 이벤트를 포함하여 AWS 계정의 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 트레일을 생성하면 기본적으로 모든 리전에 트레일이 적용됩니다. 추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음을 참조하세요.

- 트레일 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에서 Amazon SNS 알림 구성
- 여러 리전으로부터 CloudTrail 로그 파일 받기 및 여러 계정으로부터 CloudTrail 로그 파일 받기

모든 Elastic Transcoder 작업이 CloudTrail에서 로깅되고 API 참조에서 문서화됩니다. 예를 들어, `CreatePipeline`, `CreateJob` 및 `CreatePreset` 섹션을 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. ID 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 IAM 사용자 보안 인증 정보로 했는지 여부.
- 역할 또는 페더레이션 사용자의 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에서 이루어졌는지 여부입니다.

자세한 설명은 CloudTrail userIdentity 요소를 참조하세요.

Elastic Transcoder 로그 파일 항목 이해

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보를 포함하고 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출에 대한 순서 지정된 스택 추적이 아니기 때문에 특정 순서로 표시되지 않습니다.

다음은 `CreateJob`, `ListJobsByStatus`, `ReadJob`, `DeletePreset`, `DeletePipeline` 작업을 보여 주는 CloudTrail 로그 항목이 나타낸 예입니다.

```
{
    "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-f2czlr"
        },
        "responseElements": null,
        "requestID": "497b3622-d548-11e3-8fb9-4dad0example",
        "eventID": "c32289c7-005a-46f7-9801-cba41example",
        "eventType": "AwsApiCall",
        "recipientAccountId": "111122223333"
    },
    {
        "eventVersion": "1.02",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "A1B2C3D4E5F6G7EXAMPLE",
            "arn": "arn:aws:iam::111122223333:user/smithj",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
        },
        "eventTime": "2014-09-29T19:29:18Z",
        "eventSource": "elastictranscoder.amazonaws.com",
        "eventName": "DeletePreset",
        "awsRegion": "us-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 샘플 코드

이 단원에서는 Java, Ruby, PHP, Python용 AWS SDK를 사용하여 다음 작업을 수행하는 방법을 보여 주는 샘플 코드를 소개합니다.

- Amazon Elastic Transcoder에서 HLS 재생 목록 생성
- Elastic TranScoder에서 작업 만들기
- Amazon Simple Queue Service(Amazon SQS) 대기열 폴링
- Amazon Simple Notification Service(SNS)의 알림 구문 분석
- Amazon SNS의 알림 처리

다음 작업은 Elastic Transcoder 모범 사례를 나타냅니다. 대기열을 폴링하면 장기적으로 폴링이 가능하여 정기적인 폴링보다 효율적으로 규모를 조정할 수 있으며, Amazon SNS 알림을 설정하면 Elastic Transcoder가 Amazon SQS 대기열에 메시지를 보낼 수 있습니다.

이 샘플 코드는 HLS(HTTP Live Streaming)를 위한 트랜스코딩 방법과, Amazon SQS 대기열에서 Amazon SNS 알림이 제거된 후 알림을 처리하는 방법을 보여줍니다.

SDK를 포함하는 언어를 사용하는 경우, API를 통해 작업을 처음부터 끝까지 모두 수행하기보다는 SDK를 사용하는 것이 좋습니다. SDK를 사용하면 인증을 더 간단히 수행할 수 있고, 개발 환경에 쉽게 통합할 수 있으며, 관련 명령에 쉽게 액세스할 수 있습니다.

주제

- AWS SDK 샘플 코드의 배경
- 환경 설정
- Java 샘플
- Ruby 샘플
- PHP 샘플
- Python 샘플

AWS SDK 샘플 코드의 배경

HLS(HTTP Live Streaming)

가변 비트 레이트 콘텐츠를 HLS 지원 플레이어로 전송하기 위해 HTTP Live Streaming(HLS) 출력을 사용합니다. HLS 지원 디바이스의 예는 Android 4+ 디바이스, iOS 디바이스, 데스크톱 미디어 플레이어(예: QuickTime 또는 VLC) 및 브라우저 플레이어(예: jwplayer)입니다. 가변 비트 레이트는 전송되는 콘텐츠의 품질을 클라이언트 연결 품질에 따라 조정할 수 있습니다.

알림

Elastic Transcoder의 `ReadJob` API를 폴링하여 작업 상태를 추적하는 경우 제출되는 모든 작업에서 지속적으로 `ReadJob`을 호출해야 합니다. 이 방법은 트랜스코딩 작업이 증가할 경우 확장이 불가능합니다. 이 문제를 해결하기 위해 Elastic Transcoder가 작업 상태 추적을 위한 이벤트 기반 메커니즘을 제공하는 에 알림을 게시할 수 있습니다.

각 Amazon SNS 알림은 `Message` 필드에서 JSON 객체로 전송됩니다. 알림 자체가 JSON 형식으로 전송되므로 메시지 문자열에서 Amazon SNS 알림을 이스케이프해야 합니다. 알림의 형식 및 내용에 대한 자세한 내용은 알림 섹션을 참조하세요.

작업 상태 알림을 처리하는 사용자 지정 코드를 작성할 때 다음 모범 사례를 따릅니다.

- 핸들러가 idempotent여야 합니다. 알림을 여러 번 전송할 수 있습니다.
- 핸들러가 순서를 무시한 알림을 지원해야 합니다. 알림을 순서에 맞지 않게 전송할 수 있습니다.
- 핸들러가 모든 작업에 대한 알림을 처리할 수 있어야 합니다. 특정 작업에 대한 알림이 특정 작업자로 전송되도록 보장할 수는 없습니다.
- 핸들러는 짧은 작업이어야 합니다. 모든 메시지가 제한 시간 초과 이전에 처리 및 삭제되어야 합니다. 제한 시간 초과가 15초이고 Amazon SQS로부터 최대 5개 메시지가 반환될 경우 각 메시지가 3초 이내에 대기열에서 처리되고 삭제되어야 합니다. 처리가 이 시간보다 오래 걸릴 경우 처리되지 않은 메시지가 시간 초과되어 다른 작업자로 전송됩니다.

Java, Python 및 Ruby의 경우, 알림 주제를 구독하도록 등록된 Amazon SQS 대기열을 폴링하여 알림을 소비하는 것이 좋습니다. Amazon SQS는 긴 폴링 메커니즘을 사용하므로 Amazon SQS 대기열을 폴링하면 작업 알림을 소비하는 확장 가능한 방법이 제공됩니다. 또한 Amazon SQS는 호스트에 장애가 발생하거나 로드가 높은 경우 가용성과 확장을 단순화하며 일반적으로 특별한 ACL 설정이 필요하지 않습니다.

Apache에서 실행되는 PHP의 경우, 엔드포인트가 직접 Amazon SNS 주제를 구독하는 것이 좋습니다. 그러려면 Elastic Transcoder가 알림을 바로 사용자에게 푸시할 수 있어야 하므로 엔드포인트가 공개적으로 사용 가능해야 합니다.

환경 설정

샘플 코드를 실행하려면 설정된 AWS 환경이 있어야 합니다. 이 단원에서는 설정 절차를 안내하고, Elastic Transcoder를 최적으로 작동하기 위해 필요한 AWS 리소스를 만드는 방법을 보여줍니다.

주제

- Java 환경 설정
- Ruby 환경 설정
- PHP 환경 설정
- Python 환경 설정

Java 환경 설정

이 단원에서는 Java 환경을 설정하는 절차를 안내합니다.

주제

- AWS SDK for Java 설정
- Amazon S3 입력 및 출력 버킷 생성
- 작업 상태 알림을 받는 Amazon SNS 주제 생성
- 작업 상태 알림을 폴링하는 Amazon SQS 대기열 생성
- Amazon SQS 대기열에서 Amazon SNS 주제 구독
- Elastic Transcoder 파이프라인 생성
- 확장 가능한 방법으로 콘텐츠를 제공하는 Amazon CloudFront 배포 생성
- Amazon S3 버킷 정책 수정

샘플 코드는 다음 두 가지 조건을 기본 전제로 합니다.

- 샘플이 Java 버전 1.6 이상에서 사용하도록 작성되었습니다.
- 샘플을 Eclipse 및 AWS Toolkit for Eclipse를 사용하여 실행합니다.

AWS SDK for Java 설정

이들 샘플은 AWS Toolkit for Eclipse를 사용하여 실행하는 것을 전제로 합니다. AWS SDK for Java 및 다음과 같은 Jackson JSON 프로세서 JAR 파일이 모두 필요합니다.

- Jackson Core
- Jackson Databind
- Jackson Annotations

Jackson JSON 프로세서는 작업 상태 알림을 처리합니다.

또는 Maven을 사용하여 종속성을 관리하는 경우 다음 스니펫을 `pom.xml` 파일에 추가할 수 있습니다.

버전 속성:

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

종속성:

```
    <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>
```

자세한 내용은 AWS SDK for Java 설명서를 참조하세요.

Amazon S3 입력 및 출력 버킷 생성

1. Amazon S3 콘솔을 엽니다.
2. 콘솔에서 버킷 만들기를 클릭합니다.
3. 버킷 이름 필드에 입력 버킷의 원하는 이름을 입력합니다.
4. Amazon S3 콘솔에서 버킷 만들기를 다시 클릭합니다.
5. 버킷 이름 필드에 출력 버킷의 원하는 이름을 입력합니다.

작업 상태 알림을 받는 Amazon SNS 주제 생성

1. Amazon SNS 콘솔을 엽니다.
2. 콘솔에서 Create and Add(생성 및 추가)를 클릭하고 새로운 주제 생성을 선택합니다.
3. 주제 이름 필드에 **`ets-sample-topic`**을 입력하고 주제 생성을 클릭합니다.
4. `ets-sample-topic`의 ARN을 기록합니다.

작업 상태 알림을 폴링하는 Amazon SQS 대기열 생성

1. Amazon SQS 콘솔을 엽니다.
2. 콘솔에서 새로운 대기열 생성을 클릭합니다.
3. 대기열 이름 필드에 **`ets-sample-queue`**를 입력하고 대기열 생성을 클릭합니다.
4. Amazon SQS 대기열의 URL을 기록합니다.

Amazon SQS 대기열에서 Amazon SNS 주제 구독

1. Amazon SQS 콘솔을 엽니다.
2. 콘솔의 대기열 목록에서 ets-sample-queue를 선택합니다.
3. 대기열 작업에서 Amazon SNS 주제에 대한 대기열 구독을 선택합니다.
4. 주제 선택에서 ets-sample-topic을 선택하고 구독을 클릭합니다.

대기열이 성공적으로 주제를 구독하도록 등록되었다는 확인 메시지가 표시될 것입니다.

Elastic Transcoder 파이프라인 생성

1. Amazon Elastic Transcoder 콘솔을 엽니다.
2. 콘솔에서 Create New Pipeline(새 파이프라인 생성)을 클릭합니다.
3. 파이프라인 이름 필드에 **ets-sample-pipeline**을 입력합니다.
4. Input Bucket(입력 버킷) 필드에 입력 버킷의 이름을 입력합니다.
5. IAM 역할을 **Console Default Role**로 유지합니다.
6. 트랜스코딩한 파일 및 재생 목록의 Amazon S3 버킷 구성에 출력 버킷의 이름을 입력합니다.
7. 트랜스코딩한 파일 및 재생 목록의 Amazon S3 버킷 구성에서 표준 스토리지 클래스를 선택합니다.
8. 알림 드롭다운 메뉴를 확장합니다.
9. 4개 이벤트 유형 모두에 대해 Use an existing SNS topic(기존 SNS 주제 사용)을 선택하고 Select a Topic(주제 선택) 아래에서 **ets-sample-topic**을 선택합니다.
10. Create Pipeline을 클릭합니다.
11. Elastic Transcoder 파이프라인 ID를 기록합니다.

확장 가능한 방법으로 콘텐츠를 제공하는 Amazon CloudFront 배포 생성

1. Amazon CloudFront 콘솔을 엽니다.
2. 콘솔의 탐색 창에서 Private Content(프라이빗 콘텐츠)를 확장하고 원본 액세스 ID를 클릭합니다.
3. Create Origin Access Identity(원본 액세스 ID 생성)를 클릭합니다.
4. 생성을 클릭합니다.
5. 원본 액세스 ID를 만들 때 생성된 Amazon S3 정식 사용자 ID를 기록합니다.
6. 탐색 창에서 배포를 클릭합니다.
7. Create Distribution(배포 생성)을 클릭합니다.
8. 웹을 선택하고 계속을 클릭합니다.
9. 오리진 설정에서 Amazon S3 출력 버킷을 오리진 도메인 이름으로 입력합니다.
10. Origin ID(오리진 ID) 필드에 **S3-transcoder-sample-output**을 입력합니다.
11. Restrict Bucket Access(버킷 액세스 제한)에서 예를 선택합니다.

12. Create Distribution(배포 생성)을 클릭하고 배포의 도메인 이름을 기록합니다.

Amazon S3 버킷 정책 수정

1. Amazon S3 콘솔을 엽니다.
2. 출력 버킷 옆에서 확대경 아이콘을 클릭하여 버킷 속성을 불러옵니다.
3. 권한을 확장하고 Add bucket policy(버킷 정책 추가)를 클릭합니다.
4. 다음 정책 설명을 입력하고 CloudFront-oai-s3-canonical-user-id를 **`your recorded Amazon S3 canonical user ID`**로, s3-output-bucket-name을 **`the name of your output bucket`**으로 바꿉니다.
5. 저장을 클릭합니다.

```
{
  "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 샘플 섹션을 참조하세요.

Ruby 환경 설정

이 단원에서는 Ruby 환경을 설정하는 절차를 안내합니다.

주제

- AWS SDK for Ruby 설정
- Amazon S3 입력 및 출력 버킷 생성
- 작업 상태 알림을 받는 Amazon SNS 주제 생성
- 작업 상태 알림을 폴링하는 Amazon SQS 대기열 생성
- Amazon SQS 대기열에서 Amazon SNS 주제 구독

- Elastic Transcoder 파이프라인 생성
- 확장 가능한 방법으로 콘텐츠를 제공하는 Amazon CloudFront 배포 생성
- Amazon S3 버킷 정책 수정

샘플 코드는 다음 한 가지 조건을 기본 전제로 합니다.

- 샘플이 Ruby 버전 1.9와 호환되도록 작성되었습니다.

AWS SDK for Ruby 설정

AWS SDK for Ruby는 여기서 제공합니다. Ruby Gem을 사용하여 다음 명령으로 쉽게 설치할 수 있습니다.

```
gem install aws-sdk
```

자세한 내용은 AWS SDK for Ruby 설명서를 참조하세요.

Amazon S3 입력 및 출력 버킷 생성

1. Amazon S3 콘솔을 엽니다.
2. 콘솔에서 버킷 만들기를 클릭합니다.
3. 버킷 이름 필드에 입력 버킷의 원하는 이름을 입력합니다.
4. Amazon S3 콘솔에서 버킷 만들기를 다시 클릭합니다.
5. 버킷 이름 필드에 출력 버킷의 원하는 이름을 입력합니다.

작업 상태 알림을 받는 Amazon SNS 주제 생성

1. Amazon SNS 콘솔을 엽니다.
2. 콘솔에서 Create and Add(생성 및 추가)를 클릭하고 새로운 주제 생성을 선택합니다.
3. 주제 이름 필드에 **`ets-sample-topic`**을 입력하고 주제 생성을 클릭합니다.
4. `ets-sample-topic`의 ARN을 기록합니다.

작업 상태 알림을 폴링하는 Amazon SQS 대기열 생성

1. Amazon SQS 콘솔을 엽니다.

2. 콘솔에서 새로운 대기열 생성을 클릭합니다.
3. 대기열 이름 필드에 **ets-sample-queue**를 입력하고 대기열 생성을 클릭합니다.
4. Amazon SQS 대기열의 URL을 기록합니다.

Amazon SQS 대기열에서 Amazon SNS 주제 구독

1. Amazon SQS 콘솔을 엽니다.
2. 콘솔의 대기열 목록에서 ets-sample-queue를 선택합니다.
3. 대기열 작업에서 Amazon SNS 주제에 대한 대기열 구독을 선택합니다.
4. 주제 선택에서 ets-sample-topic을 선택하고 구독을 클릭합니다.

대기열이 성공적으로 주제를 구독하도록 등록되었다는 확인 메시지가 표시될 것입니다.

Elastic Transcoder 파이프라인 생성

1. Amazon Elastic Transcoder 콘솔을 엽니다.
2. 콘솔에서 Create New Pipeline(새 파이프라인 생성)을 클릭합니다.
3. 파이프라인 이름 필드에 **ets-sample-pipeline**을 입력합니다.
4. Input Bucket(입력 버킷) 필드에 입력 버킷의 이름을 입력합니다.
5. IAM 역할을 **Console Default Role**로 유지합니다.
6. 트랜스코딩한 파일 및 재생 목록의 Amazon S3 버킷 구성에 출력 버킷의 이름을 입력합니다.
7. 트랜스코딩한 파일 및 재생 목록의 Amazon S3 버킷 구성에서 표준 스토리지 클래스를 선택합니다.
8. 알림 드롭다운 메뉴를 확장합니다.
9. 4개 이벤트 유형 모두에 대해 Use an existing SNS topic(기존 SNS 주제 사용)을 선택하고 Select a Topic(주제 선택) 아래에서 **ets-sample-topic**을 선택합니다.
10. Create Pipeline을 클릭합니다.
11. Elastic Transcoder 파이프라인 ID를 기록합니다.

확장 가능한 방법으로 콘텐츠를 제공하는 Amazon CloudFront 배포 생성

1. Amazon CloudFront 콘솔을 엽니다.

2. 콘솔의 탐색 창에서 Private Content(프라이빗 콘텐츠)를 확장하고 원본 액세스 ID를 클릭합니다.
3. Create Origin Access Identity(원본 액세스 ID 생성)를 클릭합니다.
4. 생성을 클릭합니다.
5. 원본 액세스 ID를 만들 때 생성된 Amazon S3 정식 사용자 ID를 기록합니다.
6. 탐색 창에서 배포를 클릭합니다.
7. Create Distribution(배포 생성)을 클릭합니다.
8. 웹을 선택하고 계속을 클릭합니다.
9. 오리진 설정에서 Amazon S3 출력 버킷을 오리진 도메인 이름으로 입력합니다.
10. Origin ID(오리진 ID) 필드에 **`S3-transcoder-sample-output`**을 입력합니다.
11. Restrict Bucket Access(버킷 액세스 제한)에서 예를 선택합니다.
12. Create Distribution(배포 생성)을 클릭하고 배포의 도메인 이름을 기록합니다.

