



開発者ガイド

Amazon Chime SDK



Amazon Chime SDK: 開発者ガイド

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標とトレードドレスは、Amazon 以外の製品またはサービスとの関連において、顧客に混乱を招いたり、Amazon の名誉または信用を毀損するような方法で使用することはできません。Amazon が所有しない他の商標はすべてそれぞれの所有者に帰属します。所有者は必ずしも Amazon との提携や関連があるわけではありません。また、Amazon の支援を受けているとはかぎりません。

Table of Contents

Amazon Chime SDK とは	1
料金	1
リソース	1
Amazon Chime SDK を使用する	3
Amazon Chime SDK の前提条件	3
Amazon Chime SDK の概念	4
Amazon Chime SDK アーキテクチャ	4
Amazon Chime SDK サービスクォータ	6
Amazon Chime SDK のシステム要件	7
利用できるリージョン	9
コンソールリージョン	10
分析リージョンを呼び出す	11
会議リージョン	11
メディアパイプラインリージョン	13
メッセージングリージョン	15
PSTN リージョン	16
クライアントライブラリと統合する	17
SIP 統合	18
Amazon Chime SDK イベント通知	20
への通知の送信 EventBridge	21
Amazon SQS および Amazon SNS への通知の送信	21
Amazon Chime SDK に Amazon SQS と Amazon SNS へのアクセスを許可する	21
Amazon Chime 名前空間からの移行	25
エンドポイント、名前空間、CLI コマンド	25
各サービスの移行ヘルプ	26
API マッピング	26
Amazon Chime SDK ミーティングの使用	35
Amazon Chime SDK ミーティング名前空間への移行	35
移行すべき理由	36
移行する前に	36
名前空間の相違点	38
会議リージョンの使用	41
コントロールリージョンの選択	42
メディアリージョンの選択	43

最も近いメディアリージョンを見つける	44
最も近い AWS GovCloud (米国) メディアリージョンの検索	44
JavaScript 例	44
リージョンステータスを確認する	46
会議を作成する	47
会議機能の選択	49
Audio の使用。EchoReduction	49
ビデオの使用。MaxResolution	49
コンテンツの使用。MaxResolution	50
参加者の使用。MaxCount	51
クライアントアプリでの会議機能の使用	51
Amazon Chime SDK が WebRTC メディアを使用する方法	53
音声	53
動画	53
コンテンツ共有	54
データメッセージ	55
動画コーデックの設定	55
動画コーデックの環境設定	56
ネットワーク構成	57
メディアとシグナリング向けの設定	58
Amazon Voice Focus の設定	58
エコーリダクションの設定	59
背景の置換とぼかしの設定	59
ブラウザコンテンツセキュリティポリシーの設定	59
AppKeys と TenantIDs の使用	60
会議イベント	64
Amazon CloudWatch メトリクス	78
サービスマトリクス	78
API 使用状況メトリクス	79
Amazon Chime SDK メディアパイプラインを作成する	81
ChimeSdkMediaPipelines 名前空間への移行	83
パイプライン作成の概要	87
メディアキャプチャパイプラインを作成する	88
メディア連結パイプラインを作成する	96
メディアライブコネクタパイプラインを作成する	103
音声と動画を単一のビューに合成する	104

メディアストリームパイプラインを作成する	118
メディアパイプライン用のサービスにリンクされたロールを作成する	136
メディアパイプラインイベントを使用する	139
トランスクリプトを解析する	145
パイプラインを停止するためのベストプラクティス	146
Amazon Chime SDK ライブ文字起こしの使用	146
システムアーキテクチャ	147
請求と使用状況	148
アカウントの設定	148
文字起こしオプションの選択	148
文字起こしの開始と停止	153
文字起こしパラメータ	157
文字起こしイベント	157
文字起こしメッセージ	161
配信例	166
メディアレプリケーションの使用	169
インタラクティブな参加者	170
グローバル参加者	171
セッションライフサイクル	172
Amazon Chime SDK ミーティングのトラブルシューティングとデバッグ	174
システム要件の理解	174
ログ記録とモニタリングのセットアップ	175
セルフトラブルシューティング	177
一般的な問題	179
Amazon Chime SDK メッセージングを使用する	182
Amazon Chime SDK Identity 名前空間への移行	182
移行すべき理由	183
移行する前に	183
名前空間の相違点	184
Amazon Chime SDK Messaging 名前空間への移行	186
移行すべき理由	36
移行する前に	36
名前空間の相違点	38
メッセージングの前提条件	189
メッセージングの概念	190
メッセージングアーキテクチャ	191

メッセージタイプ	192
開始	192
ApplInstance の作成	193
バックエンドサービスから SDK 呼び出しを行う	194
エンドユーザークライアントアプリケーションの認証	196
チャンネルの作成	200
メッセージの送信	201
の使用 ExpirationSettings	201
WebSockets を使用したメッセージの受信	204
アタッチメントの設定	215
システムメッセージについて	216
IAM ロールの例	216
ロールごとの権限について	220
ApplInstanceAdmin	221
ChannelModerator	224
メンバー	228
非メンバー	231
メッセージングデータのストリーミング	234
Elastic チャンnelを使用してライブイベントをホストする	238
前提条件	239
Elastic チャンnelの概念	239
サポートされる追加の機能	240
Elastic チャンnelの作成	241
Elastic チャンnelメンバーの管理	241
Elastic チャンnelメッセージの送信	242
Elastic チャンnelの WebSocket システムメッセージについて	243
Kinesis ストリームを使用してシステムメッセージを受信する	243
デモアプリケーションでの Elastic チャンnelのテスト	244
モバイルプッシュ通知を使用してメッセージを受信する	244
Amazon Pinpoint アプリケーションを作成する	245
サービスロールの作成	245
モバイルデバイスのエンドポイントをアプリケーションインスタンスユーザーとして登録する	247
通知を有効にした状態でチャンネルメッセージを送信する	248
プッシュ通知の受信	249
プッシュ通知エラーのデバッグ	250

フィルタルールを使用してメッセージをフィルタリングする	250
サービスリンクロールの使用	256
データストリーミングでのサービスにリンクされたロールの使用	256
チャンネルフローを使用したメッセージの処理	259
チャンネルプロセッサの設定	261
チャンネルフローの作成	264
チャンネルフローの関連付けと関連付け解除	265
メッセージの送信	265
による自動化による障害アラートの作成 EventBridge	267
ポットをチャンネルエージェントとして使用する	268
Amazon Lex V2 ポットの作成	269
AppInstance ポットのセットアップ	272
のチャンネルメンバーシップ AppInstanceBots	272
へのメッセージの送信 AppInstanceBot	273
Amazon Lex からのメッセージの処理	274
からのレスポンスの処理 AppInstanceBot	274
ルールを使用して Amazon にイベントを送信する EventBridge	277
トラブルシューティング AppInstanceBots	278
メッセージ保持の管理	279
CLI 保持コマンドの例	279
メッセージ保持を有効にする	280
メッセージの復元と削除	280
メッセージングのユーザーインターフェイスコンポーネント	280
クライアントライブラリとの統合	280
での Amazon Chime SDK メッセージングの使用 JavaScript	281
Amazon Chime SDK PSTN Audio サービスの使用	282
Amazon Chime SDK Voice 名前空間への移行	283
移行すべき理由	283
移行する前に	284
名前空間の相違点	285
電話番号、SIP ルール、SIP メディアアプリケーション、および AWS Lambda 機能につい て	286
PSTN Audio サービスのプログラミングモデル	288
AWS Lambda 関数への呼び出しとイベントのルーティング	289
PSTN Audio サービスの通話レグの使用について	294
通話フローの例	297

PSTN Audio サービスの AWS Lambda 関数の構築	299
テレフォニーイベントについて	300
アクションについて	305
AWS Lambda 関数を呼び出すテレフォニーイベント	305
アクションリストによる呼び出しへの応答	330
PSTN Audio サービスでサポートされるアクション	332
SIP ヘッダーの使用	434
通話詳細レコードの使用	438
タイムアウトと再試行	439
デバッグとトラブルシューティング	440
VoiceFocus	451
PSTN Audio サービス用語集	457
呼び出しからのインサイトの生成	463
Amazon Chime SDK 通話分析とは	464
通話分析の用語	465
通話分析を設定する	468
前提条件	468
コンソールを使用して設定を行う	469
API を使用して通話分析の設定を行う	477
設定と Voice Connector を関連付ける	477
通話分析の設定を使用する	478
通話録音を目的としたワークフロー	479
機械学習ベースの分析を目的としたワークフロー	486
通話分析パイプラインを管理する	493
通話分析パイプラインの一時停止と再開を行う	493
通話分析のリソースアクセスロールを使用する	494
通話分析のステータスを理解する	502
Amazon での通話分析パイプラインのモニタリング CloudWatch	504
前提条件	505
コール分析メトリクス	505
CloudWatch パイプラインメトリクスの デイメンション	506
通話分析プロセッサと出力先	506
文字起こしの機能と録音シンクを連携させる	528
Amazon EventBridge 通知の使用	530
Amazon Chime SDK データレイクを作成する	549
Amazon QuickSight ダッシュボードの設定	557

通話分析のデータモデル	563
Glue データカタログのテーブル構造	563
Glue データカタログのテーブル	565
サンプルクエリ	597
Amazon Chime SDK 音声分析を使用する	602
音声分析アーキテクチャ	603
スピーカー検索ワークフローの例	605
音声トーン分析のワークフローの例	608
タスク結果のポーリング	611
通知について	611
データストレージ、オプトアウト、データ保持に関するポリシー	622
音声 API を使用して音声分析を実行する	624
通話分析のサービスクォータ	630
Android 用 Amazon Chime SDK クライアントライブラリを使用する	632
iOS 用の Amazon Chime SDK クライアントライブラリを使用する	633
での Amazon Chime SDK クライアントライブラリの使用 JavaScript	634
Amazon Chime SDK アプリケーションのコンポーネント	634
主要なコンセプト	635
サービスアーキテクチャ	636
ウェブアプリケーションのアーキテクチャ	637
サーバーアプリケーションのアーキテクチャ	637
Amazon Chime SDK メディアコントロールプレーン	638
Amazon Chime SDK メディアデータプレーン	638
ウェブアプリケーションコンポーネントのアーキテクチャ	638
サーバーアプリケーションを構築する	640
IAM ユーザーまたはロールを作成する	640
APIs を呼び出すように AWS SDK を設定する	641
会議を作成する	641
参加者を作成する	642
クライアントにレスポンスを送信する	643
クライアントアプリケーションをビルドする	643
背景フィルタをクライアントアプリケーションに統合する	643
背景フィルタの使用について	644
コンテンツセキュリティポリシーを使用する	646
アプリケーションを背景フィルタに追加する	649
背景フィルタの例	656

Windows 用の Amazon Chime SDK クライアントライブラリの使用	659
よくある質問	660
会議FAQs	660
参加者	660
セキュリティと暗号化	662
オーディオ/ビデオ	662
ライブ文字起こし	665
Service Quotas	667
名前空間の移行	667
モニタリング	668
ログ記録	669
エラーメッセージ	670
メディアパイプラインFAQs	671
PSTN オーディオFAQs	672
ドキュメント履歴	674
.....	dclxxxiv

Amazon Chime SDK とは

Amazon Chime SDK は、ウェブまたはモバイルアプリケーションにメッセージング、オーディオ、ビデオ、および画面共有機能をすばやく追加するために使用できるリアルタイム通信コンポーネントのセットです。

Amazon Chime SDK を使用することで、音声と動画を送受信してコンテンツを共有できるリアルタイムメディアアプリケーションを構築できます。Amazon Chime SDK API アクションの詳細については、「[Amazon Chime SDK API リファレンス](#)」を参照してください。

料金

Amazon Chime SDK は前払い料金なしで pay-for-use 料金を提供します。利用可能なメディアモダリティ (音声、動画、画面共有) の一部またはすべてを単一料金で実装することを選択できます。メッセージング、メディアパイプライン、音声エンハンスメント、PSTN オーディオ機能も pay-for-use 料金で利用できます。詳細については、「[Amazon Chime SDK の料金](#)」を参照してください。

リソース

このサービスを利用する際に役立つ関連リソースは次のとおりです。

- [クラスとワークショップ](#) – AWS スキルを磨き、実践的な経験を積むのに役立つセルフペースラボに加えて、ロールベースのコースや専門コースへのリンクです。
- [AWS デベロッパーセンター](#) – チュートリアルを詳しく調べたり、ツールをダウンロードしたり、デ AWS ベロッパーイベントについて学習したりします。
- [AWS デベロッパーツール](#) – AWS アプリケーションを開発および管理するためのデベロッパーツール、SDKs、IDE ツールキット、コマンドラインツールへのリンク。
- [入門リソースセンター](#) – AWS アカウントのセットアップ、AWS コミュニティへの参加、最初のアプリケーションの起動方法について説明します。
- [ハンズオンチュートリアル](#) – step-by-step チュートリアルに従って、で最初のアプリケーションを起動します AWS。
- [AWS ホワイトペーパー](#) – ソリューションアーキテクトや他の技術専門家が AWS 作成したアーキテクチャ、セキュリティ、経済などのトピックを網羅した、技術 AWS ホワイトペーパーの包括的なリストへのリンクです。

- [AWS Support センター](#) – AWS Support ケースを作成および管理するためのハブ。フォーラム、技術上のFAQs、サービスヘルスステータス、など、その他の役立つリソースへのリンクも含まれています AWS Trusted Advisor。
- [AWS Support](#) – クラウドでのアプリケーションの構築と実行に役立つ AWS Support one-on-one、高速応答サポートチャネルに関する情報のプライマリウェブページ。
- [お問い合わせ](#) - AWS の請求、アカウント、イベント、不正使用、その他の問題などに関するお問い合わせの受付窓口です。
- [AWS サイト規約](#) – 当社の著作権と商標、お客様のアカウント、ライセンス、サイトアクセス、およびその他のトピックに関する詳細情報。

Amazon Chime SDK を使用する

Amazon Chime SDK は、音声と動画を送受信し、コンテンツを共有できるリアルタイムなメディアアプリケーションを構築するために使用します。Amazon Chime SDK は Amazon Chime 管理者アカウントとは独立して動作し、Amazon Chime でホストされる会議には影響を及ぼしません。代わりに、Amazon Chime SDK には、独自の会議アプリケーションの構築に使用するビルダーツールが用意されています。

トピック

- [Amazon Chime SDK の前提条件](#)
- [Amazon Chime SDK の概念](#)
- [Amazon Chime SDK アーキテクチャ](#)
- [Amazon Chime SDK サービスクォータ](#)
- [Amazon Chime SDK のシステム要件](#)
- [利用できるリージョン](#)
- [クライアントライブラリと統合する](#)
- [Amazon Chime SDK Voice Connector を使用した SIP 統合](#)
- [Amazon Chime SDK イベント通知](#)
- [Amazon Chime 名前空間からの移行](#)

Amazon Chime SDK の前提条件

Amazon Chime SDK を使用するには、以下が必要です。

- プログラムする機能。
- AWS アカウント。
- AWS マネージド AmazonChimeSDK ポリシーなど、Amazon Chime SDK で使用される Amazon Chime API アクションへのアクセス許可を付与するポリシーを持つ IAM ロール。詳細については、「Amazon Chime SDK 管理者ガイド」の「[Amazon Chime で IAM が機能するしくみ](#)」および「[Amazon Chime SDK アクションへのアクセスをユーザーに許可する](#)」を参照してください。
- 大部分のユースケースでは、次のものも必要です。

- サーバーアプリケーション — 会議および参加者のリソースを管理し、それらのリソースをクライアントアプリケーションに提供します。サーバーアプリケーションは AWS アカウントで作成され、前述の IAM ロールにアクセスできる必要があります。
- クライアントアプリケーション — サーバーアプリケーションから会議および参加者の情報を受信し、その情報を使用してメディア接続を行います。

Amazon Chime SDK の概念

Amazon Chime SDK の使用方法を理解するうえで重要な用語と概念を、以下に示します。

会議

一意の MeetingId で識別されるエフェメラルリソース。MeetingId は、アクティブな会議をホストするメディアサービスのグループに配置されます。

メディアサービスグループ

アクティブな会議を主催するメディアサービスのグループ。

メディアの配置

メディアサービスグループを表すローカル化された一連の URL。参加者はクライアントを使用してメディアサービスグループに接続し、リアルタイムの音声や動画を送受信したり、画面を共有したりします。

参加者

一意の AttendeeId で識別される会議参加者。参加者は、Amazon Chime SDK クライアントライブラリで構築されたクライアントアプリケーションを使用して、自由に会議に参加したり退席したりできます。

参加トークン

各参加者に割り当てられる一意のトークン。参加者は参加トークンを使用してメディアサービスグループの認証を行います。

Amazon Chime SDK アーキテクチャ

以下のリストでは、Amazon Chime SDK アーキテクチャのさまざまなコンポーネントがどのように連携し、会議と参加者、音声、動画、コンテンツ共有をサポートしているのかについて説明します。

会議と参加者

サーバーアプリケーションで Amazon Chime SDK ミーティングが作成されると、ミーティングは地域固有のメディアサービスに割り当てられます。サービスのホストには、参加者のクライアント間でリアルタイムメディアを安全に転送する責任があります。作成された各参加者には、一意の参加トークン (不透明な秘密鍵) が割り当てられます。サーバーアプリケーションでは、参加者に代わって、会議への参加を許可されたクライアントにこのトークンを安全に転送する必要があります。各クライアントでは参加トークンを使用してメディアサービスグループとの認証を行います。クライアントは、Secure WebSockets と Datagram Transport Layer Security (DTLS) の組み合わせを使用して、メディアサービスグループに安全に通知し、メディアサービスグループを介して他の参加者との間でメディアを送受信します。

音声

メディアサービスでは各参加者の音声をミックスし、ミックスから各自の音声を差し引いた後、ミックスを各受信者に送信します。Amazon Chime SDK では、デバイスとブラウザでサポートされている最高レート (最大 48kHz) で音声をサンプリングします。音声のエンコードには Opus コーデックを使用します。デフォルトのビットレートは 32kbps で、ステレオでは最大 128kbps、モノラルでは最大 64kbps まで上げることができます。

動画

メディアサービスは、パブリッシュおよびサブスクライブモデルを使用する選択的転送ユニット (SFU) として機能します。各参加者は 1 つの動画ソースを公開でき、1 回の会議につき最大 25 本の動画を同時に公開できます。の Amazon Chime SDK クライアントライブラリ JavaScript は、サイマルキャストなしで 1 秒あたり 30 フレームで最大 1280 x 720 のビデオ解像度をサポートし、サイマルキャストで 1 秒あたり 15 フレームをサポートします。[iOS](#)、[Android](#)、[Windows](#) 用の Amazon Chime SDK クライアントライブラリでは、最大 1280x720 および毎秒 30 フレームの動画解像度がサポートされていますが、実際のフレームレートと解像度は Amazon Chime SDK によって自動的に管理されます。

動画サイマルキャストがアクティブな場合、各動画ストリームは 2 つの異なる解像度とビットレートで送信されます。帯域幅に制約があるクライアントは、低いビットレートのストリームを自動的にサブスクライブします。動画のエンコードとデコードでは、パフォーマンスを向上させるためにハードウェアアクセラレーション (利用可能な場合) を使用します。

データメッセージ

会議の参加者は、音声および動画コンテンツに加えて、それぞれ最大 2 KB のリアルタイムなデータメッセージを相互に送信できます。メッセージを使用して、ホワイトボード、チャット、

リアルタイムの絵文字リアクション、アプリケーション固有のフロアコントロールシグナリングなどのカスタム会議機能を実装できます。

コンテンツ共有

クライアントアプリケーションでは、スクリーンキャプチャやメディアファイルなどの音声および動画コンテンツを共有できます。コンテンツ共有では、毎秒 15 フレームで最大 1280x720 の録画済みコンテンツ動画と、64 kbps で最大 48 kHz の音声をサポートされます。コンテンツ共有用のスクリーンキャプチャは 1 秒あたり最大 15 フレームまでサポートされていますが、デバイスとブラウザの処理能力によって制限される場合があります。

Amazon Chime SDK サービスクォータ


Note

サービスクォータは API エンドポイントごとに設定されます。サービスクォータの引き上げをリクエストする際は、必ず、アプリケーションで使用するすべての API エンドポイントで引き上げをリクエストしてください。

この表には、Amazon Chime SDK ミーティングで利用できるリソースとクォータが一覧表示されています。

リソース	クォータ	調整可能
アクティブな会議	250	はい
会議ごとの参加者	250	いいえ
会議ごとの音声ストリーム	250	いいえ
会議ごとに公開される動画ストリーム	25	はい (最大 250)
参加者ごとにサブスクライブされる動画ストリーム	25	いいえ
会議ごとのコンテンツ共有	2	いいえ

リソース	クォータ	調整可能
プライマリ会議ごとのレプリカ会議	4	はい (最大 40)
会議ごとのアクティブなメディアキャプチャパイプライン	1	いいえ
アカウントごとのアクティブなメディアキャプチャパイプライン	us-east-1 エンドポイントでは 100、その他のエンドポイントでは 10	はい
API レート	1 秒あたり 10 回のリクエスト (RPS)、バーストは 20 RPS。	はい、ただし間接的

 **Note**

アクティブな会議のクォータを増やすと、API レート制限が引き上げられます。

Amazon Chime SDK のシステム要件

Amazon Chime SDK で作成されたアプリケーションには、以下のシステム要件が適用されます。

サポートされているブラウザ、用の Amazon Chime SDK クライアントライブラリ JavaScript

オペレーティングシステム	ブラウザ	サポートバージョン	メモ
Windows	Mozilla Firefox	75 以降	
	Google Chrome	78 以降	
	Chromium ベースの Edge	79 以降	

オペレーティングシステム	ブラウザ	サポートバージョン	メモ
	Chromium ベースの Electron	7 以降	Chrome バージョン 78 以降。
	Opera	66 以降	
macOS	Mozilla Firefox	75 以降	
	Google Chrome	78 以降	
	Chromium ベースの Edge	79 以降	
	Chromium ベースの Electron		
	Safari	13 以降	
	Opera	66 以降	
iOS	Mozilla Firefox	10 以降	音声と動画のみ。コンテンツ共有はできません。
	Google Chrome	78 以降	音声と動画のみ。コンテンツ共有はできません。
	Safari	13 以降	音声と動画のみ。コンテンツ共有はできません。
	WKWebView	14.3 以降	音声と動画のみ。コンテンツ共有はできません。

オペレーティングシステム	ブラウザ	サポートバージョン	メモ
Android	Google Chrome	10 以降	音声と動画のみ。コンテンツ共有はできません。
	Samsung	12 以降	音声と動画のみ。コンテンツ共有はできません。
	Chromium WebView	5 以降	音声と動画のみ。コンテンツ共有はできません。
Ubuntu LTS 16.04 以降	Google Chrome	78 以降	

iOS 用 Amazon Chime SDK クライアントライブラリ

- iOS バージョン 13 以降

Android 用 Amazon Chime SDK クライアントライブラリ

- Android OS バージョン 5 以降、ARM および ARM64 アーキテクチャ

利用できるリージョン

次の表に、Amazon Chime SDK サービスの特徴と、各サービスを提供する AWS リージョンを示します。

Note

アスタリスク (*) が付いたリージョンは、AWS アカウントで有効にする必要があります。は、デフォルトでそれらのリージョンを AWS ブロックします。リージョンの有効化の詳細

については、[「アカウント管理リファレンス」の「アカウントが 使用できる AWS リージョンを指定する」](#)を参照してください。AWS

トピック

- [コンソールリージョン](#)
- [分析リージョンを呼び出す](#)
- [会議リージョン](#)
- [メディアパイプラインリージョン](#)
- [メッセージングリージョン](#)
- [PSTN リージョン](#)

コンソールリージョン

リソースを設定し、Amazon Chime SDK サービスについて詳しく知るために、Amazon Chime SDK コンソールを使用します。

AWS リージョン	コンソール
アジアパシフィック (ソウル)	あり
アジアパシフィック (シンガポール)	
アジアパシフィック (シドニー)	はい
アジアパシフィック (東京) (ap-northeast-1)	はい
カナダ (中部) (ca-central-1)	はい
ヨーロッパ (フランクフルト) (eu-central-1)	はい
欧州 (アイルランド) (eu-west-1)	はい
ヨーロッパ (ロンドン) (eu-west-2)	はい
米国東部 (バージニア北部) (us-east-1)	はい
米国西部 (オレゴン) (us-west-2)	はい

分析リージョンを呼び出す

次の表に、分析、文字起こし、通話録音に使用できる AWS リージョンを示します。

AWS リージョン	音声分析	文字起こし	通話録音
米国東部 (バージニア北部) (us-east-1)	はい	はい	はい
米国西部 (オレゴン) (us-west-2)	はい	はい	はい
ヨーロッパ (フランクフルト) (eu-central-1)	いいえ	はい	はい

会議リージョン

Amazon Chime SDK ミーティングには、コントロールリージョンとメディアリージョンがあります。コントロールリージョンでは、会議の作成、更新、削除に使用する API エンドポイントが提供されます。コントロールリージョンでは [会議イベント](#) の受信と処理も行います。

メディアリージョンでは実際の会議を主催し、クライアントでメディアリージョンに接続します。 [CreateMeeting](#) API を呼び出すときにメディアリージョンを指定します。

コントロールリージョンは、同じ AWS パーティション内の任意のメディアリージョンで会議を作成できます。ただし、会議の作成に使用したコントロールリージョン内でのみ、会議を更新できます。

コントロールおよびメディアリージョンの選択については、[会議リージョンの使用](#) を参照してください。

次の表には、コントロール、メディア、またはその両方を提供するリージョンが一覧表示されています。

AWS リージョン	会議コントロール	会議メディア
アフリカ (ケープタウン) (af-south-1)*	はい**	はい

AWS リージョン	会議コントロール	会議メディア
アジアパシフィック (ムンバイ) (ap-south-1)	はい	はい
アジアパシフィック (ソウル) (ap-northeast-2)	はい	はい
アジアパシフィック (シンガポール) (ap-southeast-1)	はい	はい
アジアパシフィック (シドニー) (ap-southeast-2)	はい	はい
アジアパシフィック (東京) (ap-northeast-1)	はい	はい
カナダ (中部) (ca-central-1)	はい	はい
ヨーロッパ (フランクフルト) (eu-central-1)	はい	はい
欧州 (アイルランド) (eu-west-1)		はい
ヨーロッパ (ロンドン) (eu-west-2)	はい	はい
欧州 (ミラノ) (eu-south-1)*		はい
欧州 (パリ) (eu-west-3)		はい
欧州 (ストックホルム) (eu-north-1)		はい
イスラエル (テルアビブ) (il-central-1)*	Yes**	はい
南米 (サンパウロ) (sa-east-1)		はい

AWS リージョン	会議コントロール	会議メディア
米国東部 (オハイオ) (us-east-2)		はい
米国東部 (バージニア北部) (us-east-1)	はい	はい
米国西部 (北カリフォルニア) (us-west-1)		はい
米国西部 (オレゴン) (us-west-2)	はい	はい
AWS GovCloud (米国東部) (us-gov-east-1)	はい	はい
AWS GovCloud (米国西部) (us-gov-west-1)	はい	はい

* AWS アカウントでこれらのリージョンを有効にする必要があります。詳細については、「AWS 全般リファレンス」の「[リージョンを有効にする](#)」を参照してください。

**このリージョンで会議コントロールを使用する会議は、このリージョンでのみメディアをホストできます。

Note

AWS GovCloud (米国) リージョンで会議を作成するには、でコントロールリージョンを使用する必要があります GovCloud。また、のコントロールリージョン GovCloud は、AWS GovCloud (米国) リージョンでのみ会議を行うことができます。

メディアパイプラインリージョン

Amazon Chime SDK メディアパイプラインには、コントロールリージョンとメディアリージョンがあります。コントロールリージョンでは、メディアパイプラインの作成と削除に使用するメディアパ

イプライン API エンドポイントが提供されます。コントロールリージョンを使用し、[メディアパイプラインイベント](#)を受信して処理することもできます。

メディアリージョンではメディアパイプラインを実行し、システムで会議と同一のメディアリージョンを自動的に選択します。

コントロールリージョンを使用すると、どのデータリージョンでもメディアパイプラインを作成できます。メディアパイプラインでは、どの会議メディアリージョンの会議にも参加できます。

AWS リージョン	コントロール	メディア
アフリカ (ケープタウン) (af-south-1)*		はい
アジアパシフィック (ムンバイ) (ap-south-1)	はい	はい
アジアパシフィック (ソウル) (ap-northeast-2)	はい	はい
アジアパシフィック (シンガポール) (ap-southeast-1)	はい	はい
アジアパシフィック (シドニー) (ap-southeast-2)	はい	はい
アジアパシフィック (東京) (ap-northeast-1)	はい	はい
カナダ (中部) (ca-central-1)	はい	はい
ヨーロッパ (フランクフルト) (eu-central-1)	はい	はい
欧州 (アイルランド) (eu-west-1)		はい
ヨーロッパ (ロンドン) (eu-west-2)	はい	はい