Amazon S3 버킷 정책 수정

1. Amazon S3 콘솔을 엽니다.
2. 출력 버킷 옆에서 확대경 아이콘을 클릭하여 버킷 속성을 불러옵니다.
3. 권한을 확장하고 Add bucket policy(버킷 정책 추가)를 클릭합니다.
4. 다음 정책 설명을 입력하고 CloudFront-oai-s3-canonical-user-id를 **`your recorded Amazon S3 canonical user ID`**로, s3-output-bucket-name을 **`the name of your output bucket`**으로 바꿉니다.
5. 저장을 클릭합니다.

```
{
  "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 샘플 섹션으로 건너뜁니다.

PHP 환경 설정

이 단원에서는 PHP 환경을 설정하는 절차를 안내합니다.

주제

- 샘플 코드 설치
- AWS SDK for PHP 설정
- Amazon S3 입력 및 출력 버킷 생성
- 작업 상태 알림을 받는 Amazon SNS 주제 생성
- Amazon SNS에 서버 등록
- Elastic Transcoder 파이프라인 생성
- 확장 가능한 방법으로 콘텐츠를 제공하는 Amazon CloudFront 배포 생성
- Amazon S3 버킷 정책 수정

샘플 코드는 다음 세 가지 조건을 기본 전제로 합니다.

- PHP 샘플이 Apache 서버 내에서 실행됩니다.
- Apache 서버는 Amazon SNS 알림을 수신하기 위해 인터넷으로부터 입력을 받을 수 있어야 합니다.
- 샘플이 PHP 버전 5.4 이상에서 사용하도록 작성되었습니다.

샘플 코드 설치

PHP 샘플 코드를 사용하려면 먼저 다운로드하여 설치해야 합니다.

- 샘플 코드를 다운로드합니다.
  - HLS 샘플 코드.
  - 알림 샘플 코드.
- 샘플 코드의 압축을 풉니다.
- Apache 서버의 DocumentRoot 아래에 코드를 배치합니다.
- 다운로드한 샘플 코드 .zip 파일을 제거합니다.
- AWS SDK 설치 경로를 업데이트합니다.

> **Note**
>
> `HlsJobCreationSample.php` 및 `JobStatusNotificationsSample.php`에서 `autoload.php`에 대한 경로를 업데이트해야 합니다. phar 설치를 사용하는 경우 다운로드한 `aws.phar` 파일(필요한 모든 종속성이 들어 있음)의 경로를 사용할 수 있습니다.

AWS SDK for PHP 설정

AWS SDK for PHP는 여기서 받을 수 있습니다. 이 자습서에서는 phar 설치를 권장하지만, 장기 프로젝트는 composer를 사용하여 관리하는 것이 좋습니다.

자세한 내용은 AWS SDK for PHP 설명서를 참조하세요.

Amazon S3 입력 및 출력 버킷 생성

1. Amazon S3 콘솔을 엽니다.
2. 콘솔에서 버킷 만들기를 클릭합니다.
3. 버킷 이름 필드에 입력 버킷의 원하는 이름을 입력합니다.
4. Amazon S3 콘솔에서 버킷 만들기를 다시 클릭합니다.
5. 버킷 이름 필드에 출력 버킷의 원하는 이름을 입력합니다.

작업 상태 알림을 받는 Amazon SNS 주제 생성

1. Amazon SNS 콘솔을 엽니다.
2. 콘솔에서 Create and Add(생성 및 추가)를 클릭하고 새로운 주제 생성을 선택합니다.
3. 주제 이름 필드에 **`ets-sample-topic`**을 입력하고 주제 생성을 클릭합니다.
4. `ets-sample-topic`의 ARN을 기록합니다.

Amazon SNS에 서버 등록

1. PHP 샘플 코드가 설치되었는지 확인합니다.
2. Amazon SNS 콘솔을 엽니다.
3. AdditionalActions에서 Create Subscription을 클릭합니다.
4. 서버 구성에 따라 HTTP 또는 HTTPS를 선택합니다.

5. Endpoint에 서버의 엔드포인트를 입력합니다. 엔드포인트의 경로는 `JobStatusNotificationsSampleNotificationHandler.php`를 가리켜야 합니다.
6. Subscribe를 클릭합니다. 그러면 PHP 엔드포인트에 구독 요청이 전송됩니다.

PHP 샘플 코드는 구독 요청을 자동으로 처리하여 구독을 확인합니다. 구독 요청과 응답은 `/tmp/subscribe_requests.txt`에 기록됩니다

Elastic Transcoder 파이프라인 생성

1. Amazon Elastic Transcoder 콘솔을 엽니다.
2. 콘솔에서 Create New Pipeline(새 파이프라인 생성)을 클릭합니다.
3. 파이프라인 이름 필드에 **`ets-sample-pipeline`**을 입력합니다.
4. Input Bucket(입력 버킷) 필드에 입력 버킷의 이름을 입력합니다.
5. IAM 역할을 **`Console Default Role`**로 유지합니다.
6. 트랜스코딩한 파일 및 재생 목록의 Amazon S3 버킷 구성에 출력 버킷의 이름을 입력합니다.
7. 트랜스코딩한 파일 및 재생 목록의 Amazon S3 버킷 구성에서 표준 스토리지 클래스를 선택합니다.
8. 알림 드롭다운 메뉴를 확장합니다.
9. 4개 이벤트 유형 모두에 대해 Use an existing SNS topic(기존 SNS 주제 사용)을 선택하고 Select a Topic(주제 선택) 아래에서 **`ets-sample-topic`**을 선택합니다.
10. Create Pipeline을 클릭합니다.
11. Elastic Transcoder 파이프라인 ID를 기록합니다.

확장 가능한 방법으로 콘텐츠를 제공하는 Amazon CloudFront 배포 생성

1. Amazon CloudFront 콘솔을 엽니다.
2. 콘솔의 탐색 창에서 Private Content(프라이빗 콘텐츠)를 확장하고 원본 액세스 ID를 클릭합니다.
3. Create Origin Access Identity(원본 액세스 ID 생성)를 클릭합니다.
4. 생성을 클릭합니다.
5. 원본 액세스 ID를 만들 때 생성된 Amazon S3 정식 사용자 ID를 기록합니다.
6. 탐색 창에서 배포를 클릭합니다.
7. Create Distribution(배포 생성)을 클릭합니다.

8. 웹을 선택하고 계속을 클릭합니다.
9. 오리진 설정에서 Amazon S3 출력 버킷을 오리진 도메인 이름으로 입력합니다.
10. Origin ID(오리진 ID) 필드에 **`S3-transcoder-sample-output`**을 입력합니다.
11. Restrict Bucket Access(버킷 액세스 제한)에서 예를 선택합니다.
12. Create Distribution(배포 생성)을 클릭하고 배포의 도메인 이름을 기록합니다.

Amazon S3 버킷 정책 수정

1. Amazon S3 콘솔을 엽니다.
2. 출력 버킷 옆에서 확대경 아이콘을 클릭하여 버킷 속성을 불러옵니다.
3. 권한을 확장하고 Add bucket policy(버킷 정책 추가)를 클릭합니다.
4. 다음 정책 설명을 입력하고 CloudFront-oai-s3-canonical-user-id를 **`your recorded Amazon S3 canonical user ID`**로, s3-output-bucket-name을 **`the name of your output bucket`**으로 바꿉니다.
5. 저장을 클릭합니다.

```
{
  "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 샘플 섹션으로 건너뜁니다.

Python 환경 설정

이 단원에서는 Python 환경을 설정하는 절차를 안내합니다.

주제

- AWS SDK for Python 설정

- Amazon S3 입력 및 출력 버킷 생성
- 작업 상태 알림을 받는 Amazon SNS 주제 생성
- 작업 상태 알림을 폴링하는 Amazon SQS 대기열 생성
- Amazon SQS 대기열에서 Amazon SNS 주제 구독
- Elastic Transcoder 파이프라인 생성
- 확장 가능한 방법으로 콘텐츠를 제공하는 Amazon CloudFront 배포 생성
- Amazon S3 버킷 정책 수정

이 절차는 다음과 같은 한 가지 기본 전제가 있습니다.

- 샘플이 Python 버전 2.7과 호환되도록 작성되었습니다.

AWS SDK for Python 설정

AWS SDK for Python은 여기서 받을 수 있습니다. PIP를 사용하여 다음 명령으로 쉽게 설치할 수도 있습니다.

```
pip-2.7 install boto
```

자세한 내용은 AWS SDK for Python 설명서를 참조하세요.

Amazon S3 입력 및 출력 버킷 생성

1. Amazon S3 콘솔을 엽니다.
2. 콘솔에서 버킷 만들기를 클릭합니다.
3. 버킷 이름 필드에 입력 버킷의 원하는 이름을 입력합니다.
4. Amazon S3 콘솔에서 버킷 만들기를 다시 클릭합니다.
5. 버킷 이름 필드에 출력 버킷의 원하는 이름을 입력합니다.

작업 상태 알림을 받는 Amazon SNS 주제 생성

1. Amazon SNS 콘솔을 엽니다.
2. 콘솔에서 Create and Add(생성 및 추가)를 클릭하고 새로운 주제 생성을 선택합니다.
3. 주제 이름 필드에 **ets-sample-topic**을 입력하고 주제 생성을 클릭합니다.

4. ets-sample-topic의 ARN을 기록합니다.

작업 상태 알림을 폴링하는 Amazon SQS 대기열 생성

1. Amazon SQS 콘솔을 엽니다.
2. 콘솔에서 새로운 대기열 생성을 클릭합니다.
3. 대기열 이름 필드에 **ets-sample-queue**를 입력하고 대기열 생성을 클릭합니다.
4. Amazon SQS 대기열의 URL을 기록합니다.

Amazon SQS 대기열에서 Amazon SNS 주제 구독

1. Amazon SQS 콘솔을 엽니다.
2. 콘솔의 대기열 목록에서 ets-sample-queue를 선택합니다.
3. 대기열 작업에서 Amazon SNS 주제에 대한 대기열 구독을 선택합니다.
4. 주제 선택에서 ets-sample-topic을 선택하고 구독을 클릭합니다.

대기열이 성공적으로 주제를 구독하도록 등록되었다는 확인 메시지가 표시될 것입니다.

Elastic Transcoder 파이프라인 생성

1. Amazon Elastic Transcoder 콘솔을 엽니다.
2. 콘솔에서 Create New Pipeline(새 파이프라인 생성)을 클릭합니다.
3. 파이프라인 이름 필드에 **ets-sample-pipeline**을 입력합니다.
4. Input Bucket(입력 버킷) 필드에 입력 버킷의 이름을 입력합니다.
5. IAM 역할을 **Console Default Role**로 유지합니다.
6. 트랜스코딩한 파일 및 재생 목록의 Amazon S3 버킷 구성에 출력 버킷의 이름을 입력합니다.
7. 트랜스코딩한 파일 및 재생 목록의 Amazon S3 버킷 구성에서 표준 스토리지 클래스를 선택합니다.
8. 알림 드롭다운 메뉴를 확장합니다.
9. 4개 이벤트 유형 모두에 대해 Use an existing SNS topic(기존 SNS 주제 사용)을 선택하고 Select a Topic(주제 선택) 아래에서 **ets-sample-topic**을 선택합니다.
10. Create Pipeline을 클릭합니다.
11. Elastic Transcoder 파이프라인 ID를 기록합니다.

확장 가능한 방법으로 콘텐츠를 제공하는 Amazon CloudFront 배포 생성

1. Amazon CloudFront 콘솔을 엽니다.
2. 콘솔의 탐색 창에서 Private Content(프라이빗 콘텐츠)를 확장하고 원본 액세스 ID를 클릭합니다.
3. Create Origin Access Identity(원본 액세스 ID 생성)를 클릭합니다.
4. 생성을 클릭합니다.
5. 원본 액세스 ID를 만들 때 생성된 Amazon S3 정식 사용자 ID를 기록합니다.
6. 탐색 창에서 배포를 클릭합니다.
7. Create Distribution(배포 생성)을 클릭합니다.
8. 웹을 선택하고 계속을 클릭합니다.
9. 오리진 설정에서 Amazon S3 출력 버킷을 오리진 도메인 이름으로 입력합니다.
10. Origin ID(오리진 ID) 필드에 **`S3-transcoder-sample-output`**을 입력합니다.
11. Restrict Bucket Access(버킷 액세스 제한)에서 예를 선택합니다.
12. Create Distribution(배포 생성)을 클릭하고 배포의 도메인 이름을 기록합니다.

Amazon S3 버킷 정책 수정

1. Amazon S3 콘솔을 엽니다.
2. 출력 버킷 옆에서 확대경 아이콘을 클릭하여 버킷 속성을 불러옵니다.
3. 권한을 확장하고 Add bucket policy(버킷 정책 추가)를 클릭합니다.
4. 다음 정책 설명을 입력하고 CloudFront-oai-s3-canonical-user-id를 **`your recorded Amazon S3 canonical user ID`**로, s3-output-bucket-name을 **`the name of your output bucket`**으로 바꿉니다.
5. 저장을 클릭합니다.

```
{
  "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 샘플 섹션으로 건너뜁니다.

Java 샘플

모든 Java 샘플은 다음과 같은 두 가지 기본 전제가 있습니다.

- 샘플이 Java 버전 1.6 이상에서 사용하도록 작성되었습니다.
- 샘플을 Eclipse 및 AWS Toolkit for Eclipse를 사용하여 실행합니다.

HLS 샘플

주제

- Java HLS 샘플 다운로드
- 샘플 코드 조각
- 업무

이 샘플은 가변 비트 레이트 스트림을 재생하는 데 사용할 수 있는 HLS 작업과 HLS 재생목록 파일을 생성하는 방법을 보여줍니다.

Java HLS 샘플 다운로드

여기에서 샘플 코드를 다운로드할 수 있습니다.

샘플 코드 조각

Java 코드 샘플은 다음을 포함합니다.

- `HlsJobCreationSample` 클래스

업무

샘플을 실행하려면 다음과 같이 합니다.

1. Java 환경 설정의 지침에 따라 환경을 설정합니다.

2. 샘플 코드를 다운로드합니다.
3. Eclipse에서 com/amazonaws/services/elastictranscoder/samples/HlsJobCreationSample.java를 엽니다.
4. PIPELINE_ID와 INPUT_KEY를 해당 값으로 바꿉니다.
5. Eclipse에서 샘플을 실행합니다.

알림 샘플

주제

- 알림 구문 예제
- Java 알림 샘플 다운로드
- 샘플 코드 조각
- 업무

알림 구문 예제


```
{
   "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+jWzWwygXdiJXvZXl94qhd/
tLln1lxPqijjivb5RWu7n5yzdZwbod6lpLwyZ2TfWM6dZt57OzsG3GbdTxgqwVsztVSyWCYhcV8f
+CrT3IQrfrU3Me/SRYVUqrSUXXsu4Ls7A2q9mosG7v0Sn+3Z1rAa9+Rf6QmkfAg05UX0sLyA
```

```
+I2hFqTu5oAGDK4Cm6FHuIwV+oYJXNBbGWuS7tD6mRNwrYvPBlUvBLXx9m3CbqSXv5Yoj39vglv
+1djtaLA3GpwX+B1hHx8QT373lgmmsmGDRWhSQretpOTWDYb81PV2K0bg==",
   "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 알림 샘플 다운로드

여기에서 샘플 코드를 다운로드할 수 있습니다.

샘플 코드 조각

Java 코드 샘플은 다음을 포함합니다.

- `Notification` 클래스: Java의 알림을 사용하기 위해 Jackson JSON 라이브러리를 사용하여 작업 상태 알림을 POJO로 역직렬화합니다. `Notification` 클래스는 Amazon SQS 대기열로부터 돌려받는 메시지를 모델링합니다.
- `JobStatusNotification` 클래스: `JobStatusNotification` 클래스는 Amazon SNS 알림을 모델링합니다.
- `SqsQueueNotificationWorker` 클래스: `SqsQueueNotificationWorker` 클래스는 별도의 스레드로 시작하여 Amazon SQS를 폴링하고 작업 상태 알림을 처리할 수 있습니다. 이 클래스는 메시지를 수신하고, 수신한 각 알림에 대해 등록된 모든 핸들러를 호출한 후 대기열에서 메시지를 삭제합니다.
- `JobStatusNotificationHandler` 인터페이스: `JobStatusNotificationHandler` 인터페이스는 알림에 대한 사용자 지정 처리를 위해 구현할 수 있습니다.
- `JobStatusNotificationsSample` 클래스: `JobStatusNotificationsSample` 클래스는 작업을 생성하고, 완료될 때까지 기다립니다. 작업이 터미널 상태가 되면 대기열 워커와 애플리케이션이 종료됩니다.

업무

샘플을 실행하려면 다음과 같이 합니다.

1. Java 환경 설정의 지침에 따라 환경을 설정합니다.
2. 샘플 코드를 다운로드합니다.

3. JAVA 프로젝트 소스 디렉터리에 샘플 코드의 압축을 풉니다.
4. Eclipse에서 `com/amazonaws/services/elastictranscoder/samples/JobStatusNotificationsSample.java`를 엽니다.
5. PIPELINE_ID, SQS_QUEUE_URL, INPUT_KEY를 해당 값으로 바꿉니다.
6. Eclipse에서 샘플을 실행합니다.

Ruby 샘플

모든 Ruby 샘플은 다음 한 가지 조건을 기본 전제로 합니다.

- 샘플이 Ruby 버전 1.9와 호환되도록 작성되었습니다.

HLS 샘플

주제

- Ruby HLS 샘플 다운로드
- 샘플 코드 조각
- 업무

이 샘플은 가변 비트 레이트 스트림을 재생하는 데 사용할 수 있는 HLS 작업과 HLS 재생목록 파일을 생성하는 방법을 보여줍니다.

Ruby HLS 샘플 다운로드

여기에서 샘플 코드를 다운로드할 수 있습니다.

샘플 코드 조각

Ruby 샘플 코드는 다음을 포함합니다.

- `HlsJobCreationSample.rb` 파일

업무

샘플을 실행하려면 다음과 같이 합니다.

1. Ruby 환경 설정의 지침에 따라 환경을 설정합니다.

2. 샘플 코드를 다운로드합니다.
3. Ruby 프로젝트 소스 디렉터리에 샘플 코드의 압축을 풉니다.
4. HlsJobCreationSample.rb를 편집하여 pipeline_id와 input_key를 해당 값으로 바꿉니다.
5. 터미널에서 샘플 코드의 압축을 풀어 놓은 디렉터리로 이동하여 다음을 실행합니다.

```
$ruby HlsJobCreationSample.rb
```

알림 샘플

주제

- 알림 구문 예제
- Ruby 알림 샘플 다운로드
- 샘플 코드 조각
- 업무

알림 구문 예제

```
{
   "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+jWzWwygXdiJXvZXl94qhd/
tLln1lxPqijjivb5RWu7n5yzdZwbod6lpLwyZ2TfWM6dZt57OzsG3GbdTxgqwVsztVSyWCYhcV8f
```

```
+CrT3IQrfrU3Me/SRYVUqrSUXXsu4Ls7A2q9mosG7v0Sn+3Z1rAa9+Rf6QmkfAg05UX0sLyA
+I2hFqTu5oAGDK4Cm6FHuIwV+oYJXNBbGWuS7tD6mRNwrYvPBlUvBLXx9m3CbqSXv5Yoj39vglv
+1djtaLA3GpwX+B1hHx8QT373lgmmsmGDRWhSQretpOTWDYb81PV2K0bg==",
   "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 알림 샘플 다운로드

여기에서 샘플 코드를 다운로드할 수 있습니다.

샘플 코드 조각

Ruby 샘플 코드는 다음을 포함합니다.

- `SqsQueueNotificationWorker` 클래스: `SqsQueueNotificationWorker` 클래스는 에서 알림을 폴링하고, 각 알림에 대해 등록된 모든 핸들러를 호출한 후 대기열에서 메시지를 삭제합니다. 워커는 별도의 스레드에서 실행됩니다. Ruby 구현에서 "그린" 스레드를 사용한 경우 한 번에 하나의 스레드만 실행됩니다(멀티프로세싱 지원 안 됨).
- `JobStatusNotificationsSample.rb` 스크립트: 이 스크립트는 Elastic Transcoder 작업을 생성하고, Amazon SQS 워커를 시작한 후 작업이 완료될 때까지 기다립니다. 생성된 작업의 처리가 완료되면 제공된 핸들러가 Amazon SQS 워커에 중단을 지시합니다. 알림 처리는 add_handlers 메서드를 호출한 후 하나의 인수를 받는 Proc 또는 lambda를 핸들러로 제공하여 수행됩니다. 알림이 수신될 때마다, 등록된 모든 핸들러는 핸들러에 입력 인수로 제공된 알림으로 호출됩니다.

업무

샘플을 실행하려면 다음과 같이 합니다.

1. Ruby 환경 설정의 지침에 따라 환경을 설정합니다.
2. 샘플 코드를 다운로드합니다.
3. Ruby 프로젝트 소스 디렉터리에 샘플 코드의 압축을 풉니다.
4. `JobStatusNotificationsSample.rb`를 편집하여 pipeline_id, sqs_queue_url, input_key를 해당 값으로 바꿉니다.
5. 터미널에서 샘플 코드의 압축을 풀어 놓은 디렉터리로 이동하여 다음을 실행합니다.

```
$ruby JobStatusNotificationsSample.rb
```

PHP 샘플

모든 PHP 샘플은 다음 세 가지 조건을 기본 전제로 합니다.

- PHP 샘플이 Apache 서버 내에서 실행됩니다.
- Apache 서버는 Amazon SNS 알림을 수신하기 위해 인터넷으로부터 입력을 받을 수 있어야 합니다.
- 샘플이 PHP 버전 5.4 이상에서 사용하도록 작성되었습니다.

HLS 샘플

주제

- PHP HLS 샘플 다운로드
- 샘플 코드 조각
- 업무

이 샘플은 가변 비트 레이트 스트림을 재생하는 데 사용할 수 있는 HLS 작업과 HLS 재생목록 파일을 생성하는 방법을 보여줍니다.

PHP HLS 샘플 다운로드

여기에서 샘플 코드를 다운로드할 수 있습니다.

샘플 코드 조각

PHP 샘플 코드는 다음을 포함합니다.

- `HlsJobCreationSample.php` 파일

업무

샘플을 실행하려면 다음과 같이 합니다.

1. PHP 환경 설정의 지침에 따라 환경을 설정합니다.

2. 샘플 코드를 다운로드합니다.
3. 브라우저에서 http://<your-endpoint>/transcoder-samples/HlsJobCreationSample.php 페이지를 엽니다.
4. 파이프라인 ID와 입력 키를 입력하고 양식을 제출하여 작업을 생성합니다.

알림 샘플

주제

- 알림 구문 예제
- PHP 알림 샘플 다운로드
- 샘플 코드 조각
- 업무

알림 구문 예제


```
{
   "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+jWzWwygXdiJXvZXl94qhd/
tLln1lxPqijjivb5RWu7n5yzdZwbod6lpLwyZ2TfWM6dZt57OzsG3GbdTxgqwVsztVSyWCYhcV8f
+CrT3IQrfrU3Me/SRYVUqrSUXXsu4Ls7A2q9mosG7v0Sn+3Z1rAa9+Rf6QmkfAg05UX0sLyA
+I2hFqTu5oAGDK4Cm6FHuIwV+oYJXNBbGWuS7tD6mRNwrYvPBlUvBLXx9m3CbqSXv5Yoj39vglv
+1djtaLA3GpwX+B1hHx8QT373lgmmsmGDRWhSQretpOTWDYb81PV2K0bg==",
```

```
   "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 알림 샘플 다운로드

여기에서 샘플 코드를 다운로드할 수 있습니다.

샘플 코드 조각

PHP 샘플 코드는 다음을 포함합니다.

- `JobStatusNotificationsSampleNotificationHandler.php` 파일: 알림이 서버에 POST되면 알림 핸들러가 /tmp/<job-id> 아래의 파일에 상태를 기록합니다.
- `JobStatusNotificationsSample.php` 파일: 알림 핸들러가 /tmp/<job-id> 아래의 파일에 상태를 기록하면, 할당된 쿼리 문자열의 Id로 지정된 /tmp/<job-id> 상태 파일이 `JobStatusNotificationsSample.php`에 로드됩니다.

업무

샘플을 실행하려면 다음과 같이 합니다.

1. PHP 환경 설정의 지침에 따라 환경을 설정합니다.
2. 샘플 코드를 다운로드합니다.
3. PHP 프로젝트 소스 디렉터리에 샘플 코드의 압축을 풉니다.
4. 브라우저에서 http://<your-endpoint>/transcoder-samples/JobStatusNotificationsSample.php 페이지를 엽니다.
5. 트랜스코딩하려는 파이프라인 ID와 입력 키를 입력하고 Create Job 버튼을 누릅니다.

Python 샘플

모든 Python 샘플은 다음 한 가지 조건을 기본 전제로 합니다.

- 샘플이 Python 버전 2.7과 호환되도록 작성되었습니다.

HLS 샘플

주제

- Python HLS 샘플 다운로드
- 샘플 코드 조각
- 업무

이 샘플은 가변 비트 레이트 스트림을 재생하는 데 사용할 수 있는 HLS 작업과 HLS 재생목록 파일을 생성하는 방법을 보여줍니다.

Python HLS 샘플 다운로드

여기에서 샘플 코드를 다운로드할 수 있습니다.

샘플 코드 조각

Python 샘플 코드는 다음을 포함합니다.

- `HlsJobCreationSample.py` 파일

업무

샘플을 실행하려면 다음과 같이 합니다.

1. Python 환경 설정의 지침에 따라 환경을 설정합니다.
2. 샘플 코드를 다운로드합니다.
3. 샘플 코드의 압축을 풉니다.
4. `HlsJobCreationSample.py`를 편집하여 pipeline_id와 input_key를 해당 값으로 바꿉니다.
5. 터미널에서 샘플 코드의 압축을 풀어 놓은 디렉터리로 이동하여 다음을 실행합니다.

```
$python HlsJobCreationSample.py
```

알림 샘플

주제

- 알림 구문 예제
- Python 알림 샘플 다운로드
- 샘플 코드 조각
- 업무

알림 구문 예제

```
{
   "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+jWzWwygXdiJXvZXl94qhd/
tLln1lxPqijjivb5RWu7n5yzdZwbod6lpLwyZ2TfWM6dZt57OzsG3GbdTxgqwVsztVSyWCYhcV8f
+CrT3IQrfrU3Me/SRYVUqrSUXXsu4Ls7A2q9mosG7v0Sn+3Z1rAa9+Rf6QmkfAg05UX0sLyA
+I2hFqTu5oAGDK4Cm6FHuIwV+oYJXNBbGWuS7tD6mRNwrYvPBlUvBLXx9m3CbqSXv5Yoj39vglv
+1djtaLA3GpwX+B1hHx8QT373lgmmsmGDRWhSQretpOTWDYb81PV2K0bg==",
   "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 알림 샘플 다운로드

여기에서 샘플 코드를 다운로드할 수 있습니다.

샘플 코드 조각

Python 샘플 코드는 다음을 포함합니다.

- `SqsQueueNotificationWorker.py` 클래스: `SqsQueueNotificationWorker` 클래스는 Amazon SQS를 폴링하여 개별 프로세스로 알림을 처리합니다. 따라서 Python에서는 멀티프로세싱이 가능합니다. 이 클래스는 메시지를 수신하고, 수신한 각 알림에 대해 등록된 모든 핸들러를 호출한 후 대기열에서 메시지를 삭제합니다. 이 클래스는 또한 Elastic Transcoder 알림 처리를 위한 프레임워크를 위해 제공된 `JobStatusNotificationHandler` 클래스에 대한 정의를 포함합니다. 이 클래스는 확장할 수 있으며 사용자 지정 작업 처리를 제공하기 위해 핸들 메서드를 덮어씁니다. 이 클래스는 Pythons 덕 타이핑(duck typing) 표준을 따르지 않지만, 이 샘플을 위해 공식적인 핸들러 정의를 제공합니다.
- `JobStatusNotificationSample.py` 스크립트: `JobStatusNotificationSample.py` 스크립트는 Elastic Transcoder에서 작업을 생성하고, 완료될 때까지 기다립니다. 작업이 완료되면 대기열 폴링 프로세스가 중지되고 종료됩니다. 핸들러는 폴링하는 대기열의 별도 프로세스에서 실행되므로 멀티프로세싱 값 객체를 사용한 프로세스 간 동기화가 필요합니다.

업무

샘플을 실행하려면 다음과 같이 합니다.

1. Python 환경 설정의 지침에 따라 환경을 설정합니다.
2. 샘플 코드를 다운로드합니다.
3. 로컬 폴더에 샘플 코드의 압축을 풉니다.
4. `JobStatusNotificationsSample.py`를 편집하여 pipeline_id, sqs_queue_url, input_key를 해당 값으로 바꿉니다.
5. 터미널에서 샘플 코드의 압축을 풀어 놓은 디렉터리로 이동하여 다음을 실행합니다.

```
$python JobStatusNotificationsSample.py
```

API 참조

AWS SDK가 있는 언어를 사용한다면, API를 통해 작업을 하려고 애쓰기보다는 SDK를 이용하십시오. SDK는 인증을 더 간단하게 만들고, 개발 환경에 쉽게 통합되며, Elastic Transcoder 명령에 쉽게 액세스할 수 있게 해줍니다. 환경 설정 방법, SDK 문서로 연결되는 링크, 샘플 코드 등 AWS SDK에 대한 자세한 내용은 AWS SDK 샘플 코드 단원을 참조하십시오.

주제

- AWS SDK 사용
- Elastic Transcoder에 대한 HTTP 요청 만들기
- 요청에 서명하기
- Elastic Transcoder에서의 오류 처리
- 파이프라인 작업
- 작업
- 프리셋 작업

AWS SDK 사용

AWS SDK가 있는 언어를 사용한다면, API를 통해 작업을 하려고 애쓰기보다는 SDK를 이용하십시오. SDK는 인증을 더 간단하게 만들고, 개발 환경에 쉽게 통합되며, Elastic Transcoder 명령에 쉽게 액세스할 수 있게 해줍니다. 환경 설정 방법, SDK 문서로 연결되는 링크, 샘플 코드 등 AWS SDK에 대한 자세한 내용은 AWS SDK 샘플 코드 단원을 참조하십시오.

Elastic Transcoder에 대한 HTTP 요청 만들기

RFC 2616에 정의되어 있는 것처럼, Elastic Transcoder REST 요청은 HTTPS 요청입니다. (자세한 내용은 http://www.ietf.org/rfc/rfc2616.txt를 참조하세요.) 이 섹션에서는 Elastic Transcoder REST 요청의 구조를 설명합니다. 수행할 수 있는 작업에 대한 자세한 설명은 파이프라인 작업, 작업, 프리셋 작업을 참조하세요.

일반적인 REST 작업은 HTTPS 요청을 Elastic Transcoder로 보내고 응답을 기다리는 과정으로 구성됩니다. HTTP 요청과 마찬가지로, Elastic Transcoder에 대한 REST 요청에는 요청 방법, URI, 요청 헤더, 그리고 때로는 쿼리 문자열이나 요청 본문이 포함됩니다. 응답에는 HTTP 상태 코드, 응답 헤더, 그리고 때로는 응답 본문이 포함됩니다.

주제

- HTTP 헤더 콘텐츠
- HTTP 요청 본문
- HTTP 응답

HTTP 헤더 콘텐츠

Elastic Transcoder 작업 시 HTTP 요청의 헤더에 다음 정보를 제공해야 합니다.

호스트(필수)

리소스가 생성되는 위치를 지정하는 Elastic Transcoder 엔드포인트입니다. 이 값은 명명된 리전 엔드포인트여야 합니다. Amazon S3 버킷과 Elastic Transcoder 파이프라인 및 작업에 동일한 엔드포인트를 사용하는 것이 좋습니다. 서로 다른 엔드포인트를 사용하면 Amazon S3 버킷이 포함된 리전과 Elastic Transcoder가 인코딩하는 리전 간에 전송되는 데이터에 대해 추가 요금이 발생합니다. 또한 데이터 전송에 필요한 시간으로 인해 트랜스코딩된 파일에 대한 액세스가 지연됩니다.

지원되는 모든 Elastic Transcoder 엔드포인트 목록은 Amazon Web Services 일반 참조에서 리전 및 엔드포인트 섹션을 참조하세요.

리전 간 요금에 대한 자세한 내용은 Amazon S3 요금의 데이터 전송 요금을 참조하세요.

x-amz-date 또는 날짜(필수)

`Authorization` 헤더에 포함된 서명을 만드는 데 사용되는 날짜. ISO 8601 표준 형식을 사용하여 UTC 시간으로 날짜를 지정합니다. 예: `X-Amz-Date: 20130613T203622Z`.

`x-amz-date` 또는 `Date`를 포함시켜야 합니다. 일부 HTTP 클라이언트 라이브러리에서는 `Date` 헤더를 설정할 수 없습니다. `x-amz-date` 헤더가 있으면 요청 인증 시 모든 `Date` 헤더가 무시됩니다.

타임스탬프는 요청 수신 시 AWS 시스템 시간이 15분 이내여야 합니다. 그렇지 않으면 다른 사람이 요청을 재생하는 것을 방지하기 위해 `RequestExpired` 오류 코드와 함께 요청이 실패합니다.

권한 부여(필수)

요청 인증에 필요한 정보. 이 헤더를 구성하는 방법에 대한 자세한 내용은 요청에 서명하기 섹션을 참조하세요.

Content-Type(조건부)

JSON 및 버전을 지정합니다. 예: `Content-Type: application/x-amz-json-1.0`.

조건: POST 요청에 필수입니다.

Content-Length(조건부)

RFC 2616에 따른 메시지의 길이(헤더 제외).

조건: 요청 본문 자체에 정보가 포함되어 있는 경우에 필요합니다(대부분의 도구 키트는 이 헤더를 자동으로 추가함).

다음은 파이프라인을 생성하기 위한 HTTP 요청의 헤더 예제입니다.

```
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 요청 본문

많은 Elastic Transcoder API 작업에는 JSON 형식의 데이터를 요청 본문에 포함시켜야 합니다. JSON은 Elastic Transcoder 스키마를 준수합니다.

> **Note**
>
> 요청 본문 내 JSON 값은 문자열입니다.

Example요청

다음 예제 요청은 간단한 JSON 문을 사용하여 `sample.mp4`라는 파일을 트랜스코딩하고 이를 `sams-birthday.mp4`로 저장하는 작업을 만듭니다.

```
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 응답

모든 Elastic Transcoder API 작업에서는 JSON 형식의 데이터가 응답에 포함됩니다. JSON은 Elastic Transcoder 스키마를 준수합니다.

> **Note**
>
> 응답 내 JSON 값은 문자열입니다.

다음은 HTTP 응답의 몇 가지 중요 헤더와 해당되는 경우 애플리케이션에서 이 헤더를 처리하는 방법입니다.

HTTP/1.1

이 헤더 다음에는 상태 코드가 이어집니다. 상태 코드 200은 작업 성공을 나타냅니다. 오류 코드에 대한 자세한 내용은 API 오류 코드(클라이언트 및 서버 오류) 섹션을 참조하세요.

유형: 문자열

x-amzn-RequestId

요청을 고유하게 식별하기 위해 Elastic Transcoder가 생성한 값입니다(예: K2QH8DNOU907N97FNA2GDLL8OBVV4KQNSO5AEMVJF66Q9ASUAAJG). Elastic Transcoder에 문제가 있는 경우 AWS는 이 값을 사용하여 문제를 해결할 수 있습니다. 이러한 값을 로깅할 것을 권장합니다.

유형: 문자열

Content-Length

응답 본문의 길이(바이트)입니다.

유형: 문자열

날짜

Elastic Transcoder가 응답한 날짜 및 시간입니다(예: `Sun, 25 Mar 2012 12:00:00 GMT`). 날짜 형식은 RFC 2616 섹션 3.3에 지정된 전체 날짜 형식 중 하나여야 합니다.

유형: 문자열

요청에 서명하기

AWS에서 SDK를 제공하는 언어를 사용한다면 SDK를 사용하는 것이 좋습니다. 모든 AWS SDK는 요청 서명 프로세스를 대폭 간소화하므로 Elastic Transcoder API를 사용할 때보다 상당히 시간이 절약됩니다. 또한 SDK는 개발 환경에 쉽게 통합되며 관련 명령에 쉽게 액세스할 수 있습니다.

Elastic Transcoder는 보내는 모든 요청에 서명하여 요청을 인증하도록 요구합니다. 요청에 서명하려면 입력을 바탕으로 해시 값을 반환하는 암호화 해시 함수를 사용하여 디지털 서명을 계산합니다. 입력에는 요청 텍스트와 보안 액세스 키가 포함됩니다. 해시 함수는 요청에 서명으로 포함하는 해시 값을 반환합니다. 서명은 요청에서 `Authorization` 헤더의 일부입니다.

Elastic Transcoder는 요청을 수신한 후 사용자가 요청에 서명할 때와 동일한 해시 함수 및 입력을 사용하여 서명을 재계산합니다. 결과 서명이 요청 서명과 일치할 경우 Elastic Transcoder가 요청을 처리합니다. 그렇지 않으면 요청이 거부됩니다.

Elastic Transcoder는 AWS 서명 버전 4를 이용한 인증을 지원합니다. 서명을 계산하기 위한 프로세스는 다음 세 작업으로 나뉠 수 있습니다.

- 작업 1: 정식 요청 생성

https://docs.aws.amazon.com/general/latest/gr/sigv4-create-canonical-request.html의Amazon Web Services 일반 참조작업 1: Signature Version 4에 대한 정식 요청 생성에 설명되어 있는 대로 정규 형식으로 HTTP 요청을 생성합니다.

- 작업 2: 서명할 문자열 생성

  암호화 해시 함수에 대한 입력 값 중 하나로 사용할 문자열을 만듭니다. 서명할 문자열이라는 문자열은 해시 알고리즘의 이름, 요청 날짜, 자격 증명 범위 문자열, 이전 작업에서 정규화된 요청을 연결한 것입니다. 자격 증명 범위 문자열 자체는 날짜, 리전 및 서비스 정보를 연결한 것입니다.

  `X-Amz-Credential` 파라미터에 대해 다음을 지정합니다.

  - 요청(예: `us-east-1`)을 전송할 엔드포인트에 대한 코드. Elastic Transcoder의 리전 및 엔드포인트 목록은 Amazon Web Services 일반 참조의 리전 및 엔드포인트 섹션을 참조하세요. 엔드포인트에 대해 코드를 지정할 때는 `elastictranscoder.`와 `.amazonaws.com` 사이 부분만 포함시키세요.
  - 서비스 약어에 대한 `elastictranscoder`

  예시:

  ```
  X-Amz-Credential=AKIAIOSFODNN7EXAMPLE/20130501/us-east-1/
  elastictranscoder/aws4_request
  ```

- 작업 3: 서명 생성

  서명할 문자열과 파생된 의 두 입력 문자열을 허용하는 암호화 해시 함수를 사용하여 요청에 대한 서명을 만듭니다. 파생된 키는 보안 액세스 키로 시작해서 자격 증명 범위 문자열을 사용하여 일련의 해시 기반 메시지 인증 코드(HMAC)를 만들어 계산됩니다.

Elastic Transcoder에서의 오류 처리

주제

- API 오류 코드(클라이언트 및 서버 오류)
- 작업 처리 중 오류 발생
- 오류 발견
- 오류 재시도 횟수 및 지수 백오프

Elastic Transcoder API로 요청을 전송하고 응답을 수신할 때 두 유형의 API 오류가 발생할 수 있습니다.

- 클라이언트 오류: 클라이언트 오류는 4xx HTTP 응답 코드로 알 수 있습니다. 클라이언트 오류는 인증 실패, 필수 파라미터 누락 등 Elastic Transcoder가 클라이언트 요청에서 문제를 발견했다는 것을 의미합니다. 따라서 요청을 다시 제출하기 전에 클라이언트 애플리케이션의 문제를 해결해야 합니다.
- 서버 오류 서버 오류는 5xx HTTP 응답 코드로 알 수 있으며, Amazon 측에서 해결해야 합니다. 사용자는 성공할 때까지 요청을 다시 제출하거나 재시도할 수 있습니다.

각 API 오류에 대해 Elastic Transcoder는 다음 값을 반환합니다.

- 상태 코드, 예: `400`
- 오류 코드, 예: `ValidationException`)
- 오류 메시지, 예: `Supplied AttributeValue is empty, must contain exactly one of the supported datatypes`

Elastic Transcoder가 클라이언트 및 서버 오류에 대해 반환하는 오류 코드의 목록은 API 오류 코드(클라이언트 및 서버 오류)를 참조하세요.

Elastic Transcoder가 작업을 처리하는 동안 오류가 발생할 수도 있습니다. 자세한 내용은 작업 처리 중 오류 발생 섹션을 참조하세요.

API 오류 코드(클라이언트 및 서버 오류)

HTTP 상태 코드는 작업의 성공 여부를 나타냅니다.

응답 코드 `200`은 작업 성공을 나타냅니다. 다른 오류 코드는 클라이언트 오류(4xx) 또는 서버 오류(5xx)를 나타냅니다.

다음 표는 Elastic Transcoder에서 반환되는 오류를 나열한 것입니다. 일부는 동일한 요청을 재시도하는 것만으로도 해결됩니다. 연이어 재시도하여 해결 가능성이 있는 오류는 따로 표시되어 있습니다. 재시도 열의 값은 다음을 의미합니다.

- Yes: 동일한 요청을 다시 제출합니다.
- No 새 요청을 제출하기 전에 클라이언트 측에서 문제를 해결합니다.

요청 재시도에 대한 자세한 내용은 오류 재시도 횟수 및 지수 백오프 섹션을 참조하세요.

HTTP 상태 코드	오류 코드	메시지	원인	다시 해 보 십 시 오
400	Conditional Check Failed Exception	조건부 요청이 실패하였습니다.	예: 예상 값이 시스템에 저장된 값과 일치하지 않습니다.	아니요
400	Incomplete Signature Exception	요청 서명이 AWS 표준을 준수하지 않습니다.	요청 서명에 일부 필요한 구성 요소가 빠져있습니다. HTTP 헤더 콘텐츠 섹션을 참조하세요.	아니요
403	Missing Authentication Token Exception	요청에는 유효한(등록된) AWS 액세스 키 ID가 포함되어야 합니다.	요청에 필수 `x-amz-security-token` 이 포함되어 있지 않습니다. Elastic Transcoder에 대한 HTTP 요청 만들기 섹션을 참조하세요.	아니요
400	Validation Exception	다양한.	요청에서 하나 이상의 값이 누락되거나 유효하지 않습니다. 예를 들어 값이 비어 있거나 최대 유효 값보다 큽니다.	아니요
403	AccessDenied Exception	• 시스템 프리셋 삭제가 허용되지 않음: account=<accountId>, presetId=<presetId>. • 일반 인증 실패. 클라이언트가 요청에 정확히 서명하지 않았습니다. 요청에 서명하기 섹션을 참조하세요.	시스템 프리셋을 삭제하려고 시도했거나, Elastic Transcoder API에 대한 호출의 서명이 유효하지 않거나, IAM 사용자가 작업을 수행할 수 있는 권한이 없습니다.	아니요

HTTP 상태 코드	오류 코드	메시지	원인	다시 해 보 십 시 오
404	ResourceNot Found Exception	• 지정한 <resource>를 찾을 수 없음: <resourceId>. • 지정한 작업을 찾을 수 없음: account=<accountId>, jobId=<jobId>. • 지정한 파이프라인을 찾을 수 없음: account=<accountId>, pipelineId=<pipelineId> • 지정한 프리셋을 찾을 수 없음: account=<accountId>, presetId=<presetId>	예: 작업을 추가하려는 파이프라인이 존재하지 않거나, 현재 생성 중입니다.	아니요
409	Resource InUse Exception	• <resource>가 이미 사용 중임: accountId=<accountId>, resourceId=<resourceId>. • 파이프라인에 활성 작업이 있음: account=<accountId>, pipeline=<pipelineId>.	예: 현재 사용 중인 파이프라인을 삭제하려고 시도했습니다.	아니요

HTTP 상태 코드	오류 코드	메시지	원인	다시 해보십시오
429	Limit Exceeded Exception	• 계정의 파이프라인 수가 이미 최대 허용 개수임: 계정=<accountId>, 최대 파이프라인 수=<maximum> • 계정의 프리셋 수가 이미 최대 허용 개수임: 계정=<accountId>, 최대 프리셋 수=<maximum> • 계정의 작업 수가 이미 백로그의 파이프라인당 최대 개수임: 계정=<accountId>, 파이프라인에 대한 백로그의 최대 작업 수=<maximum>	현재 AWS 계정이 Elastic Transcoder 객체 한도를 초과했습니다. 자세한 내용은 Elastic Transcoder 파이프라인, 작업, 프리셋 개수 제한 섹션을 참조하세요.	

HTTP 상태 코드	오류 코드	메시지	원인	다시 해 보십시오
429	ProvisionedThroughput ExceededException	허용되는 최대 프로비저닝 처리량을 초과하였습니다.	예: 요청량이 너무 높습니다. Elastic Transcoder용 AWS SDK가 예외가 발생한 요청을 자동으로 재시도합니다. 재시도 대기열이 너무 많아 완료하지 못하는 경우를 제외하고 결국 요청이 성공합니다. 요청 횟수를 줄입니다. 자세한 내용은 오류 재시도 횟수 및 지수 백오프 섹션을 참조하세요. 요청 상태를 확인하기 위해 폴링할 경우 알림을 사용하여 상태를 확인해 보세요. 자세한 내용은 작업 상태 알림 섹션을 참조하세요.	예
429	ThrottlingException	요청량이 허용 처리량을 초과하였습니다.	요청을 너무 빠른 속도로 제출하고 있습니다. 예: 새 작업을 생성하는 요청. 요청 상태를 확인하기 위해 폴링할 경우 알림을 사용하여 상태를 확인해 보세요. 자세한 내용은 작업 상태 알림 섹션을 참조하세요.	예
500	InternalFailure	요청을 실행하는 중 서버 내부에서 오류가 발생하였습니다.	요청을 처리하는 도중 서버에서 오류가 발생하였습니다.	예

HTTP 상태 코드	오류 코드	메시지	원인	다시 해 보십시오
500	내부 서버 오류	요청을 실행하는 중 서버 내부에서 오류가 발생하였습니다.	요청을 처리하는 도중 서버에서 오류가 발생하였습니다.	예
500	Internal Service Exception		서비스가 요청을 충족하는 동안 예기치 못한 예외가 발생했습니다.	예
500	Service Unavailable Exception	현재 서비스를 이용할 수 없거나 사용 중입니다.	요청을 처리하는 도중 예상하지 못한 오류가 서버에서 발생하였습니다.	예

샘플 오류 응답

다음은 `inputBucket` 값이 null이었음(유효한 값이 아님)을 나타내는 HTTP 응답입니다.

```
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"}
```

작업 처리 중 오류 발생

Elastic Transcoder에서 작업을 처리하는 도중 오류가 발생할 경우 두 가지 방법으로 오류가 보고됩니다.

- 작업 상태 및 출력 상태: Elastic Transcoder가 실패한 출력에 대해 `Job:Status` 객체와 `Outputs:Status` 객체를 `Error`로 설정합니다. 또한 Elastic Transcoder는 실패한 출력의 `Outputs:StatusDetail` JSON 객체를 실패를 설명하는 값으로 설정합니다.
- SNS 알림: Elastic Transcoder에서 처리 중 오류가 발생할 경우 SNS 알림을 보내도록 파이프라인을 구성한 경우, Elastic Transcoder는 다음 형식으로 알림에 JSON 객체를 포함시킵니다.

```
{
   "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"
   }
}
```

errorCode 값	messageDetails 값	원인
1000	Validation Error	Elastic Transcoder가 작업 처리 중 요청의 값 중 한 개 이상이 유효하지 않음을 확인했습니다.
1001	Dependency Error	재생 목록 종속성 중 하나 이상에 오류가 발생하여 Elastic Transcoder가 재생 목록을 생성할 수 없습니다.
2000	Cannot Assume Role	Elastic Transcoder는이 작업에 대한 파이프라인의 `Role` 객체에 지정된 AWS Identity and Access Management 역할을 수임할 수 없습니다.
3000	Unclassified Storage Error	
3001	Input Does Not Exist	이 작업의 `Input:Key` 객체에 지정된 이름을 가진 파일이 없습니다. 이 작업에 대한 파이프라인의 `InputBucket` 객체에 지정된 Amazon S3 버킷에 파일이 있어야 합니다.
3002	Output Already Exists	이 작업의 `Outputs:Key` (또는 `Output:Key`) 객체에 지정된 이름을 가진 파일이 이미 있습니다. 이 작업에 대한 파이프라인의 `OutputBucket` 객체에 지정된 Amazon S3 버킷에 파일이 있을 수 없습니다.
3003	Does Not Have Read Permission	이 작업에 사용한 파이프라인의 `Role` 객체에 지정된 IAM 역할에, 트랜스코딩하려는 파일을 포함하는 Amazon S3 버킷을 읽을 수 있는 권한이 없습니다.
3004	Does Not Have Write Permission	이 작업에 사용한 파이프라인의 `Role` 객체에 지정된 IAM 역할에, 트랜스코딩한 파일이나 썸네일 파일을 저장하려는 Amazon S3 버킷에 쓸 수 있는 권한이 없습니다.
3005	Bucket Does Not Exist	지정한 S3 버킷이 존재하지 않습니다. 버킷={1}입니다.
3006	Does Not Have Write Permission	키가 버킷과 동일한 리전에 있지 않기 때문에 Elastic Transcoder가 키={1}을(를) 버킷={2}에 쓰지 못했습니다.

errorCode 값	messageDetails 값	원인
4000	Bad Input File	이 작업의 Input:Key 객체에 지정한 파일이 Elastic Transcoder에서 현재 지원하지 않는 형식입니다.
4001	Bad Input File	이 작업의 Input:Key 객체에 지정한 파일의 너비 x 높이가 최대 허용 너비 x 높이를 초과합니다.
4002	Bad Input File	이 작업의 Input:Key 객체에 지정한 파일의 크기가 최대 허용 크기를 초과합니다.
4003	Bad Input File	이 작업의 Outputs:Watermarks:InputKey 객체 중 하나에 지정한 파일을 Elastic Transcoder가 해석할 수 없습니다.
4004	Bad Input File	이 작업의 Outputs:Watermarks:InputKey 객체 중 하나에 지정한 파일의 너비 x 높이가 최대 허용 너비 x 높이를 초과합니다.
4005	Bad Input File	{1} 객체 중 하나에 지정한 파일 크기가 최대 허용 크기: 버킷={2}, 키={3}, 크기{4}, 최대 크기={5}를 초과합니다.
4006	Bad Input File	입력 파일 형식이 지원되지 않기 때문에 Elastic Transcoder가 입력 파일을 트랜스코딩할 수 없습니다.
4007	처리되지 않은 입력 파일	Elastic Transcoder에서 일반적으로 지원하는 파일 형식이지만 제대로 처리하지 못한 파일이 있습니다. 이 오류는 자동으로 지원 요청을 개설하므로 문제 원인의 조사를 시작했습니다.

errorCode 값	messageDetails 값	원인
4008	Bad Input File	이 오류의 근본 원인이 프리셋과 입력 파일 간에 일치하지 않습니다. 그러한 예는 다음과 같습니다. • 프리셋에 오디오 설정이 포함되지만 입력 파일에 오디오가 없습니다. • 프리셋에 비디오 설정이 포함되지만 입력 파일에 비디오가 없습니다.
4009	Bad Input File	아트워크 스트림 수가 최대 수를 초과했기 때문에 Elastic Transcoder가 일부 앨범 아트를 출력 파일에 삽입하지 못했습니다.
4010	Bad Input File	`AlbumArt:Artwork:InputKey` 에 지정한 그래픽 파일을 Elastic Transcoder가 해석할 수 없습니다.
4011	Bad Input File	Elastic Transcoder가 임베디드 아트워크 스트림을 감지했지만 해석할 수 없습니다.
4012	Bad Input File	`AlbumArt:Artwork` 에 지정한 이미지가 최대 허용 너비 x 높이: 4096 x 3072를 초과합니다.
4013	Bad Input File	임베디드 아트워크의 너비 x 높이가 최대 허용 너비 x 높이: 4096 x 3072를 초과합니다.
4014	Bad Input	클립의 시작 시간으로 지정한 값이 입력 파일의 끝보다 나중입니다. Elastic Transcoder가 출력 파일을 생성할 수 없습니다. Elastic Transcoder가 출력 파일을 생성할 수 없습니다.
4015	Bad Input	생성된 세그먼트가 일치하지 않기 때문에 Elastic Transcoder가 매니페스트 파일을 생성할 수 없습니다.
4016	Bad Input	Elastic Transcoder가 {1}의 입력 파일을 {2}을(를) 사용하여 해독할 수 없습니다.

errorCode 값	messageDetails 값	원인
4017	Bad Input	AES 키가 {2}비트 암호화 키를 사용하여 암호화되었습니다. AES는 128비트, 192비트, 256비트 암호화 키만 지원합니다. MD5={1}.
4018	Bad Input	Elastic Transcoder가 MD5={1}을(를) 사용하여 암호화된 키를 해독하지 못했습니다.
4019	Bad Input	Elastic Transcoder가 KMS 키 ARN {0}을(를) 사용하여 데이터 키를 생성하지 못했습니다.
4020	Bad Input	AES-128 암호화를 위한 128비트 키여야 합니다. MD5={1}, {2} 비트.
4021	Bad Input	PlayReady DRM을 위한 128비트 키여야 합니다. MD5={1}, 강도={2} 비트.
4022	Bad Input	{1} 지정 미디어 파일의 통합 크기가 최대 허용 크기: 버킷 ={2}, 크기={3}를 초과합니다.
4023	Bad Input	연결에 지정된 {1} 입력 파일이 지정된 프리셋과 일치하는 해상도로 출력을 생성하지 않습니다. 다른 `PaddingPolicy` , `SizingPolicy` , `MaxWidth`, `MaxHeight` 설정의 프리셋을 사용하세요.
4024	Bad Input	연결에 지정된 {1} 입력 파일이 지정된 프리셋과 일치하는 해상도로 썸네일을 생성하지 않습니다. 다른 `PaddingPolicy` , `SizingPolicy` , `MaxWidth`, `MaxHeight` 설정의 썸네일을 사용하세요.
4025	Bad Input	한 개 이상의 미디어 파일(입력 #{1})이 다른 파일과 일치하지 않습니다. 모든 미디어 파일이 비디오이거나 비디오가 아니어야 합니다.
4026	Bad Input	한 개 이상의 미디어 파일(입력 #{1})이 다른 파일과 일치하지 않습니다. 모든 미디어 파일이 오디오이거나 오디오가 아니어야 합니다.

errorCode 값	messageDetails 값	원인
4100	Bad Input File	Elastic Transcoder가 임베디드 캡션 트랙을 감지했지만 해석할 수 없습니다.
4101	Bad Input File	Elastic Transcoder가 Amazon S3 버킷={1}, 키={2}에 지정된 캡션 파일을 해석할 수 없습니다.
4102	Bad Input File	지정된 캡션 파일이 UTF-8 인코딩 형식이 아니기 때문에 Elastic Transcoder가 해석할 수 없음: Amazon S3 버킷={1}, 키={2}.
4103	Bad Input File	최대 허용 캡션 트랙 수인 {1}을(를) 초과했기 때문에 Elastic Transcoder가 캡션 트랙을 일부 처리하지 못했습니다.
4104	Bad Input File	임베디드 캡션의 최대 개수는 4개인데 해당 출력에 {1}개의 임베디드 캡션이 있기 때문에 Elastic Transcoder가 마스터 재생 목록을 생성할 수 없습니다.
4105	Bad Input File	CEA-708에는 {1}의 프레임 속도가 지원되지 않기 때문에 Elastic Transcoder가 캡션 트랙을 임베딩할 수 없습니다. [29.97, 30]의 프레임 속도만 지원됩니다.
4106	Bad Input File	{1} 형식은 {2}개의 캡션 트랙만 지원하기 때문에 Elastic Transcoder가 캡션 트랙을 임베딩할 수 없습니다.
9000	내부 서버 오류	
9001	내부 서버 오류	
9999	내부 서버 오류	

오류 발견

애플리케이션의 원활한 실행을 위해서는 오류를 발견하여 대응할 수 있는 로직을 내장해야 합니다. 일반적으로 `try` 블록 또는 `if-then` 문 내에서 요청을 실행하는 방법이 많이 사용됩니다.

AWS SDK는 자체적으로 재시도 및 오류 검사를 실행합니다. AWS SDK 사용 중에 오류가 발생하면 해당 오류 코드와 설명을 확인해야 합니다. 또한 Request ID 값도 확인해야 합니다. Request ID 값은 Elastic Transcoder 지원 문제를 해결하는 데 도움이 됩니다.

다음은 Java용 AWS SDK를 사용하여 try 블록 내 항목을 삭제하려다가 catch 블록을 사용하여 오류에 대응하는 예제입니다. 이 예제에서는 요청 실패를 경고합니다. 이 예제에서는 AmazonServiceException 클래스를 사용해 Request ID를 포함한 모든 작업 오류에 대한 정보를 가져옵니다. 또한 AmazonClientException 클래스를 사용해 다른 원인으로 인해 요청이 실패할 경우를 대비합니다.