AWS リージョン	コントロール	メディア
欧州 (ミラノ) (eu-south-1)*		はい
欧州 (パリ) (eu-west-3)		はい
欧州 (ストックホルム) (eu-north-1)		はい
南米 (サンパウロ) (sa-east-1)		はい
米国東部 (オハイオ) (us-east-2)		はい
米国東部 (バージニア北部) (us-east-1)	はい	はい
米国西部 (北カリフォルニア) (us-west-1)		はい
米国西部 (オレゴン) (us-west-2)	はい	はい

* AWS アカウントでこれらのリージョンを有効にする必要があります。詳細については、「AWS 全般リファレンス」の「[リージョンを有効にする](#)」を参照してください。

メッセージングリージョン

Amazon Chime SDK メッセージングには、コントロールリージョンとデータリージョンがあります。コントロールリージョンではメッセージング API エンドポイントが公開され、データリージョンではメッセージが保存されます。Amazon Kinesis を使用してメッセージングデータ、またはチャネルフローの AWS Lambda 関数をストリーミングする場合、それらはコントロールリージョンに存在する必要があります。

AWS リージョン	コントロール	[データ]
ヨーロッパ (フランクフルト) (eu-central-1)	はい	はい

AWS リージョン	コントロール	[データ]
米国東部 (バージニア北部) (us-east-1)	はい	はい

PSTN リージョン

Amazon Chime SDK SIP (セッション開始プロトコル) の機能には、API リージョンとメディアリージョン、および PSTN リージョンがあります。API リージョンでは SIP 機能を作成および設定するための API エンドポイントが提供されます。メディアリージョンには Amazon Chime SDK Voice Connector と SIP メディアアプリケーションが含まれます。PSTN リージョンを使用すると、ユーザーはオンプレミスの電話システムを公衆電話回線に接続できます。さらに、PSTN リージョンは電話番号のプロビジョニングと管理をサポートしています。

AWS リージョン	API	メディア	PSTN
アジアパシフィック (ソウル) (ap-north-east-2)	はい	はい	
アジアパシフィック (シンガポール) (ap-southeast-1)	はい	はい	
アジアパシフィック (シドニー) (ap-south-east-2)	はい	はい	
アジアパシフィック (東京) (ap-north-east-1)	はい	はい	
カナダ (中部) (ca-central-1)	はい	はい	
ヨーロッパ (フランクフルト) (eu-central-1)	はい	はい	

AWS リージョン	API	メディア	PSTN
欧州 (アイルランド) (eu-west-1)	はい	はい	
ヨーロッパ (ロンドン) (eu-west-2)	はい	はい	
米国東部 (バージニア 北部) (us-east-1)	はい	はい	はい*
米国西部 (オレゴン) (us-west-2)	はい	はい	はい*

*特定の AWS リージョンでの電話番号の可用性については、[Amazon Chime SDK の料金](#) ページを参照してください。

クライアントライブラリと統合する

Amazon Chime SDK を使用してリアルタイム会議クライアントを構築する前に、クライアントアプリケーションを Amazon Chime SDK クライアントライブラリと統合する必要があります。以下のクライアントライブラリを使用できます。

- [Android 用 Amazon Chime SDK クライアントライブラリ](#) – サポートされている Android デバイスで Amazon Chime SDK アプリケーションを構築するのに役立つ Kotlin ライブラリ。
- [C++ 用 Amazon Chime SDK シグナリングクライアントライブラリ](#) – 組み込みデバイスの Amazon Chime SDK 会議へのシグナリング接続をセットアップするのに役立つ C++ ライブラリ。
- [iOS 用 Amazon Chime SDK クライアントライブラリ](#) – サポートされている iOS デバイスで Amazon Chime SDK アプリケーションを構築するのに役立つ Swift ライブラリ。
- [JavaScript 用 Amazon Chime SDK クライアントライブラリ \(NPM\)](#) – WebRTC 対応ブラウザで Amazon Chime SDK アプリケーションを構築するのに役立つ、TypeScript 型定義を含む JavaScript ライブラリ。
- [Windows 用 Amazon Chime SDK クライアントライブラリ](#)。サポートされているデバイスで Amazon Chime SDK アプリケーションを構築するのに役立つ C++ ライブラリ。

クライアントアプリケーションを Amazon Chime SDK と統合する方法について知るには、クライアントライブラリ README.md ファイルのアクションを参照してください。アプリケーション用の特定のメディアコンポーネントを構築する方法について知るには、デモを使用してください。

Amazon Chime SDK Voice Connector を使用した SIP 統合

SIP 対応の音声インフラストラクチャを Amazon Chime SDK Voice Connector と統合して、SIP 音声通話を行うことができます。us-east-1 または us-east-2 リージョンを使用する必要があります。IP Private Branch Exchange (PBX)、Session Border Controller (SBC)、あるいは Session Initiation Protocol (SIP) をサポートしているインターネットアクセスのある他の音声インフラストラクチャを用意する必要があります。詳細については、「Amazon Chime SDK 管理者ガイド」の「[開始する前に](#)」を参照してください。

音声インフラストラクチャを統合するには

1. AWS アカウントで Amazon Chime SDK Voice Connector を作成します。詳細については、「Amazon Chime SDK 管理者ガイド」の「[Amazon Chime Voice Connector の作成](#)」を参照してください。
2. Amazon Chime SDK Voice Connector 設定を編集して、音声インフラストラクチャから AWS への呼び出しを許可します。詳細については、「Amazon Chime SDK 管理者ガイド」の「[Amazon Chime Voice Connector 設定の編集](#)」を参照してください。
 - a. [終了設定] で、[有効] を選択します。
 - b. [許可リスト] で、[新規] を選択します。
 - c. 内部 SIP インフラストラクチャの IP アドレスの CIDR 表記を入力します。これにより、ご使用のインフラストラクチャで Amazon Chime SDK Voice Connector にアクセスできるようになります。例えば、IP アドレス 10.24.34.0 からのトラフィックを許可するには、CIDR 表記 10.24.34.0/32 を許可リストに登録します。
 - d. [追加] を選択します。
 - e. [Calling plan (通話プラン)] で、通話プランに追加する 1 つ以上の国を選択します。
 - f. 必要に応じて他の設定を編集し、[保存] を選択します。
3. Amazon Chime SDK コンソールで、[音声コネクタ] の下に Amazon Chime SDK Voice Connector の [アウトバウンドホスト名] が表示されます。例:
abcdefghijklmno3pqr4.voiceconnector.chime.aws
4. Amazon Chime SDK を使用して会議に参加するには、SIP URI を使用して Amazon Chime SDK Voice Connector のアウトバウンドホスト名に SIP リクエストを出します。SIP URI では電話番

号 **+17035550122** を使用します。transport パラメータを設定して、TLS プロトコルを使用します。最後に、[CreateAttendee](#) API アクションを呼び出して生成された一意の結合トークンを使用します。詳細については、次の例を参照してください。

Example 例: SIP リクエスト

次の例は、Amazon Chime SDK Voice Connector に SIP リクエストを出すために使用する SIP URI のコンテンツを示しています。

```
sip:+17035550122@abcdefghijklmno3pqr4.voiceconnector.chime.aws;transport=tls;X-chime-join-token=join-token
```

以下の例は、Amazon Chime SDK ミーティングに参加するための SIP INVITE メッセージのサンプルを示しています。

```
INVITE sip:
+17035550122@abcdefghijklmno3pqr4.voiceconnector.chime.aws;transport=tls;X-chime-
join-token=join-token SIP/2.0
Via: SIP/2.0/TLS IPaddress:12345;rport;branch=branch;alias
Max-Forwards: 70
From: sip:+12065550100@IPaddress;tag=tag
To: sip:+17035550122@abcdefghijklmno3pqr4.voiceconnector.chime.aws;X-chime-join-
token=join-token
Contact: <sip:+12065550100@IPaddress:54321;transport=TLS;ob>
Call-ID: a1234567-89b0-1c2d-e34f-5gh678j9k2lm
CSeq: 6214 INVITE
Allow: PRACK, INVITE, ACK, BYE, CANCEL, UPDATE, INFO, SUBSCRIBE, NOTIFY, REFER,
MESSAGE, OPTIONS
Supported: replaces, 100rel, timer, norefersub
Session-Expires: 1800
Min-SE: 90
Content-Type: application/sdp
Content-Length: 991

v=0
o=- 3775321410 3775321410 IN IP4 IPaddress
s=pjmedia
b=AS:117
t=0 0
a=X-nat:0
m=audio 4000 RTP/SAVP 0 3 8 9 125 101
```

```
c=IN IP4 IPaddress
b=TIAS:96000
a=rtcp:4001 IN IP4 IPaddress
a=sendrecv
a=rtpmap:0 PCMU/8000
a=rtpmap:3 GSM/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:9 G722/8000
a=rtpmap:125 opus/48000/2
a=fmtp:125 useinbandfec=1
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=crypto:1 AEAD_AES_256_GCM inline:EXAMPLE
a=crypto:2 AEAD_AES_256_GCM_8 inline:EXAMPLE
a=crypto:3 AES_256_CM_HMAC_SHA1_80 inline:EXAMPLE
a=crypto:4 AES_256_CM_HMAC_SHA1_32 inline:EXAMPLE
a=crypto:5 AES_CM_128_HMAC_SHA1_80 inline:EXAMPLE
a=crypto:6 AES_CM_128_HMAC_SHA1_32 inline:EXAMPLE
```

Note

Amazon Chime SDK では E.164 形式の電話番号のみを認識します。From ヘッダーに E.164 の電話番号が含まれていることを確認してください。

Amazon Chime SDK イベント通知

Amazon Chime SDK は、Amazon、Amazon Simple Queue Service (SQS) EventBridge、および Amazon Simple Notification Service (SNS) への会議イベント通知の送信をサポートしています。

Note

デフォルトの Amazon Chime SDK ミーティング名前空間は、ChimeSDKMeetings エンドポイントを使用します。従来の Chime 名前空間は、1 つのエンドポイントを使用します。名前空間とエンドポイントの詳細については、このガイドの前半にある [Amazon Chime SDK ミーティング名前空間への移行](#)「」を参照してください。

への通知の送信 EventBridge

Amazon Chime SDK イベント通知を に送信できます EventBridge。で Amazon Chime SDK を使用する方法の詳細については EventBridge、[「Amazon Chime SDK 管理者ガイド」の「で EventBridge Amazon Chime SDK を自動化する」](#)を参照してください。の詳細については EventBridge、[「Amazon ユーザーガイド EventBridge」](#)を参照してください。

Amazon SQS および Amazon SNS への通知の送信

「Amazon Chime SDK [CreateMeeting](#) API リファレンス」の「API」を使用して、Amazon Chime SDK ミーティングイベント通知を、会議ごとに 1 つの Amazon SQS キューと 1 つの Amazon SNS トピックに送信できます。これにより、通知のレイテンシーを低減できます。Amazon SQS の詳細については、[「Amazon Simple Queue Service Developer Guide」](#)を参照してください。Amazon SNS の詳細については、[Amazon Simple Notification Service デベロッパーガイド](#)を参照してください。

Amazon SQS および Amazon SNS に送信される通知には、Amazon Chime SDK が に送信する通知と同じ情報が含まれています EventBridge。Amazon Chime SDK は、会議の作成に使用された API リージョンのキューとトピックへの会議イベント通知の送信をサポートします。イベント通知は、発生順とは異なる順序で配信される場合があります。

Amazon Chime SDK に Amazon SQS と Amazon SNS へのアクセスを許可する

Amazon Chime SDK が Amazon SQS キューまたは Amazon SNS トピックを介して通知を送信できるようにするには、キューまたはトピックの Amazon リソースネーム (ARN) にメッセージを発行するためのアクセス許可を Amazon Chime SDK に付与する必要があります。これを行うには、Amazon Chime SDK に適切なアクセス許可を付与する AWS Identity and Access Management (IAM) ポリシーをキューまたはトピックにアタッチします。詳細については、「Amazon Simple Queue Service 開発者ガイド」の「[Amazon SQS での Identity and Access Management](#)」および「Amazon Simple Notification Service デベロッパーガイド」の「[Amazon SNS アクセスコントロールのケース例](#)」を参照してください。

Note

Amazon SQS キューまたは Amazon SNS トピックは、Amazon Chime SDK API エンドポイントと同じ AWS リージョンを使用する必要があります。

Example Amazon Chime SDK が Amazon SQS キューにイベントを発行することを許可する

次の IAM ポリシーの例は、指定した Amazon SQS キューに会議イベント通知を発行するためのアクセス許可を Amazon Chime SDK に付与します。aws:SourceArn および aws:SourceAccount の条件ステートメントに注意してください。これらは「[混乱した代理](#)」の潜在的な問題に対処していません。

Note

- 以下のポリシーを作成する際に aws:SourceArn または aws:SourceAccount を使用できます。両方を使用する必要はありません。
- これらの例では、ChimeSDKMeetings 名前空間および対応するエンドポイントを使用しています。Chime 名前空間を使用する場合は、chime.amazonaws.com エンドポイントを使用する必要があります。

```
{
  "Version": "2008-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "example-statement-ID",
      "Effect": "Allow",
      "Principal": {
        "Service": "meetings.chime.amazonaws.com"
      },
      "Action": [
        "sqs:SendMessage",
        "sqs:GetQueueUrl"
      ],
      "Resource": "arn:aws:sqs:eu-central-1:111122223333:queueName",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:partition:chime::111122223333:*"
        },
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        }
      }
    }
  ]
}
```



```
}
```

この例は、Amazon Chime SDK が会議イベント通知を SNS トピックに送信することを許可する Amazon SNS ポリシーを示しています。

```
{
  "Version": "2008-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "allow-chime-sdk-access-statement-id",
      "Effect": "Allow",
      "Principal": {
        "Service": "meetings.chime.amazonaws.com"
      },
      "Action": [
        "SNS:Publish"
      ],
      "Resource": "arn:aws:sns:eu-central-1:111122223333:topicName",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:partition:chime::111122223333:*"
        }
      },
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      }
    }
  ]
}
```

Amazon SQS キューでサーバー側の暗号化 (SSE) が有効になっている場合は、追加のステップを実行する必要があります。Amazon Chime SDK にキューに追加されたデータの暗号化に必要な AWS KMS アクションへのアクセス許可を付与する IAM ポリシーを、関連付けられた AWS KMS キーにアタッチします。

```
{
  "Version": "2012-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "example-statement-ID",
```

```
    "Effect": "Allow",
    "Principal": {
      "Service": "meetings.chime.amazonaws.com"
    },
    "Action": [
      "kms:GenerateDataKey",
      "kms:Decrypt"
    ],
    "Resource": "*"
  }
]
```

Example Amazon Chime SDK が Amazon SNS トピックにイベントを発行することを許可する

次の IAM ポリシーの例は、指定した Amazon SNS トピックに会議イベント通知を発行するためのアクセス許可を Amazon Chime SDK に付与します。

```
{
  "Version": "2008-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "allow-chime-sdk-access-statement-id",
      "Effect": "Allow",
      "Principal": {
        "Service": "meetings.chime.amazonaws.com"
      },
      "Action": [
        "SNS:Publish"
      ],
      "Resource": "arn:aws:sns:eu-central-1:111122223333:topicName",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:partition:chime::111122223333:*"
        }
      },
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      }
    }
  ]
}
```

Amazon Chime 名前空間からの移行

Amazon Chime SDK は、一連のエンドポイントで API を公開します。HTTPS リクエストをエンドポイントに対して直接行うこともできますが、多くのお客様はアプリケーション内で AWS SDK を使用してサービス API を呼び出します。AWS SDK はさまざまな言語で使用でき、リクエスト署名と再試行ロジックをカプセル化することで API 呼び出しを簡素化します。AWS SDK には、各サービスエンドポイントの名前空間が含まれます。

最初に発表されたときには、Amazon Chime SDK は Amazon Chime アプリケーションと 1 つのエンドポイントを共有していました。その結果、ソリューションは AWS SDK の名前空間 Chime を使用して Amazon Chime アプリケーションと Amazon Chime SDK APIs。

現在は、Amazon Chime SDK は、各サブサービス (会議や PSTN Audio など) に対して専用のエンドポイントを提供するようになりました。各エンドポイントは AWS SDK の専用名前空間を介してアドレス指定できます。

以下のトピックでは、サービス、名前空間、エンドポイントを一覧表示し、それらをコードおよび AWS CLI で使用する方法について説明します。

トピック

- [エンドポイント、名前空間、CLI コマンド](#)
- [各サービスの移行ヘルプ](#)
- [API マッピング](#)

エンドポイント、名前空間、CLI コマンド

次の表は、専用の Amazon Chime SDK 名前空間、エンドポイント、および CLI コマンドのリストです。リンクをクリックすると、各サービスの詳細が表示されます。

エンドポイント	AWS SDK 名前空間	AWS SDK CLI
identity-chime	ChimeSDKIdentity	chime-sdk-identity
media-pipelines-chime	ChimeSDKMediaPipelines	chime-sdk-media-pipelines
meetings-chime	ChimeSDKMeetings	chime-sdk-meetings
messaging-chime	ChimeSDKMessaging	chime-sdk-messaging

エンドポイント	AWS SDK 名前空間	AWS SDK CLI
voice-chime	ChimeSDKVoice	chime-sdk-voice

各サービスの移行ヘルプ

すべてのお客様は、最新の Amazon Chime SDK 機能、APIs エンドポイントを使用することを検討する必要があります。AWS Chime 名前空間で共有エンドポイントを使用している場合、以下の移行ガイドを参照すると、移行前に技術的な違いを理解するのに役立ちます。

- [Amazon Chime SDK Identity 名前空間への移行](#)
- [Amazon Chime SDK Media Pipelines 名前空間への移行](#)
- [Amazon Chime SDK ミーティング 名前空間への移行](#)
- [Amazon Chime SDK メッセージング 名前空間への移行](#)
- [Amazon Chime SDK Voice 名前空間への移行](#)

API マッピング

次の表は、Chime 名前空間の API と、それに対応する専用の名前空間と API のリストです。専用 API の一部は Chime API と異なっており、表にはそれらのインスタンスが示されています。

Chime 名前空間の API	専用の名前空間	専用の名前空間の API
AssociatePhoneNumbersWithVoiceConnector	voice-chime	AssociatePhoneNumbersWithVoiceConnector
AssociatePhoneNumbersWithVoiceConnectorGroup	voice-chime	AssociatePhoneNumbersWithVoiceConnectorGroup
BatchCreateAttendee	meetings-chime	BatchCreateAttendee
BatchCreateChannelMembership	messaging-chime	BatchCreateChannelMembership
CreateAppInstance	identity-chime	CreateAppInstance
CreateAppInstanceAdmin	identity-chime	CreateAppInstanceAdmin

Chime 名前空間の API	専有の名前空間	専有の名前空間の API
CreateAppInstanceUser	identity-chime	CreateAppInstanceUser
CreateAttendee	meetings-chime	CreateAttendee
CreateChannel	messaging-chime	CreateChannel
CreateChannelBan	messaging-chime	CreateChannelBan
CreateChannelMembership	messaging-chime	CreateChannelMembership
CreateChannelModerator	messaging-chime	CreateChannelModerator
CreateMediaCapturePipeline	media-pipelines-chime	CreateMediaCapturePipeline
CreateMeeting	meetings-chime	CreateMeeting
CreateMeetingWithAttendees	meetings-chime	CreateMeetingWithAttendees
CreateMeetingDialOut*	該当なし	
CreateProxySession	voice-chime	CreateProxySession
CreateSipMediaApplication	voice-chime	CreateSipMediaApplication
CreateSipMediaApplicationCa ll	voice-chime	CreateSipMediaApplicationCa ll
CreateSipRule	voice-chime	CreateSipRule
CreateVoiceConnector	voice-chime	CreateVoiceConnector
CreateVoiceConnectorGroup	voice-chime	CreateVoiceConnectorGroup
DeleteAppInstance	identity-chime	DeleteAppInstance
DeleteAppInstanceAdmin	identity-chime	DeleteAppInstanceAdmin
DeleteAppInstanceS treamingConfigurations	messaging-chime	DeleteAppInstanceS treamingConfigurations

Chime 名前空間の API	専有の名前空間	専有の名前空間の API
DeleteAppInstanceUser	identity-chime	DeleteAppInstanceUser
DeleteAttendee	meetings-chime	DeleteAttendee
DeleteChannel	messaging-chime	DeleteChannel
DeleteChannelBan	messaging-chime	DeleteChannelBan
DeleteChannelMembership	messaging-chime	DeleteChannelMembership
DeleteChannelMessage	messaging-chime	DeleteChannelMessage
DeleteChannelModerator	messaging-chime	DeleteChannelModerator
DeleteMediaCapturePipeline	media-pipelines-chime	DeleteMediaCapturePipeline
DeleteMeeting	meetings-chime	DeleteMeeting
DeleteProxySession	voice-chime	DeleteProxySession
DeleteSipMediaApplication	voice-chime	DeleteSipMediaApplication
DeleteSipRule	voice-chime	DeleteSipRule
DeleteVoiceConnector	voice-chime	DeleteVoiceConnector
DeleteVoiceConnect orEmergencyCallingConfigura tion	voice-chime	DeleteVoiceConnect orEmergencyCallingConfigura tion
DeleteVoiceConnectorGroup	voice-chime	DeleteVoiceConnectorGroup
DeleteVoiceConnectorOrigina tion	voice-chime	DeleteVoiceConnectorOrigina tion
DeleteVoiceConnectorProxy	voice-chime	DeleteVoiceConnectorProxy
DeleteVoiceConnectorStreami ngConfiguration	voice-chime	DeleteVoiceConnectorStreami ngConfiguration

Chime 名前空間の API	専有の名前空間	専有の名前空間の API
DeleteVoiceConnectorTermination	voice-chime	DeleteVoiceConnectorTermination
DeleteVoiceConnectorTerminationCredentials	voice-chime	DeleteVoiceConnectorTerminationCredentials
DescribeAppInstance	identity-chime	DescribeAppInstance
DescribeAppInstanceAdmin	identity-chime	DescribeAppInstanceAdmin
DescribeAppInstanceUser	identity-chime	DescribeAppInstanceUser
DescribeChannel	messaging-chime	DescribeChannel
DescribeChannelBan	messaging-chime	DescribeChannelBan
DescribeChannelMembership	messaging-chime	DescribeChannelMembership
DescribeChannelMembershipForAppInstanceUser	messaging-chime	DescribeChannelMembershipForAppInstanceUser
DescribeChannelModeratedByAppInstanceUser	messaging-chime	DescribeChannelModeratedByAppInstanceUser
DescribeChannelModerator	messaging-chime	DescribeChannelModerator
DisassociatePhoneNumbersFromVoiceConnector	voice-chime	DisassociatePhoneNumbersFromVoiceConnector
DisassociatePhoneNumbersFromVoiceConnectorGroup	voice-chime	DisassociatePhoneNumbersFromVoiceConnectorGroup
GetAppInstanceRetentionSettings	identity-chime	GetAppInstanceRetentionSettings
GetAppInstanceStreamingConfigurations	messaging-chime	GetMessagingStreamingConfigurations

Chime 名前空間の API	専有の名前空間	専有の名前空間の API
GetAttendee	meetings-chime	GetAttendee
GetChannelMessage	messaging-chime	GetChannelMessage
GetMediaCapturePipeline	media-pipelines-chime	GetMediaCapturePipeline
GetMeeting	meetings-chime	GetMeeting
GetMessagingSessionEndpoint	messaging-chime	GetMessagingSessionEndpoint
GetProxySession	voice-chime	GetProxySession
GetSipMediaApplication	voice-chime	GetSipMediaApplication
GetSipMediaApplicationLoggingConfiguration	voice-chime	GetSipMediaApplicationLoggingConfiguration
GetSipRule	voice-chime	GetSipRule
GetVoiceConnector	voice-chime	GetVoiceConnector
GetVoiceConnectorEmergencyCallingConfiguration	voice-chime	GetVoiceConnectorEmergencyCallingConfiguration
GetVoiceConnectorGroup	voice-chime	GetVoiceConnectorGroup
GetVoiceConnectorLoggingConfiguration	voice-chime	GetVoiceConnectorLoggingConfiguration
GetVoiceConnectorOrigination	voice-chime	GetVoiceConnectorOrigination
GetVoiceConnectorProxy	voice-chime	GetVoiceConnectorProxy
GetVoiceConnectorStreamingConfiguration	voice-chime	GetVoiceConnectorStreamingConfiguration
GetVoiceConnectorTermination	voice-chime	GetVoiceConnectorTermination

Chime 名前空間の API	専有の名前空間	専有の名前空間の API
GetVoiceConnectorTerminationHealth	voice-chime	GetVoiceConnectorTerminationHealth
ListAppInstanceAdmins	identity-chime	ListAppInstanceAdmins
ListAppInstances	identity-chime	ListAppInstances
ListAppInstanceUsers	identity-chime	ListAppInstanceUsers
ListAttendees	meetings-chime	ListAttendees
ListAttendeeTags*	該当なし	
ListChannelBans	messaging-chime	ListChannelBans
ListChannelMemberships	messaging-chime	ListChannelMemberships
ListChannelMembershipsForAppInstanceUser	messaging-chime	ListChannelMembershipsForAppInstanceUser
ListChannelMessages	messaging-chime	ListChannelMessages
ListChannelModerators	messaging-chime	ListChannelModerators
ListChannels	messaging-chime	ListChannels
ListChannelsModeratedByAppInstanceUser	messaging-chime	ListChannelsModeratedByAppInstanceUser
ListMediaCapturePipelines	media-pipelines-chime	ListMediaCapturePipelines
ListMeetings*	該当なし	
ListMeetingTags+	meetings-chime	ListTagsForResource
ListProxySessions	voice-chime	ListProxySessions
ListSipMediaApplications	voice-chime	ListSipMediaApplications
ListSipRules	voice-chime	ListSipRules

Chime 名前空間の API	専有の名前空間	専有の名前空間の API
ListTagsForResource	identity-chime	ListTagsForResource
ListVoiceConnectorGroups	voice-chime	ListVoiceConnectorGroups
ListVoiceConnectors	voice-chime	ListVoiceConnectors
ListVoiceConnectorTerminationCredentials	voice-chime	ListVoiceConnectorTerminationCredentials
PutAppInstanceRetentionSettings	identity-chime	PutAppInstanceRetentionSettings
PutAppInstanceStreamingConfigurations	messaging-chime	PutMessagingStreamingConfigurations
PutSipMediaApplicationLoggingConfiguration	voice-chime	PutSipMediaApplicationLoggingConfiguration
PutVoiceConnectorEmergencyCallingConfiguration	voice-chime	PutVoiceConnectorEmergencyCallingConfiguration
PutVoiceConnectorLoggingConfiguration	voice-chime	PutVoiceConnectorLoggingConfiguration
PutVoiceConnectorOrigination	voice-chime	PutVoiceConnectorOrigination
PutVoiceConnectorProxy	voice-chime	PutVoiceConnectorProxy
PutVoiceConnectorStreamingConfiguration	voice-chime	PutVoiceConnectorStreamingConfiguration
PutVoiceConnectorTermination	voice-chime	PutVoiceConnectorTermination
PutVoiceConnectorTerminationCredentials	voice-chime	PutVoiceConnectorTerminationCredentials
RedactChannelMessage	messaging-chime	RedactChannelMessage

Chime 名前空間の API	専有の名前空間	専有の名前空間の API
SendChannelMessage	messaging-chime	SendChannelMessage
StartMeetingTranscription	meetings-chime	StartMeetingTranscription
StopMeetingTranscription	meetings-chime	StopMeetingTranscription
TagAttendee*	該当なし	
TagMeeting+	meetings-chime	TagResource
TagResource	identity-chime	TagResource
	media-pipelines-chime	TagResource
	meetings-chime	TagResource
	messaging-chime	TagResource
	voice-chime	TagResource
UntagAttendee*	該当なし	
UntagMeeting+	meetings-chime	UntagResource
UntagResource	identity-chime	UntagResource
	media-pipelines-chime	UntagResource
	meetings-chime	UntagResource
	messaging-chime	UntagResource
	voice-chime	UntagResource
UpdateAppInstance	identity-chime	UpdateAppInstance
UpdateAppInstanceUser	identity-chime	UpdateAppInstanceUser
UpdateChannel	messaging-chime	UpdateChannel

Chime 名前空間の API	専有の名前空間	専有の名前空間の API
UpdateChannelMessage	messaging-chime	UpdateChannelMessage
UpdateChannelReadMarker	messaging-chime	UpdateChannelReadMarker
UpdateProxySession	voice-chime	UpdateProxySession
UpdateSipMediaApplication	voice-chime	UpdateSipMediaApplication
UpdateSipMediaApplicationCa ll	voice-chime	UpdateSipMediaApplicationCa ll
UpdateSipRule	voice-chime	UpdateSipRule
UpdateVoiceConnector	voice-chime	UpdateVoiceConnector
UpdateVoiceConnectorGroup	voice-chime	UpdateVoiceConnectorGroup
ValidateE911Address	voice-chime	ValidateE911Address

+ API は別の名前の API に置き換えられました。

* API は使用できなくなりました。

Amazon Chime SDK ミーティングの使用

このセクションのトピックでは、Amazon Chime SDK ミーティングを使用してカスタム会議アプリケーションを作成する方法について説明します。以下のトピックを、記載されている順序で実行することをお勧めします。

トピック

- [Amazon Chime SDK ミーティング名前空間への移行](#)
- [会議リージョンの使用](#)
- [会議を作成する](#)
- [会議機能の選択](#)
- [Amazon Chime SDK が WebRTC メディアを使用する方法](#)
- [動画コーデックの設定](#)
- [ネットワーク構成](#)
- [会議イベント](#)
- [Amazon CloudWatch メトリクス](#)
- [Amazon Chime SDK メディアパイプラインを作成する](#)
- [Amazon Chime SDK ライブ文字起こしの使用](#)
- [メディアレプリケーションの使用](#)
- [Amazon Chime SDK ミーティングのトラブルシューティングとデバッグ](#)

Amazon Chime SDK ミーティング名前空間への移行

[Amazon Chime SDK ミーティング](#) 名前空間は、Amazon Chime SDK ミーティングのリソースを作成および管理する API 専用の場所です。この名前空間を使用すると、任意の AWS リージョンの Amazon Chime SDK ミーティング API エンドポイントをアドレス指定できます (そのリージョンでそれらのエンドポイントを利用できる場合)。Amazon Chime SDK を使い始めたばかりの場合は、この名前空間を使用してください。リージョンの詳細については、このガイドの「[利用できるリージョン](#)」を参照してください。

[Amazon Chime](#) 名前空間を使用する既存のアプリケーションでは、最新の API と機能を使用するために、専用の名前空間への移行を計画する必要があります。

トピック

- [移行すべき理由](#)
- [移行する前に](#)
- [名前空間の相違点](#)

移行すべき理由

以下の理由から、[Amazon Chime SDK ミーティング](#) 名前空間に移行することをお勧めします。

API エンドポイントの選択

Amazon Chime SDK ミーティング名前空間は、[API エンドポイントを利用できる任意のリージョン](#)の API エンドポイントを使用できる唯一の API 名前空間です。us-east-1 以外の API エンドポイントを使用する場合は、Amazon Chime SDK ミーティング名前空間を使用する必要があります。

Amazon Chime SDK 会議で AWS リージョンを使用する方法の詳細については、このガイドの「[会議リージョン](#)」を参照してください。

会議 API の更新と新規追加

Amazon Chime SDK ミーティング名前空間の会議 API のみを追加または更新します。

移行する前に

移行する前に、名前空間の相違点に注意してください。以下の表では、名前空間の一覧と説明を示しています。

	Amazon Chime SDK ミーティング名前空間	Amazon Chime 名前空間
AWS SDK 名前空間	ChimeSDKMeetings	Chime
リージョン	複数	us-east-1 のみ
エンドポイント	https://meetings-chime.region.amazonaws.com	https://service.chime.amazonaws.com
サービスプリンシパル	meetings.chime.amazonaws.com	chime.amazonaws.com

	Amazon Chime SDK ミーティング名前空間	Amazon Chime 名前空間
API	会議用の API のみ	会議用の API と Amazon Chime のその他の部分
CreateMeeting	ExternalMeetingId と MediaRegion は必須です。	ExternalMeetingId および MediaRegion はオプションです。
CreateMeetingWithAttendees	ExternalMeetingId と MediaRegion は必須です。	ExternalMeetingId および MediaRegion はオプションです。
ListMeetings	利用不可	利用可能
ExternalMeetingId	検証にはパターンマッチングが含まれます	利用可能
ExternalUserId	検証にはパターンマッチングが含まれます	利用可能
会議タグ APIs	TagResource , UntagResource , ListTagsForResource	TagMeeting , UntagMeeting , ListMeetingTags
参加者タグ	利用不可	利用可能
エコーリダクション	利用可能	利用不可
ライブ文字起こし言語識別	利用可能	利用不可
参加者の機能	利用可能	利用不可
メディアレプリケーション	利用可能	利用不可
AppKeys および TenantIds	利用可能	利用不可

	Amazon Chime SDK ミーティング名前空間	Amazon Chime 名前空間
メディアパイプライン	メディアパイプラインは、Amazon Chime SDK ミーティング名前空間で複数のリージョンをサポートします。詳細については、「 ChimeSdkMediaPipelines 名前空間への移行 」を参照してください。	us-east-1 エンドポイント経由で利用可能
SIP メディアアプリケーション	JoinChimeMeeting アクションに MeetingId は必須	JoinChimeMeeting アクションに MeetingId は不要
SIP の直接統合	利用不可	利用可能

名前空間の相違点

以下のセクションでは、Amazon Chime 名前空間と Amazon Chime SDK Meetings 名前空間の相違点について説明します。

AWS SDK 名前空間

Amazon Chime SDK 名前空間では Chime という正式名を使用します。Amazon Chime SDK ミーティング名前空間では、ChimeSDKMeetings という正式な名前を使用します。名前の正確な形式はプラットフォームによって異なります。

例えば、Node.js の AWS SDK を使用して会議を作成する場合は、コード行を使用して名前空間に対処します。

```
const chimeMeetings = AWS.Chime();
```

Amazon Chime 会議 SDK に移行するには、この行のコードを新しい名前空間とエンドポイントリージョンで更新します。

```
const chimeMeetings = AWS.ChimeSDKMeetings({ region: "eu-central-1" });
```


リージョン

[Amazon Chime](#) 名前空間は、us-east-1 リージョンの API エンドポイントのみをアドレス指定します。[Amazon Chime SDK ミーティング](#) 名前空間は、任意のリージョンの Amazon Chime SDK ミーティング API エンドポイントをアドレス指定できます (そのリージョンでそれらのエンドポイントを利用できる場合)。会議リージョンの最新リストについては、本ガイドの「[利用できるリージョン](#)」を参照してください。

エンドポイント

[Amazon Chime SDK ミーティング](#) 名前空間は、[Amazon Chime](#) 名前空間とは異なる API エンドポイントを使用します。

会議を変更するために使用できるのは、会議の作成に使用したエンドポイントだけです。つまり、EU-CENTRAL-1 のエンドポイントで作成された会議は、EU-CENTRAL-1 でのみ変更できます。また、Chime 名前空間で作成された会議を、US-EAST-1 の ChimeSDKMeetings 名前空間を使用してアドレス指定することはできません。現在のエンドポイントの詳細については、このガイドの「[API マッピング](#)」を参照してください。

サービスプリンシパル

[Amazon Chime SDK ミーティング](#) 名前空間は、新しいサービスプリンシパル `meetings.chime.amazonaws.com` を使用します。サービスへのアクセスを許可する SQS、SNS またはその他の IAM アクセスポリシーがある場合は、それらのポリシーを更新して新しいサービスプリンシパルへのアクセスを許可する必要があります。

API

[Amazon Chime SDK ミーティング](#) 名前空間には、会議を作成および管理するための API のみが含まれています。[Amazon Chime](#) 名前空間には、会議用および Amazon Chime サービスのその他の部分用の API が含まれています。

CreateMeeting 必須フィールド

Amazon Chime SDK ミーティング名前空間では、[CreateMeeting](#) および [CreateMeetingWithAttendees](#) APIs には ExternalMeetingId および MediaRegion フィールドを指定する必要があります。

外部 ID 値

[Amazon Chime SDK ミーティング](#) 名前空間は、ExternalMeetingId および ExternalUserId に使用できる値に対して追加の検証を実施します。

エコーリダクション

[Amazon Chime SDK ミーティング](#) 名前空間には機械学習ベースのエコーリダクション機能が備わっており、ローカルのラウドスピーカーから発生するノイズや音が会議の音声に入らないように取り除くのに役立ちます。詳細については、GitHub「」のガイドを参照してください。

参加者の機能

[Amazon Chime SDK ミーティング](#) 名前空間では、会議の参加者が音声、動画、コンテンツを送受信する機能をきめ細かく制御できます。

メディアレプリケーション

[Amazon Chime SDK ミーティング](#) 名前空間には、プライマリ会議をレプリカ会議にリンクするメディアレプリケーション機能が備わっており、最大 10,000 人が同時にリアルタイムセッションに参加できます。レプリカセッションに接続した参加者は、プライマリセッションに接続したプレゼンターのメディアを受け取りますが、これらの参加者をプライマリ会議の参加者に昇格させることもできます。詳細については、本ガイドの「[メディアレプリケーションの使用](#)」を参照してください。

AppKeys および TenantIds

[Amazon Chime SDK ミーティング](#) 名前空間は、特定の Amazon Chime SDK ミーティングに対するネットワークからのアクセスを制限する方法を提供します。詳細については、本ガイドの「[AppKeys と TenantIDsの使用](#)」を参照してください。

メディアパイプライン

Amazon Chime SDK メディアパイプラインは、[Amazon Chime SDK ミーティング](#) 名前空間または [Amazon Chime](#) 名前空間のどちらを使用したかにかかわらず、どの会議エンドポイントによって作成された会議でも動作します。メディアパイプラインリージョンの最新リストについては、「[利用できるリージョン](#)」を参照してください。

SIP メディアアプリケーション

Amazon Chime SDK SIP メディアアプリケーションは、[Amazon Chime SDK ミーティング](#) 名前空間または [Amazon Chime](#) 名前空間のどちらを使用したかにかかわらず、どの会議エンドポイントによって作成された会議でも動作します。Amazon Chime SDK ミーティング名前空間を使用して作成された会議で SIP メディアアプリケーションを使用する場合、[JoinChimeMeeting](#) アクションには MeetingId パラメータが必要です。

その他の API

会議名前空間には、Chime 名前空間にはない API が増え続けています。Amazon Chime SDK の使用を開始する場合は、会議名前空間を使用して最新の機能にアクセスしてください。

会議リージョンの使用

Amazon Chime SDK ミーティングには、コントロールリージョンとメディアリージョンがあります。コントロールリージョンには、会議の作成、更新、削除に使用される API エンドポイントがあります。メディアリージョンは、実際の会議をホストします。

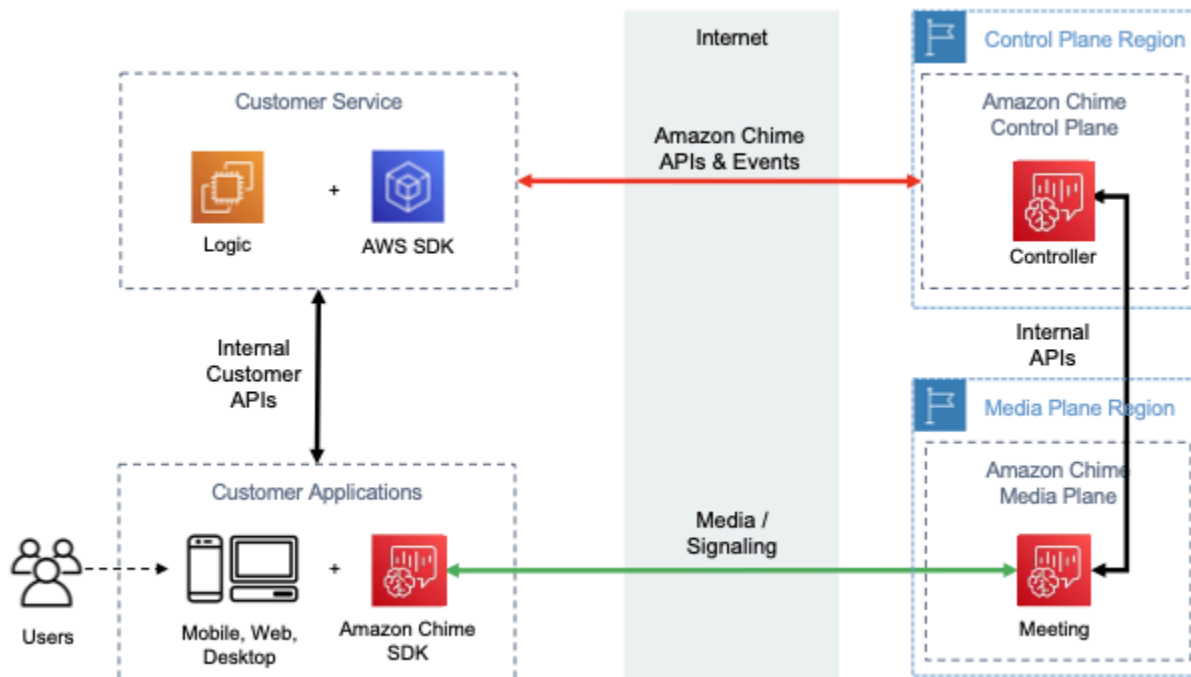
通常、アプリケーションサービスは [AWS SDK](#) を使用して、コントロールリージョンの API の [署名と呼び出し](#) を行います。アプリケーションクライアントは、[iOS JavaScript](#)、または [Android](#) 用の Amazon Chime SDK クライアントライブラリを使用して、メディアリージョンの会議に接続します。

コントロールリージョンは、同じ AWS パーティション内の任意のメディアリージョンで会議を作成できます。ただし、会議の更新は、その会議の作成に使用されたコントロールリージョン内で行うことができません。顧客に最も近いメディアリージョンを見つけるには、<https://nearest-media-region.l.chime.aws> を呼び出します。

会議コントロールリージョンの [EventBridge](#)、[Amazon Simple Queue Service \(SQS\)](#)、[Amazon Simple Notification Service \(SNS\)](#) の `AttendeeJoined` 呼び出しなどの会議 [イベント](#)。

利用可能な Amazon Chime SDK ミーティングコントロールリージョンとメディアリージョンのリストについては、本ガイドの「[利用できるリージョン](#)」を参照してください。

この図は、コントロールリージョンとメディアリージョンを介した一般的なデータフローを示しています。



コントロールリージョンの選択

Amazon Chime SDK ミーティングのコントロールリージョンを選択する際には、以下の要素に留意してください。

- 規制要件。アプリケーションは地政学的な境界内にある必要があるか、あるいは FIPS 140-2 で検証済みの暗号化モジュールを備えたエンドポイントを使用する必要があるか。
- API のレイテンシー。アプリケーションサービスのリージョンに最も近いコントロール AWS リージョンを使用すると、APIsのネットワークレイテンシーを低減できます。その結果、会議の作成に必要な時間が短縮され、ユーザーはより迅速に会議に参加できるようになります。
- 高可用性。複数のコントロールリージョンを使用して、高可用性アーキテクチャを実装できます。ただし、各コントロールリージョンは独立して動作します。また、会議の更新は、その会議の作成に使用されたコントロールリージョン内でのしか行うことができません。さらに、[Amazon Simple Queue Service \(SQS\) EventBridge](#)、または [Amazon Simple Notification Service \(SNS\)](#) で会議イベントを使用するには、同じリージョンを使用する必要があります。

メディアリージョンの選択

Note

[CreateMeeting](#) API アクションの `MediaRegion` パラメータで常に値を指定することをお勧めします。リージョンの詳細については、「[利用できるリージョン](#)」を参照してください。

Amazon Chime SDK ミーティングに使用するメディアリージョンを選択する際には、以下の一般的な要素を考慮してください。

規制要件

Amazon Chime SDK ミーティングが、地政学的な境界内でホストすることを義務付ける規制の対象となる場合は、固定のアプリケーションロジックに基づいて会議リージョンをハードコーディングすることを検討してください。

例えば、遠隔医療アプリケーションでは、すべての会議を医師の管轄区域でホストすることが義務付けられている場合があります。アプリケーションがヨーロッパと米国の両方にある診療所をサポートしている場合は、各診療所の住所を使用して、その管轄区域内のリージョンを選択できます。

会議の品質

Amazon Chime SDK ミーティングがメディアリージョンでホストされた場合、各参加者の音声と動画はそのリージョンで送受信されます。参加者とリージョンの間の距離が長くなると、会議の品質はネットワークレイテンシーの影響を受ける可能性があります。Amazon Chime SDK ミーティングのリージョンを指定すると、参加者が相互に地理的に近くにいるか分散しているかにかかわらず、会議の品質を向上させることができます。

Amazon Chime SDK ミーティングのメディアリージョンを選択する際には、以下のいずれかの方法を使用できます。

メディアリージョンをハードコーディングする

Amazon Chime SDK ミーティングがすべて特定の AWS リージョン内でホストされている場合に推奨されます。

最も近いメディアリージョンを選択する

Amazon Chime SDK 会議の参加者が同じ AWS リージョンに配置されているが、会議が別のリージョンでホストされている場合に推奨されます。

最も近いメディアリージョンを見つける

Amazon Chime SDK ミーティングをホストできる最も近いメディアリージョンを見つけるには、<https://nearest-media-region.l.chime.aws> を呼び出します。このエンドポイントは、1つのリージョン ({"region": "us-west-2"} など) を返します。クライアントアプリケーションから URL を呼び出して、ユーザーに最も近いリージョンを特定し、[CreateMeeting](#) API の `MediaRegion` パラメータの結果を使用して、そのリージョンで会議を作成します。

通常、URL を呼び出すのは、クライアントアプリケーションの起動時か、またはネットワーク接続の変更時です。最も近いリージョンを事前に確認しておくことで、会議の作成時に呼び出しのレイテンシーが増えるのを防ぐことができます。

最も近い AWS GovCloud (米国) メディアリージョンの検索

Amazon Chime SDK 会議をホストできる最も近い AWS GovCloud (米国) リージョンを見つけるには、<https://nearest-us-gov-media-region.l.chime.aws> を呼び出します。このエンドポイントは、最も近いリージョン ({"region": "us-gov-west-1"} など) を返します。クライアントアプリケーションから URL を呼び出して、ユーザーに最も近い AWS GovCloud (米国) を特定し、[CreateMeeting](#) API の `MediaRegion` パラメータの結果を使用して、そのリージョンで会議を作成します。

通常、URL を呼び出すのは、クライアントアプリケーションの起動時か、またはネットワーク接続の変更時です。最も近いリージョンを事前に確認しておくことで、会議の作成時に呼び出しのレイテンシーが増えるのを防ぐことができます。

JavaScript 例

次の例では、HTML とを使用して JavaScript、最も近いメディアリージョンと AWS GovCloud (米国) メディアリージョンを返します。

```
<html>
<head>
  <title>Amazon Chime SDK - Nearest Media Region</title>
  <script>
```

```
async function getNearestMediaRegion(partition) {

    console.log('Nearest media region partition: ' + partition);

    const url = ('aws-us-gov' == partition) ? 'https://nearest-us-gov-media-
region.l.chime.aws' : 'https://nearest-media-region.l.chime.aws';
    let result = ('aws-us-gov' == partition) ? 'us-gov-west-1' : 'us-west-2';

    try { //Find the nearest media region
        console.log('Nearest media region URL: ' + url);
        const response = await fetch(url, {method: 'GET'} );
        const body = await response.json();
        result = body.region;
    } catch (error) {
        console.log(error.message);
    } finally {
        console.log('Nearest media region found: ' + result);
        return result;
    }
}

async function findRegions(partition) {
    aws.innerText = await getNearestMediaRegion();
    awsusgov.innerText = await getNearestMediaRegion('aws-us-gov');
}
</script>
</head>
<body>
<h3>Nearest media region, by AWS partition</h3>
<table>
<tr><th>Partition</th><th>Media Region</th></tr>
<tr><td>aws</td><td id="aws">Finding...</td></tr>
<tr><td>aws-us-gov</td><td id="awsusgov">Finding...</td></tr>
</table>
<script>
    findRegions();
</script>
</body>
</html>
```

リージョンステータスを確認する

<https://region.status.chime.aws/> にアクセスして、各リージョンの Amazon Chime SDK サービスの状態を取得します。結果には、推奨リージョンが表示されます。メディアリージョンのステータスが [推奨] 以外の場合、最も近いメディアリージョンのエンドポイントはそのリージョンを返しません。

次の例は一般的な結果を示しています。

```
{
  "MeetingsControlRegions": {
    "us-east-1": "recommended",
    "us-west-2": "recommended",
    "ap-southeast-1": "recommended",
    "eu-central-1": "recommended"
  },
  "MeetingsMediaRegions": {
    "af-south-1": "recommended",
    "ap-northeast-1": "recommended",
    "ap-northeast-2": "recommended",
    "ap-south-1": "recommended",
    "ap-southeast-1": "recommended",
    "ap-southeast-2": "recommended",
    "ca-central-1": "recommended",
    "eu-central-1": "recommended",
    "eu-north-1": "recommended",
    "eu-south-1": "recommended",
    "eu-west-1": "recommended",
    "eu-west-2": "recommended",
    "eu-west-3": "recommended",
    "sa-east-1": "recommended",
    "us-east-1": "recommended",
    "us-east-2": "recommended",
    "us-west-1": "recommended",
    "us-west-2": "recommended"
  },
  "MediaPipelineControlRegions": {
    "ap-southeast-1": "recommended",
    "eu-central-1": "recommended",
    "us-east-1": "recommended",
    "us-west-2": "recommended"
  },
  "MediaPipelineDataRegions": {
    "af-south-1": "recommended",
```



```
"ap-northeast-1": "recommended",
"ap-northeast-2": "recommended",
"ap-south-1": "recommended",
"ap-southeast-1": "recommended",
"ap-southeast-2": "recommended",
"ca-central-1": "recommended",
"eu-central-1": "recommended",
"eu-north-1": "recommended",
"eu-south-1": "recommended",
"eu-west-1": "recommended",
"eu-west-2": "recommended",
"eu-west-3": "recommended",
"sa-east-1": "recommended",
"us-east-1": "recommended",
"us-east-2": "recommended",
"us-west-1": "recommended",
"us-west-2": "recommended"
}
}
```

会議を作成する

次の手順では、サーバーアプリケーションとクライアントアプリケーション用に、音声と動画を使用する会議を作成する方法を説明します。開始する前に、クライアントアプリケーションを Amazon Chime SDK のクライアントライブラリと統合する必要があります。詳細については、「[クライアントライブラリと統合する](#)」を参照してください。

音声と動画を使用する会議を作成する方法

1. サーバーアプリケーションから以下のステップを実行します。
 - a. Amazon Chime SDK [CreateMeeting](#) API リファレンスの API アクションを使用して会議を作成します。MediaRegion パラメータを使用して AWS リージョンを指定します。会議リージョンを選択する方法の詳細については、「[会議リージョン](#)」を参照してください。
 - b. API アクションまたは [CreateAttendee](#) API [BatchCreateAttendee](#) アクションを使用して会議に参加者を追加します。会議と参加者に関する情報を、サーバーアプリケーションからそれぞれの参加者として承認されたクライアントに安全に転送します。会議と参加者の詳細については、「Amazon Chime SDK API リファレンス [Attendee](#)」の [Meeting](#) 「」と「」を参照してください。
2. クライアントアプリケーションから以下のステップを実行します。

- a. Amazon Chime SDK クライアントライブラリを使用して、MeetingSessionConfiguration オブジェクトを作成します。上記のステップの会議と参加者の情報を使用します。
- b. AudioVideoObserver インターフェイスを実装します。
- c. MeetingSessionConfiguration を使用して MeetingSession を作成します。
- d. MeetingSession の AudioVideoFacade を使用して、リアルタイムメディアを制御します。
 - i. AudioVideoObserver インターフェイスのインスタンスを登録します。これにより、会議の状態が変化したときにイベントを受信できます。
 - ii. 音声入力、音声出力、動画入力の初期デバイスを選択します。
 - iii. オーディオビジュアルセッションを開始します。
 - iv. ユーザーが動画を共有したいときに、ローカル動画キャプチャを開始します。
 - v. 動画タイルを表示するには、動画タイルイベントを管理し、クライアントアプリケーションでタイルを動画サーフェスにバインドします。
 - vi. ミュートやミュート解除、ローカル動画キャプチャの開始や停止など、その他のユーザー操作を管理します。
 - vii. 会議から退出するには、オーディオビジュアルセッションを停止します。
- e. (オプション) MeetingSession の AudioVideoFacade を使用して、画面キャプチャなどのメディアコンテンツを他のクライアントと共有します。
 - i. 画面共有セッションを開始します。コンテンツは追加の参加者として会議に参加します。
 - ii. 共有コンテンツを表示するには、動画タイルイベントを管理し、クライアントアプリケーションでタイルをサーフェスにバインドします。
 - iii. コンテンツ共有の一時停止、再開、停止など、その他の操作を管理します。

会議は [DeleteMeeting](#) API アクションを実行すると終了します。また、以下の場合に会議は自動的に終了します。

- 会議時間が 24 時間を超えた場合。
- その会議が [レプリカ会議](#) であり、プライマリ会議が終了した場合。
- 非レプリカ会議で、接続している参加者が 5 分間連続して存在しない場合。

会議機能の選択

[CreateMeeting](#) API を呼び出すときに、セッションに参加するクライアントが使用できるようにする機能を指定できます。一部の機能オプションには追加料金が発生することに注意してください。

セッションでは次の機能を使用できます。

- `Audio.EchoReduction` - 機械学習エコーリダクション。
- `Video.MaxResolution` - ウェブカメラ動画の最大解像度。
- `Content.MaxResolution` - コンテンツ共有の最大解像度。
- `Attendees.MaxCount` - 参加者の最大数。

トピック

- [Audio の使用。EchoReduction](#)
- [ビデオの使用。MaxResolution](#)
- [コンテンツの使用。MaxResolution](#)
- [参加者の使用。MaxCount](#)
- [クライアントアプリでの会議機能の使用](#)

Audio の使用。EchoReduction

ユーザーのスピーカーからの音が会議に逆流するのを防ぐために `Audio.EchoReduction` を使用します。

エコーリダクションは、会議音声の主出力デバイスをユーザーのスピーカーにするような状況に最適です。例えば、複数のユーザーが会議室の同じデバイスから会議に参加する場合や、リモートの参加者がヘッドフォンをつけていない場合などです。

エコーリダクションは、JavaScript および React クライアントライブラリで使用できます。詳細については、「」のドキュメント [GitHub](#) を参照してください。追加料金がかかります。詳細については、「[Amazon Chime SDK の料金](#)」を参照してください。

ビデオの使用。MaxResolution

`Video.MaxResolution` は会議のウェブカメラ動画の最大解像度を指定する場合に使用します。この機能には次のようなオプションがあります。

- None: カメラ動画は使用できません。
- HD: 高解像度カメラ動画 (1280 x 720p)
- FHD: full-high-definition カメラビデオ (1920 x 1080)

FHD (1080p) 動画がリクエストされると、高解像度 WebRTC セッションが作成されます。詳細については、「[Amazon Chime SDK の料金](#)」を参照してください。

クライアントがウェブカメラ動画を指定された上限を超えて送信しようとする、サービスは動画を拒否し、次のエラーを送信します。

```
Disabled video/content send capability, reason: Video resolution is above limit of current meeting feature selection.
```

コンテンツの使用。MaxResolution

`Content.MaxResolution` は会議のコンテンツ共有の最大解像度を指定するために使用します。この機能には次のようなオプションがあります。

- None: コンテンツ共有は許可されません。
- FHD : full-high-definition コンテンツ共有 (1920 x 1080)
- UHD : ultra-high-definition コンテンツ共有 (3840 x 2160)

UHD (4K) コンテンツがリクエストされると、高解像度 WebRTC セッションが作成されます。

クライアントが最大解像度を超えてコンテンツ共有を送信しようとする、その解像度は指定された最大解像度まで縮小されます。`MediaTrackConstraints` をコンテンツ共有トラックに適用してスケールリングできます。次の例は、共有トラックをスケールリングする方法を示しています。

```
const constraint: MediaTrackConstraints = {
  width: { ideal: videoQualitySettings.videoWidth },
  height: { ideal: videoQualitySettings.videoHeight },
  frameRate: { ideal: videoQualitySettings.videoFrameRate },
};
this.context.logger.info(
  `Video track (content = ${isContentAttendee}) with constraint: ${JSON.stringify(
    constraint
  )}, trackSettings: ${JSON.stringify(trackSettings)}`
);
```

```
try {
  await mediaStreamTrack.applyConstraints(constraint);
} catch (error) {
  this.context.logger.info(
    `Could not apply constraint for video track (content = ${isContentAttendee})`
  );
}
```

次の表に、予想されるコンテンツ共有の動作を示します。

コンテンツ機能	コンテンツ共有ネイティブ解像度	スケーリング	コンテンツコーディング解像度
FHD	1280x720	いいえ	1280x720
FHD	1920x1080	いいえ	1920x1080
FHD	3840x2160	はい	1920x1080
UHD	1920x1080	いいえ	1920x1080
UHD	3840x2160	いいえ	3840x2160
UHD	4200 x 2400	はい	3780 x 2160