```
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());
   }
```

오류 재시도 횟수 및 지수 백오프

DNS 서버, 스위치, 로드 밸런서 등 수많은 네트워크 구성 요소들은 요청이 이루어지는 모든 단계에서 오류를 일으킬 수 있습니다.

네트워크 환경에서는 클라이언트 애플리케이션의 재시도 기술이 이러한 오류 응답을 처리하는 데 가장 많이 사용되고 있습니다. 이 기술은 애플리케이션의 안정성을 높일 뿐만 아니라 개발자의 운영 비용을 절감하는 효과가 있습니다.

Elastic Transcoder을 지원하는 AWS SDK는 모두 자동 재시도 로직을 구현합니다. Java용 AWS SDK 역시 자동으로 요청을 재시도하며, 이러한 재시도 설정은 `ClientConfiguration` 클래스를 사용해 구성할 수 있습니다. 예를 들어 최소의 지연 시간으로 재시도 없이 요청하는 웹 페이지의 경우에는 재시도 로직을 비활성화하면 됩니다. 이때는 `ClientConfiguration` 클래스를 사용하여 `maxErrorRetry` 값을 `0`으로 입력하면 재시도가 비활성화됩니다.

AWS SDK를 사용하지 않을 때는 서버 오류(5xx)가 수신된 최초 요청을 재시도해야 합니다. 하지만 클라이언트 오류(4xx, `ThrottlingException` 또는 `ProvisionedThroughputExceededException` 제외)는 요청 자체를 수정하여 문제를 해결해야만 재시도가 가능합니다.

> **Note**
>
> 요청 상태를 확인하기 위해 폴링하는 경우 Elastic Transcoder가 HTTP 상태 코드 429와 오류 코드 `ProvisionedThroughputExceededException` 또는 `ThrottlingException`을 반환하면 폴링 대신 알림을 사용하여 상태를 확인해 보세요. 자세한 내용은 작업 상태 알림 섹션을 참조하세요.

간단한 재시도 방법 외에도 흐름 제어가 탁월한 지수 백오프 알고리즘을 권장합니다. 지수 백오프의 기본 개념은 오류 응답이 연이어 나올 때마다 재시도 간 대기 시간을 점진적으로 늘리는 것입니다. 예를 들어 1차 재시도 시 1초, 2차 재시도 시 4초, 3차 재시도 시 16초 등으로 지연 시간을 설정할 수 있습니다. 하지만 요청이 1분 동안 성공하지 못할 경우 문제는 요청 속도가 아니라 하드 제한일 수 있습니다. 예를 들어 허용되는 최대 파이프라인 수에 도달했을 수 있습니다. 따라서 약 1분 정도에서 멈추도록 최대 재시도 횟수를 설정하세요.

다음은 재시도 로직을 나타낸 워크플로입니다. 워크플로 로직이 먼저 서버 오류(5xx) 여부를 결정합니다. 서버 오류인 경우에는 코드가 최초 요청을 재시도합니다.

```
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
```

파이프라인 작업

주제

- 파이프라인 생성
- 파이프라인 목록 조회
- 파이프라인 읽기
- 파이프라인 업데이트
- 파이프라인 상태 업데이트
- 파이프라인 알림 업데이트
- 파이프라인 삭제
- 테스트 역할

파이프라인은 트랜스코딩 작업을 관리하는 대기열입니다. 작업을 생성할 때 작업을 추가할 파이프라인을 지정합니다. Elastic Transcoder는 작업을 추가한 순서대로 파이프라인에서 작업을 처리하기 시작합니다.

이 섹션에서는 Elastic Transcoder API를 사용해 파이프라인에 수행할 수 있는 작업을 설명합니다. Elastic Transcoder 콘솔을 사용해 같은 작업을 수행하는 방법 등 파이프라인에 대한 자세한 내용은 파이프라인 작업 섹션을 참조하세요.

파이프라인 생성

주제

- 설명
- 요청
- 응답
- 오류

- 예시

설명

파이프라인을 생성하려면 /2012-09-25/pipelines/ 리소스에 POST 요청을 보냅니다.

요청

구문

```
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"
   }
}
```

요청 파라미터

이 작업은 요청 파라미터를 사용하지 않습니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

요청 본문의 JSON 문자열에는 CreatePipeline 작업에 대한 입력 객체가 포함되어 있습니다. 입력 객체에 대한 자세한 내용은 Elastic Transcoder 파이프라인을 생성할 때 지정하는 설정 섹션을 참조하세요.

응답

구문

```
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"
   },
   {...}
]
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

파이프라인을 만들면 Elastic Transcoder는 사용자가 요청에 지정한 값을 반환합니다. 자세한 내용은 요청 본문 단원을 참조하십시오.

또한 Elastic Transcoder는 다음 값을 반환합니다.

(자동) Id

파이프라인의 식별자. 이 값을 사용하여 작업 또는 프리셋 생성과 같은 다양한 작업을 수행하려는 파이프라인을 식별합니다.

(자동) Status

파이프라인의 현재 상태입니다.

- `Active`: 파이프라인이 작업을 처리 중입니다.
- `Paused`: 파이프라인이 현재 작업을 처리하지 않습니다.

(자동) Warnings

다른 리전의 리소스를 사용하는 파이프라인을 생성하면 Elastic Transcoder가 하나 이상의 경고를 반환합니다. 그래도 파이프라인은 생성되지만 처리 시간이 증가하고 교차 리전 비용이 발생할 수 있습니다. 경고는 다음 형식으로 제공됩니다.

코드 - 경고 코드입니다.

메시지 - 경고 코드와 관련된 메시지입니다.

반환될 수 있는 경고 코드와 메시지는 다음과 같습니다.

6000

입력 버킷과 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 입력 버킷과 파이프라인에 동일한 리전을 사용하세요.

6001

ContentConfig 버킷과 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 ContentConfig 버킷과 파이프라인에 동일한 리전을 사용하세요.

6002

ThumbnailConfig 버킷과 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 ThumbnailConfig 버킷과 파이프라인에 동일한 리전을 사용하세요.

6003

진행 이벤트에 대한 SNS 알림 주제와 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 SNS 알림 주제와 파이프라인에 동일한 리전을 사용하세요.

6004

경고 이벤트에 대한 SNS 알림 주제와 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 SNS 알림 주제와 파이프라인에 동일한 리전을 사용하세요.

6005

완료 이벤트에 대한 SNS 알림 주제와 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 SNS 알림 주제와 파이프라인에 동일한 리전을 사용하세요.

6006

오류 이벤트에 대한 SNS 알림 주제와 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 SNS 알림 주제와 파이프라인에 동일한 리전을 사용하세요.

6007

이 파이프라인에 지정된 AWS KMS 키와 ContentConfig 버킷이 서로 다른 리전에 위치하여 s3-aws-kms 암호화 모드를 사용한 출력이 실패합니다. s3-aws-kms 암호화 모드를 사용하려면 KMS 키와 ContentConfig 버킷에 동일한 리전을 사용하세요.

6008

이 파이프라인에 지정된 AWS KMS 키와 ThumbnailConfig 버킷이 서로 다른 리전에 위치하여 s3-aws-kms 암호화 모드를 사용한 출력이 실패합니다. s3-aws-kms 암호화 모드를 사용하려면 KMS 키와 ThumbnailConfig 버킷에 동일한 리전을 사용하세요.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예시

다음 예제 요청은 Default라는 파이프라인을 생성합니다.

샘플 요청

```
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"
   }
}
```

샘플 응답

```
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."
      },
      {...}
   ]
}
```

파이프라인 목록 조회

주제

- 설명
- 요청
- 응답
- 오류
- 예시

설명

현재 AWS 계정에 연결된 파이프라인의 목록을 확인하려면 /2012-09-25/pipelines/ 리소스에 GET 요청을 보내세요.

요청

구문

현재 AWS 계정에 연결된 모든 파이프라인에 대한 정보를 확인하려면 다음 GET 요청을 보내세요.

```
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
```

요청 파라미터

이 작업은 다음 요청 파라미터를 사용합니다. Elastic Transcoder는 모든 파이프라인을 반환합니다.

Ascending

파이프라인을 제출 날짜 및 시간순으로 나열하려면 `true`를 입력하세요. 파이프라인을 역순으로 나열하려면 `false`를 입력하세요.

PageToken

Elastic Transcoder가 반환하는 결과 페이지가 여러 개라면 후속 GET 요청에서 PageToken을 사용해 이어지는 각 결과 페이지를 확인할 수 있습니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

이 작업에는 요청 본문이 없습니다.

응답

구문

```
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
```

```
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

응답 본문의 JSON 문자열은 파이프라인을 생성할 때 지정한 값을 반환합니다. 각 객체에 대한 자세한 내용은 Elastic Transcoder 파이프라인을 생성할 때 지정하는 설정 섹션을 참조하세요.

또한 Elastic Transcoder는 다음 값을 반환합니다.

(자동) Id

파이프라인의 식별자. 이 값을 사용하여 작업 또는 프리셋 생성과 같은 다양한 작업을 수행하려는 파이프라인을 식별합니다.

(자동) Status

파이프라인의 현재 상태입니다.

- `Active`: 파이프라인이 작업을 처리 중입니다.
- `Paused`: 파이프라인이 현재 작업을 처리하지 않습니다.

(자동) NextPageToken

결과의 두 번째 및 이후 페이지(있는 경우)에 액세스하는 데 사용하는 값. 파이프라인이 한 페이지에 표시될 수 있거나 결과의 마지막 페이지에 도달한 경우 `NextPageToken` 값은 `null`입니다.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예시

다음 예제 요청은 현재 AWS 계정에 연결된 파이프라인의 목록을 조회합니다.

샘플 요청

```
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
```

샘플 응답

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT
{
   "Pipelines":[
      {
         "Id":"1111111111111-abcde1",
         "Name":"Tokyo-Default",
         "InputBucket":"salesoffice-tokyo.example.com-source",
         "OutputBucket":"salesoffice-tokyo.example.com-output",
         "Role":"arn:aws:iam::123456789012:role/Elastic_Transcoder_Default
            _Role",
         "AwsKmsKeyArn":"base64 encoded key from KMS",
         "Notifications":{
            "Progressing":"",
            "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":"2222222222222-abcde2",
         "Name":"Amsterdam-Default",
         "InputBucket":"salesoffice-amsterdam.example.com-source",
         "OutputBucket":"salesoffice-amsterdam.example.com-output",
         "Role":"arn:aws:iam::123456789012:role/Elastic_Transcoder_Default
            _Role",
         "AwsKmsKeyArn":"base64 encoded key from KMS",
         "Notifications":{
            "Progressing":"",
            "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"
      }
   ]
}
```

파이프라인 읽기

주제

- 설명
- 요청
- 응답
- 오류
- 예시

설명

파이프라인에 대한 세부 정보를 확인하려면 /2012-09-25/pipelines/*pipelineId* 리소스에 GET 요청을 보냅니다.

요청

구문

```
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
```

요청 파라미터

이 작업은 다음 요청 파라미터를 사용합니다.

pipelineId

세부 정보를 조회하려는 파이프라인의 식별자입니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

이 작업에는 요청 본문이 없습니다.

응답

구문

```
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"
    },
    {...}
  ]
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

응답 본문의 JSON 문자열은 파이프라인을 생성할 때 지정한 값을 반환합니다. 각 객체에 대한 자세한 내용은 Elastic Transcoder 파이프라인을 생성할 때 지정하는 설정 섹션을 참조하세요.

Elastic Transcoder는 해당되는 경우 다음 값과 경고를 포함합니다.

(자동) Id

파이프라인의 식별자. 이 값을 사용하여 작업 또는 프리셋 생성과 같은 다양한 작업을 수행하려는 파이프라인을 식별합니다.

(자동) Status

파이프라인의 현재 상태입니다.

- `Active`: 파이프라인이 작업을 처리 중입니다.
- `Paused`: 파이프라인이 현재 작업을 처리하지 않습니다.

(자동) Warnings

다른 리전의 리소스를 사용하는 파이프라인을 생성하면 Elastic Transcoder가 하나 이상의 경고를 반환합니다. 그래도 파이프라인은 생성되지만 처리 시간이 증가하고 교차 리전 비용이 발생할 수 있습니다. 경고는 다음 형식으로 제공됩니다.

코드

메시지 - 경고 코드와 관련된 메시지입니다.

다음은 유효한 경고 코드와 그 메시지 목록입니다.

6000

입력 버킷과 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 입력 버킷과 파이프라인에 동일한 리전을 사용하세요.

6001

ContentConfig 버킷과 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 ContentConfig 버킷과 파이프라인에 동일한 리전을 사용하세요.

6002

ThumbnailConfig 버킷과 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 ThumbnailConfig 버킷과 파이프라인에 동일한 리전을 사용하세요.

6003

진행 이벤트에 대한 SNS 알림 주제와 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 SNS 알림 주제와 파이프라인에 동일한 리전을 사용하세요.

6004

경고 이벤트에 대한 SNS 알림 주제와 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 SNS 알림 주제와 파이프라인에 동일한 리전을 사용하세요.

6005

완료 이벤트에 대한 SNS 알림 주제와 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 SNS 알림 주제와 파이프라인에 동일한 리전을 사용하세요.

6006

오류 이벤트에 대한 SNS 알림 주제와 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 SNS 알림 주제와 파이프라인에 동일한 리전을 사용하세요.

6007

이 파이프라인에 지정된 AWS KMS 키와 ContentConfig 버킷이 서로 다른 리전에 위치하여 s3-aws-kms 암호화 모드를 사용한 출력이 실패합니다. s3-aws-kms 암호화 모드를 사용하려면 KMS 키와 ContentConfig 버킷에 동일한 리전을 사용하세요.

6008

이 파이프라인에 지정된 AWS KMS 키와 ThumbnailConfig 버킷이 서로 다른 리전에 위치하여 s3-aws-kms 암호화 모드를 사용한 출력이 실패합니다. s3-aws-kms 암호화 모드를 사용하려면 KMS 키와 ThumbnailConfig 버킷에 동일한 리전을 사용하세요.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예시

다음 예제 요청은 ID가 1111111111111-abcde1인 파이프라인을 가져옵니다.

샘플 요청

```
GET /2012-09-25/pipelines/1111111111111-abcde1 HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder endpoint/
elastictranscoder/aws4_request,
```

```
                SignedHeaders=host;x-amz-date;x-amz-target,
                Signature=calculated-signature
```

샘플 응답

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Pipeline":{
      "Id":"1111111111111-abcde1",
      "Name":"Default",
      "InputBucket":"salesoffice.example.com-source",
      "OutputBucket":"salesoffice.example.com-output",
     "Role":"arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
      "AwsKmsKeyArn":"base64 encoded key from KMS",
      "Notifications":{
         "Progressing":"",
         "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."
       },
       {...}
    ]
}
```

파이프라인 업데이트

주제

- 설명
- 값을 업데이트하고 삭제하는 방법
- 요청
- 응답
- 오류
- 예시

설명

파이프라인의 설정을 업데이트하려면 /2012-09-25/pipelines/*pipelineId* 리소스에 PUT 요청을 보냅니다.

> **Important**
>
> 파이프라인 설정을 변경하면 변경 사항이 즉시 적용됩니다. 설정을 변경한 후 제출한 작업 외에도 이미 제출했으나 Elastic Transcoder가 처리를 시작하지 않은 작업도 영향을 받습니다.

값을 업데이트하고 삭제하는 방법

값을 업데이트하려면 예를 들어 다음과 같이 새 값을 지정합니다.

`"GranteeType":"Canonical"`

값을 변경하지 않고 그대로 두려면 다음 중 하나를 수행합니다.

- 현재 값을 지정합니다.
- null 값을 지정합니다. 이는 개별 객체(예: `"Role":null`) 및 어레이(예: `"ContentConfig":null`)에 모두 적용됩니다.
- 요청에서 JSON 객체를 뺍니다. 이는 개별 객체(예: `Role`) 및 어레이(예: `ContentConfig`)에 모두 적용됩니다.

현재 값을 삭제하려면 빈 값을 지정합니다. 예를 들어 알림을 삭제하려면 `"Progressing":""`과 같이 빈 스트링을 지정합니다. 어레이의 값을 모두 삭제하려면 `"Permissions":[]`과 같이 빈 어레이를 지정합니다.

예를 들어 작업을 처리하는 중에 Elastic Transcoder에 경고가 발생했을 때 더 이상 알림을 받고 싶지 않으며 다른 알림 설정을 변경하지 않으려는 경우, 요청의 `Notifications`에 다음을 지정합니다.

```
...
"Notifications":{
   "Warning":""
},
...
```

요청

구문

```
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"
}
```

요청 파라미터

이 작업은 요청 파라미터를 사용하지 않습니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

요청 본문의 JSON 문자열에는 UpdatePipeline 작업에 대한 입력 객체가 포함되어 있습니다. 입력 객체에 대한 자세한 내용은 Elastic Transcoder 파이프라인을 생성할 때 지정하는 설정 섹션을 참조하세요.

Id

파이프라인의 식별자. 이 값을 사용하여 작업 또는 프리셋 생성과 같은 다양한 작업을 수행하려는 파이프라인을 식별합니다.

상태

파이프라인의 현재 상태입니다.

- Active: 파이프라인이 작업을 처리 중입니다.
- Paused: 파이프라인이 현재 작업을 처리하지 않습니다.