参加者の使用。MaxCount

Attendee.MaxCount は会議に参加できる最大参加者数を指定する場合に使用します。Attendee の上限はMaxCount、セッションタイプによって異なります。標準セッションの場合、最大 250 人の出席者を選択できます。高解像度セッションでは、最大 25 人の参加者を選択する必要があります。

FHD (1080p) 動画または UHD (4K) コンテンツをリクエストした場合、セッションは高解像度セッションになります。

高解像度セッションには出席者定員料金が適用されます。詳細については、「[Amazon Chime SDK の料金](#)」を参照してください。

クライアントアプリでの会議機能の使用

指定した機能を使用した会議の作成

会議を作成するには、[CreateMeeting](#) API を呼び出し、目的の会議機能を指定します。次の例は、すべての機能を指定する方法を示しています。

```
// You must migrate to the Amazon Chime SDK Meetings namespace.
const chime = AWS.ChimeSDKMeetings({ region: "eu-central-1" });

// Create meeting
const meetingInfo = await chime.createMeeting({
  ...
  MeetingFeatures: {
    Audio: {
      EchoReduction: 'AVAILABLE'
    },
    Video: {
      MaxResolution: 'FHD'
    },
    Content: {
      MaxResolution: 'UHD'
    },
    Attendee: {
      MaxCount: 25
    },
  },
}).promise();
```

クライアントでの会議機能の使用

必要な機能を備えた会議を作成したら、`MeetingSessionConfiguration` オブジェクトを作成するときに `joinInfo` を渡すことができます。会議機能は `meetingSession` 作成時にウェブカメラの動画解像度とビットレート、コンテンツ共有の解像度とビットレートを設定するために使用されません。

```
const configuration = new MeetingSessionConfiguration(this.joinInfo.Meeting,
  this.joinInfo.Attendee);

this.meetingSession = new DefaultMeetingSession(
  configuration,
  this.meetingLogger,
  this.deviceController,
  new DefaultEventController(configuration, this.meetingLogger, this.eventReporter)
```

```
);
```

Amazon Chime SDK が WebRTC メディアを使用する方法

Amazon Chime SDK は、標準と高解像度の 2 種類の WebRTC セッションをサポートしています。以下のトピックでは、React、iOS JavaScript、および Android 用の Amazon Chime SDK クライアントライブラリを使用する際に、各タイプのセッションで使用できるメディアについて説明します。

トピック

- [音声](#)
- [動画](#)
- [コンテンツ共有](#)
- [データメッセージ](#)

音声

Amazon Chime クライアントはそれぞれ 1 つの音声ストリームをセッションに送信し、セッションから 1 つの音声ストリームを受信します。通常、ローカルデバイスのマイクがオーディオを生成します。受信したオーディオは、他のセッションクライアントから送信されたオーディオとミックスされたものです。

どちらのセッションタイプも、Opus コーデックを使用して最大 48 kHz のサンプルレートと最大 128 kbps のビットレートでエンコードされた 2 チャンネル (ステレオ) をサポートします。ただし、送受信される音声ストリームはクライアントライブラリのタイプによって異なります。

- JavaScript および React 用の Amazon Chime SDK クライアントライブラリは、デバイスとブラウザでサポートされている最大サンプルレートで、最大 48kHz のモノラルオーディオとステレオオーディオの送受信をサポートします。
- iOS および Android 用の Amazon Chime SDK クライアントライブラリは、最大 48 kHz のモノラルオーディオの送信と 48 kHz のステレオオーディオの受信をサポートしています。

動画

Amazon Chime クライアントはそれぞれ 1 つの動画ストリームをセッションに送信し、セッションから最大 25 個の動画ストリームを受信できます。送信される動画は、通常、ローカルデバイスの

ウェブカメラから送信されます。各クライアントは、受信する動画ストリームを最大 25 個まで選択でき、セッション中いつでも選択を変更できます。

標準セッションでは、H.264、VP8、VP9、AV1 を使用して最大 1500 kbps のビットレートでエンコードされた、毎秒 30 フレームで最大 1280 x 720 の動画解像度がサポートされます。

高解像度セッションでは、H.264、VP8、VP9、AV1 を使用して最大 2500 kbps のビットレートでエンコードされた、毎秒 30 フレームで最大 1920 x 1080 の動画解像度がサポートされます。

JavaScript および React 用の Amazon Chime SDK クライアントライブラリは、1 秒あたり 15 フレームのサイマルキャスト、またはスケーラブルビデオコーディング (SVC) によるビデオの送信をサポートします。SVC は、3 つの空間レイヤーと 3 つの時間レイヤーを含む 1 つの動画ストリームをターゲット値の 100%、50%、25% でエンコードします。このサービスは、視聴者が利用できる帯域幅に基づいて、各視聴者に送信するレイヤーを自動的に選択します。

iOS および Android 用の Amazon Chime SDK クライアントライブラリは、1 秒あたり最大 15 フレームの送信をサポートします。ただし、実際のフレームレートと解像度は Amazon Chime SDK によって自動的に管理されます。

動画のエンコードとデコードでは、パフォーマンスを向上させるためにハードウェアアクセラレーション (利用可能な場合) を使用します。

クライアントが最大許容ビットレートを超えるビットレートで動画を送信すると、セッションはまず、リアルタイム制御プロトコルを介してクライアントの受信者推定最大ビットレートメッセージの送信を開始します。クライアントが最大許容ビットレートを超えるビットレートで動画を送信し続けると、セッションは受信した動画ストリームパケットを破棄します。

コンテンツ共有

最大 2 つのクライアントがセッションでコンテンツを共有できます。コンテンツ共有には、動画トラック、音声トラック、またはその両方を含めることができます。コンテンツ共有の一般的な例としては、スクリーンキャプチャをコンテンツのソースとして使用する画面共有があります。もう 1 つの例として、収録済みのコンテンツを動画トラックや音声トラックと共有することが挙げられます。

コンテンツオーディオは、セッションによって送信された音声ストリームにミックスされます。コンテンツオーディオは、Opus コーデックを使用して最大 48 kHz のサンプルレートと最大 128 kbps のビットレートでエンコードされた 2 チャンネル (ステレオ) をサポートします。

動画コンテンツはセッションに送信され、別の動画ストリームでクライアントに転送されます。標準セッションは、毎秒 30 フレームで最大 1920 x 1080 の解像度をサポートします。高解像度セッションは、30 フレーム/秒で最大 3840 x 2160 のコンテンツ動画をサポートします。

コンテンツ共有用のスクリーンキャプチャでは、キャプチャされる画面またはウィンドウの解像度が、セッションの種類に応じた最大コンテンツ解像度、1秒あたり最大30フレームまで使用されます。ただし、デバイスとブラウザの機能によってこれらの値が制限される場合があります。

JavaScript および React 用の Amazon Chime SDK クライアントライブラリは、画面キャプチャやその他のソースからのコンテンツ共有をサポートします。

iOS および Android 用の Amazon Chime SDK クライアントライブラリでは、スクリーンキャプチャからのコンテンツ共有のみをサポートします。

データメッセージ

データメッセージは、クライアントがセッション中の他のクライアントに情報をブロードキャストする方法を提供します。例えば、アプリケーションはデータメッセージを使ってセッション中の絵文字リアクションを共有できます。

各データメッセージには以下が含まれます。

- トピック (最大 64 文字の文字列)。
- トピックを含めて最大 2 KB のデータ。

クライアントはセッションにデータメッセージを送信し、セッションは接続しているすべてのクライアントにデータメッセージを送信します。

セッションでは、オプションでデータメッセージを最大 5 分間キャッシュできます。クライアントがセッションに参加したり、セッションに再接続したりすると、会議は、以前に送信されていないキャッシュされたデータメッセージをクライアントに自動的に送信します。セッションキャッシュには最大 1024 件のデータメッセージが保存されます。

セッションでは、毎秒最大 100 件のデータメッセージの送信がサポートされます。[ライブ文字起こし](#)を使用すると、各クライアントはデータメッセージを介して[文字起こしメッセージ](#)を受信します。この文字起こしメッセージは 1 秒あたりの送信メッセージ総数にカウントされます。

動画コーデックの設定

クライアントデバイスは動画コーデックを使用して raw 動画をサービスに送信する前に圧縮し、受信した動画をレンダリングする前に解凍します。

JavaScript 用の Amazon Chime SDK クライアントライブラリを使用する場合、動画を送信するためのコーデックプリファレンスを指定できます。

iOS および Android 用の Amazon Chime SDK クライアントライブラリは、デバイスの機能に基づいて自動的にコーデックを選択します。

動画コーデックの環境設定

JavaScript 用の Amazon Chime SDK クライアントライブラリでは、ウェブカメラとコンテンツ動画用に独立した動画コーデック環境設定を指定できます。

[AudioVideoControllerFacade.setVideoCodecSendPreferences](#) 関数を使用して、ウェブカメラ動画を送信するためのコーデック環境設定を設定します。リンク先は GitHub です。

設定は順序付けられた配列として渡され、最も優先度の高いコーデックが最初で、最も優先度低いコーデックが最後になります。

複数のコーデック設定を指定すると、サービスはセッション参加者全員がデコードできる最も好ましいコーデックを自動的に選択します。

次の例は、VP9 の動画コーデック環境設定をフォールバックオプションとして VP8 に設定する方法を示しています。

```
// A meeting session has already been created and stored in `this.meetingSession`
this.meetingSession.audioVideo.setVideoCodecSendPreferences(
  [
    VideoCodecCapability.vp9(),
    VideoCodecCapability.vp8()
  ]
);
```

環境設定には次のシナリオが適用されます。

- 最適 - クライアントは VP9 コーデックを使用して動画をエンコードします。
- ローカルフォールバック - クライアントが VP9 エンコーディングをサポートしていない場合、VP8 エンコーディングにフォールバックします。クライアントが VP8 エンコーディングをサポートしていない場合は、ブラウザとサービスでサポートされているコーデックにフォールバックします。
- リモートフォールバック - セッション内の別のクライアントに VP9 デコーダーがない場合、このクライアントは VP8 エンコーディングにフォールバックします。
- ローカル障がい - クライアントが VP9 または VP8 エンコーディングをサポートしていない場合、動画は送信されません。

コンテンツ共有のプリファレンスを設定するに

は、[ContentShareControllerFacade.setContentShareVideoCodecPreferences](#) 関数を使用して、コンテンツ動画を送信するためのコーデック環境設定を設定します。このリンク先は GitHub です。

次の例では、VP9 のコンテンツ動画コーデック環境設定を VP8 のフォールバックオプションで設定しています。

```
// A meeting session has already been created and stored in `this.meetingSession`
this.meetingSession.audioVideo.setContentShareVideoCodecPreferences(
  [
    VideoCodecCapability.vp9(),
    VideoCodecCapability.vp8()
  ]
);
```

ネットワーク構成

Amazon Chime SDK をクライアントアプリケーションに統合すると、SDK はバックエンドサービスに接続して、音声、動画、コンテンツ共有、データメッセージを送受信します。ユーザーのネットワークで Amazon Chime SDK サービスへのトラフィックがブロックされている場合、ユーザーはサービスを使用できなくなります。ネットワーク管理者はこの情報を使用して、Amazon Chime SDK サービスにアクセスできるようにネットワークを再設定できます。

Note

ネットワークを設定する際には、DNS の拡張メカニズム (EDNS0) をデフォルトで有効にする必要があります。これによってホスト情報が UDP パケットに適したサイズになることで、アプリケーションが Amazon Chime SDK サービスにアクセスできるようになります。

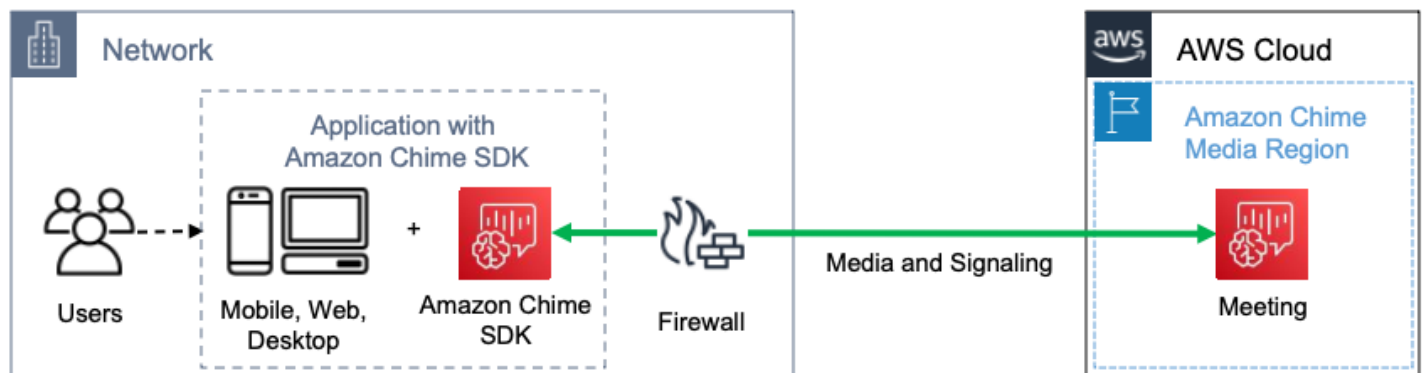
トピック

- [メディアとシグナリング向けの設定](#)
- [Amazon Voice Focus の設定](#)
- [エコーリダクションの設定](#)
- [背景の置換とぼかしの設定](#)
- [ブラウザコンテンツセキュリティポリシーの設定](#)
- [AppKeys と TenantIDs の使用](#)

メディアとシグナリング向けの設定

Amazon Chime SDK の音声、動画、コンテンツは、可能な限りユーザーデータグラムプロトコル (UDP) トランスポートを使用します。UDP がブロックされている場合、Amazon Chime SDK は、双方向メディアトランスポート用の Transport Layer Security (TLS) 接続の確立を試みます。Amazon Chime SDK のシグナリングとデータメッセージでは、Transmission Control Protocol (TCP) 接続と WebSocket 接続が使用されます。

次の図は、Amazon Chime SDK を実行するアプリケーションを含む一般的なネットワークを示しています。



Amazon Chime SDK は、メディアとシグナリングに以下の宛先とポートを使用します。

ドメイン	サブネット	ポート
*.chime.aws	99.77.128.0/18	TCP: 443 UDP:3478

このサブネットは [AWS IP アドレス範囲](#) 内の CHIME_MEETINGS サービスです。

Amazon Voice Focus の設定

iOS および Android 用の Amazon Chime SDK クライアントライブラリには、Amazon Voice Focus モジュールが含まれています。JavaScript 用の Amazon Chime SDK クライアントライブラリは、Amazon CloudFront から Amazon Voice Focus モジュールをダウンロードします。Windows 用の Amazon Chime SDK クライアントライブラリは、Voice Focus をサポートしていません。

Amazon Voice Focus は、以下の宛先とポートを使用します。

ドメイン	ポート
*.sdkassets.chime.aws	TCP: 443

このサブネットは [AWS IP アドレス範囲](#)内の CLOUDFRONT サービスです。

エコーリダクションの設定

JavaScript 用の Amazon Chime SDK クライアントライブラリは、Amazon CloudFront からエコーリダクションモジュールをダウンロードします。

エコーリダクションは、以下の宛先とポートを使用します。

ドメイン	ポート
*.sdkassets.chime.aws	TCP: 443

このサブネットは [AWS IP アドレス範囲](#)内の CLOUDFRONT サービスです。

背景の置換とぼかしの設定

JavaScript 用の Amazon Chime SDK クライアントライブラリは、Amazon CloudFront から背景の置換とぼかしモジュールをダウンロードします。

背景の置換とぼかしは、以下の宛先とポートを使用します。

ドメイン	ポート
*.sdkassets.chime.aws	TCP: 443

このサブネットは [AWS IP アドレス範囲](#)内の CLOUDFRONT サービスです。

ブラウザコンテンツセキュリティポリシーの設定

JavaScript 用の Amazon Chime SDK クライアントライブラリを使用してアプリケーションを構築する場合、アプリケーション内でブラウザコンテンツセキュリティポリシーを設定する必要があります。詳細については、GitHub の [コンテンツセキュリティポリシーのガイド](#)を参照してください。

AppKeys と TenantIDs の使用

AppKeys および TenantIDs を使用して、ネットワークから特定のアプリケーションの Amazon Chime SDK WebRTC メディアセッションへのアクセスを制限できます。

開発者は、UDP 経由でリアルタイムな動画を送受信するアプリケーションを作成するために、Amazon Chime SDK を使用します。アプリケーションユーザーには [CHIME_MEETINGS](#) サブネットへの UDP アクセスが必要です。組織 (ネットワーク所有者) は、AppKeys と TenantIDs を使用して、ネットワークからのアクセスを特定のアプリケーションの WebRTC メディアセッションのみに制限できます。

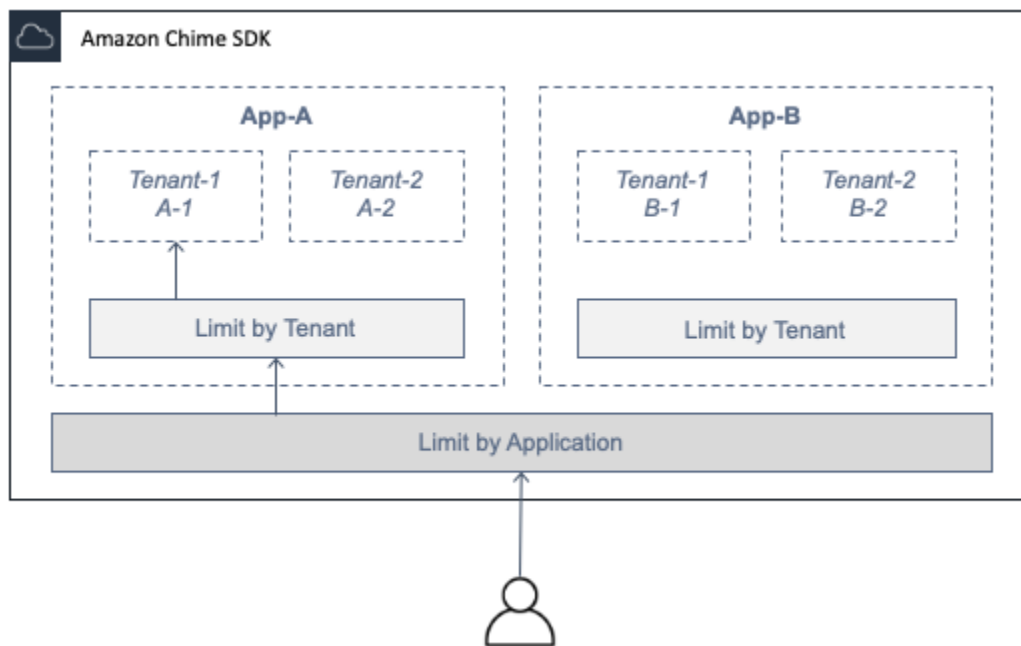
例 1: の使用 AppKeys

App-A と App-B が Amazon Chime SDK を使用する場合、組織は App-A がネットワークから WebRTC メディアセッションにアクセスすることを許可できますが、App-B と Amazon Chime SDK を使用するその他のアプリケーションをブロックできます。組織は、App-A の AppKey と HTTPS プロキシを使用してこれを行うことができます。詳細については、このトピックで後述する「[アクセスを特定のアプリケーションに制限する](#)」を参照してください。

例 2: AppKeys と TenantIDs

アプリ A が一般に公開され、多くの顧客によって使用されている場合は、組織でユーザーがセッションに参加している場合にのみ、アプリ A で自社のネットワークから WebRTC メディアセッションにアクセスできるようにして、他のすべてのアプリ A セッションへのアクセスをブロックするとよいでしょう。組織は、アプリケーションの AppKey、組織の TenantID、および HTTPS プロキシを使用してこれを行うことができます。詳細については、このトピックで後述する「[アクセスを特定のテナントに制限する](#)」を参照してください。

AppKeys および TenantIDs を使用するには、HTTPS ヘッダーをリクエストに追加できる HTTPS プロキシサーバーが必要です。次の図は、AppKeys と TenantIDs の仕組みを示しています。



この図では、アプリ A にはテナント A-1 と A-2 があり、アプリ B にはテナント B-1 と B-2 があります。この場合、AppKey のみが App-A に WebRTC メディアセッションへの接続を許可し、テナント ID はテナント A-1 のみをセッションに許可します。

トピック

- [アクセスを特定のアプリケーションに制限する](#)
- [アクセスを特定のテナントに制限する](#)
- [HTTPS ヘッダーの例](#)

アクセスを特定のアプリケーションに制限する

AppKey は、Amazon Chime が AWS アカウントごとに作成する、一貫性のある一意の 256 ビット値です。をお持ちでない場合は AppKey、Amazon Support にリクエストできます。複数の AWS アカウントがある場合は、すべてのアカウントに共通 AppKey をリクエストできます。

i Note

を AppKeys パブリックに安全に共有し、他の組織がネットワークからのアクセスを制限できるようにします。

Amazon Chime SDK は、セッションの作成に使用された AWS アカウント ID AppKey に基づいて、各 WebRTC メディアセッションを自動的に関連付けます。ネットワークからのアクセスを特定のアプリケーションに制限するには、次の手順を実行します。

1. すべてのアウトバウンドリクエストを HTTPS プロキシサーバー経由で CHIME_MEETINGS サブネットにルーティングします。
2. CHIME_MEETINGS サブネットへのすべてのアウトバウンドリクエストに次のヘッダーを追加するようにプロキシサーバーを設定します。

X-Amzn-Chime-App-Keys: ##### ##### AppKeys。

例えば、X-Amzn-Chime-App-Keys:AppKey-A,AppKey-B,AppKey-Cは、それらに関連付けられているアプリがサブネットにアクセス AppKeys できるようにします。

Amazon Chime SDK では X-Amzn-Chime-App-Keys ヘッダーに対するインバウンド WebRTC メディアセッションの接続を確認し、次のロジックを適用します。

1. X-Amzn-Chime-App-Keys ヘッダーが存在し、セッションの が含まれている場合は AppKey、接続を受け入れます。
2. X-Amzn-Chime-App-Keys ヘッダーは存在するが、セッションの が含まれていない場合は AppKey、403 エラーで接続を拒否します。
3. X-Amzn-Chime-App-Keys ヘッダーが存在しない場合、接続を受け入れます。ユーザーが組織のネットワークの外部からアプリケーションにアクセスできる場合は、セッションにもアクセスできます。

アクセスを特定のテナントに制限する

TenantID は開発者が作成した不透明な識別子です。TenantID については、次の点に注意してください。

- TenantIDsはアプリケーション間で一意である保証がないため、テナント TenantID AppKeyリストごとに を指定する必要があります。
- TenantID では大文字と小文字が区別されます。開発者の指示どおりに正確に入力してください。
- 組織で複数のアプリケーションへのアクセスを制限できますが、それらのアプリケーションの一部には TenantID しか指定できません。TenantID のないアプリケーションは、すべての WebRTC メディアセッションに接続できます。

メディアセッションを TenantIDs、デベロッパーはまず TenantIdsプロパティと TenantIDsのリストを [CreateMeeting](#) または [CreateMeetingWithAttendees](#) リクエストに追加する必要があります。

例:

```
CreateMeeting(..., TenantIds : [ tenantId1, tenantId2 ] )
```

組織のネットワークからのアクセスを特定のアプリケーションにおける自社の WebRTC メディアセッションに制限するには、次の手順を実行します。

1. 「[アクセスを特定のアプリケーションに制限する](#)」の手順を実行します。
2. アウトバウンド接続に X-Amzn-Chime-Tenants ヘッダーを追加するように HTTPS プロキシサーバーを設定します。この例に示すように、AppKeys と TenantIDsのリストを含めます。X-Amzn-Chime-Tenants: *AppKey-A:tenantId-A-1,tenantId-A-2;AppKey-B:tenantId-B-1,tenantId-B-2*

Amazon Chime SDK では X-Amzn-Chime-Tenants ヘッダーに対するインバウンド WebRTC メディアセッションの接続を確認し、次のロジックを適用します。

- ヘッダーにセッションの AppKey:tenantId が含まれている場合は、接続を承認します。
- ヘッダーにセッションの AppKey が含まれているものの、tenantId に一致しない場合は、403 エラーで接続を拒否します。
- ヘッダーにセッションの AppKey が含まれていない場合は、接続を承認します。
- ヘッダーにセッションの AppKey が含まれていても、そのセッションに 1 つ以上の許可された tenantId がない場合は、403 エラーで接続を拒否します。これは開発者のバグである可能性があります。
- ヘッダーが存在しない場合、接続を受け入れます。ユーザーが組織のネットワークの外部からアプリケーションにアクセスできる場合は、すべてのセッションにもアクセスできます。

HTTPS ヘッダーの例

次の例は、HTTPS ヘッダーで AppKeys および TenantIDs を使用する方法の一部を示しています。

1 つのアプリと 1 つのテナント

```
X-Amzn-Chime-App-Keys: AppKey
```

```
X-Amzn-Chime-Tenants: AppKey:orgId
```

ユーザーは、指定されたアプリで組織の WebRTC メディアセッションにのみアクセスできます。その他のアプリはすべてブロックされます。

1 つのアプリと 2 つのテナント

X-Amzn-Chime-App-Keys: *AppKey*

X-Amzn-Chime-Tenants: *AppKey:engineeringId,salesId*

ユーザーは、指定されたアプリでエンジニアリングとセールスのメディアセッションにのみアクセスできます。その他のアプリはすべてブロックされます。

2 つのアプリ、1 つは単一のテナントに限定

X-Amzn-Chime-App-Keys: *AppKey1,AppKey2*

X-Amzn-Chime-Tenants: *AppKey1:orgId*

ユーザーは、アプリ 1 では組織のメディアセッションにのみアクセスでき、アプリ 2 ではどのセッションにもアクセスできます。その他のアプリはすべてブロックされます。

会議イベント

Amazon Chime SDK は会議のライフサイクルイベントを送信します。これを使用して通知をトリガーし、ダウンストリームワークフローを開始できます。会議イベントを使用する例をいくつか次に示します。

- 参加者が Amazon Chime SDK ミーティングに参加または退席したときにメタデータを更新する。
- Amazon Chime SDK ミーティングのプッシュ通知または名簿を実装する。
- Amazon Chime SDK ミーティングでの動画およびコンテンツ共有の使用状況を測定する。

Amazon、Amazon Simple Notification Service (SNS) EventBridge、および Amazon Simple Queue Service (SQS) にイベントを送信できます。詳細については、「Amazon ユーザーガイド」の「[AWS のサービスからのイベント](#)」を参照してください。 EventBridge

Amazon Chime SDK ミーティングの開始

Amazon Chime SDK は、新しい会議の開始時に、このイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:MeetingStarted",
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "externalMeetingId": "87654321-4321-4321-1234-111122223333",
    "mediaRegion": "us-east-1"
  }
}
```

Amazon Chime SDK ミーティングの終了

Amazon Chime SDK は、アクティブな会議が終了すると、このイベントを送信します。

Note

効率を高めるため、サービスは [DeleteMeeting](#) API を呼び出すときにもこのイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
```

```
"version": "0",
"eventType": "chime:MeetingEnded",
"timestamp": 12344566754,
"meetingId": "87654321-4321-4321-1234-111122223333",
"externalMeetingId": "87654321-4321-4321-1234-111122223333",
"mediaRegion": "us-east-1"
}
}
```

Amazon Chime SDK 参加者の追加

Amazon Chime SDK は、新しい参加者がアクティブな会議に追加されると、このイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:AttendeeAdded",
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "attendeeId": "87654321-4321-4321-1234-111122223333",
    "externalUserId": "87654321-4321-4321-1234-111122223333",
    "externalMeetingId": "87654321-4321-4321-1234-111122223333",
    "mediaRegion": "us-east-1"
  }
}
```

Amazon Chime SDK 参加者が削除されます

Amazon Chime SDK は、[DeleteAttendee](#) API を使用してアクティブな会議から参加者を削除すると、このイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:AttendeeDeleted",
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "attendeeId": "87654321-4321-4321-1234-111122223333",
    "externalUserId": "87654321-4321-4321-1234-111122223333",
    "externalMeetingId": "87654321-4321-4321-1234-111122223333",
    "mediaRegion": "us-east-1"
  }
}
```

Amazon Chime SDK 参加者の許可

Amazon Chime SDK は、すでに会議に参加しているユーザーが同じ参加トークンを使用して会議に再度参加したときに、このイベントを送信します。例えば、ユーザーはデスクトップマシンからモバイルデバイスに切り替えることができます。これにより、会議が新しいデバイスに効果的に「引き継がれ」ます。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
```

```
"resources": []
"detail": {
  "version": "0",
  "eventType": "chime:AttendeeAuthorized",
  "timestamp": 12344566754,
  "meetingId": "87654321-4321-4321-1234-111122223333",
  "attendeeId": "87654321-4321-4321-1234-111122223333",
  "externalUserId": "87654321-4321-4321-1234-111122223333",
  "externalMeetingId": "87654321-4321-4321-1234-111122223333",
  "mediaRegion": "us-east-1"
}
```

Amazon Chime SDK 参加者の会議への参加

Amazon Chime SDK は、既存の参加者が指定したネットワークトランスポートを使用して Amazon Chime SDK ミーティングに参加したときに、このイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:AttendeeJoined",
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "attendeeId": "87654321-4321-4321-1234-111122223333",
    "externalUserId": "87654321-4321-4321-1234-111122223333",
    "networkType": "Voip",
    "externalMeetingId": "87654321-4321-4321-1234-111122223333",
    "mediaRegion": "us-east-1"
  }
}
```

Amazon Chime SDK 参加者の会議からの退席

Amazon Chime SDK は、既存の参加者が指定したネットワークトランスポートを使用して Amazon Chime SDK ミーティングから退席したときに、このイベントを送信します。

Note

サービスは、同じ「leave」アクションの `chime:AttendeeLeft` AND `chime:AttendeeDropped` イベントを送信しません。ドロップと脱退は異なるアクションであり、システムは各アクションに対応するイベントを送信します。例えば、接続が不十分な参加者が午前 11 時に会議に参加するとします。次のアクションが期待できます。

```
11:00 API - CreateAttendee, CreateMeetingWithAttendee, or BatchCreateAttendee
11:00 Event - chime:AttendeeAdded
11:01 Action - user joins meeting
11:01 Event - chime:AttendeeJoined
11:02 Action - user's connection drops
11:02 Event - chime:AttendeeDropped
11:03 Action - user's connection restored
11:03 Event - chime:AttendeeJoined
11:30 Action - user leaves meeting
11:30 Event - chime:AttendeeLeft
```

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:AttendeeLeft",
    "timestamp": 12344566754,
```

```
"meetingId": "87654321-4321-4321-1234-111122223333",
"attendeeId": "87654321-4321-4321-1234-111122223333",
"externalUserId": "87654321-4321-4321-1234-111122223333",
"networkType": "Voip",
"externalMeetingId": "87654321-4321-4321-1234-111122223333",
"mediaRegion": "us-east-1"
}
}
```

Amazon Chime SDK 参加者が会議から削除されました

Amazon Chime SDK は、現在の参加者が Amazon Chime SDK ミーティングから切断されたときに、通常、接続が不十分なためにこのイベントを送信します。サービスがパケットを 10~15 秒間受信しない場合、サービスは削除されたサイエンティフィックを考慮し、イベントを発行します。

サービスは通常、ドロップアクションをトリガーしますが、クライアントはアクションをトリガーすることもできます。例えば、ユーザーがラップトップを Wi-Fi からイーサネットに切り替えたとします。これはネットワークアダプタの変更を構成し、接続はリセットされます。次に、は WebSocket をリセットし、結合されたドロップ結合アクションをトリガーします。

Note

サービスは、同じ「leave」アクションの `chime:AttendeeLeft` AND `chime:AttendeeDropped` イベントを送信しません。ドロップと脱退は異なるアクションであり、システムは各アクションに対応するイベントを送信します。例えば、接続が不十分な参加者が午前 11 時に会議に参加するとします。次のアクションが期待できます。

```
11:00 API - CreateAttendee, CreateMeetingWithAttendee, or BatchCreateAttendee
11:00 Event - chime:AttendeeAdded
11:01 Action - user joins meeting
11:01 Event - chime:AttendeeJoined
11:02 Action - user's connection drops
11:02 Event - chime:AttendeeDropped
11:03 Action - user's connection restored
11:03 Event - chime:AttendeeJoined
11:30 Action - user leaves meeting
11:30 Event - chime:AttendeeLeft
```


Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:AttendeeDropped",
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "attendeeId": "87654321-4321-4321-1234-111122223333",
    "externalUserId": "87654321-4321-4321-1234-111122223333",
    "networkType": "Voip",
    "externalMeetingId": "87654321-4321-4321-1234-111122223333",
    "mediaRegion": "us-east-1"
  }
}
```

Amazon Chime SDK 参加者による動画ストリームの開始

Amazon Chime SDK は、既存の参加者が動画ストリームを開始したときに、このイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
```

```
"detail": {
  "version": "0",
  "eventType": "chime:AttendeeVideoStarted",
  "timestamp": 12344566754,
  "meetingId": "87654321-4321-4321-1234-111122223333",
  "attendeeId": "87654321-4321-4321-1234-111122223333",
  "externalUserId": "87654321-4321-4321-1234-111122223333",
  "externalMeetingId": "87654321-4321-4321-1234-111122223333",
  "mediaRegion": "us-east-1"
}
```

Amazon Chime SDK 参加者による動画ストリームの停止

Amazon Chime SDK は、既存の参加者が動画ストリームを停止したときに、このイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:AttendeeVideoStopped",
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "attendeeId": "87654321-4321-4321-1234-111122223333",
    "externalUserId": "87654321-4321-4321-1234-111122223333",
    "externalMeetingId": "87654321-4321-4321-1234-111122223333",
    "mediaRegion": "us-east-1"
  }
}
```

Amazon Chime SDK 参加者による共有画面の開始

Amazon Chime SDK は、既存の参加者が画面の共有を開始したときに、このイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:AttendeeContentJoined",
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "attendeeId": "87654321-4321-4321-1234-111122223333",
    "externalUserId": "87654321-4321-4321-1234-111122223333",
    "externalMeetingId": "87654321-4321-4321-1234-111122223333",
    "mediaRegion": "us-east-1"
  }
}
```

Amazon Chime SDK 参加者による共有画面の停止

Amazon Chime SDK は、既存の参加者が画面の共有を停止したときに、このイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
```

```
"detail-type": "Chime Meeting State Change",
"time": "yyyy-mm-ddThh:mm:ssZ",
"resources": []
"detail": {
  "version": "0",
  "eventType": "chime:AttendeeContentLeft",
  "timestamp": 12344566754,
  "meetingId": "87654321-4321-4321-1234-111122223333",
  "attendeeId": "87654321-4321-4321-1234-111122223333",
  "externalUserId": "87654321-4321-4321-1234-111122223333",
  "externalMeetingId": "87654321-4321-4321-1234-111122223333",
  "mediaRegion": "us-east-1"
}
}
```

Amazon Chime SDK 出席者の機能が更新されました

Amazon Chime SDK は、既存の出席者の機能が更新されたときにこのイベントを送信します。

Example イベントデータ

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "id": "12345678-1234-1234-1234-111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": [],
  "detail": {
    "version": "0",
    "eventType": "chime:AttendeeCapabilitiesUpdated",
    "success": "1", // value can be 1 or 0. 1 means success, 0 means failure
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "externalMeetingId": "mymeeting",
    "attendeeId": "attendeeId",
    "externalUserId": "externalUserId"
    "mediaRegion": "us-east-1"
    "attendeeCapabilities": {
      "audio": "SendReceive",
      "video": "SendReceive",

```

```
    "content": "SendReceive"  
  }  
}  
}
```

Amazon Chime SDK 参加者によるコンテンツの会議への参加

Amazon Chime SDK は、指定したネットワークトランスポートを使用してコンテンツ共有が Amazon Chime SDK ミーティングに参加したときに、このイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{  
  "version": "0",  
  "source": "aws.chime",  
  "account": "111122223333",  
  "region": "us-east-1",  
  "detail-type": "Chime Meeting State Change",  
  "time": "yyyy-mm-ddThh:mm:ssZ",  
  "resources": [],  
  "detail": {  
    "version": "0",  
    "eventType": "chime:AttendeeContentJoined",  
    "timestamp": 12344566754,  
    "meetingId": "87654321-4321-4321-1234-111122223333",  
    "attendeeId": "87654321-4321-4321-1234-111122223333",  
    "externalUserId": "87654321-4321-4321-1234-111122223333",  
    "networkType": "Voip",  
    "externalMeetingId": "87654321-4321-4321-1234-111122223333",  
    "mediaRegion": "us-east-1"  
  }  
}
```

Amazon Chime SDK 参加者によるコンテンツの会議からの退席

Amazon Chime SDK は、指定したネットワークトランスポートを使用してコンテンツ共有が Amazon Chime SDK ミーティングから退席したときに、このイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:AttendeeContentLeft",
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "attendeeId": "87654321-4321-4321-1234-111122223333",
    "externalUserId": "87654321-4321-4321-1234-111122223333",
    "networkType": "Voip",
    "externalMeetingId": "87654321-4321-4321-1234-111122223333",
    "mediaRegion": "us-east-1"
  }
}
```

Amazon Chime SDK 参加者によるコンテンツの会議からの途中退席

Amazon Chime SDK は、コンテンツ共有が Amazon Chime SDK ミーティングから切断されたときに、通常、低帯域幅が原因でこのイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:AttendeeContentDropped",
    "timestamp": 12344566754,
```

```
"meetingId": "87654321-4321-4321-1234-111122223333",
"attendeeId": "87654321-4321-4321-1234-111122223333",
"externalUserId": "87654321-4321-4321-1234-111122223333",
"networkType": "Voip",
"externalMeetingId": "87654321-4321-4321-1234-111122223333",
"mediaRegion": "us-east-1"
}
}
```

Amazon Chime SDK 参加者によるコンテンツの動画ストリームの開始

Amazon Chime SDK は、コンテンツ共有がビデオのストリーミングを開始すると、このイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:AttendeeContentVideoStarted",
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "attendeeId": "87654321-4321-4321-1234-111122223333",
    "externalUserId": "87654321-4321-4321-1234-111122223333",
    "externalMeetingId": "87654321-4321-4321-1234-111122223333",
    "mediaRegion": "us-east-1"
  }
}
```

Amazon Chime SDK 参加者によるコンテンツの動画ストリームの停止

Amazon Chime SDK は、コンテンツ共有がビデオのストリーミングを停止すると、このイベントを送信します。

Example イベントデータ

次の例は、このイベントのデータを示しています。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:AttendeeContentVideoStopped",
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "attendeeId": "87654321-4321-4321-1234-111122223333",
    "externalUserId": "87654321-4321-4321-1234-111122223333",
    "externalMeetingId": "87654321-4321-4321-1234-111122223333",
    "mediaRegion": "us-east-1"
  }
}
```

Amazon CloudWatch メトリクス

Amazon Chime SDK を使用すると、サービスと使用状況のメトリクスが に送信されます CloudWatch。メトリクスを使用すると、CloudWatch グラフとダッシュボードを使用して Amazon Chime SDK サービスの使用状況をモニタリングできます。メトリクスは、呼び出す各 API のデータをキャプチャします。

以下のセクションでは、メトリクスをリストして説明します。

トピック

- [サービスマトリクス](#)
- [API 使用状況メトリクス](#)

サービスマトリクス

Amazon Chime SDK は、以下のサービスマトリクスを AWS/ChimeSDK 名前空間に公開します。

メトリクス	単位	説明
AttendeeAuthorizationSuccess	カウント	成功した承認試行の総数。成功とは、参加者が会議への参加を許可されたことを意味します。
AttendeeAuthorizationError	カウント	承認失敗の総数。これは、参加者が会議に参加できなかったことを示します。
AttendeeAudioDrops	カウント	音声ドロップの総数。
AttendeeContentDrops	カウント	コンテンツ共有ドロップの総数。
MeetingSQSNotificationErrors	カウント	SQS 通知エラーの総数。
MeetingSNSNotificationErrors	カウント	SNS 通知エラーの総数。

API 使用状況メトリクス

API 使用状況メトリクスは、AWS サービスクォータに対応します。使用量がサービスクォータに近づいたときに警告するアラームを設定することもできます。サービスクォータとの統合の詳細については CloudWatch、「Amazon ユーザーガイド」の「[AWS 使用状況メトリクス](#)」を参照してください。 CloudWatch

Amazon Chime SDK は、以下の API メトリクスを ChimeSDK サービス名と共に AWS/Usage 名前空間に公開します。

メトリクス	説明
CallCount	Amazon Chime SDK 内の API に対して行われた呼び出しの総数。SUM は、指定した期間に

メトリクス	説明
	API に対して行われた呼び出しの総数を表します。
ErrorCount	Amazon Chime SDK 内の API によってスローされたエラーの総数。SUM は、指定した期間に API に対して行われた呼び出しの総数を表します。
ThrottleCount	Amazon Chime SDK 内の API によってスローされたスロットリングエラーの総数。SUM は、指定した期間に API に対して行われた呼び出しの総数を表します。

Amazon Chime SDK は、使用状況メトリクスを以下のディメンションと共に AWS/Usage 名前空間に公開します。

ディメンション	説明
Service	リソースを含む AWS サービスの名前。Amazon Chime SDK 使用状況メトリクスの場合、このディメンションの値は ChimeSDK です。
タイプ	報告されるエンティティのタイプ。Amazon Chime SDK の使用状況メトリクスの有効な値は API のみです。
リソース	メトリクスを報告するリソースのタイプ。Amazon Chime SDK 使用状況メトリクスの場合、このディメンションの値は API の名前です。
Class	追跡されているリソースのクラス。Amazon Chime SDK のメトリクスの有効な値は None のみです。

Amazon Chime SDK メディアパイプラインを作成する

⚠ Important

お客様とそのエンドユーザーは、Amazon Chime SDK ミーティングのレコーディングが電子通信記録に関する法律または規制の対象となる場合があることを理解する必要があります。レコーディングに関するすべての適用法を遵守することは、お客様とそのエンドユーザーの責任です。これには、録画されたセッションのすべての参加者にセッションまたは通信がレコーディングされていることを適切に通知し、同意を得ることが含まれます。

お客様とそのエンドユーザーは、メディアライブコネクタサービスを使用してストリーミングされるすべてのコンテンツについて責任を負い、かかるコンテンツが法律に違反したり、第三者の権利を侵害または悪用したり、Amazon との契約の重要な条件に違反したりしないようにする必要があります。

Amazon Chime SDK ミーティングをキャプチャまたはストリーミングするには、メディアストリームパイプラインを作成します。メディアパイプラインは、以下のパイプラインのいずれかで構成できます。

- **メディアキャプチャ** – メディアキャプチャパイプラインを使用し、音声、動画、コンテンツ共有ストリームに加えて、会議イベントやデータメッセージをキャプチャします。すべてのメディアキャプチャパイプラインでは、作成した [Amazon Simple Storage Service \(S3\)](#) バケットにデータを保存します。Amazon Chime SDK ミーティングごとに 1 つのメディアキャプチャパイプラインを作成できます。詳細については、このセクションで後述する「[パイプライン作成の概要](#)」を参照してください。
- **メディア連結** – メディア連結パイプラインを使用して、メディアキャプチャパイプラインからのアーティファクトを連結します。連結パイプラインは、メディアキャプチャおよびライブコネクタパイプラインとは独立して機能します。詳細については、このセクションで後述する「[メディア連結パイプラインを作成する](#)」を参照してください。
- **メディアライブコネクタ** – メディアライブコネクタパイプラインを使用して、Amazon Chime SDK ミーティングを RTMP エンドポイントにストリーミングできるサービスに接続します。Amazon Chime SDK ミーティングごとにメディアライブコネクタパイプラインを 1 つまで作成できます。詳細については、このセクションで後述する「[メディアライブコネクタパイプラインを作成する](#)」を参照してください。
- **メディアストリーム** – メディアストリームパイプラインを使用して、会議の全参加者の個別の音声と、メディア連結パイプラインによって生成された混合音声をキャプチャします。すべてのメ

ディアストリームパイプラインでは、データを [Amazon Kinesis Video Streams \(KVS\)](#) に保存します。詳細については、このセクションで後述する「[メディアストリームパイプラインを作成する](#)」を参照してください。

作成されるパイプラインは、使用する名前空間によって異なります。Chime 名前空間を使用する場合、作成できるのはメディアキャプチャパイプラインのみです。ChimeSdkMediaPipelines 名前空間を使用する場合は、メディア連結パイプラインとメディアライブコネクタパイプラインの作成や合成機能の使用も可能です。ChimeSdkMediaPipelines 名前空間に移行する必要がある場合は、[ChimeSdkMediaPipelines 名前空間への移行](#) を参照してください。

次の表では、各リージョンにおけるアクティブなメディアパイプラインのデフォルトの上限が一覧表示されています。それぞれのパイプラインのタイプは上限に加算されます。

リージョン	アクティブなパイプラインのデフォルトの上限
us-east-1	100
us-west-2	10
ap-northeast-1	10
ap-northeast-2	10
ap-south-1	10
ap-southeast-1	10
ap-southeast-2	10
ca-central-1	10
eu-central-1	10
eu-west-2	10

Note

リージョンの制限を超える
と、[CreateMediaCapturePipeline](#)、[CreateMediaConcatenationPipeline](#)、および
[CreateMediaLiveConnectorPipeline](#) APIs はリソース制限超過の例外をスローします。
AWS コンソールの Service Quotas ページを使用してアクティブなパイプラインの制限を調
整するか、[カスタマーサポート担当者](#) にお問い合わせください。Amazon Chime SDK ミー
ティングでの上限の詳細については、「[Amazon Chime SDK サービスクォータ](#)」を参照して
ください。

開始する前に、クライアントアプリケーションを Amazon Chime SDK クライアントライブラリと
統合する必要があります。詳細については、「[クライアントライブラリと統合する](#)」を参照してく
ださい。メディアパイプラインの詳細については、「[Capture Amazon Chime SDK Meetings Using
media pipelines](#)」を参照してください。

トピック

- [ChimeSdkMediaPipelines 名前空間への移行](#)
- [パイプライン作成の概要](#)
- [メディアキャプチャパイプラインを作成する](#)
- [メディア連結パイプラインを作成する](#)
- [メディアライブコネクタパイプラインを作成する](#)
- [音声と動画を単一のビューに合成する](#)
- [メディアストリームパイプラインを作成する](#)
- [メディアパイプライン用のサービスにリンクされたロールを作成する](#)
- [メディアパイプラインイベントを使用する](#)
- [トランスクリプトを解析する](#)
- [パイプラインを停止するためのベストプラクティス](#)

ChimeSdkMediaPipelines 名前空間への移行

ChimeSdkMediaPipelines 名前空間を使用すると、任意の AWS リージョンのメディアパイプ
ライン API エンドポイントをアドレス指定できます (そのリージョンでそれらのエンドポイントを利用
できる場合)。Amazon Chime SDK を使い始めたばかりの場合は、この名前空間を使用してくださ
い。リージョンの詳細については、このガイドの「[利用できるリージョン](#)」を参照してください。

[Amazon Chime](#) 名前空間を使用する既存のアプリケーションでは、専用の名前空間への移行を計画する必要があります。

トピック

- [パイプラインを移行する理由](#)
- [パイプラインを移行する前に](#)

パイプラインを移行する理由

以下の理由から、ChimeSdkMediaPipelines 名前空間に移行することをお勧めします。

API エンドポイントの選択

Amazon Chime SDK メディアキャプチャ名前空間は、API エンドポイントを利用できる任意のリージョンの API エンドポイントを使用できる唯一の API 名前空間です。リージョンの詳細については、「[利用できるリージョン](#)」を参照してください。us-east-1 以外の API エンドポイントを使用する場合は、ChimeSdkMediaPipelines 名前空間を使用する必要があります。現在のエンドポイントの詳細については、このガイドの「[API マッピング](#)」を参照してください。

更新および新規追加されたメディアパイプライン API

ChimeSdkMediaPipelines 名前空間のメディアパイプライン API のみを追加または更新します。

パイプラインを移行する前に

移行する前に、名前空間の相違点に注意してください。以下の表では、名前空間の一覧と説明を示しています。

項目	メディアパイプライン名前空間	Chime 名前空間
名前空間名	ChimeSdkMediaPipelines	Chime
リージョン	複数	us-east-1 のみ
エンドポイント	https://media-pipelines-chime.region.amazonaws.com	https://service.chime.amazonaws.com
サービスプリンシパル	mediapipelines.chime.amazonaws.com	chime.amazonaws.com

項目	メディアパイプライン名前空間	Chime 名前空間
API	メディアパイプラインの API のみ	メディアパイプラインの API と Amazon Chime のその他の部分
会議	us-west-2 、 ap-southeast-1 、 eu-central-1 リージョンのメディアパイプラインは、Amazon Chime SDK 会議の名前空間で作成された会議でのみ機能します。us-east-1 リージョンのメディアパイプラインは、どちらの名前空間のどの会議エンドポイントによって作成された会議でも動作します。	メディアパイプラインは、どちらの名前空間のどの会議エンドポイントによって作成された会議でも動作します。
デフォルトのアクティブメディアパイプライン	us-east-1 リージョンでは 100、us-west-2、ap-southeast-1、および eu-central-1 の各リージョンでは 10。	us-east-1 の 100 のみ。
サービスリンクロール	AWSServiceRoleForAmazonChimeSDKMediaPipelines	
タグ	使用可能	メディアパイプライン API では使用できません。
CloudTrail イベントソース	chime-sdk-media-pipelines.amazonaws.com	chime.amazonaws.com
メディアライブコネクタ	使用可能	メディアパイプライン API では使用できません。

項目	メディアパイプライン名前空間	Chime 名前空間
コンポジット	使用可能	メディアパイプライン API では使用できません。
連結	使用可能	利用不可。

以下のリストは、Chime 名前空間と AWSChimeSdkMediaPipelines 名前空間の違いの詳細を示しています。

名前空間名

Amazon Chime SDK 名前空間では `AWS.Chime` という正式名を使用します。Amazon Chime SDK メディアパイプライン名前空間では、`AWS.ChimeSDKMediaPipelines` という正式な名前を使用します。名前の正確な形式はプラットフォームによって異なります。

例えば、Node.js コードの次の行は `chime` 名前空間をアドレス指定します。

```
const chimeMediaPipelines = AWS.Chime();
```

メディアパイプライン SDK 名前空間に移行するには、そのコードを新しい名前空間とエンドポイントリージョンで更新します。

```
const chimeMediaPipelines = AWS.ChimeSDKMediaPipelines({ region: "eu-central-1" });
```

リージョン

Amazon Chime 名前空間は、US-EAST-1 リージョンの API エンドポイントのみをアドレス指定します。Amazon Chime SDK メディアパイプライン名前空間は、任意のリージョンの Amazon Chime SDK メディアパイプライン API エンドポイントをアドレス指定できます (そのリージョンにそれらのエンドポイントが存在する場合)。メディアパイプラインのリージョンの最新リストについては、本ガイドの「[利用できるリージョン](#)」を参照してください。

エンドポイント

メディアキャプチャパイプラインを変更するには、パイプラインを作成したのと同じエンドポイントを使用する必要があります。例えば、eu-central-1 のエンドポイント経由でパイプラインを作成した場合は、eu-central-1 を使用してそのパイプラインを操作する必要があります。現在のエンドポイントの詳細については、このガイドの「[API マッピング](#)」を参照してください。

サービスプリンシパル

[Amazon Chime SDK メディアパイプライン](#) 名前空間は、新しいサービスプリンシパル `mediapipelines.chime.amazonaws.com` を使用します。サービスへのアクセスを許可する Amazon S3 バケットポリシーまたはその他の IAM ポリシーがある場合は、それらのポリシーを更新して新しいサービスプリンシパルへのアクセスを許可する必要があります。

例えば、メディアパイプラインを作成する場合、「[Amazon S3 バケットの作成](#)」に記載されているポリシー権限を新しいサービスプリンシパルに追加する必要があります。ポリシーの詳細については、「IAM ユーザーガイド」の「[AWS JSON ポリシーの要素: プリンシパル](#)」を参照してください。

API

Amazon Chime SDK メディアパイプライン名前空間には、メディアパイプラインを作成および管理する API のみが含まれています。Amazon Chime 名前空間には、メディアパイプライン、会議、および Amazon Chime サービスのその他の部分用の API が含まれています。

会議

IAD リージョンのメディアパイプラインは、どちらの名前空間のどの会議エンドポイントによって作成された会議でも動作します。

サービスリンクロール

Amazon Chime SDK メディアパイプライン名前空間専用。AWS `ServiceRoleForAmazonChimeSDKMediaPipelines` ロールを作成します。

タグ

[Amazon Chime SDK メディアパイプライン](#) 名前空間は、タグをサポートします。1 つ以上のタグを含む [CreateMediaCapturePipeline](#) API または [CreateMediaLiveConnectorPipeline](#) API を呼び出す場合、ロールには `TagResource` オペレーションを呼び出すアクセス許可が必要です。

パイプライン作成の概要

Amazon Chime SDK メディアパイプラインを作成するには、複数の手順から成るプロセスに従います。また、複数のタイプのパイプラインを作成できます。次のリストでは、作成プロセスについて概説し、さまざまなタイプのパイプラインの作成に関する詳細情報へのリンクを提示しています。

- Amazon S3 バケットを作成する。バケットは、会議と同じ AWS リージョンに作成する必要があります。詳細については、「[Amazon S3 バケットの作成](#)」を参照してください。

- `AWSServiceRoleForAmazonChimeSDKMediaPipelines` という名前のサービスにリンクされたロールを作成します。これにより、メディアパイプラインでユーザーに代わって会議にアクセスできるようになります。詳細については、「[メディアパイプライン用のサービスにリンクされたロールを作成する](#)」を参照してください。
- [Amazon Chime SDK メディアパイプライン API](#) を操作するための十分なアクセス許可を持つ IAM ロールを作成します。このロールを作成するには、IAM コンソールから [AmazonChimeSDK](#) 管理ポリシーを追加することをお勧めします。このポリシーには必要な API が含まれています。

IAM ロールには、すべてのリソースで Amazon S3 API を呼び出すアクセス許可も必要です。[GetBucketPolicy](#) 次の例は、そのための一般的なバケットポリシーを示しています。

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

それらの項目を取得したら、これらのトピックを参照してパイプラインの作成に関する情報を確認します。

- [メディアキャプチャパイプラインを作成する](#)
- [メディア連結パイプラインを作成する](#)
- [メディアライブコネクタパイプラインを作成する](#)
- [メディアストリームパイプラインを作成する](#)

メディアキャプチャパイプラインを作成する

メディアキャプチャパイプラインでは、音声、動画、コンテンツ共有ストリームに加えて、会議イベントやデータメッセージをキャプチャします。すべてのメディアキャプチャパイプラインでは、作成した [Amazon Simple Storage Service](#) (S3) バケットにデータを保存します。Amazon Chime SDK ミーティングごとに 1 つのメディアキャプチャパイプラインを作成できます。

以下のセクションでは、メディアキャプチャパイプラインを作成する方法について説明します。記載されている順序に従ってください。

トピック

- [Amazon S3 バケットの作成](#)
- [Amazon S3 バケットのサーバー側の暗号化を有効にする](#)
- [メディアキャプチャパイプラインを作成する](#)
- [メディアキャプチャアーティファクトを処理する](#)
- [音声フォルダを設定する](#)
- [動画フォルダを設定する](#)
- [データチャンネルフォルダ内のメッセージについて](#)
- [Amazon S3 バケットフォルダ構造について](#)
- [会議イベントファイルについて](#)
- [文字起こしファイルについて](#)
- [データストリームを連結する](#)

Amazon S3 バケットの作成

Amazon S3 コンソール、AWS SDKs、または AWS CLI を使用して Amazon S3 バケットを作成できます。詳細については、「Amazon Simple Storage Service (S3) ユーザーガイド」の「[バケットの作成](#)」を参照してください。

メディアキャプチャパイプラインの Amazon S3 バケットは、Amazon Chime SDK ミーティングと同じ AWS アカウントに属している必要があります。さらに、Amazon Chime SDK サービスプリンシパル mediapipelines.chime.amazonaws.com に `s3:PutObject` および `s3:PutObjectAcl` アクセス許可を付与する必要があります。これを行うには、Amazon S3 コンソールまたは AWS Command Line Interface () を使用します。AWS CLI。Amazon S3 バケットは、利用可能な [Amazon Chime SDK メディアリージョン](#) のいずれかに属している必要があります。

Note

必ず IAM ユーザーにポリシーを追加して、バケットへのアクセスを許可してください。また、デフォルトで AWS を無効にするリージョンを使用する場合は、そのリージョンに Amazon S3 バケットが必要です。デフォルトでは、AWS によって以下のリージョンが無効になっています。有効にするまで、これらのリージョンで会議リソースをホストすることはできません。

- アフリカ (ケープタウン)
- アジアパシフィック (香港)
- アジアパシフィック (ジャカルタ)
- ヨーロッパ (ミラノ)
- 中東 (バーレーン)

これらのリージョンのいずれかを使用する場合、Amazon S3 バケットが必要です。これは、Amazon S3 API を使用し、デフォルトでブロックされておらず、既にバケットがあるリージョンと通信する場合にも適用されます。ブロックされたリージョンの有効化に関する詳細については、「AWS 全般リファレンス」の「[AWS リージョンの管理](#)」を参照してください。