응답

구문

```
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"
      },
      {...}
   ]
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

파이프라인을 만들면 Elastic Transcoder는 사용자가 요청에 지정한 값을 반환합니다. 자세한 내용은 요청 본문 단원을 참조하십시오.

Elastic Transcoder는 해당되는 경우 다음 경고를 포함합니다.

(자동) Warnings

다른 리전의 리소스를 사용하는 파이프라인을 생성하면 Elastic Transcoder가 하나 이상의 경고를 반환합니다. 그래도 파이프라인은 생성되지만 처리 시간이 증가하고 교차 리전 비용이 발생할 수 있습니다. 경고는 다음 형식으로 제공됩니다.

코드

메시지 - 경고 코드와 관련된 메시지입니다.

다음은 유효한 경고 코드와 그 메시지 목록입니다.

6000

입력 버킷과 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 입력 버킷과 파이프라인에 동일한 리전을 사용하세요.

6001

ContentConfig 버킷과 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 ContentConfig 버킷과 파이프라인에 동일한 리전을 사용하세요.

6002

ThumbnailConfig 버킷과 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 ThumbnailConfig 버킷과 파이프라인에 동일한 리전을 사용하세요.

6003

진행 이벤트에 대한 SNS 알림 주제와 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 SNS 알림 주제와 파이프라인에 동일한 리전을 사용하세요.

6004

경고 이벤트에 대한 SNS 알림 주제와 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 SNS 알림 주제와 파이프라인에 동일한 리전을 사용하세요.

6005

완료 이벤트에 대한 SNS 알림 주제와 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 SNS 알림 주제와 파이프라인에 동일한 리전을 사용하세요.

6006

오류 이벤트에 대한 SNS 알림 주제와 파이프라인이 서로 다른 리전에 위치하여 파이프라인 내 작업 처리 시간이 증가하며 추가 요금이 부과될 수 있습니다. 처리 시간을 단축하고 교차 리전 요금을 방지하려면 SNS 알림 주제와 파이프라인에 동일한 리전을 사용하세요.

6007

이 파이프라인에 지정된 AWS KMS 키와 ContentConfig 버킷이 서로 다른 리전에 위치하여 s3-aws-kms 암호화 모드를 사용한 출력이 실패합니다. s3-aws-kms 암호화 모드를 사용하려면 KMS 키와 ContentConfig 버킷에 동일한 리전을 사용하세요.

6008

이 파이프라인에 지정된 AWS KMS 키와 ThumbnailConfig 버킷이 서로 다른 리전에 위치하여 s3-aws-kms 암호화 모드를 사용한 출력이 실패합니다. s3-aws-kms 암호화 모드를 사용하려면 KMS 키와 ThumbnailConfig 버킷에 동일한 리전을 사용하세요.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예시

다음 예제 요청은 ID가 `1111111111111-abcde1`인 파이프라인을 업데이트합니다.

샘플 요청

```
PUT /2012-09-25/pipelines/1111111111111-abcde1 HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
```

```
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder 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":"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"
}
```

샘플 응답

```
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."
      },
      {...}
   ]
}
```

파이프라인 상태 업데이트

주제

- 설명
- 요청
- 응답
- 오류
- 예제

설명

파이프라인을 일시 중지하거나 다시 활성화하기 위해 파이프라인은 작업 처리를 중지하거나 다시 시작하고 파이프라인의 상태를 업데이트합니다. `/2012-09-25/pipelines/`*`pipelineId`*`/status` 리소스에 POST 요청을 보냅니다.

작업을 하나 이상 취소하려는 경우 파이프라인 상태를 변경하는 것이 유용합니다. Elastic Transcoder 가 작업을 처리하기 시작한 후에는 작업을 취소할 수 없습니다. 작업을 제출한 파이프라인을 일시 중지

하는 경우, 취소하려는 작업의 작업 ID를 확인한 후 Delete Job 요청을 보내야 하므로 시간이 오래 걸립니다.

요청

구문

```
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"
}
```

요청 파라미터

이 작업은 다음 요청 파라미터를 사용합니다.

pipelineId

일시 중지하거나 다시 활성화하려는 파이프라인의 식별자입니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

요청 본문의 JSON 문자열에는 다음 객체가 포함되어 있습니다.

상태

파이프라인의 새 상태입니다.

- Active: 파이프라인이 작업 처리를 시작할 수 있도록 활성화합니다.
- Paused: 파이프라인이 작업 처리를 중지할 수 있도록 비활성화합니다.

응답

구문

```
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"
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

파이프라인의 상태를 업데이트하면 Elastic Transcoder는 사용자가 요청에 지정한 값을 반환합니다. 자세한 내용은 요청 본문 단원을 참조하십시오.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예제

다음 예제 요청은 ID가 1111111111111-abcde1인 파이프라인을 활성화합니다.

샘플 요청

```
POST /2012-09-25/pipelines/1111111111111-abcde1/status HTTP/1.1
```

```
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder 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"
}
```

샘플 응답

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT
{
   "Id":"1111111111111-abcde1",
   "Status":"Active"
}
```

파이프라인 알림 업데이트

주제

- 설명
- 요청
- 응답
- 오류
- 예시

설명

파이프라인의 Amazon Simple Notification Service(SNS) 알림만 업데이트하려면 `/2012-09-25/pipelines/pipelineId/notifications` 리소스에 POST 요청을 보냅니다.

> **Important**
>
> 알림을 변경하면 변경 사항이 즉시 적용됩니다. 알림을 변경한 후 제출한 작업 외에도 이미 제출했으나 Elastic Transcoder가 처리를 시작하지 않은 작업도 영향을 받습니다.

요청

구문

```
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"
   }
}
```

요청 파라미터

이 작업은 다음 요청 파라미터를 사용합니다.

pipelineId

알림 설정을 변경하려는 파이프라인의 식별자입니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

요청 본문의 JSON 문자열에는 다음 객체가 포함되어 있습니다.

id

업데이트할 파이프라인의 ID입니다.

알림: 진행 중

Elastic Transcoder가 이 파이프라인에서 작업을 처리하기 시작하면 알릴 Amazon Simple Notification Service(SNS) 주제의 주제 ARN입니다. 이 값은 주제를 생성할 때 Amazon SNS가 반환한 ARN입니다. 자세한 내용은 Amazon Simple Notification Service 개발자 가이드의 주제 생성을 참조하세요.

> **Important**
>
> 알림을 수신하려면 Amazon SNS 콘솔에서 새 주제를 구독해야 합니다.

Amazon SNS는 Amazon Simple Queue Service 대기열에 Amazon SNS 메시지를 전송하는 기능을 포함하여 다양한 알림 옵션을 제공합니다. 자세한 내용은 Amazon Simple Notification Service Developer Guide를 참조하십시오.

Notifications:Complete

Elastic Transcoder가 이 파이프라인에서 작업 처리를 완료하면 알릴 Amazon SNS 주제의 주제 ARN입니다. 이 값은 주제를 생성할 때 Amazon SNS가 반환한 ARN입니다.

Notifications:Warning

Elastic Transcoder가 이 파이프라인에서 작업을 처리하는 동안 경고 상태가 발생하면 알릴 Amazon SNS 주제의 주제 ARN입니다. 이 값은 주제를 생성할 때 Amazon SNS가 반환한 ARN입니다.

Notifications:Error

Elastic Transcoder가 이 파이프라인에서 작업을 처리하는 동안 오류 상태가 발생하면 알릴 Amazon SNS 주제의 주제 ARN입니다. 이 값은 주제를 생성할 때 Amazon SNS가 반환한 ARN입니다.

응답

구문

```
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"
      }
   }
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

파이프라인의 알림을 업데이트하면 Elastic Transcoder는 사용자가 요청에 지정한 값을 반환합니다. 자세한 내용은 요청 본문 단원을 참조하십시오.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예시

다음 예제 요청은 파이프라인의 알림을 업데이트합니다.

샘플 요청

```
POST /2012-09-25/pipelines/1111111111111-abcde1/notifications HTTP/1.1
Content-Type: application/json; charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder 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":"1111111111111-abcde1",
   "Notifications":{
      "Progressing":"",
      "Complete":"",
      "Warning":"",
      "Error":"arn:aws:sns:us-east-1:111222333444:ETS_Errors"
   }
}
```

샘플 응답

```
Status: 202 Accepted
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Id":"1111111111111-abcde1",
```

```
   "Notifications":{
      "Progressing":"",
      "Complete":"",
      "Warning":"",
      "Error":"arn:aws:sns:us-east-1:111222333444:ETS_Errors"
   }
}
```

파이프라인 삭제

주제

- 설명
- 요청
- 응답
- 오류
- 예제

설명

파이프라인을 삭제하려면 /2012-09-25/pipelines/*pipelineId* 리소스에 DELETE 요청을 보냅니다. 사용한 적이 없거나 현재 사용 중이지 않은 파이프라인(활성 작업이 포함되어 있지 않음)만 삭제할 수 있습니다. 파이프라인을 현재 사용 중인 경우 Delete Pipeline이 오류를 반환합니다.

요청

구문

```
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
```

요청 파라미터

이 작업은 다음 요청 파라미터를 사용합니다.

pipelineId

삭제하려는 파이프라인의 식별자입니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

이 작업에는 요청 본문이 없습니다.

응답

구문

```
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"
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

응답 본문에는 다음 JSON 객체가 포함됩니다.

Success

파이프라인이 삭제된 경우 `Success`의 값은 `true`입니다.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예제

다음 예제 요청은 파이프라인 1111111111111-abcde1을 삭제합니다.

샘플 요청

```
DELETE /2012-09-25/pipelines/1111111111111-abcde1 HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder endpoint/
elastictranscoder/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

샘플 응답

```
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"
}
```

테스트 역할

주제

- 설명
- 요청
- 응답
- 오류

- 예

설명

Elastic Transcoder가 작업을 만들고 처리할 수 있는지 확인하기 위해 파이프라인의 설정을 테스트하려면 /2012-09-25/roleTests 리소스에 POST 요청을 보냅니다.

요청

구문

```
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"
   ]
}
```

요청 파라미터

이 작업은 요청 파라미터를 사용하지 않습니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

요청 본문의 JSON 문자열에는 다음 객체가 포함되어 있습니다.

InputBucket

트랜스코딩할 미디어 파일을 저장한 Amazon S3 버킷입니다. `Test Role`은 이 버킷에서 읽기를 시도합니다.

OutputBucket

Elastic Transcoder가 트랜스코딩된 파일을 저장할 Amazon S3 버킷입니다. `Test Role`은 이 버킷에서 읽기를 시도합니다.

역할

Elastic Transcoder가 작업을 트랜스코딩하는 데 사용할 역할의 IAM Amazon 리소스 이름(ARN)입니다. `Test Role`은 지정된 역할을 맡으려고 시도합니다.

주제

`Test Role`에서 테스트 알림을 보내려는 하나 이상의 Amazon Simple Notification Service(SNS) 주제에 대한 ARN입니다. Amazon SNS 알림을 사용하지 않는 경우 빈 목록을 지정할 수 있습니다.

응답

구문

```
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"
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

파이프라인의 설정을 테스트하면 Elastic Transcoder가 다음 값을 반환합니다.

메시지

Success의 값이 false인 경우, Messages에 실패한 테스트를 설명하는 하나 이상의 메시지 배열이 포함됩니다.

Success

작업에 성공하면 이 값은 true입니다. 그렇지 않은 경우 값은 false입니다.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예

예 요청

```
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"]
}
```

샘플 응답

```
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"
}
```

작업

주제

- 작업 생성
- 파이프라인별 작업 나열
- 상태별 작업 나열
- 작업 읽기
- 작업 취소

작업은 트랜스코딩을 수행합니다. 각 작업은 하나의 파일을 한 형식으로 변환합니다. 예를 들어 미디어 파일을 6가지 다른 형식으로 변환하려면 작업을 6개 만듭니다.

이 섹션에서는 Elastic Transcoder API를 사용하여 작업에 대해 수행할 수 있는 작업을 설명합니다. Elastic Transcoder 콘솔을 사용해 같은 작업을 수행하는 방법 등 작업에 대한 자세한 내용은 작업 사용 섹션을 참조하세요.

작업 생성

주제

- 설명
- 요청

- 응답
- 오류
- 예시

설명

작업을 만들려면 /2012-09-25/jobs 리소스에 POST 요청을 보냅니다. 작업은 만드는 즉시 시작됩니다.

> **Note**
>
> 작업의 상태가 변경되거나(Elastic Transcoder가 작업 처리를 시작 및 종료하는 경우 등) Elastic Transcoder에 경고 또는 오류 조건이 발생하면 이를 알리도록 Elastic Transcoder를 구성할 수 있습니다. 자세한 내용은 파이프라인 생성 섹션을 참조하세요.

요청

구문

```
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 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",
      "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|mxf|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"
}
```

요청 파라미터

이 작업은 요청 파라미터를 사용하지 않습니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

요청 본문의 JSON 문자열에는 CreateJob 작업에 대한 입력 객체가 포함되어 있습니다. 입력 객체에 대한 자세한 내용은 Elastic Transcoder 작업을 만들 때 지정하는 설정 섹션을 참조하세요.

응답

구문

```
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|mxf|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"
   }
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

작업을 만들면 Elastic Transcoder는 사용자가 요청에 지정한 값을 반환합니다. 자세한 내용은 Elastic Transcoder 작업을 만들 때 지정하는 설정 섹션을 참조하세요.

또한 Elastic Transcoder는 다음 값을 반환합니다.

(자동) Id

Elastic Transcoder가 작업에 할당한 식별자입니다. 이 값을 사용하여 작업의 설정을 가져오거나 작업을 삭제합니다.

(자동) Outputs:Id

현재 작업의 출력 중에서 출력을 식별하는 1부터 시작하는 순차 카운터. Output 구문에서 이 값은 항상 1입니다.

(자동) Outputs:Duration

출력 파일의 지속 시간(초)으로, 반올림됩니다.

(자동) Outputs:Width

출력 파일의 너비(픽셀)입니다.

(자동) Outputs:Height

출력 파일의 높이(픽셀)입니다.

(자동) Outputs:Status

작업 내 단일 출력의 상태입니다. 작업에 지정한 출력이 하나뿐일 경우 Outputs:Status는 항상 Job:Status와 동일합니다. 여러 개의 출력을 지정한 경우:

- Elastic Transcoder가 첫 번째 출력을 처리하기 시작할 때까지 모든 출력의 Job:Status 및 Outputs:Status는 Submitted입니다.
- Elastic Transcoder가 첫 번째 출력을 처리하기 시작하면 해당 출력의 Outputs:Status와 Job:Status가 모두 Progressing으로 바뀝니다. 각 출력에서 Outputs:Status 값은 Elastic Transcoder가 해당 출력을 처리하기 시작할 때까지 Submitted를 유지합니다.
- 모든 출력이 최종 상태(Job:Status 또는 Progressing)에 도달할 때까지 Complete는 Error을 유지합니다.
- 모든 출력이 최종 상태에 도달하면 모든 출력의 Outputs:Status가 Complete인 경우에만 Job:Status가 Complete로 바뀝니다. 하나 이상의 출력에서 Outputs:Status가 Error일 경우Job:Status의 최종 상태도 Error입니다.

Status 값은 다음 중 하나입니다. Submitted, Progressing, Complete, Canceled 또는 Error.

(자동) Outputs:StatusDetail

Outputs:Status를 더 자세히 설명하는 정보입니다.

(자동) Outputs:AppliedColorSpaceConversion

Elastic Transcoder가 ColorSpaceConversionMode를 포함한 프리셋을 사용하여 출력 파일을 트랜스코딩한 경우 AppliedColorSpaceConversion 파라미터가 사용된 변환을 표시합니다. 프리셋에 ColorSpaceConversionMode가 정의되지 않은 경우 이 파라미터는 작업 응답에 포함되지 않습니다.

(자동) Status

작업에 여러 개의 출력을 지정한 경우 전체 작업의 상태. Elastic Transcoder가 작업을 처리하기 시작하면 Job:Status가 Progressing으로 바뀐 후 Elastic Transcoder가 모든 출력을 처리할 때까지 바뀌지 않습니다. 처리가 완료되면 Job:Status가 Complete로 바뀌거나, 실패한 출력이 있을 경우 Error로 바뀝니다.

작업에 지정한 출력이 하나뿐일 경우 Job:Status는 Outputs:Status와 동일합니다.

Job:Status 값은 다음 중 하나입니다. Submitted, Progressing, Complete, Canceled 또는 Error.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

> **Note**
>
> 작업이 Access Denied 오류로 인해 실패한 경우, Test Role API 작업을 실행하여 오류의 원인을 확인하는 것이 좋습니다. 자세한 내용은 테스트 역할 섹션을 참조하세요.

예시

다음 예제 요청은 출력이 두 개인 작업을 생성합니다.

샘플 요청

```
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"
```

```
}
```

샘플 응답

```
Status: 201 Created
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Job":{
      "Id":"3333333333333-abcde3",
      "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":"1111111111111-abcde1",
      "Status":"Progressing"
   }
}
```


파이프라인별 작업 나열

주제

- 설명
- 요청
- 응답
- 오류
- 예시

설명

파이프라인에 할당된 작업 목록을 가져오려면 /2012-09-25/jobsByPipeline/*pipelineId* 리소스에 GET 요청을 보냅니다. 파이프라인별로 작업을 나열하면, Elastic Transcoder는 사용자가 지정된 파이프라인에 대해 최근 생성한 작업을 나열합니다.

요청

구문

파이프라인에 현재 있는 작업에 대한 정보를 확인하려면 다음 GET 요청을 보냅니다.

```
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
```

요청 파라미터

이 작업은 다음 요청 파라미터를 사용합니다. Elastic Transcoder는 지정된 파이프라인에서 지난 6개월 분량의 작업을 반환합니다.

PipelineId

지정된 파이프라인에 추가한 작업에 대한 정보를 확인하려면 해당 파이프라인의 `Id` 객체의 값을 지정합니다.

Ascending

작업을 제출 날짜 및 시간순으로 나열하려면 `true`를 입력합니다. 작업을 시간 역순으로 나열하려면 `false`를 입력합니다.

PageToken

Elastic Transcoder가 반환하는 결과 페이지가 여러 개라면 후속 GET 요청에서 PageToken을 사용해 이어지는 각 결과 페이지를 확인할 수 있습니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

이 작업은 요청 본문을 사용하지 않습니다.

응답

구문

```
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|mxf|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
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

응답 본문에는 각 작업에 대해 검색 조건을 충족하는 요소 한 개가 포함되며, 각 요소에는 작업을 만들 때 지정한 값이 포함됩니다. 작업 값에 대한 자세한 내용은 Elastic Transcoder 작업을 만들 때 지정하는 설정 섹션을 참조하세요.

또한 Elastic Transcoder는 다음 값을 반환합니다.

(자동) Id

Elastic Transcoder가 작업에 할당한 식별자입니다. 이 값을 사용하여 작업의 설정을 가져오거나 작업을 삭제합니다.

(자동) Inputs:DetectedProperties

입력 파일의 감지된 속성입니다. Elastic Transcoder는 입력 파일에서 이러한 값을 식별합니다.

(자동) Inputs:Width

입력 파일의 감지된 너비(픽셀 단위)입니다.

(자동) Inputs:Height

입력 파일의 감지된 높이(픽셀 단위)입니다.

(자동) Inputs:FrameRate

입력 파일의 감지된 프레임 속도(초당 프레임 수)입니다.

(자동) Inputs:FileSize

입력 파일의 감지된 크기(바이트)입니다.

(자동) Inputs:DurationMillis

입력 파일의 감지된 지속 시간(밀리초)입니다.

(자동) Outputs:Id

현재 작업의 출력 중에서 출력을 식별하는 1부터 시작하는 순차 카운터. `Output` 구문에서 이 값은 항상 1입니다.

(자동) Outputs:Duration

출력 파일의 지속 시간(초)으로, 반올림됩니다.

(자동) Outputs:DurationMillis

출력 파일의 기간(밀리초)입니다.

(자동) Outputs:Width

출력 파일의 너비(픽셀)입니다.

(자동) Outputs:Height

출력 파일의 높이(픽셀)입니다.

(자동) Outputs:FrameRate

출력 파일의 프레임 속도(초당 프레임 수)입니다.

(자동) Outputs:FileSize

출력 파일의 파일 크기(바이트)입니다.

(자동) Outputs:Status

작업 내 단일 출력의 상태입니다. 작업에 지정한 출력이 하나뿐일 경우 Outputs:Status는 항상 Job:Status와 동일합니다. 여러 개의 출력을 지정한 경우:

- Elastic Transcoder가 첫 번째 출력을 처리하기 시작할 때까지 모든 출력의 Job:Status 및 Outputs:Status는 Submitted입니다.
- Elastic Transcoder가 첫 번째 출력을 처리하기 시작하면 해당 출력의 Outputs:Status와 Job:Status가 모두 Progressing으로 바뀝니다. 각 출력에서 Outputs:Status 값은 Elastic Transcoder가 해당 출력을 처리하기 시작할 때까지 Submitted를 유지합니다.
- 모든 출력이 최종 상태(Job:Status 또는 Progressing)에 도달할 때까지 Complete는 Error을 유지합니다.
- 모든 출력이 최종 상태에 도달하면 모든 출력의 Outputs:Status가 Complete인 경우에만 Job:Status가 Complete로 바뀝니다. 하나 이상의 출력에서 Outputs:Status가 Error일 경우Job:Status의 최종 상태도 Error입니다.

Status 값은 다음 중 하나입니다. Submitted, Progressing, Complete, Canceled 또는 Error.

(자동) Outputs:StatusDetail

Outputs:Status를 더 자세히 설명하는 정보입니다.

(자동) Outputs:AppliedColorSpaceConversion

Elastic Transcoder가 ColorSpaceConversionMode를 포함한 프리셋을 사용하여 출력 파일을 트랜스코딩한 경우 AppliedColorSpaceConversion 파라미터가 사용된 변환을 표시합니다. 프리셋에 ColorSpaceConversionMode가 정의되지 않은 경우 이 파라미터는 작업 응답에 포함되지 않습니다.

(자동) Status

작업에 여러 개의 출력을 지정한 경우 전체 작업의 상태. Elastic Transcoder가 작업을 처리하기 시작하면 Job:Status가 Progressing으로 바뀐 후 Elastic Transcoder가 모든 출력을 처리할 때까

지 바뀌지 않습니다. 처리가 완료되면 `Job:Status`가 `Complete`로 바뀌거나, 실패한 출력이 있을 경우 `Error`로 바뀝니다.

작업에 지정한 출력이 하나뿐일 경우 `Job:Status`는 `Outputs:Status`와 동일합니다.

`Job:Status` 값은 다음 중 하나입니다. `Submitted`, `Progressing`, `Complete`, `Canceled` 또는 `Error`.

(자동) Timing

작업 타이밍에 대한 세부 정보입니다.

(자동) Timing:SubmitTimeMillis

작업이 Elastic Transcoder에 제출된 시간(epoch 밀리초)입니다.

(자동) Timing:StartTimeMillis

작업이 트랜스코딩을 시작한 시간(epoch 밀리초)입니다.

(자동) Timing:FinishTimeMillis

작업이 트랜스코딩을 마친 시간(epoch 밀리초)입니다.

epoch 시간에 대한 자세한 내용은 Wikipedia의 Epoch Computing 페이지를 참조하세요.

(자동) NextPageToken

결과의 두 번째 및 이후 페이지(있는 경우)에 액세스하는 데 사용하는 값. 지정된 파이프라인의 작업이 한 페이지에 표시될 수 있거나 결과의 마지막 페이지에 도달한 경우 `NextPageToken` 값은 `null`입니다.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예시

다음 예제 요청은 작업을 만듭니다.

샘플 요청

다음 예제 요청은 지난 6개월 동안 파이프라인 ID `1111111111111-abcde1`에 할당한 모든 작업의 목록을 가져옵니다.

```
GET /2012-09-25/jobsByPipeline/1111111111111-abcde1?Ascending=true HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder endpoint/
elastictranscoder/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

샘플 응답

```
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",
         "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":"4444444444444-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":"1111111111111-abcde1",
      "Timing":{
         "SubmitTime":"1427212800000",
         "StartTime":"1427212856000",
         "FinishTime":"1427212875000"
      }
   }
],
"NextPageToken":null
```

```
}
```

상태별 작업 나열

주제

- 설명
- 요청
- 응답
- 오류
- 예시

설명

지정된 상태의 작업 목록을 얻으려면 /2012-09-25/jobsByStatus/Submitted 리소스에 GET 요청을 보냅니다. Elastic Transcoder는 최근에 생성한 작업과 현재 상태를 지정한 작업을 나열합니다.

요청

구문

현재 AWS 계정에 연결되었으며 지난 6개월 동안 지정된 상태의 모든 작업에 대한 정보를 확인하려면 다음 GET 요청을 보냅니다.

```
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
```

요청 파라미터

이 작업은 다음 요청 파라미터를 사용합니다. Elastic Transcoder는 지난 6개월 동안 상태가 지정된 모든 작업을 반환합니다.

상태

현재 AWS 계정에 연결되었으며 지난 6개월 동안 지정된 상태의 모든 작업에 대한 정보를 확인하려면 상태 `Submitted`, `Progressing`, `Complete`, `Canceled` 또는 `Error`를 지정합니다.

Ascending

작업을 제출 날짜 및 시간순으로 나열하려면 `true`를 입력합니다. 작업을 시간 역순으로 나열하려면 `false`를 입력합니다.

PageToken

Elastic Transcoder가 반환하는 결과 페이지가 여러 개라면 후속 GET 요청에서 PageToken을 사용해 이어지는 각 결과 페이지를 확인할 수 있습니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

이 작업은 요청 본문을 사용하지 않습니다.

응답

구문

```
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|mxf|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
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

응답 본문에는 각 작업에 대해 검색 조건을 충족하는 요소 한 개가 포함되며, 각 요소에는 작업을 만들 때 지정한 값이 포함됩니다. 작업 값에 대한 자세한 내용은 Elastic Transcoder 작업을 만들 때 지정하는 설정 섹션을 참조하세요.

또한 Elastic Transcoder는 다음 값을 반환합니다.

(자동) Id

Elastic Transcoder가 작업에 할당한 식별자입니다. 이 값을 사용하여 작업의 설정을 가져오거나 작업을 삭제합니다.

(자동) Inputs:DetectedProperties

입력 파일의 감지된 속성입니다. Elastic Transcoder는 입력 파일에서 이러한 값을 식별합니다.

(자동) Inputs:Width

입력 파일의 감지된 너비(픽셀 단위)입니다.

(자동) Inputs:Height

입력 파일의 감지된 높이(픽셀 단위)입니다.

(자동) Inputs:FrameRate

입력 파일의 감지된 프레임 속도(초당 프레임 수)입니다.

(자동) Inputs:FileSize

입력 파일의 감지된 크기(바이트)입니다.

(자동) Inputs:DurationMillis

입력 파일의 감지된 지속 시간(밀리초)입니다.

(자동) Outputs:Id

현재 작업의 출력 중에서 출력을 식별하는 1부터 시작하는 순차 카운터. Output 구문에서 이 값은 항상 1입니다.

(자동) Outputs:Duration

출력 파일의 지속 시간(초)으로, 반올림됩니다.

(자동) Outputs:DurationMillis

출력 파일의 기간(밀리초)입니다.

(자동) Outputs:Width

출력 파일의 너비(픽셀)입니다.

(자동) Outputs:Height

출력 파일의 높이(픽셀)입니다.

(자동) Outputs:FrameRate

출력 파일의 프레임 속도(초당 프레임 수)입니다.

(자동) Outputs:FileSize

출력 파일의 파일 크기(바이트)입니다.

(자동) Outputs:Status

작업 내 단일 출력의 상태입니다. 작업에 지정한 출력이 하나뿐일 경우 Outputs:Status는 항상 Job:Status와 동일합니다. 여러 개의 출력을 지정한 경우:

- Elastic Transcoder가 첫 번째 출력을 처리하기 시작할 때까지 모든 출력의 Job:Status 및 Outputs:Status는 Submitted입니다.
- Elastic Transcoder가 첫 번째 출력을 처리하기 시작하면 해당 출력의 Outputs:Status와 Job:Status가 모두 Progressing으로 바뀝니다. 각 출력에서 Outputs:Status 값은 Elastic Transcoder가 해당 출력을 처리하기 시작할 때까지 Submitted를 유지합니다.
- 모든 출력이 최종 상태(Job:Status 또는 Progressing)에 도달할 때까지 Complete는 Error을 유지합니다.

- 모든 출력이 최종 상태에 도달하면 모든 출력의 Outputs:Status가 Complete인 경우에만 Job:Status가 Complete로 바뀝니다. 하나 이상의 출력에서 Outputs:Status가 Error일 경우Job:Status의 최종 상태도 Error입니다.

Status 값은 다음 중 하나입니다. Submitted, Progressing, Complete, Canceled 또는 Error.

(자동) Outputs:StatusDetail

Outputs:Status를 더 자세히 설명하는 정보입니다.

(자동) Outputs:AppliedColorSpaceConversion

Elastic Transcoder가 ColorSpaceConversionMode를 포함한 프리셋을 사용하여 출력 파일을 트랜스코딩한 경우 AppliedColorSpaceConversion 파라미터가 사용된 변환을 표시합니다. 프리셋에 ColorSpaceConversionMode가 정의되지 않은 경우 이 파라미터는 작업 응답에 포함되지 않습니다.

(자동) Status

작업에 여러 개의 출력을 지정한 경우 전체 작업의 상태. Elastic Transcoder가 작업을 처리하기 시작하면 Job:Status가 Progressing으로 바뀐 후 Elastic Transcoder가 모든 출력을 처리할 때까지 바뀌지 않습니다. 처리가 완료되면 Job:Status가 Complete로 바뀌거나, 실패한 출력이 있을 경우 Error로 바뀝니다.

작업에 지정한 출력이 하나뿐일 경우 Job:Status는 Outputs:Status와 동일합니다.

Job:Status 값은 다음 중 하나입니다. Submitted, Progressing, Complete, Canceled 또는 Error.

(자동) Timing

작업 타이밍에 대한 세부 정보입니다.

(자동) Timing:SubmitTimeMillis

작업이 Elastic Transcoder에 제출된 시간(epoch 밀리초)입니다.

(자동) Timing:StartTimeMillis

작업이 트랜스코딩을 시작한 시간(epoch 밀리초)입니다.

(자동) Timing:FinishTimeMillis

작업이 트랜스코딩을 마친 시간(epoch 밀리초)입니다.

epoch 시간에 대한 자세한 내용은 Wikipedia의 Epoch Computing 페이지를 참조하세요.

(자동) NextPageToken

결과의 두 번째 및 이후 페이지(있는 경우)에 액세스하는 데 사용하는 값. 지정된 파이프라인의 작업이 한 페이지에 표시될 수 있거나 결과의 마지막 페이지에 도달한 경우 NextPageToken 값은 null입니다.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예시

다음 예제 요청은 작업을 만듭니다.

샘플 요청

다음 예제 요청은 지난 6개월 동안 생성한 작업 중 상태가 Complete인 모든 작업의 목록을 가져옵니다.

```
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
```

샘플 응답

```
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":"1111111111111-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":"4444444444444-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
}
```

작업 읽기

주제

- 설명
- 요청
- 응답
- 오류
- 예

설명

작업에 대한 세부 정보를 확인하려면 /2012-09-25/jobs/*jobId* 리소스에 GET 요청을 보냅니다.

요청

구문

```
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
```

요청 매개변수

이 작업은 다음 요청 파라미터를 사용합니다.

jobId

세부 정보를 조회하려는 작업의 식별자입니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

이 작업에는 요청 본문이 없습니다.

응답

구문

```
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|mxf|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"
   }
 }
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

검색 조건을 만족하는 각 작업에 대해 응답 본문에는 작업을 만들 때 지정한 값이 포함됩니다. 작업 값에 대한 자세한 내용은 Elastic Transcoder 작업을 만들 때 지정하는 설정 섹션을 참조하세요.

또한 Elastic Transcoder는 다음 값을 반환합니다.

(자동) Id

Elastic Transcoder가 작업에 할당한 식별자입니다. 이 값을 사용하여 작업의 설정을 가져오거나 작업을 삭제합니다.

(자동) Inputs:DetectedProperties

입력 파일의 감지된 속성입니다. Elastic Transcoder는 입력 파일에서 이러한 값을 식별합니다.

(자동) Inputs:Width

입력 파일의 감지된 너비(픽셀 단위)입니다.

(자동) Inputs:Height

입력 파일의 감지된 높이(픽셀 단위)입니다.

(자동) Inputs:FrameRate

입력 파일의 감지된 프레임 속도(초당 프레임 수)입니다.

(자동) Inputs:FileSize

입력 파일의 감지된 크기(바이트)입니다.

(자동) Inputs:DurationMillis

입력 파일의 감지된 지속 시간(밀리초)입니다.

(자동) Outputs:Id

현재 작업의 출력 중에서 출력을 식별하는 1부터 시작하는 순차 카운터. `Output` 구문에서 이 값은 항상 1입니다.

(자동) Outputs:Duration

출력 파일의 지속 시간(초)으로, 반올림됩니다.

(자동) Outputs:DurationMillis

출력 파일의 기간(밀리초)입니다.

(자동) Outputs:Width

출력 파일의 너비(픽셀)입니다.

(자동) Outputs:Height

출력 파일의 높이(픽셀)입니다.

(자동) Outputs:FrameRate

출력 파일의 프레임 속도(초당 프레임 수)입니다.

(자동) Outputs:FileSize

출력 파일의 파일 크기(바이트)입니다.

(자동) Outputs:Status

작업 내 단일 출력의 상태입니다. 작업에 지정한 출력이 하나뿐일 경우 Outputs:Status는 항상 Job:Status와 동일합니다. 여러 개의 출력을 지정한 경우:

- Elastic Transcoder가 첫 번째 출력을 처리하기 시작할 때까지 모든 출력의 Job:Status 및 Outputs:Status는 Submitted입니다.
- Elastic Transcoder가 첫 번째 출력을 처리하기 시작하면 해당 출력의 Outputs:Status와 Job:Status가 모두 Progressing으로 바뀝니다. 각 출력에서 Outputs:Status 값은 Elastic Transcoder가 해당 출력을 처리하기 시작할 때까지 Submitted를 유지합니다.
- 모든 출력이 최종 상태(Job:Status 또는 Progressing)에 도달할 때까지 Complete는 Error을 유지합니다.
- 모든 출력이 최종 상태에 도달하면 모든 출력의 Outputs:Status가 Complete인 경우에만 Job:Status가 Complete로 바뀝니다. 하나 이상의 출력에서 Outputs:Status가 Error일 경우Job:Status의 최종 상태도 Error입니다.

Status 값은 다음 중 하나입니다. Submitted, Progressing, Complete, Canceled 또는 Error.

(자동) Outputs:StatusDetail

Outputs:Status를 더 자세히 설명하는 정보입니다.

(자동) Outputs:AppliedColorSpaceConversion

Elastic Transcoder가 ColorSpaceConversionMode를 포함한 프리셋을 사용하여 출력 파일을 트랜스코딩한 경우 AppliedColorSpaceConversion 파라미터가 사용된 변환을 표시합니다. 프리셋에 ColorSpaceConversionMode가 정의되지 않은 경우 이 파라미터는 작업 응답에 포함되지 않습니다.

(자동) Status

작업에 여러 개의 출력을 지정한 경우 전체 작업의 상태. Elastic Transcoder가 작업을 처리하기 시작하면 Job:Status가 Progressing으로 바뀐 후 Elastic Transcoder가 모든 출력을 처리할 때까지 바뀌지 않습니다. 처리가 완료되면 Job:Status가 Complete로 바뀌거나, 실패한 출력이 있을 경우 Error로 바뀝니다.

작업에 지정한 출력이 하나뿐일 경우 Job:Status는 Outputs:Status와 동일합니다.

Job:Status 값은 다음 중 하나입니다. Submitted, Progressing, Complete, Canceled 또는 Error.

(자동) Timing

작업 타이밍에 대한 세부 정보입니다.

(자동) Timing:SubmitTimeMillis

작업이 Elastic Transcoder에 제출된 시간(epoch 밀리초)입니다.

(자동) Timing:StartTimeMillis

작업이 트랜스코딩을 시작한 시간(epoch 밀리초)입니다.

(자동) Timing:FinishTimeMillis

작업이 트랜스코딩을 마친 시간(epoch 밀리초)입니다.

epoch 시간에 대한 자세한 내용은 Wikipedia의 Epoch Computing 페이지를 참조하세요.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예

예 요청

다음 예제 요청은 작업 ID가 3333333333333-abcde3인 작업을 가져옵니다.

```
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
```

샘플 응답

```
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":"1111111111111-abcde1",
      "Status":"Progressing",
      "Timing":{
         "SubmitTime":"1427212800000",
         "StartTime":"1427212856000",
         "FinishTime":"1427212875000"
      }
   }
```

```
}
```

작업 취소

주제

- 설명
- 요청
- 응답
- 오류
- 예시

설명

Elastic Transcoder가 처리를 시작하지 않은 작업 취소 시 `/2012-09-25/jobs/`*`jobId`* 리소스에 삭제 요청을 보내세요. `Read Job`, `List Job by Pipeline` 또는 `List Job by Status` 요청을 제출해도 Elastic Transcoder는 여전히 작업을 반환할 수 있지만 Elastic Transcoder는 입력 파일을 트랜스코딩하지 않으므로 작업에 대한 요금이 부과되지 않습니다.

> **Note**
>
> 상태가 Submitted인 작업만 취소할 수 있습니다. 작업 식별자를 가져오는 동안 파이프라인이 작업 처리를 시작하지 못하게 하려면 파이프라인 상태 업데이트를 사용하여 파이프라인을 일시적으로 중지합니다.

요청

구문

```
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
```

요청 매개변수

이 작업은 다음 요청 파라미터를 사용합니다.

jobId

취소하려는 작업의 식별자입니다.

상태가 Submitted인 작업 목록(jobId 포함)을 가져오려면 상태별 작업 나열 API 작업을 사용합니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

이 작업에는 요청 본문이 없습니다.