バケットを作成したら、その ARN を記録します。これを使用して、メディアキャプチャパイプラインを作成します。

次の例は、Amazon S3 バケットのポリシーを示しています。

```
{
  "Version": "2012-10-17",
  "Id": "AWSChimeMediaCaptureBucketPolicy",
  "Statement": [
    {
      "Sid": "AWSChimeMediaCaptureBucketPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "mediapipelines.chime.amazonaws.com"
      },
      "Action": [ "s3:PutObject", "s3:PutObjectAcl" ],
      "Resource": "arn:aws:s3:::Bucket_Name/*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "Account_Id"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:chime:*:Account_Id:*"
        }
      }
    }
  ]
}
```

```
]
}
```

Amazon S3 バケットのサーバー側の暗号化を有効にする

Amazon Simple Storage Service (Amazon S3) バケットのサーバー側の暗号化を有効にするには、以下のタイプの暗号化キーを使用します。

- Amazon S3 マネージドキー
- Key Management Service (KMS) のカスタマーマネージド AWS キー

Note

Key Management Service は、カスタマーマネージドキーと AWS マネージドキーの 2 種類のキーをサポートしています。Amazon Chime SDK ミーティングは、カスタマーマネージドキーのみをサポートしています。

Amazon S3 マネージドキーを使用する

Amazon S3 コンソール、CLI、REST API を使用して、Amazon S3 バケットのサーバー側の暗号化を有効にします。いずれの場合でも、暗号化キータイプとして Amazon S3 キーを選択します。これ以上の操作は不要です。バケットをメディアキャプチャに使用すると、アーティファクトがサーバー側でアップロードされ、暗号化されます。詳細については、「[Amazon S3 ユーザーガイド](#)」の「[Amazon S3 の暗号化の指定](#)」をご参照ください。

所有しているキーを使用する

管理しているキーで暗号化を有効にするには、カスタマーマネージドキーで Amazon S3 バケットのサーバー側の暗号化を有効にしてから、Amazon Chime でそのキーを使用してアップロードされたアーティファクトを暗号化できるようにするステートメントをキーポリシーに追加する必要があります。

1. KMS でカスタマーマネージドキーを作成します。詳細については、「[Amazon S3 ユーザーガイド](#)」の [AWS KMS 「\(SSE-KMS\) によるサーバー側の暗号化」の指定](#) を参照してください。

Amazon S3

2. `GenerateDataKey` アクションを実行し、Amazon Chime SDK サービスプリンシパルである `mediapipelines.chime.amazonaws.com` で使用するキーの生成を可能にするステートメントをキーポリシーに追加します。

この例は、典型的なステートメントを示しています。

```
...
{
  "Sid": "MediaPipelineSSEKMS",
  "Effect": "Allow",
  "Principal": {
    "Service": "mediapipelines.chime.amazonaws.com"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:SourceAccount": "Account_Id"
    },
    "ArnLike": {
      "aws:SourceArn": "arn:aws:chime:*:Account_Id:*"
    }
  }
}
...
```

3. メディア連結パイプラインを使用する場合は、Amazon Chime SDK サービスプリンシパルである `mediapipelines.chime.amazonaws.com` に `kms:Decrypt` アクションの使用を許可するステートメントをキーポリシーに追加します。
4. キーによるサーバー側の暗号化を有効にするように Amazon S3 バケットを設定します。

メディアキャプチャパイプラインを作成する

Amazon S3 バケットを作成して設定した後、メディアキャプチャパイプラインを作成します。

メディアキャプチャパイプラインを作成するには

- [CreateMediaCapturePipeline](#) API を呼び出します。

バケット ARN を `SinkArn` パラメータとして使用します。

成功すると、会議に参加してキャプチャする参加者が Amazon Chime SDK によって作成されます。

メディアキャプチャパイプラインを作成してそのアクセス許可を設定した後、5 秒間のメディアチャンクを 1 つのファイルに連結するメディア連結パイプラインを作成します。詳細については、このセクションで後述する「[メディア連結パイプラインを作成する](#)」を参照してください。

メディアキャプチャアーティファクトを処理する

Amazon Chime SDK ミーティング中に、メディアキャプチャパイプラインでは次のタイプのアーティファクトを作成します。

- 音声
- 動画
- データチャネルメッセージ
- 会議イベント
- 文字起こしメッセージ

パイプラインで Amazon S3 バケット内の一連のフォルダにアーティファクトを作成します。また、特定のタイプのアーティファクトを制限するように音声および動画フォルダを設定できます。以下のセクションでは、フォルダ構造、フォルダの設定方法、Amazon S3 バケットのアクセス許可の設定方法、アーティファクトファイルの連結方法について説明します。

音声フォルダを設定する

音声フォルダには、混合音声ストリームの 5 秒の MP4 ファイルが含まれています。つまり、参加者全員の音声に加えて、発言中のスピーカーの動画が含まれます。このフォルダには、会議全体のファイルが含まれています。必要に応じて、音声アーティファクトのみを含むようにフォルダを設定できます。各ファイル名には yyyy-mm-dd-hour-min-seconds-milleseconds のタイムスタンプが含まれています。タイムスタンプは UTC で、開始時刻を示しています。音声アーティファクトのみを含むようにフォルダを設定できます。

```
"ArtifactsConfiguration": {
  "Audio": {
    "MuxType": "AudioOnly"
  },
  "Content": {
    "State": "Disabled"
  },
  "Video": {
    "State": "Disabled"
  }
}
```

```
    }  
  }  
}
```

動画フォルダを設定する

動画フォルダには、動画ストリーム (および、API リクエストで指定されている場合はコンテンツ共有ストリーム) を含む 5 秒の MP4 ファイルが含まれています。各ファイル名には、<yyyy-mm-dd-hour-min-seconds-milliseconds>-<attendeelD> という参加者 ID 付きのタイムスタンプが含まれます。コンテンツ共有の動画チャUNKは <yyyy-mm-dd-hour-min-seconds-milliseconds>-<attendeelD>#content.mp4 として追加されます。動画アーティファクトのみを含むようにフォルダを設定できます。

```
"ArtifactsConfiguration": {  
  "Audio": {  
    "MuxType": "AudioOnly"  
  },  
  "Content": {  
    "State": "Disabled"  
  },  
  "Video": {  
    "MuxType": "VideoOnly"  
    "State": "Enabled"  
  }  
}
```

データチャンネルフォルダ内のメッセージについて

data-channel フォルダには .txt 形式のデータメッセージが含まれており、各メッセージは JSON オブジェクトです。メッセージはすべての設定オプションと共に表示されます。ファイル名には yyyy-mm-dd-hour-min-seconds-milliseconds のタイムスタンプが含まれます。この例は、メッセージ内のデータフィールドを示しています。

```
{  
  "Timestamp": "string",  
  "Topic": "string",  
  "Data": "string",  
  "SenderAttendeeId": "string"  
}
```


Amazon S3 バケットフォルダ構造について

メディアキャプチャパイプラインの Amazon S3 バケットでは、このようなフォルダ構造が使用されています。

```
S3 bucket path/
audio
video
data-channel
meeting-events
transcription-messages
```

会議イベントファイルについて

meeting-events フォルダには .txt 形式の会議イベントが含まれており、各イベントは JSON オブジェクトです。メッセージはすべての設定オプションと共に表示されます。ファイル名には <yyyy-mm-dd-hour-min-seconds-milleseconds> のタイムスタンプが含まれます。この例は、一般的なイベントファイルのフィールドとデータを示しています。

```
{
  "Timestamp": "string",
  "EventType": "AttendeeJoined | AttendeeLeft | AttendeeVideoJoined |
AttendeeVideoLeft | ActiveSpeaker | CaptureStarted | CaptureEnded | AudioTrackMute |
AudioTrackUnmute",
  "EventParameters": {
    # ...
  }
}
```

文字起こしファイルについて

transcription-messages フォルダには、.txt 形式の文字起こしファイルが含まれています。ただし、このフォルダではライブ文字起こしが有効になっている場合にのみ、ファイルを受信します。ライブ文字起こしの有効化に関する詳細については、「[Amazon Chime SDK ライブ文字起こしの使用](#)」を参照してください。

このフォルダには、部分的な文字起こしメッセージと完全な文字起こしメッセージがすべて含まれており、各メッセージは JSON オブジェクトです。ファイル名には <yyyy-mm-dd-hour-min-seconds-milleseconds> のタイムスタンプが含まれます。文字起こしファイルの例は、「[配信例](#)」で確認できます。

データストリームを連結する

Note

メディアキャプチャアーティファクトを連結するプロセスを自動化するには、このガイドの「[メディア連結パイプラインを作成する](#)」を参照してください。

この例では、ffmpeg を使用して動画または音声ファイルを 1 つの mp4 ファイルに連結します。まず、すべての入力ファイルを含む filelist.txt ファイルを作成します。以下の形式を使用します。

```
file 'input1.mp4'  
file 'input2.mp4'  
file 'input3.mp4'
```

次に、以下のコマンドを使用して入力ファイルを連結します。

```
ffmpeg -f concat -i filelist.txt -c copy output.mp4
```

メディア連結パイプラインの詳細については、このガイドの「[メディア連結パイプラインを作成する](#)」を参照してください。

メディア連結パイプラインを作成する

メディア連結パイプラインを使用し、メディアキャプチャパイプラインで生成したアーティファクト(ファイル)を連結します。

メディアキャプチャパイプラインは、メディアストリームをチャンク化し、それらのアーティファクトを Amazon S3 バケットに保存することで、会議の内容を取得します。メディアキャプチャパイプラインでは、次のタイプのアーティファクトを作成します。

- 音声
- 動画
- 共有コンテンツ
- データチャネルメッセージ
- 文字起こしメッセージ
- 会議イベント

- 合成された動画 (共有コンテンツと複数の動画ストリームが動画タイルとしてグリッドに表示されます)。

メディア連結パイプラインを使用すると、各タイプのアーティファクトを1つのファイルに連結し、それらの大きなファイルを Amazon S3 バケットに保存できます。メディアキャプチャイベントの終了を待たずにメディア連結パイプラインを作成できますが、連結パイプラインではキャプチャパイプラインが停止したときにのみ連結を開始します。

Note

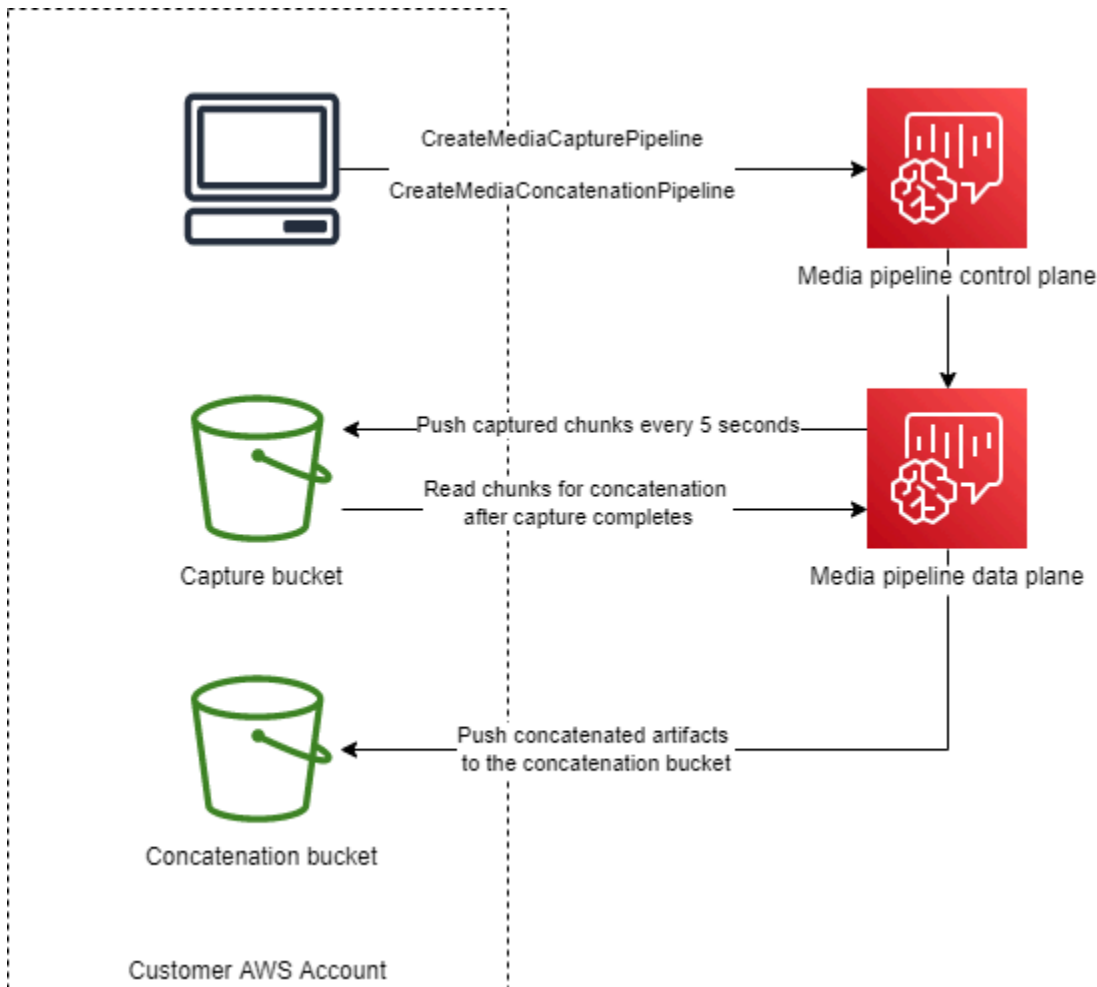
メディアキャプチャパイプライン、メディア連結パイプライン、および Amazon S3 バケットは、同じ AWS アカウントに存在する必要があります。

トピック

- [連結パイプラインのアーキテクチャ](#)
- [メディア連結パイプラインを構築する](#)
- [Amazon S3 バケットフォルダ構造について](#)

連結パイプラインのアーキテクチャ

次の図は、メディア連結パイプラインのアーキテクチャを示しています。



この図では、[CreateMediaCapturePipeline](#) リクエストを受信すると、メディアパイプラインコントロールプレーンはメディアパイプラインデータプレーンでメディアキャプチャパイプラインを開始します。その後、データプレーンで5秒ごとにキャプチャしたチャンクをキャプチャバケットにプッシュします。[CreateMediaConcatenationPipeline](#) リクエストを受信すると、メディアパイプラインコントロールプレーンは、指定されたメディアキャプチャパイプラインが終了するのを待ってから、メディアパイプラインデータプレーンでメディア連結パイプラインを開始します。次に、データプレーンでバケット内のキャプチャされたチャンクを読み取り、連結されたアーティファクトを連結バケットにプッシュします。

メディア連結パイプラインを構築する

Amazon Chime SDK メディア連結パイプラインを作成するには、複数のステップから成るプロセスに従います。このプロセスのステップを以下に示します。

1. メディアキャプチャパイプラインのデータシンクとして使用するための Amazon S3 バケットを作成した後、バケットポリシーを設定します。Amazon S3 バケットのサーバー側の暗号化を有効に

する方法については、このガイドの[Amazon S3バケットのサーバー側の暗号化を有効にする](#)」を参照してください。メディアキャプチャパイプラインで使用する Amazon S3 バケットを作成した場合は、そのバケットのポリシーに s3:GetObject および s3:ListBucket アクションを追加する必要があります。s3:ListBucket アクションには、バケットに対するアクセス許可が必要です。その他のアクションには、バケット内のオブジェクトに対するアクセス許可が必要です。バケットレベルとオブジェクトレベルのアクセス許可を指定するには、2 つの異なる Amazon リソースネーム (ARN) を使用する必要があります。

次の例は、バケットポリシーを示しています。必要に応じてこの例をコピーして貼り付けてください。

```
{
  "Version": "2012-10-17",
  "Id": "AWSChimeMediaCaptureBucketPolicy",
  "Statement": [
    {
      "Sid": "AWSChimeMediaCaptureBucketPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": ["mediapipelines.chime.amazonaws.com"]
      },
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl",
        "s3:GetObject",
        "s3:ListBucket",
      ],
      "Resource": [
        "arn:aws:s3:::[Bucket-Name]/*",
        "arn:aws:s3:::[Bucket-Name]",
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "[Account-Id]"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:chime:*:[Account-Id]:*"
        }
      }
    }
  ]
},
```

```
}  
}
```

2. メディア連結パイプラインのデータシンクとして使用する Amazon S3 バケットを作成し、バケットポリシーを設定します。Amazon S3 バケットのサーバー側の暗号化を有効にする方法については、このガイドの[Amazon S3バケットのサーバー側の暗号化を有効にする](#)を参照してください。

次の例はポリシーを示しています。

```
{  
  "Version": "2012-10-17",  
  "Id": "AWSChimeMediaConcatenationBucketPolicy",  
  "Statement": [  
    {  
      "Sid": " AWSChimeMediaConcatenationBucketPolicy ",  
      "Effect": "Allow",  
      "Principal": {  
        "Service": ["mediapipelines.chime.amazonaws.com"]  
      },  
      "Action": [  
        "s3:PutObject",  
        "s3:PutObjectAcl"  
      ],  
      "Resource": "arn:aws:s3:::[Bucket-Name]/*",  
      "Condition": {  
        "StringEquals": {  
          "aws:SourceAccount": "[Account-Id]"  
        },  
        "ArnLike": {  
          "aws:SourceArn": "arn:aws:chime:*:[Account-Id]:*"  
        }  
      }  
    }  
  ],  
}
```

Note

メディアキャプチャおよびメディア連結パイプラインには、単一の Amazon S3 バケットを使用できます。ただし、そうする場合は、ステップ 2 で示したメディア連結バケットポリシーに `s3:GetObject` および `s3:ListBucket` アクセス許可を追加する必要があります。

す。連結バケットポリシーにこれらのアクセス許可を付与したくない場合は、パイプラインごとに個別のバケットを作成します。

3. [CreateMediaCapturePipeline](#) API を使用して、メディアキャプチャパイプラインを作成します。その一環として、パイプラインの ARN を取得します。ARN の取得に関する詳細については、「[パイプライン作成の概要](#)」を参照してください。ARN は次のステップで使用します。
4. [CreateMediaConcatenationPipeline](#) API を使用して連結パイプラインを作成します。

次の例はリクエストの本文を示しています。*Path* フィールドはオプションで、デフォルトでは連結パイプラインの ID になります。

Note

過去 30 日間に作成された `MediaPipelineArn` を使用する必要があります。

```
{
  "Sources": [
    {
      "Type": "MediaCapturePipeline",
      "MediaCapturePipelineSourceConfiguration": {
        "MediaPipelineArn": "Media_Pipeline_Arn", //must be <30 days old
        "ChimeSdkMeetingConfiguration": {
          "ArtifactsConfiguration": {
            "Audio": {
              "State": "Enabled"
            },
            "Video": {
              "State": "Enabled | Disabled"
            },
            "Content": {
              "State": "Enabled | Disabled"
            },
            "DataChannel": {
              "State": "Enabled | Disabled"
            },
            "TranscriptionMessages": {
              "State": "Enabled | Disabled"
            },
            "MeetingEvents": {
              "State": "Enabled | Disabled"
            }
          }
        }
      }
    }
  ]
}
```

```
        },
        "CompositedVideo": {
            "State": "Enabled | Disabled"
        }
    }
}
],
"Sinks": [
    {
        "Type": "S3Bucket",
        "S3BucketSinkConfiguration": {
            "Destination": "arn:aws:s3:::[Bucket_Name]/[Path]"
        }
    }
]
}
```

キャプチャパイプラインが停止するたびに連結が開始されます。連結パイプラインは、連結の完了後に停止します。

Amazon S3 バケットフォルダ構造について

メディア連結パイプラインの Amazon S3 バケットは、次のフォルダ構造を使用します。

S3 bucket path/

```
audio
video
composited-video
data-channel
meeting-events
transcription-messages
```

Note

メディアパイプラインの作成時にプレフィックスを指定すると、フォルダへのパスはバケット名/プレフィックスになります。プレフィックスなしの場合、パスはバケット名/メディアパイプライン ID になります。S3BucketSinkConfiguration オブジェクトの Destination フィールドでプレフィックスを指定します。連結されたファイル名は、メ

ディアファイルの場合はメディアパイプライン ID.mp4、テキストファイルの場合はメディアパイプライン ID.txt になります。

メディアライブコネクタパイプラインを作成する

次のセクションでは、メディアライブコネクタパイプラインの Real-Time Messaging Protocol (RTMP)、音声、動画の設定を一覧表示して説明します。

RTMP 設定

メディアライブコネクタパイプラインは TLS/SSL 接続を介した RTMP をサポートしています。シンク URL はストリーム URL とストリームキーで構成されています。URL は次の形式に従います。

```
rtmp(s)://stream-server/stream-key
```

以下の例は、一般的なストリーミングプラットフォームに接続する方法を示しています。

- Amazon Interactive Video Service (IVS) – `rtmps://a1b2c3d4e5f6.global-contribute.live-video.net:443/app/IVS-stream-key`
- YouTube – `rtmps://a.youtube.com/live2/stream-key`
- Twitch – `rtmps://live.twitch.tv/app/primary-stream-key`

Important

RTMPS では暗号化を使用し、ストリームが無許可のエンティティによって傍受されないようにします。ベストプラクティスとして、データセキュリティの強化が必要な場合は RTMPS を使用することが挙げられます。

オーディオ設定

メディアライブコネクタパイプラインは、以下の音声設定をサポートしています。

- コーデック – AAC
- サンプルレート – 44100 Hz または 48000 Hz。デフォルトは 44100 Hz。
- チャンネル – モノラルまたはステレオ。デフォルトはモノラルです。

ビデオ設定

メディアライブコネクタパイプラインでは H264 エンコーダーを使用します。1280x720 (HD) または 1920x1080 (FHD) を使用できます。どちらの解像度も毎秒 30 フレームで、2 秒ごとに 1 つのキーフレームを使用します。

メディアライブコネクタパイプラインを停止する

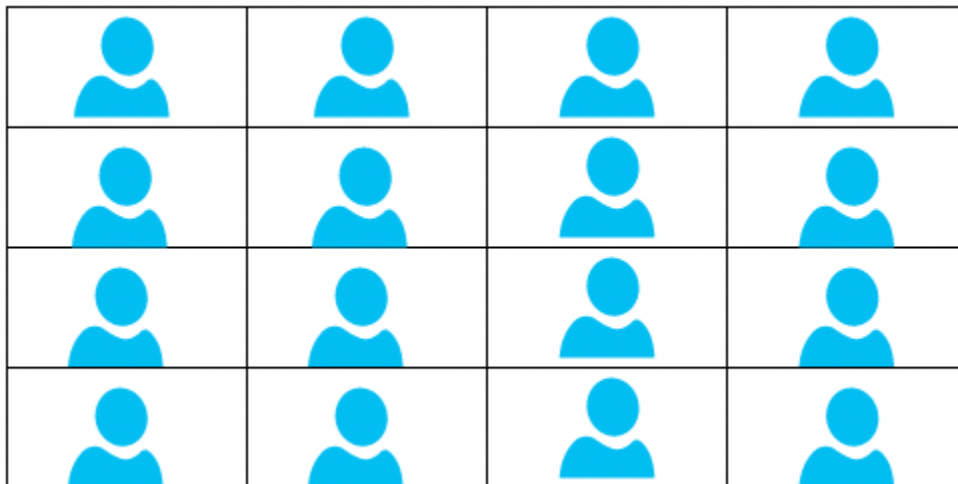
メディアライブコネクタパイプラインを停止するためのベストプラクティスとして、[DeleteMediaPipeline](#) API を呼び出します。IVS などのストリーミングプラットフォームでストリームを終了しても、メディアライブコネクタパイプラインは停止しません。

音声と動画を単一のビューに合成する

Amazon Chime SDK メディアパイプラインは、音声、ウェブカメラ動画、コンテンツ共有動画ストリームの単一のビューへの合成をサポートしています。その後、ライブコネクタを使用して、その単一ビューを Amazon インタラクティブビデオサービス、Twitch、YouTube Live などのストリーミングサービスに送信できます。合成された動画は Amazon Simple Storage Service にキャプチャして、保存したり、再利用したりすることもできます。

合成時には GridView というデフォルトの画面レイアウトが使用され、これは次のような動作をします。

- ウェブカメラ動画だけがアクティブな場合、GridView ではストリームを次のグリッドパターンで編成します。



グリッドには最大 25 件のウェブカメラストリームが表示され、ユーザーがカメラをオンにした時点の順にタイルが並び替えられます。

- GridView には、Landscape と Portrait という 2 つのキャンバスの向きがあります。デフォルトの向きである横向きでは、1280x720 と 1920x1080 (FHD の場合) の動画解像度をサポートしています。縦向きでは、720x1280 と 1080x1920 (FHD の場合) の解像度がサポートされています。
- 動画タイトルの順序、位置、総数、タイトルのアスペクト比、角の半径、枠の色、枠線の太さ、ハイライトの色を構成できます。
- 会議中に参加者が画面を共有すると、コンテンツ共有用のスペースを確保するためにウェブカメラの動画タイトルが動的に変化します。次のセクションで説明するレイアウト構成のいずれかを使用して、これらの変化と動画タイトルの位置を制御します。

レイアウト構成について

参加者がコンテンツ共有を開始したときに、次のレイアウト構成のいずれかを使用して、共有コンテンツとウェブカメラの動画ストリームを合成する方法を選択できます。

- `ActiveSpeakerOnlyConfiguration` では、動画コンテンツを全画面表示にして、発言中のスピーカーのウェブカメラ動画を隅に重ねて合成します。隅の位置は指定できます。
- `PresenterOnlyConfiguration` では動画コンテンツを全画面表示にして、プレゼンターのウェブカメラ動画を隅に重ねて合成します。隅の位置は指定できます。
- `VerticalLayoutConfiguration` では、動画コンテンツをウェブカメラ動画と隣接する垂直の列に合成します。この列は共有コンテンツの右または左に表示できます。
- `HorizontalLayoutConfiguration` では、動画コンテンツとウェブカメラ動画を隣接する水平の行に合成します。この行は共有コンテンツの上または下に表示できます。

合成されたレイアウトは、コンテンツ共有がアクティブか否かに基づき、GridView と選択したレイアウト間で自動的に切り替わります。

以下のトピックでは、グローバルな GridView 設定と各構成レイアウトの使用方法を説明します。

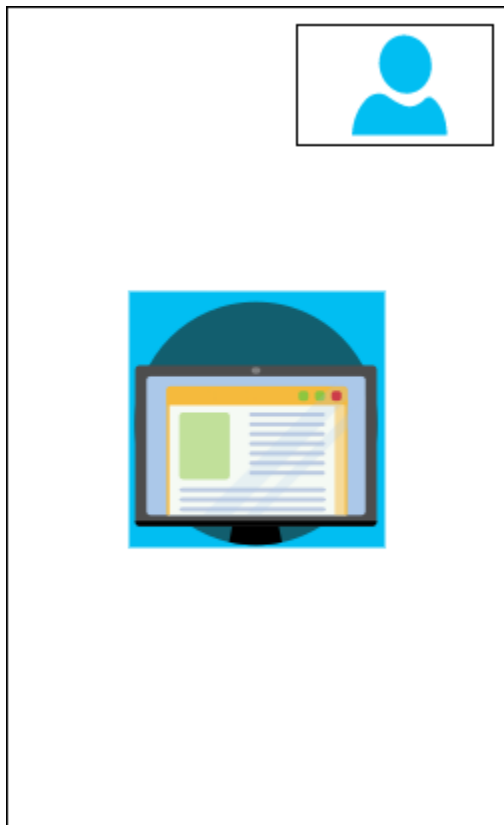
トピック

- [キャンバスの向きの設定](#)
- [枠線と角の属性の設定](#)
- [レイアウト構成の使用](#)

キャンバスの向きの設定

合成時には、キャンバスにすべての動画ストリームを含めます。キャンバスでは Landscape または Portrait の向きを指定できます。横向きのアスペクト比は 16:9 です。縦向きのアスペクト比は 9:16 です。

次の図では、縦方向の向きを示しています。



次の例は、右上隅に動画タイルがある縦向きキャンバスを実装する方法を示しています。この例では、発言中のスピーカーがタイルに表示されます。詳細については、「[ActiveSpeakerOnlyConfiguration](#)」を参照してください。

```
{
  "CompositedVideo":{
    "Layout":"GridView",
    "Resolution":"FHD",
    "GridViewConfiguration":{
      "ContentShareLayout":"ActiveSpeakerOnly",
      "ActiveSpeakerOnlyConfiguration":{
        "ActiveSpeakerPosition":"TopRight"
      }
    }
  },
}
```

```
"CanvasOrientation": "Portrait"  
}  
}
```

次の図では、横向きを示しています。



CanvasOrientation

説明 – 向きの設定 (横向きまたは縦向き)。

使用できる値 – Landscape | Portrait

必須 – いいえ

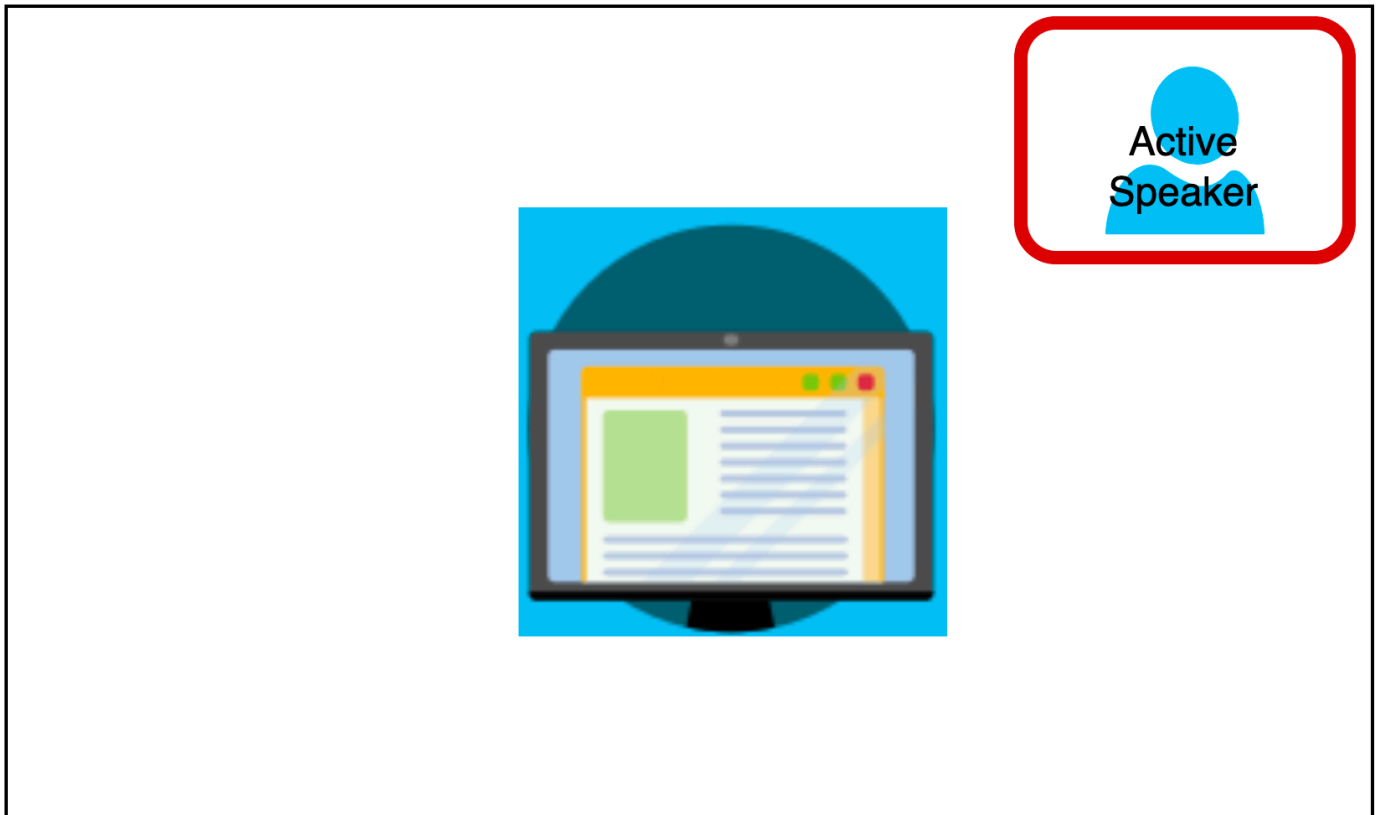
デフォルト – 横向き

枠線と角の属性の設定

必要に応じて、VideoAttribute パラメータを使用して動画タイトルの枠線と角の設定を指定できます。色、幅、角の丸みを指定できます。ハイライトの色も指定でき、参加者が発言すると枠線がその色に変わります。

属性の設定は、共有するコンテンツを問わず、すべてのレイアウトに適用されます。

次の画像は、枠線の色と角の半径が適用された動画タイトルを示しています。



次の例は、各属性の使用方法を示しています。この場合、動画タイルの角は丸く、半径は 5 ピクセルです。タイルの枠線は緑色で、幅は 5 ピクセルです。スピーカーが話すと、HighlightColor属性の境界線の色が赤に変わります。

```
{
  "CompositedVideo":{
    "Layout":"GridView",
    "Resolution":"FHD",
    "GridViewConfiguration":{
      "ContentShareLayout":"ActiveSpeakerOnly",
      "ActiveSpeakerOnlyConfiguration":{
        "ActiveSpeakerPosition":"TopRight"
      }
    }
    "VideoAttribute": {
      "CornerRadius" : 10,
      "BorderColor" : "Green",
      "HighlightColor" : "Red",
      "BorderThickness": 5
    },
  },
}
```

```
}  
}
```

VideoAttribute

説明 – 動画タイトルの枠線と丸い角の設定を指定します

使用できる値 – `BorderColor` | `BorderThickness` | `CornerRadius` | `HighlightColor`

必須 – いいえ

VideoAttribute.BorderColor

説明 – あらゆる動画タイトルの枠線の色を定義します

使用できる値 – 赤、緑、青などの色の名前

必須 – いいえ

VideoAttribute.BorderThickness

説明 – あらゆる動画タイトルの枠線の太さをピクセル単位で定義します

タイプ – 整数

使用できる値 – 1 ~ 20

必須 – いいえ

VideoAttribute.CornerRadius

説明 – あらゆる動画タイトルの角の半径をピクセル単位で定義します。

タイプ – 整数

使用できる値 – 1 ~ 20

必須 – いいえ

VideoAttribute.HighlightColor

説明 – プレゼンターまたはスピーカーが話すときに表示される枠線の色を定義します

使用できる値 – 赤、緑、青などの色の名前

必須 - いいえ

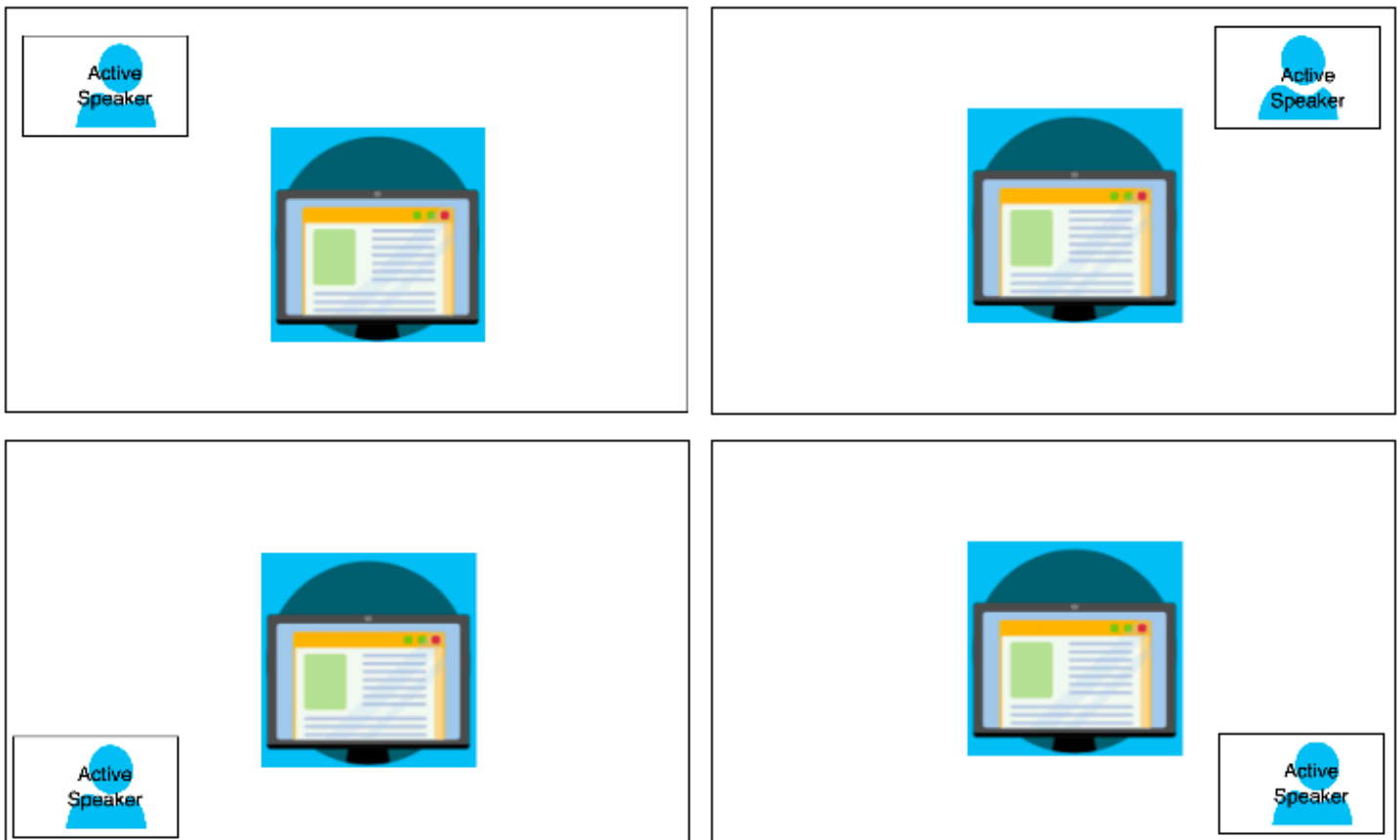
レイアウト構成の使用

以下のトピックでは、さまざまな構成レイアウトの使用方法を説明します。レイアウトは、参加者がコンテンツ共有を開始したときにのみ有効になります。詳細については、各セクションを展開してください。

ActiveSpeakerOnlyConfiguration

ActiveSpeakerOnlyConfiguration では、共有コンテンツと発言中のスピーカーの動画を表示します。つまり、話している人がコンテンツ共有ストリームに重なる小さな動画タイルに表示されます。

次の図は、スピーカータイルの構成と利用可能な位置を示しています。



次の例は、ActiveSpeakerOnly レイアウトをプログラムで実装する方法を示しています。この場合、プレゼンタータイルは左上隅に表示されます。

```
{
```



```
"CompositedVideo":{
  "Layout":"GridView",
  "Resolution":"FHD",
  "GridViewConfiguration":{
    "ContentShareLayout":"ActiveSpeakerOnly",
    "ActiveSpeakerOnlyConfiguration":{
      "ActiveSpeakerPosition":"TopLeft"
    }
  }
}
```

ActiveSpeakerOnlyConfiguration

説明 – ActiveSpeakerOnly 動画タイトルの構成の設定

タイプ — ActiveSpeakerOnlyConfiguration オブジェクト

必須 – いいえ

ActiveSpeakerOnlyConfiguration.ActiveSpeakerPosition

説明 – 発言中のスピーカーの動画タイトルの位置

タイプ – 文字列

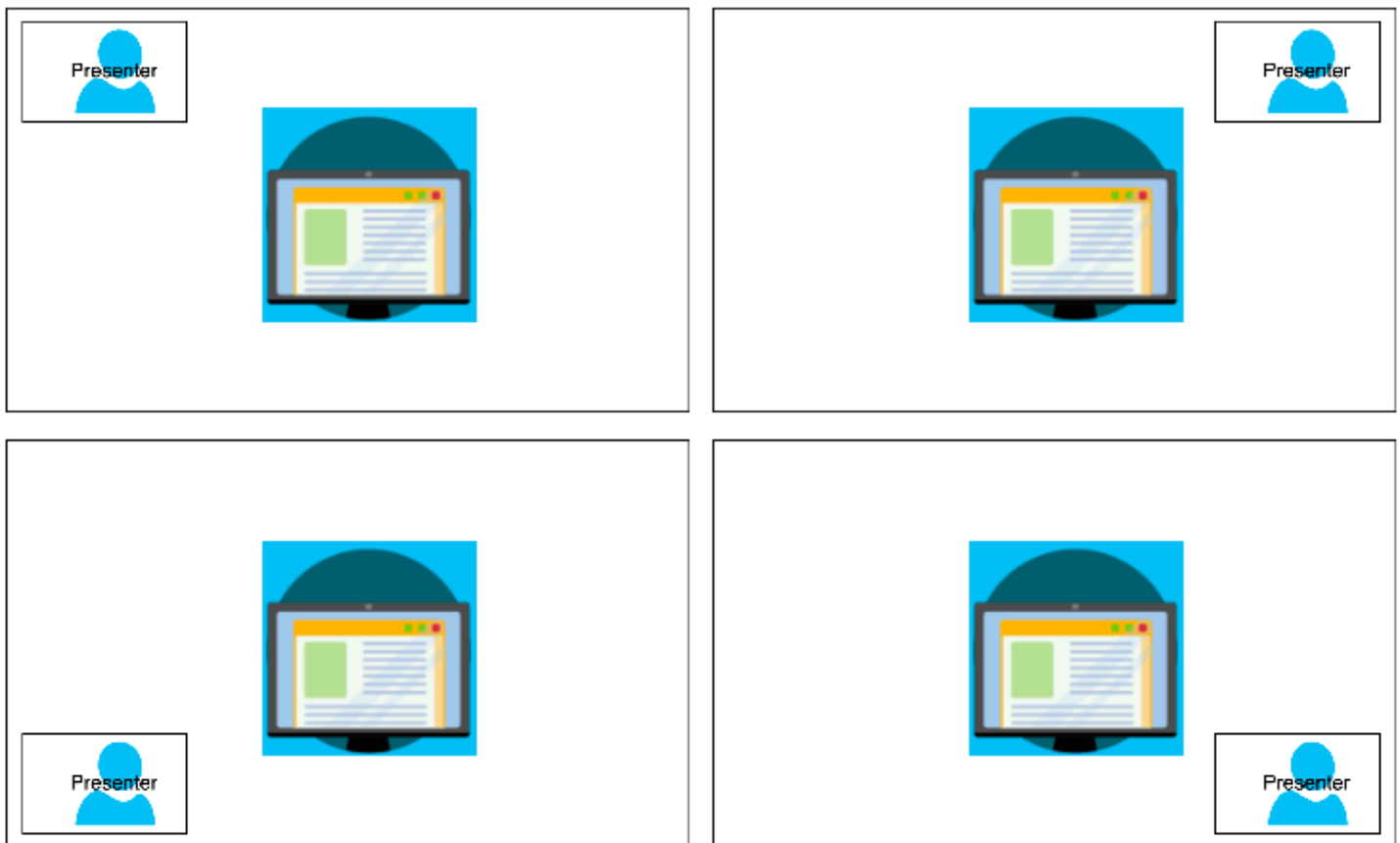
有効な値 – TopLeft | TopRight | BottomLeft | BottomRight

必須 – いいえ

デフォルト – TopRight

PresenterOnlyConfiguration

PresenterOnlyConfiguration では、発言者を問わず、共有コンテンツとプレゼンターの動画のみを表示します。次の画像は、構成を示しています。



次の例は、プレゼンターを右上に配置してレイアウトをプログラムで実装する方法を示しています。

```
{
  "CompositedVideo": {
    "Layout": "GridView",
    "Resolution": "FHD",
    "GridViewConfiguration": {
      "ContentShareLayout": "PresenterOnly",
      "PresenterOnlyConfiguration": {
        "PresenterPosition": "TopRight"
      }
    }
  }
}
```

PresenterOnlyConfiguration

説明 – PresenterOnly レイアウトの構成の設定

タイプ – PresenterOnlyConfiguration オブジェクト

必須 – いいえ

PresenterOnlyConfiguration.PresenterPosition

説明 – プレゼンターの動画タイトルの位置

タイプ – 文字列

有効な値 – TopLeft | TopRight | BottomLeft | BottomRight

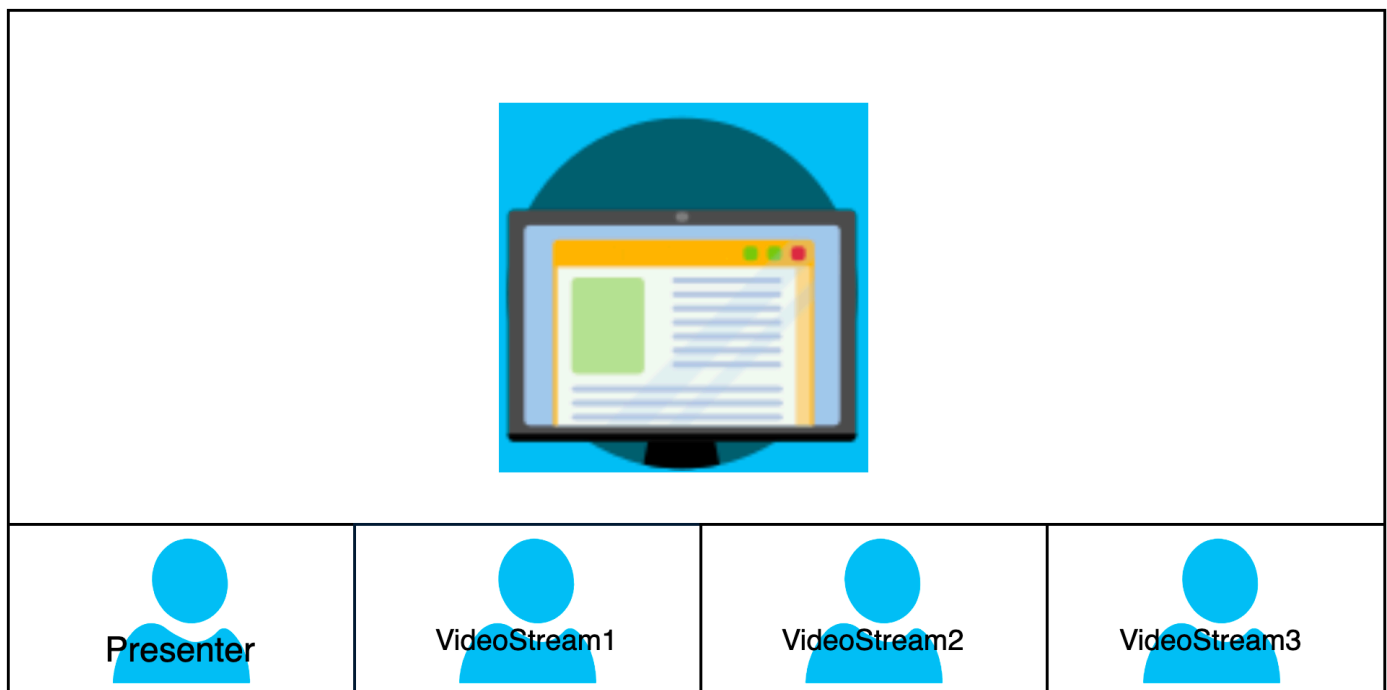
必須 – いいえ

デフォルト – TopRight

HorizontalLayoutConfiguration

HorizontalLayoutConfiguration では、共有コンテンツと動画ストリームを水平に表示します。TilePosition 設定を使用して、コンテンツ共有ストリームの上または下にタイトルを表示することができます。プレゼンターは常に左側に表示されます。追加のタイトルは、JoinSequence で指定された順序で表示されます。

次の図は、コンテンツ共有ストリームの下にタイトルを示しています。



次の例は、水平レイアウトをプログラムで実装する方法を示しています。この場合、レイアウトではタイルを SpeakerSequence の順番に並べ替えただけで、共有画面の下に配置します。このレイアウトでは最大 4 つのタイルを使用でき、16/9 のアスペクト比が適用されます。

```
{
  "CompositedVideo":{
    "Layout":"GridView",
    "Resolution":"FHD",
    "GridViewConfiguration":{
      "ContentShareLayout":"Horizontal",
      "HorizontalLayoutConfiguration":{
        "TileOrder":"SpeakerSequence",
        "TilePosition":"Bottom",
        "TileCount":4,
        "TileAspectRatio":"16/9"
      }
    }
  }
}
```

HorizontalLayoutConfiguration

説明 – 水平レイアウトの構成の設定

タイプ — HorizontalLayoutConfiguration オブジェクト

必須 – いいえ

HorizontalLayoutConfiguration.TilePosition

説明 – 共有コンテンツの上または下にタイルを配置します。

タイプ - 文字列

有効な値 – Bottom | Top

必須 – いいえ

デフォルト — 下部

HorizontalLayoutConfiguration.TileOrder

説明 – ユーザーが参加したとき、または発言したときにタイルの順番を並べ替えます。

タイプ – 文字列

有効な値 – JoinSequence | SpeakerSequence

必須 – いいえ

デフォルト – JoinSequence

HorizontalLayoutConfiguration.TileCount

説明 – 画面共有中に引き続き表示されるタイルの数を指定します

タイプ – 整数

有効な値 – 1 ~ 10

必須 – いいえ

デフォルト – 4

HorizontalLayoutConfiguration.TileAspectRatio

説明 – タイルのアスペクト比を指定します

タイプ – 整数

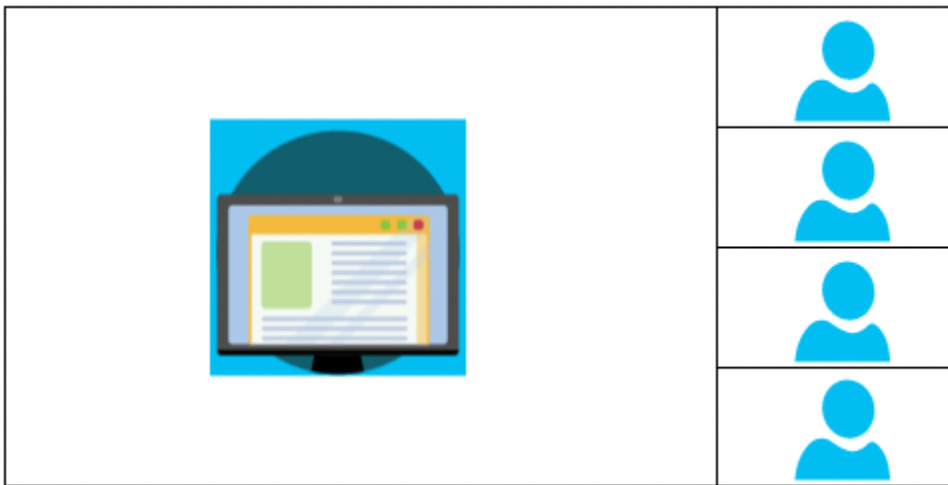
有効な値 – n/n

必須 – いいえ

デフォルト – 16/9、値はすべてのタイルに適用されます

VerticalLayoutConfiguration

VerticalLayoutConfiguration では、共有コンテンツと最新の 4 つの動画を右側に積み重ねて表示します。プレゼンターは常に一番上に表示されます。他の出席者は指定された順序で表示されます。TileOrder



次の例は、垂直レイアウトをプログラムで実装する方法を示しています。この場合、JoinSequence レイアウトはタイルを並べて画面共有の右に配置します。このレイアウトでは最大 4 つのタイルを使用でき、16/9 のアスペクト比が適用されます。

```
{
  "CompositedVideo":{
    "Layout": "GridView",
    "Resolution": "FHD",
    "GridViewConfiguration":{
      "ContentShareLayout": "Vertical",
      "VerticalLayoutConfiguration":{
        "TileOrder": "JoinSequence",
        "TilePosition": "Right",
        "TileCount": 4,
        "TileAspectRatio": "16/9"
      }
    }
  }
}
```

VerticalLayoutConfiguration

説明 – 垂直レイアウトの構成の設定

タイプ — VerticalLayoutConfiguration オブジェクト

必須 – いいえ

VerticalLayoutConfiguration.TilePosition

説明 – 共有コンテンツの右または左にタイルを配置します。

タイプ – 文字列

有効な値 – Bottom | Top

必須 – いいえ

デフォルト – 下部

VerticalLayoutConfiguration.TileOrder

説明 – ユーザーが参加したとき、または発言したときにタイルの順番を並べ替えます。

タイプ – 文字列

有効な値 – JoinSequence | SpeakerSequence

必須 – いいえ

デフォルト – JoinSequence

VerticalLayoutConfiguration.TileCount

説明 – タイルの数を指定します

タイプ – 整数

有効な値 – 1 ~ 10

必須 – いいえ

デフォルト – 4

VerticalLayoutConfiguration.TileAspectRatio

説明 – タイルのアスペクト比を指定します

タイプ – 整数

有効な値 – n/n

必須 – いいえ

デフォルト – 9/16、値はすべてのタイルに適用されます

メディアストリームパイプラインを作成する

メディアストリームパイプラインでは、会議での全参加者の個別の音声のほか、メディア連結パイプラインで生成された混合音声をキャプチャします。すべてのメディアストリームパイプラインでは、データを [Amazon Kinesis Video Streams](#) (KVS) に保存します。

ビデオストリームを作成するには、[CreateMediaPipelineKinesisVideoStreamPool](#) API を呼び出します。Amazon Chime SDK ミーティングごとに 1 つのメディアストリームパイプラインを作成できます。

Note

会議でオプトインリージョンをとして使用している場合 [MediaRegion](#)、KVS ストリームは同じリージョンに存在する必要があります。例えば、会議で af-south-1リージョンを使用する場合、KVS ストリームも がある必要がありますaf-south-1。ただし、会議で AWS がデフォルトでオンになっているリージョンを使用している場合、KVS ストリームはオプトインリージョンを含む利用可能な任意のリージョンに存在することができます。例えば、会議で使用する場合ca-central-1、KVS ストリームはeu-west-2、af-south-1、または Amazon Chime SDK us-east-1がサポートする他のリージョンにあることができます。会議が使用する AWS リージョンを確認するには、[GetMeeting](#) API を呼び出し、レスポンスから [MediaRegion](#)パラメータを使用します。

オプトインリージョンの詳細については、このガイド[利用できるリージョンの「」、および「アカウント管理リファレンスガイド」の「アカウントが 利用できる AWS リージョンを指定する」](#)を参照してください。AWS

次のセクションでは、メディアストリームパイプラインを作成する方法について説明します。記載されている順序に従ってください。

トピック

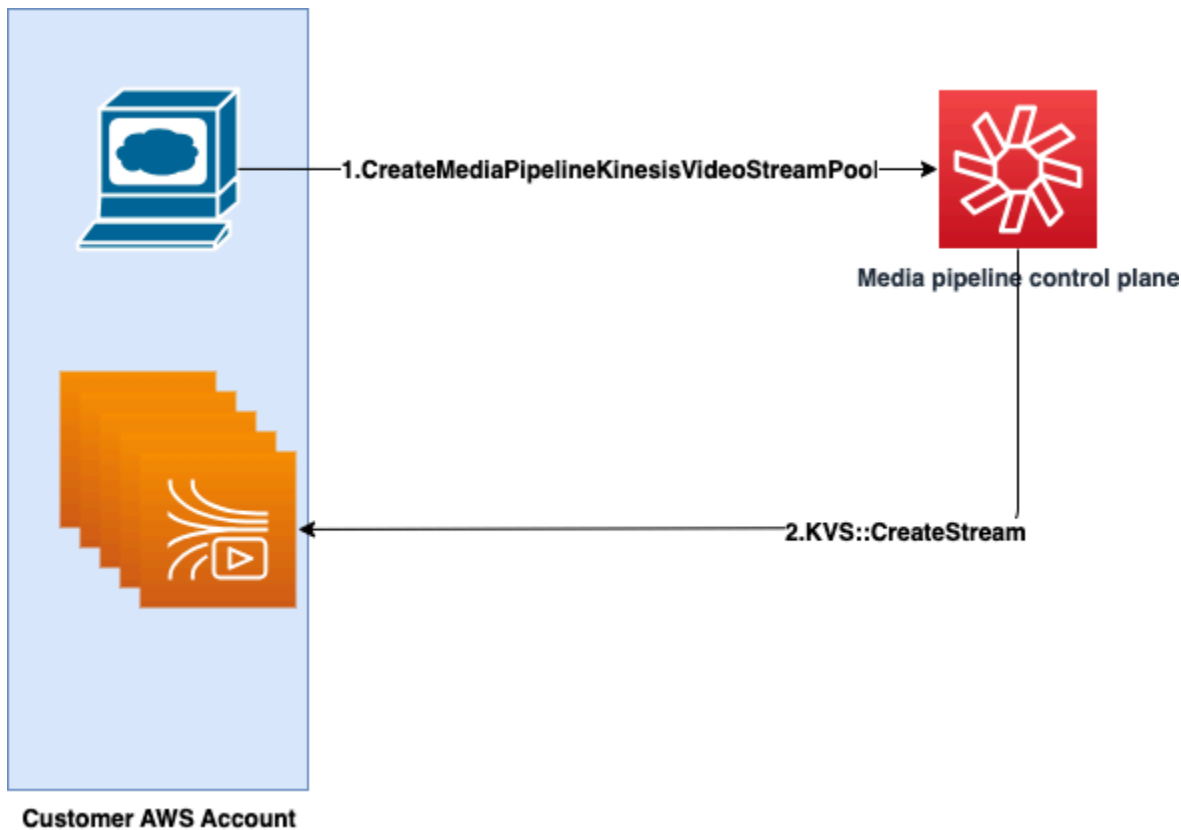
- [Kinesis Video Streams プールの作成](#)
- [Kinesis Video Streams プールのコードの例](#)
- [メディアストリームパイプラインを作成する](#)
- [メディアストリームパイプラインのコードの例](#)
- [Event Bridge 通知の使用](#)