응답

구문

```
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"
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

응답 본문에는 다음 JSON 객체가 포함됩니다.

Success

작업이 취소된 경우 Success의 값은 true입니다.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예시

다음 예제 요청은 ID가 3333333333333-abcde3인 작업을 취소합니다.

샘플 요청

```
DELETE /2012-09-25/jobs/3333333333333-abcde3 HTTP/1.1
Content-Type: charset=UTF-8
Accept: */*
Host: elastictranscoder.Elastic Transcoder endpoint.amazonaws.com:443
x-amz-date: 20130114T174952Z
Authorization: AWS4-HMAC-SHA256
               Credential=AccessKeyID/request-date/Elastic Transcoder endpoint/
elastictranscoder/aws4_request,
               SignedHeaders=host;x-amz-date;x-amz-target,
               Signature=calculated-signature
```

샘플 응답

```
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"
}
```

프리셋 작업

주제

- 프리셋 만들기
- 프리셋 나열
- 프리셋 읽기
- 프리셋 삭제

프리셋은 미디어 파일을 한 형식에서 다른 형식으로 트랜스코딩하기 위한 대부분의 설정이 들어 있는 템플릿입니다. Elastic Transcoder에는 일반적인 형식을 위한 몇 가지 기본 프리셋이 포함되어 있습니다(예: 여러 iPod 및 iPhone 버전). 기본 프리셋에 포함되지 않은 형식의 고유 프리셋을 생성할 수도 있습니다. 작업을 만들 때 사용할 프리셋을 지정합니다.

이 섹션에서는 Elastic Transcoder API를 사용해 프리셋에 수행할 수 있는 작업을 설명합니다. Elastic Transcoder 콘솔을 사용해 같은 작업을 수행하는 방법 등 프리셋에 대한 자세한 내용은 프리셋 작업 섹션을 참조하세요.

프리셋 만들기

주제

- 설명
- 요청
- 응답
- 오류
- 예시

설명

프리셋을 생성하려면 `/2012-09-25/presets` 리소스에 POST 요청을 보냅니다.

> **Important**
>
> Elastic Transcoder는 지정한 설정이 Elastic Transcoder 요구 사항을 충족하고 적용되는 표준을 준수하는지 확인하기 위해 지정한 설정을 확인합니다. 설정이 Elastic Transcoder에 유효하지 않은 경우, Elastic Transcoder는 HTTP 400 응답(`ValidationException`)을 반환하고 프리셋을 생성하지 않습니다. 설정이 Elastic Transcoder에 유효하나 표준을 엄격하게 준수하지

않은 경우, Elastic Transcoder는 프리셋을 생성하고 경고 메시지를 응답으로 반환합니다. 이를 통해 설정이 표준을 준수하는지 여부를 확인할 수 있고, Elastic Transcoder가 생성하는 파일에 대해 유연성을 높일 수 있습니다.

> **Note**
>
> Elastic Transcoder에는 일반적인 출력 형식의 여러 시스템 프리셋이 포함되어 있습니다. 자세한 내용은 작업 생성 주제에서 PresetId 섹션을 참조하세요.

요청

구문

```
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|mxf|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 pixelsxheight in pixels" <not recommended>,
      "AspectRatio":"auto|1:1|4:3|3:2|16:9" <not recommended>
   }
}
```

요청 파라미터

이 작업은 요청 파라미터를 사용하지 않습니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

요청 본문의 JSON 문자열에는 CreatePreset 작업에 대한 입력 객체가 포함되어 있습니다. 입력 객체에 대한 자세한 내용은 Elastic Transcoder 프리셋을 생성할 때 지정하는 설정 섹션을 참조하세요.

응답

구문

```
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|mxf|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"
       }
    },
    "Warning":"message about codec compatibility"
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

작업을 만들면 Elastic Transcoder는 사용자가 요청에 지정한 값을 반환합니다. 자세한 내용은 요청 본문 단원을 참조하십시오.

또한 Elastic Transcoder는 다음 값을 반환합니다.

(자동) Id

새로운 프리셋의 식별자입니다. 이 값을 사용하여 프리셋의 설정을 가져오거나 프리셋을 삭제할 수 있습니다.

(자동) Type

프리셋이 Elastic Transcoder가 제공하는 기본 프리셋(System) 또는 사용자가 정의한 프리셋(Custom)인지 여부입니다.

(자동) Warning

프리셋에 대한 설정이 Codec의 표준과 호환되지는 않지만 해당 설정이 허용 가능한 출력을 생성할 수 있는 경우 Elastic Transcoder는 프리셋을 생성하고 프리셋 설정이 호환되지 않는 이유를 설명하는 경고를 포함합니다.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예시

다음 예제 요청은 DefaultPreset이라는 프리셋을 생성합니다.

샘플 요청

```
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"
   }
}
```

샘플 응답

```
Status: 201 Created
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Preset":{
      "Id":"5555555555555-abcde5",
      "Type":"Custom",
      "Name":"DefaultPreset",
      "Description":"Use for published videos",
      "Container":"mp4",
      "Audio":{
         "Codec":"AAC",
         "CodecOptions":{
            "Profile":"AAC-LC"
         },
         "SampleRate":"44100",
         "BitRate":"96",
         "Channels":"2"
      },
      "Video":{
         "Codec":"H.264",
         "CodecOptions":{
            "Profile":"main",
            "Level":"2.2",
            "MaxReferenceFrames":"3",
            "MaxBitRate":"",
            "BufferSize":"",
            "InterlacedMode":"Progressive",
            "ColorSpaceConversionMode":"None|Bt709ToBt601|Bt601ToBt709|Auto"
         },
         "KeyframesMaxDist":"240",
         "FixedGOP":"false",
         "BitRate":"1600",
         "FrameRate":"auto",
         "MaxFrameRate":"30",
         "MaxWidth":"auto",
         "MaxHeight":"auto",
         "SizingPolicy":"Fit",
         "PaddingPolicy":"Pad",
```

```
         "DisplayAspectRatio":"auto",
         "Watermarks":[
            {
               "Id":"company logo",
               "MaxWidth":"20%",
               "MaxHeight":"20%",
               "SizingPolicy":"ShrinkToFit",
               "HorizontalAlign":"Right",
               "HorizontalOffset":"10px",
               "VerticalAlign":"Bottom",
               "VerticalOffset":"10px",
               "Opacity":"55.5",
               "Target":"Content"
            }
         ]
      },
      "Thumbnails":{
         "Format":"png",
         "Interval":"120",
         "MaxWidth":"auto",
         "MaxHeight":"auto",
         "SizingPolicy":"Fit",
         "PaddingPolicy":"Pad"
      }
   },
   "Warning":""
}
```

프리셋 나열

주제

- 설명
- 요청
- 응답
- 오류
- 예시

설명

현재 AWS 계정에 연결된 모든 프리셋의 목록을 확인하려면 /2012-09-25/presets 리소스에 GET 요청을 보내세요.

요청

구문

```
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
```

요청 파라미터

이 작업은 다음 요청 파라미터를 사용합니다. Elastic Transcoder는 사용 가능한 모든 프리셋을 반환합니다.

Ascending

프리셋을 제출 날짜 및 시간순으로 나열하려면 true를 입력하세요. 프리셋을 역순으로 나열하려면 false를 입력하세요.

PageToken

Elastic Transcoder가 반환하는 결과 페이지가 여러 개라면 후속 GET 요청에서 PageToken을 사용해 이어지는 각 결과 페이지를 확인할 수 있습니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

요청 본문의 JSON 문자열에는 다음 객체가 포함되어 있습니다.

응답

구문

```
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|mxf|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
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

응답 본문의 JSON 문자열은 프리셋을 생성할 때 지정한 값을 반환합니다. 각 객체에 대한 자세한 내용은 Elastic Transcoder 프리셋을 생성할 때 지정하는 설정 섹션을 참조하세요.

또한 Elastic Transcoder는 다음 값을 반환합니다.

(자동) Id

프리셋의 식별자. 이 값을 사용하여 프리셋의 설정을 가져오거나 프리셋을 삭제할 수 있습니다.

(자동) Type

프리셋이 Elastic Transcoder가 제공하는 기본 프리셋(System) 또는 사용자가 정의한 프리셋(Custom)인지 여부입니다.

(자동) NextPageToken

결과의 두 번째 및 이후 페이지(있는 경우)에 액세스하는 데 사용하는 값. 프리셋이 한 페이지에 표시될 수 있거나 결과의 마지막 페이지에 도달한 경우 NextPageToken 값은 null입니다.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예시

다음 예제 요청은 DefaultPreset이라는 프리셋을 생성합니다.

샘플 요청

```
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
```

샘플 응답

```
Status: 200 OK
x-amzn-RequestId: c321ec43-378e-11e2-8e4c-4d5b971203e9
Content-Type: application/json
Content-Length: number of characters in the response
Date: Mon, 14 Jan 2013 06:01:47 GMT

{
   "Presets":[
      {
         "Id":"5555555555555-abcde5",
         "Type":"Custom",
         "Name":"DefaultPreset",
         "Description":"Use for published videos",
         "Container":"mp4",
         "Audio":{
            "BitRate":"96",
            "Channels":"2",
            "Codec":"AAC",
            "CodecOptions":{
               "Profile":"AAC-LC"
            },
            "SampleRate":"44100"
         },
         "Video":{
            "Codec":"H.264",
            "CodecOptions":{
               "Profile":"main",
               "Level":"2.2",
               "MaxReferenceFrames":"3",
               "MaxBitRate":"",
               "BufferSize":"",
```

```
               "InterlacedMode":"Progressive",
               "ColorSpaceConversionMode":"None"
            },
            "KeyframesMaxDist":"240",
            "FixedGOP":"false",
            "BitRate":"1600",
            "FrameRate":"auto",
            "MaxFrameRate":"30",
            "MaxWidth":"auto",
            "MaxHeight":"auto",
            "SizingPolicy":"Fit",
            "PaddingPolicy":"Pad",
            "DisplayAspectRatio":"auto",
            "Watermarks":[
               {
                  "Id":"company logo",
                  "MaxWidth":"20%",
                  "MaxHeight":"20%",
                  "SizingPolicy":"ShrinkToFit",
                  "HorizontalAlign":"Right",
                  "HorizontalOffset":"10px",
                  "VerticalAlign":"Bottom",
                  "VerticalOffset":"10px",
                  "Opacity":"55.5",
                  "Target":"Content"
               }
            ]
         }
         "Thumbnails":{
            "Format":"png",
            "Interval":"120",
            "MaxWidth":"auto",
            "MaxHeight":"auto",
            "SizingPolicy":"Fit",
            "PaddingPolicy":"Pad"
         },
      },
      {...}
   ]
}
```

프리셋 읽기

주제

- 설명
- 요청
- 응답
- 오류
- 예시

설명

프리셋에 대한 세부 정보를 보려면 /2012-09-25/presets/*presetId* 리소스에 GET 요청을 보냅니다.

요청

구문

```
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
```

요청 매개변수

이 작업은 다음 요청 파라미터를 사용합니다.

presetId

세부 정보를 조회하려는 프리셋의 식별자입니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

이 작업에는 요청 본문이 없습니다.

응답

구문

```
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|mxf|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"
      }
   }
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

프리셋을 조회하면, Elastic Transcoder는 프리셋을 만들 때 지정한 값을 반환합니다. 자세한 내용은 Elastic Transcoder 프리셋을 생성할 때 지정하는 설정 단원을 참조하십시오.

또한 Elastic Transcoder는 다음 값을 반환합니다.

(자동) Id

프리셋의 식별자. 이 값을 사용하여 프리셋의 설정을 가져오거나 프리셋을 삭제할 수 있습니다.

(자동) Type

프리셋이 Elastic Transcoder가 제공하는 기본 프리셋(System) 또는 사용자가 정의한 프리셋(Custom)인지 여부입니다.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예시

다음 예제 요청은 ID가 5555555555555-abcde5인 프리셋을 가져옵니다.

샘플 요청

```
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
```

샘플 응답

```
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":"5555555555555-abcde5",
      "Type":"Custom",
      "Name":"DefaultPreset",
      "Description":"Use for published videos",
      "Container":"mp4",
      "Audio":{
         "Codec":"AAC",
         "CodecOptions":{
            "Profile":"AAC-LC"
         },
         "SampleRate":"44100",
         "BitRate":"96",
         "Channels":"2"
      },
      "Video":{
         "Codec":"H.264",
         "CodecOptions":{
```

```
            "Profile":"main",
            "Level":"2.2",
            "MaxReferenceFrames":"3",
            "MaxBitRate":"",
            "BufferSize":"",
            "InterlacedMode":"Progressive",
            "ColorSpaceConversionMode":"None"
         },
         "KeyframesMaxDist":"240",
         "FixedGOP":"false",
         "BitRate":"1600",
         "FrameRate":"auto",
         "MaxFrameRate":"30",
         "MaxWidth":"auto",
         "MaxHeight":"auto",
         "SizingPolicy":"Fit",
         "PaddingPolicy":"Pad",
         "DisplayAspectRatio":"auto",
         "Watermarks":[
            {
               "Id":"company logo",
               "MaxWidth":"20%",
               "MaxHeight":"20%",
               "SizingPolicy":"ShrinkToFit",
               "HorizontalAlign":"Right",
               "HorizontalOffset":"10px",
               "VerticalAlign":"Bottom",
               "VerticalOffset":"10px",
               "Opacity":"55.5",
               "Target":"Content"
            }
         ]
      }
      "Thumbnails":{
         "Format":"png",
         "Interval":"120",
         "MaxHeight":"auto",
         "MaxWidth":"auto",
         "SizingPolicy":"Fit",
         "PaddingPolicy":"Pad"
      },
   },
   "Warning":""
```

```
}
```

프리셋 삭제

주제

- 설명
- 요청
- 응답
- 오류
- 예시

설명

프리셋을 삭제하려면 /2012-09-25/presets/*presetId* 리소스에 DELETE 요청을 보냅니다.

> **Note**
>
> 사용한 프리셋은 삭제할 수 없습니다.

요청

구문

```
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
```

요청 파라미터

이 작업은 다음 요청 파라미터를 사용합니다.

presetId

세부 정보를 조회하려는 프리셋의 식별자입니다.

요청 헤더

이 작업은 모든 작업에 일반적인 요청 헤더만 사용합니다. 일반적인 요청 헤더에 대한 내용은 HTTP 헤더 콘텐츠 섹션을 참조하세요.

요청 본문

이 작업에는 요청 본문이 없습니다.

응답

구문

```
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"
}
```

응답 헤더

이 작업은 대부분의 응답에 일반적인 응답 헤더만 사용합니다. 일반적인 응답 헤더에 대한 내용은 HTTP 응답 섹션을 참조하세요.

응답 본문

응답 본문에는 다음 JSON 객체가 포함됩니다.

Success

프리셋이 삭제된 경우 Success의 값은 true입니다.

오류

Elastic Transcoder 예외 및 오류 메시지에 대한 자세한 내용은 Elastic Transcoder에서의 오류 처리 섹션을 참조하세요.

예시

다음 예제 요청은 ID가 5555555555555-abcde5인 프리셋을 삭제합니다.

샘플 요청

```
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
```

샘플 응답

```
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"
}
```

문서 기록

다음 표에서는 본 Amazon Elastic Transcoder 릴리스의 설명서에서 변경된 중요 사항에 대해 설명합니다.

- API 버전: 2012-09-25
- 마지막 설명서 업데이트: 2016년 11월 17일

변경 사항	설명	날짜
지원 종료 알림	지원 종료 공지: 2025년 11월 13일에는 Amazon Elastic Transcoder에 대한 지원을 중단할 AWS 예정입니다. 2025년 11월 13일 이후에는 더 이상 Elastic Transcoder 콘솔 또는 Elastic Transcoder 리소스에 액세스할 수 없습니다. 로 전환하는 방법에 대한 자세한 내용은이 블로그 게시물을 AWS Elemental MediaConvert참조하세요.	2024년 11월 12일
새 기능	리전 간 경고: 리전 간 설정으로 파이프라인을 생성하거나 업데이트하거나 검색하면 Elastic Transcoder가 이를 경고합니다.	2019년 2월 19일
새 기능	클립 스티칭: 이제 Elastic Transcoder를 사용하여 여러 파일의 부분들을 연결하여 하나의 출력 파일로 만들 수 있습니다.	2017년 11월 17일

새 기능	AWS 아시아 태평양(뭄바이) 리전: 이제 AWS 리전 ap-south-1에서 Elastic Transcoder를 사용할 수 있습니다.	2016년 8월 30일
새 기능	CloudWatch를 사용하여 모니터링: 이제 계정을 통해 Elastic Transcoder 리소스 및 API 직접 호출에 대한 모니터링을 설정할 수 있습니다.	2016년 8월 20일
새 기능	확장된 오디오(WAV): Elastic Transcoder가 이제 오디오용 WAV 컨테이너를 지원합니다.	2016년 7월 19일
새 기능	확장된 재생 목록 지원(MPEG-DASH): Elastic Transcoder가 이제 FMP4 컨테이너용 MPEG-DASH 재생 목록을 지원합니다.	2016년 5월 24일
새 기능	확장된 비디오 지원(VP9): Elastic Transcoder가 이제 WEBM 컨테이너용 VP9 비디오 코덱을 지원합니다.	2016년 20월 4일
새 기능	확장된 캡션 지원(CEA-708): Elastic Transcoder가 이제 MP4 및 MPEG-TS 컨테이너용 CEA-708 임베디드 캡션 출력을 지원합니다.	2015년 10월 21일
새 기능	확장된 형식 지원(MXF, FLAC 등): Elastic Transcoder가 이제 MXF, FLAC, OGA를 포함한 추가 형식을 지원하고, 출력 오디오의 유연성을 한층 높여줍니다.	2015년 5월 20일

새 기능	PlayReady DRM: Elastic Transcoder가 이제 파일의 PlayReady DRM을 지원합니다.	2015년 3월 31일
새 기능	감지된 파라미터 및 작업 타이밍: Elastic Transcoder가 이제 입력 파일, 출력 파일 및 작업을 처리하는 데 걸린 시간(epoch 밀리초 단위)의 감지된 속성을 기록합니다.	2015년 3월 24일
새 기능	확장된 형식 지원(PAL, NTSC 등): Elastic Transcoder가 이제 추가 형식, 인터레이스 신호, 크로마 하위 샘플링, 색상 변환을 지원합니다.	2015년 3월 17일
새 기능	HLS 콘텐츠 보호: 이제 스트리밍 미디어의 콘텐츠 보호를 설정할 수 있습니다.	2015년 1월 13일
새 기능	사용자 정의 메타데이터: 이제 파일의 사용자 지정 메타데이터를 만들 수 있으며, Elastic Transcoder가 트랜스코딩 프로세스를 통해 이를 그대로 전달합니다.	2014년 12월 10일
새 기능	암호화된 파일: 이제 출력 파일을 암호화하고 암호화된 파일을 트랜스코딩할 수 있습니다.	2014년 11월 24일
새 기능	CloudTrail로 로깅: 이제 계정을 통해 Elastic Transcoder api 호출에 대한 로깅을 설정할 수 있습니다.	2014년 10월 27일

새 기능	HTTP Live Streaming v4: 이제 HTTP Live Streaming v4 스트리밍 파일을 생성할 수 있습니다.	2014년 10월 13일
새 기능	Smooth Streaming: 이제 Microsoft Smooth Streaming 파일을 생성할 수 있습니다.	2014년 10월 1일
새 기능	선택 캡션: 이제 트랜스코딩된 출력에 선택 캡션을 포함시킬 수 있습니다.	2014년 6월 18일
새 기능	오디오 프로필: 이제 AAC-LC, HE-AAC, HE-AACv2를 포함한 다양한 오디오 압축 구조를 사용하도록 Elastic Transcoder를 구성할 수 있습니다.	2013년 12월 18일
새 기능	클립 생성: Elastic Transcoder를 사용하여 소스 미디어에서 콘텐츠의 부분 발췌, 즉 "클립"을 생성할 수 있습니다. 이 기능을 사용하여 소스 미디어의 일부만 트랜스코딩할 수 있습니다.	2013년 11월 8일
새 기능	새로운 오디오 파라미터	2013년 10월 9일
새 기능	새 작업 프레임 속도 파라미터	2013년 7월 8일
새 기능	새 작업 파라미터입니다.	2013년 5월 16일
최초 릴리스	이 설명서는 Amazon Elastic Transcoder 개발자 안내서의 최초 릴리스입니다.	2013년 1월 28일

AWS 용어집

비어 있음

placeholder 이 페이지는 AWS 일반 참조의 AWS 용어집으로 리디렉션됩니다.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.