- [メディアストリームパイプラインデータの使用](#)

Kinesis Video Streams プールの作成

メディアストリームパイプラインの Kinesis Video Streams (KVS) プールは、Amazon Chime SDK ミーティングと同じ AWS アカウントに属している必要があります。Kinesis Video Streams プールを作成するには、[CreateMediaPipelineKinesisVideoStreamPool](#) API を呼び出します。

次の図は、メディアパイプラインの Kinesis Video Streams プールのアーキテクチャを示しています。画像内の数字は、以下の番号付きテキストに対応しています。



図中の手順を説明します。

1. [CreateMediaPipelineKinesisVideoStreamPool](#) API を呼び出します。
2. メディアパイプラインコントロールプレーンでは、ご使用のアカウントに代わって Kinesis Video Streams (KVS) とプールを作成および管理します。

KVS プールの操作、つまりプールでストリームを作成、更新、削除するプロセスは非同期的に行われます。そのため、Event Bridge 通知では Chime Media Pipeline Kinesis Video Pool State Change 詳細タイプを使用してプールでのストリームの状態を伝えます。

プールを一度作成すると、他の会議にまたいで再利用できます。また、必要に応じて他のプールを作成し、不要になったらプールを削除することもできます。

プールは、同時通話件数に基づいて自動的にスケールアップされます。不要なプールは削除できません。

Note

プールを削除する場合、プールが完全に削除されるのを待ってから、プール内の KVS ストリームを削除する必要があります。プールが完全に削除されると、Event Bridge 通知が表示されます。これは、プールを使用する会議がすべて終了した後に行われます。[GetMediaPipelineKinesisVideoStreamPool](#) API を呼び出して、特定の KVS プール `PoolId` の を表示することもできます。

Kinesis Video Streams [DeleteStream](#) API を呼び出すと、その命名文字列を使用してプール内のストリームを検索および削除できます。[GetMediaPipelineKinesisVideoStreamPool](#) API を呼び出して、特定の KVS プール `PoolId` の を表示することもできます。次のセクションの例でその方法について説明します。

Kinesis Video Streams プールのコードの例

次の例は、Kinesis Video Streams (KVS) プールを作成、更新、取得、一覧表示、削除する方法を示しています。詳細については、各セクションを展開してください。

インポートと一般的な変数

```
...
Define imports and common variables
...

import boto3
from uuid import uuid4
import json

client = boto3.client("chime-sdk-media-pipelines", region_name='us-east-1')
pool_name = 'MyDemoKvsPool'

def pretty_print_json(obj):
    print(json.dumps(obj, default=str, indent=4))
```

CreateMediaPipelineKinesisVideoStreamPool

```
response = client.create_media_pipeline_kinesis_video_stream_pool(  
    StreamConfiguration={  
        'Region': 'us-east-1',  
        'DataRetentionInHours': 24  
    },  
    PoolName=pool_name,  
    ClientRequestToken=str(uuid4()),  
    Tags=[  
        {  
            'Key': 'MyTagForAccessControl',  
            'Value': 'SomeTagValue'  
        },  
    ],  
)  
  
pretty_print_json(response['KinesisVideoStreamPoolConfiguration'])
```

出力:

```
{  
    "PoolArn": "arn:aws:chime:us-east-1:account-ID:media-pipeline-kinesis-video-stream-pool/MyDemoKvsPool",  
    "PoolName": "MyDemoKvsPool",  
    "PoolId": "ChimeMediaPipelines-MyDemoKvsPool-1f4e1a69-e718-4884-bf92-8a393ac0405b",  
    "PoolStatus": "CREATING",  
    "StreamConfiguration": {  
        "Region": "us-east-1",  
        "DataRetentionInHours": 24  
    },  
    "CreatedTimestamp": "2023-10-13 01:26:09.979000+00:00",  
    "UpdatedTimestamp": "2023-10-13 01:26:09.979000+00:00"  
}
```

GetMediaPipelineKinesisVideoStream

```
response = client.get_media_pipeline_kinesis_video_stream_pool(  
    Identifier=pool_name  
)  
  
pretty_print_json(response['KinesisVideoStreamPoolConfiguration'])
```

出力:

```
{
  "PoolArn": "arn:aws:chime:us-east-1:account-ID:media-pipeline-kinesis-video-stream-pool/MyDemoKvsPool",
  "PoolName": "MyDemoKvsPool",
  "PoolId": "ChimeMediaPipelines-MyDemoKvsPool-1f4e1a69-e718-4884-bf92-8a393ac0405b",
  "PoolStatus": "ACTIVE",
  "StreamConfiguration": {
    "Region": "us-east-1",
    "DataRetentionInHours": 24
  },
  "CreatedTimestamp": "2023-10-13 01:26:09.979000+00:00",
  "UpdatedTimestamp": "2023-10-13 01:26:09.979000+00:00"
}
```

UpdateMediaPipelineKinesisVideoStream

```
response = client.update_media_pipeline_kinesis_video_stream_pool(
    Identifier=pool_name,
    StreamConfiguration={
        'DataRetentionInHours': 48
    }
)
pretty_print_json(response['KinesisVideoStreamPoolConfiguration'])
```

出力:

```
{
  "PoolArn": "arn:aws:chime:us-east-1:account-ID:media-pipeline-kinesis-video-stream-pool/MyDemoKvsPool",
  "PoolName": "MyDemoKvsPool",
  "PoolId": "ChimeMediaPipelines-MyDemoKvsPool-d08c26ae-0336-4e2e-acdf-805a7d71b891",
  "PoolStatus": "UPDATING",
  "PoolSize": 40,
  "StreamConfiguration": {
    "Region": "us-east-1",
    "DataRetentionInHours": 48
  },
  "CreatedTimestamp": "2023-10-13 01:44:23.010000+00:00",
  "UpdatedTimestamp": "2023-10-13 01:44:28.486000+00:00"
}
```

ListMediaPipelineKinesisVideoStream

```
list_of_pools = []
max_results = 100
next_token = None
while(True):
    if next_token:
        response = client.list_media_pipeline_kinesis_video_stream_pools(
            NextToken=next_token,
            MaxResults=max_results
        )
    else:
        response = client.list_media_pipeline_kinesis_video_stream_pools(
            MaxResults=max_results
        )

    list_of_pools.extend(response['KinesisVideoStreamPools'])
    next_token = response.get('NextToken')
    if not next_token:
        break
pretty_print_json(list_of_pools)
```

出力:

```
[
  {
    "PoolName": "MyDemoKvsPool",
    "PoolId": "ChimeMediaPipelines-MyDemoKvsPool-6588e703-f046-4288-
ba7f-0c03de76a6bb",
    "PoolArn": "arn:aws:chime:us-east-1:account-ID:media-pipeline-kinesis-video-
stream-pool/MyDemoKvsPool"
  }
]
```

DeleteMediaPipelineKinesisVideoStream

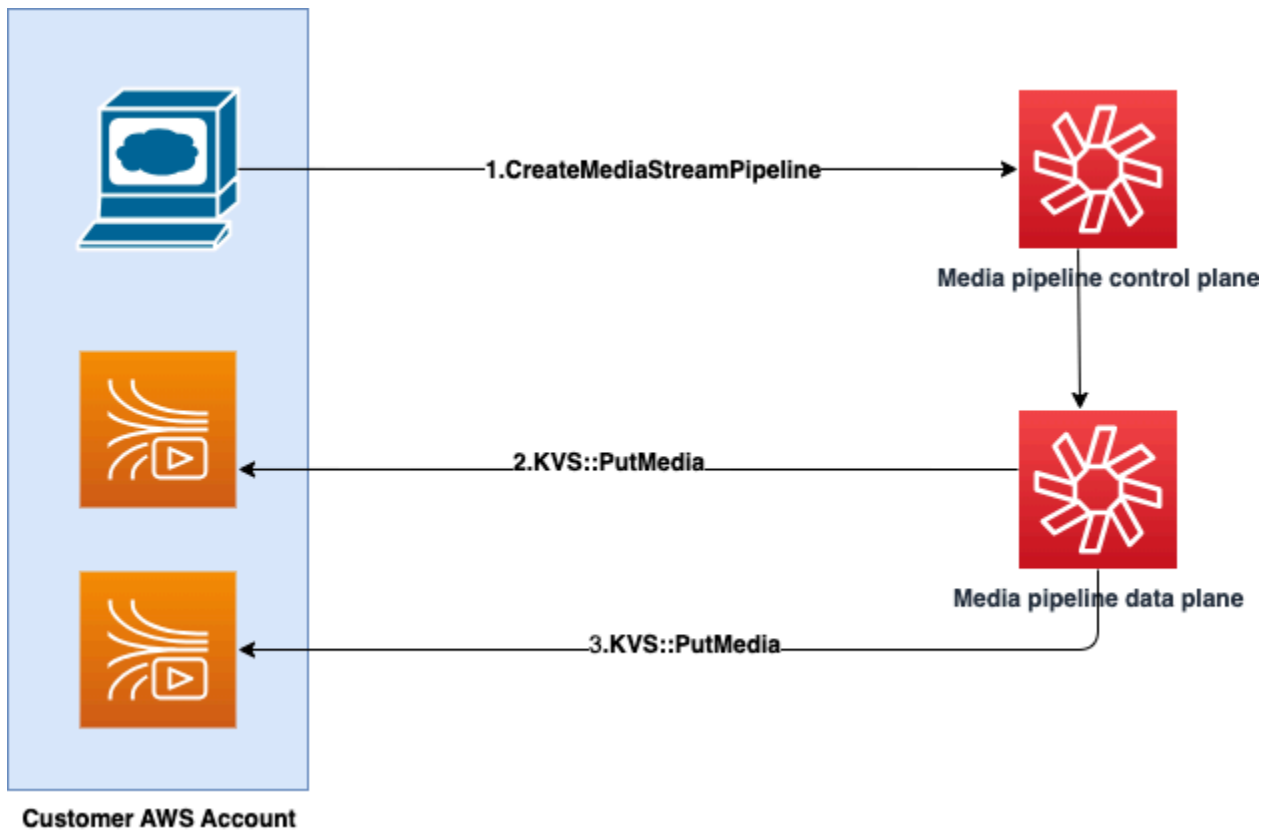
```
client.delete_media_pipeline_kinesis_video_stream_pool(
    Identifier=pool_name
)
```

出力: 成功した `delete_media_pipeline_kinesis_video_stream_pool` リクエストには本文はありません。

メディアストリームパイプラインを作成する

チャイムメディアストリームパイプラインは、Amazon Chime SDK ミーティングと同じ AWS アカウントに属している必要があります。Amazon Chime SDK メディアストリームパイプラインを作成するには、[CreateMediaStreamPipeline](#) API を呼び出し、ソースとシンクを指定します。

次の図は、Amazon Chime SDK メディアストリームパイプラインのアーキテクチャを示しています。図の数字は、次の各説明の番号に対応しています。



図中の手順を説明します。

1. `CreateMediaStreamPipeline` API を呼び出します。リクエストでは、ストリームのソースとシンクを指定します。個別の音声、混合音声、またはその両方のどれをキャプチャするのかを指定します。KVS プールの ARN をリクエストに含めます。
 - ソース配列は `SourceType` と `SourceArn` で構成されます。ChimeSdkMeeting `SourceType` を使用する必要があり、`SourceArn` は ChimeSdkMeeting の ARN です。
 - シンク配列は `SinkType`、`SinkArn`、`ReservedStreamCapacity`、`MediaStreamType` で構成されます。KinesisVideoStreamPoolSinkType のみがサポートされており、`SinkArn` は KinesisVideoStreamPool の ARN です。MediaStreamType では、シンクにストリーミングされるメディアのタイプ (MixedAudio または IndividualAudio)

を制御します。ReservedStreamCapacity では、KinesisVideoStreamPool から MediaStreamType に対して割り当てるストリームの数を設定します。

- IndividualAudio と MixedAudio の両方をストリーミングしたい場合は、Sinks 配列に 2 つのシンクオブジェクト (1 つは IndividualAudio 用、もう 1 つは MixedAudio 用) を作成します。SinkArn (KinesisVideoStreamPool の ARN) はシンクごとに異なる場合があります。
- 個別の音声または混合音声のみをストリーミングするには、目的の MediaStreamType でシンクオブジェクトを 1 つ作成します。
- 次の点に注意してください。
 - を KinesisVideoStreamPool として [CreateMediaStreamPipeline](#) API を呼び出す場合 SinkType、CreateMediaStreamPipeline は呼び出されるコントロールプレーンリージョンに属している SinkARN 必要があります。

例えば、us-east-1 でメディアストリームパイプラインを作成する場合は、us-east-1 で KinesisVideoStreamPool を使用する必要があります。

- ReservedStreamCapacity は、MixedAudio MediaStreamType を指定する場合は **1** に、IndividualAudio MediaStreamType を指定する場合は **1-10** の間にする必要があります。
2. メディアパイプラインデータプレーンは KVS [PutMedia](#) API を呼び出して、指定した KVS プールに属する KVS ストリームに個々のオーディオを保存します。
 3. メディアパイプラインデータプレーンでは KVS PutMedia API を呼び出して、指定した KVS プールに属するストリームに混合音声を保存します。

Note

[CreateMediaStreamPipeline](#) API を呼び出すと、ビルダーは [メディアパイプラインイベント](#) を使用するか、[GetMediaPipeline](#) API を呼び出してパイプラインの状態が `InProgress` を判断できます。

パイプラインの状態が `InProgress` に達すると、メディア (IndividualAudio と MixedAudio を任意に組み合わせたもの) が KVS にストリーミングされます。

IndividualAudio ストリームタイプでは、参加者 ID と KinesisVideoStreamPool から割り当てられた KVS ストリーム間に 1:1 のマッピングが存在します。このマッピングは、メディアパイプラインの存続期間にわたって適用されます。

どの KVS ストリームが参加者 ID にマッピングされるか、またはに割り当てられているかを知るには MixedAudio、次のいずれかの方法を使用します。

- [Event Bridge 通知](#)を使用する。それぞれの通知には、参加者 ID や参加者の音声をストリーミングする KVS ARN などの情報が表示されます。IndividualAudio または MixedAudio ストリーミングセッションが開始されると、chime:MediaPipelineKinesisVideoStreamStart イベントが送信されます。ストリーミングセッションは、参加者が通話から退出するか (IndividualAudio の場合)、会議が終了すると終了します。
- Kinesis Video Streams で各フラグメントと共に送信される永続メタデータを使用します。このメタデータには、Event Bridge で送信されるものと同様の情報が含まれています。ビルダーは、このソリューションを使用して Kinesis Video Streams API [ListStreams](#) のプレフィックスとしてプール名を指定KinesisVideoStreamPoolして、のすべてのストリームを解析する必要があります。

メディアストリームパイプラインの終了は、会議が削除されるか、そのメディアストリームパイプラインの [DeleteMediaPipeline](#) API が呼び出されたときに発生します。メディアパイプラインの終了を示す [Event Bridge 通知](#)も送信されます。

メディアストリームパイプラインのコードの例

以下の例は、混合音声、個別の音声、またはその両方のメディアストリームパイプラインを作成する方法を示しています。詳細については、各セクションを展開してください。

CreateMediaStreamPipeline 混合オーディオ用

```
response = client.create_media_stream_pipeline(  
    Sources=[  
        {  
            'SourceType': 'ChimeSdkMeeting',  
            'SourceArn': 'arn:aws:chime:us-east-1:account-  
ID:meeting/bed804cf-8cf0-4991-9b8d-d1acc2987433'  
        },  
    ],  
    Sinks=[  
        {  
            'SinkArn': 'arn:aws:chime:us-east-1:account-ID:media-pipeline-kinesis-  
video-stream-pool/foo',  
            'SinkType': 'KinesisVideoStreamPool',  
            'ReservedStreamCapacity': 1,  
            'MediaStreamType': 'MixedAudio'        }  
    ]  
)
```



```
    },
  ],
  ClientRequestToken='sample token',
  Tags=[
    {
      'Key': 'sample key',
      'Value': 'sample value'
    }
  ]
)
```

レスポンス:

```
{
  'MediaStreamPipeline': {
    'MediaPipelineId': '45bc79a0-4591-4ebe-a642-d42c4e279f2d',
    'MediaPipelineArn': 'arn:aws:chime:us-east-1:account-ID:media-
pipeline/45bc79a0-4591-4ebe-a642-d42c4e279f2d',
    'CreatedTimestamp': '2023-07-25T21:48:48.265Z',
    'UpdatedTimestamp': '2023-07-25T21:48:48.376Z',
    'Status': 'Initializing',
    'Sources': [
      {
        'SourceType': 'ChimeSdkMeeting',
        'SourceArn': 'arn:aws:chime:us-east-1:account-
ID:meeting/bed804cf-8cf0-4991-9b8d-d1acc2987433'
      }
    ],
    'Sinks': [
      {
        'SinkArn': 'arn:aws:chime:us-east-1:account-ID:media-pipeline-kinesis-
video-stream-pool/foo',
        'SinkType': 'KinesisVideoStreamPool',
        'ReservedStreamCapacity': 1,
        'MediaStreamType': 'MixedAudio'
      }
    ]
  }
}
```

CreateMediaStreamPipeline 個々のオーディオ用

```
response = client.create_media_stream_pipeline(
```

```

Sources=[
  {
    'SourceType': 'ChimeSdkMeeting',
    'SourceArn': 'arn:aws:chime:us-east-1:account-
ID:meeting/bed804cf-8cf0-4991-9b8d-d1acc2987433'
  },
],
Sinks=[
  {
    'SinkArn': 'arn:aws:chime:us-east-1:account-ID:media-pipeline-kinesis-
video-stream-pool/foo',
    'SinkType': 'KinesisVideoStreamPool',
    'ReservedStreamCapacity': 5,
    'MediaStreamType': 'IndividualAudio'
  },
],
ClientRequestToken='sample token',
Tags=[
  {
    'Key': 'sample key',
    'Value': 'sample value'
  },
]
)

```

レスポンス:

```

{
  'MediaStreamPipeline': {
    'MediaPipelineId': '45bc79a0-4591-4ebe-a642-d42c4e279f2d',
    'MediaPipelineArn': 'arn:aws:chime:us-east-1:account-ID:media-
pipeline/45bc79a0-4591-4ebe-a642-d42c4e279f2d',
    'CreatedTimestamp': '2023-07-25T21:48:48.265Z',
    'UpdatedTimestamp': '2023-07-25T21:48:48.376Z',
    'Status': 'Initializing',
    'Sources': [
      {
        'SourceType': 'ChimeSdkMeeting',
        'SourceArn': 'arn:aws:chime:us-east-1:account-
ID:meeting/bed804cf-8cf0-4991-9b8d-d1acc2987433'
      },
    ],
    'Sinks': [

```

```

    {
      'SinkArn': 'arn:aws:chime:us-east-1:account-ID:media-pipeline-kinesis-
video-stream-pool/foo',
      'SinkType': 'KinesisVideoStreamPool',
      'ReservedStreamCapacity': 5,
      'MediaStreamType': 'IndividualAudio'
    },
  ]
}

```

CreateMediaStreamPipeline 混合オーディオと個々のオーディオ用

```

response = client.create_media_stream_pipeline(
  Sources=[
    {
      'SourceType': 'ChimeSdkMeeting',
      'SourceArn': 'arn:aws:chime:us-east-1:account-
ID:meeting/bed804cf-8cf0-4991-9b8d-d1acc2987433'
    },
  ],
  Sinks=[
    {
      'SinkArn': 'arn:aws:chime:us-east-1:account-ID:media-pipeline-kinesis-
video-stream-pool/foo',
      'SinkType': 'KinesisVideoStreamPool',
      'ReservedStreamCapacity': 1,
      'MediaStreamType': 'MixedAudio'
    },
    {
      'SinkArn': 'arn:aws:chime:us-east-1:account-ID:media-pipeline-kinesis-
video-stream-pool/foo',
      'SinkType': 'KinesisVideoStreamPool',
      'ReservedStreamCapacity': 5,
      'MediaStreamType': 'IndividualAudio'
    },
  ],
  ClientRequestToken='sample token',
  Tags=[
    {
      'Key': 'sample key',
      'Value': 'sample value'
    },
  ],
)

```

```
]
)
```

レスポンス:

```
{
  'MediaStreamPipeline': {
    'MediaPipelineId': '45bc79a0-4591-4ebe-a642-d42c4e279f2d',
    'MediaPipelineArn': 'arn:aws:chime:us-east-1:account-ID:media-
pipeline/45bc79a0-4591-4ebe-a642-d42c4e279f2d',
    'CreatedTimestamp': '2023-07-25T21:48:48.265Z',
    'UpdatedTimestamp': '2023-07-25T21:48:48.376Z',
    'Status': 'Initializing',
    'Sources': [
      {
        'SourceType': 'ChimeSdkMeeting',
        'SourceArn': 'arn:aws:chime:us-east-1:account-
ID:meeting/bed804cf-8cf0-4991-9b8d-d1acc2987433'
      },
    ],
    'Sinks': [
      {
        'SinkArn': 'arn:aws:chime:us-east-1:account-ID:media-pipeline-kinesis-
video-stream-pool/foo',
        'SinkType': 'KinesisVideoStreamPool',
        'ReservedStreamCapacity': 1,
        'MediaStreamType': 'MixedAudio'
      },
      {
        'SinkArn': 'arn:aws:chime:us-east-1:account-ID:media-pipeline-kinesis-
video-stream-pool/foo',
        'SinkType': 'KinesisVideoStreamPool',
        'ReservedStreamCapacity': 5,
        'MediaStreamType': 'IndividualAudio'
      }
    ]
  }
}
```

Event Bridge 通知の使用

[メディアパイプラインイベントを使用する](#) の他にも、メディアストリームパイプラインでは、KVS へのストリーミングを開始および停止したとき、および動画プールの状態が変化したときに Event Bridge 通知を送信します。

トピック

- [メディアストリームパイプラインイベント](#)
- [Media Pipeline Kinesis Video Pool イベント](#)

メディアストリームパイプラインイベント

メディアストリームパイプラインでは次のイベントを送信します。詳細については、各セクションを展開してください。

Amazon Chime Media Stream Pipeline Kinesis Video Stream Start

Amazon Chime SDK メディアパイプラインでは、メディアストリームパイプラインで会議から音声を受信し、その音声を KVS にストリーミングし始めると、このイベントが送信されます。空の `AttendeeId` および `ExternalUserId` フィールドは、メディアパイプラインによって混合音声が KVS ストリームに送信されたことを示します。

```
{
  "version": "0",
  "id": "5ee6265a-0a40-104e-d8fd-a3b4bdd78483",
  "detail-type": "Chime Media Pipeline State Change",
  "source": "aws.chime",
  "account": "111122223333",
  "time": "2021-07-28T20:20:49Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "eventType": "chime:MediaPipelineKinesisVideoStreamStart",
    "timestamp": 1627503649251,
    "meetingId": "1e6bf4f5-f4b5-4917-b8c9-bda45c340706",
    "externalMeetingId": "Meeting_Id",
    "mediaPipelineId": "e40ee45e-2ed1-408e-9156-f52b8208a491",
    "mediaRegion": "ap-southeast-1",

    "attendeeId": "Attendee_Id",
```

```
"externalUserId": "External_User_Id",

"kinesisVideoStreamArn": "arn:aws:kinesisvideo:us-east-1:123456:stream/Chime*",
"startFragmentNumber": "1234567899444",
"startTime": "yyyy-mm-ddThh:mm:ssZ"

}
}
```

Amazon Chime Media Stream Pipeline Kinesis Video Stream End

KVS へのストリーミングが終了すると、メディアパイプラインによってこのイベントが Event Bridge に送信されます。

```
{
  "version": "0",
  "id": "5ee6265a-0a40-104e-d8fd-a3b4bdd78483",
  "detail-type": "Chime Media Pipeline State Change",
  "source": "aws.chime",
  "account": "111122223333",
  "time": "2021-07-28T20:20:49Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "eventType": "chime:MediaPipelineKinesisVideoStreamEnd",
    "timestamp": 1627503649251,
    "meetingId": "1e6bf4f5-f4b5-4917-b8c9-bda45c340706",
    "externalMeetingId": "Meeting_Id",
    "mediaPipelineId": "e40ee45e-2ed1-408e-9156-f52b8208a491",
    "mediaRegion": "ap-southeast-1",

    "attendeeId": "Attendee_Id",
    "externalUserId": "External_User_Id",

    "kinesisVideoStreamArn": "arn:aws:kinesisvideo:us-east-1:123456:stream/Chime*",
    "startFragmentNumber": "1234567899444",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "endFragmentNumber": "1234567899555"
  }
}
```

Media Pipeline Kinesis Video Pool イベント

プールの状態が変化すると、メディアパイプラインによって次のイベントが Event Bridge に送信されます。詳細については、各セクションを展開してください。

Amazon Chime Media Pipeline Kinesis Video Pool Active

メディアパイプラインは、[CreateMediaPipelineKinesisVideoStreamPoolAPI](#) がプールを作成した後にこのイベントを送信します。

```
{
  "version": "0",
  "id": "5ee6265a-0a40-104e-d8fd-a3b4bdd78483",
  "detail-type": "Chime Media Pipeline Kinesis Video Pool State Change",
  "source": "aws.chime",
  "account": "111122223333",
  "time": "2021-07-28T20:20:49Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "eventType": "chime:MediaPipelineKinesisVideoStreamPoolActive",
    "timestamp": 1627503649251,
    "mediaRegion": "ap-southeast-1",
    "poolArn" : "ARN of the KVS Pool"
  }
}
```

Amazon Chime Chime Media Pipeline Kinesis Video Pool Updated

メディアパイプラインは、[UpdateMediaPipelineKinesisVideoStreamPoolAPI](#) がプールを更新した後にこのイベントを送信します。

```
{
  "version": "0",
  "id": "5ee6265a-0a40-104e-d8fd-a3b4bdd78483",
  "detail-type": "Chime Media Pipeline Kinesis Video Pool State Change",
  "source": "aws.chime",
  "account": "111122223333",
  "time": "2021-07-28T20:20:49Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
```

```

    "eventType": "chime:MediaPipelineKinesisVideoStreamPoolUpdated",
    "timestamp": 1627503649251,
    "mediaRegion": "ap-southeast-1",
    "poolArn" : "ARN of the KVS Pool"
  }
}

```

Amazon Chime Media Pipeline Kinesis Video Pool Deleted

がプール [DeleteMediaPipelineKinesisVideoStreamPool](#) を削除すると、メディアパイプラインはこのイベントを Event Bridge に送信します。

プールの削除について詳しくは、このセクションの「[Kinesis Video Streams プールの作成](#)」を参照してください。

```

{
  "version": "0",
  "id": "5ee6265a-0a40-104e-d8fd-a3b4bdd78483",
  "detail-type": "Chime Media Pipeline Kinesis Video Pool State Change",
  "source": "aws.chime",
  "account": "111122223333",
  "time": "2021-07-28T20:20:49Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {If the attendeeId and externalUserId fields are empty, the media
    pipeline sends mixed audio to the KVS stream.
    "eventType": "chime:MediaPipelineKinesisVideoStreamPoolDeleted",
    "timestamp": 1627503649251,
    "mediaRegion": "ap-southeast-1",
    "poolArn" : "ARN of the KVS Pool"
  }
}

```

Amazon Chime Media Pipeline Kinesis Video Pool Temporary Failure

動画プールで一時的に障害が発生すると、メディアパイプラインによって次のイベントが Event Bridge に送信されます。

```

{
  "version": "0",
  "id": "5ee6265a-0a40-104e-d8fd-a3b4bdd78483",
  "detail-type": "Chime Media Pipeline Kinesis Video Pool State Change",

```



```
"source": "aws.chime",
"account": "111122223333",
"time": "2021-07-28T20:20:49Z",
"region": "us-east-1",
"resources": [],
"detail": {
  "eventType": "chime:MediaPipelineKinesisVideoStreamPoolTemporaryFailure",
  "timestamp": 1627503649251,
  "mediaRegion": "ap-southeast-1",
  "poolArn" : "ARN of the KVS Pool"
}
}
```

Amazon Chime Media Pipeline Kinesis Video Pool Permanent Failure

動画プールに恒久的に障害が発生すると、メディアパイプラインによって次のイベントが Event Bridge に送信されます。

```
{
  "version": "0",
  "id": "5ee6265a-0a40-104e-d8fd-a3b4bdd78483",
  "detail-type": "Chime Media Pipeline Kinesis Video Pool State Change",
  "source": "aws.chime",
  "account": "111122223333",
  "time": "2021-07-28T20:20:49Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "eventType": "chime:MediaPipelineKinesisVideoStreamPoolPermanentFailure",
    "timestamp": 1627503649251,
    "mediaRegion": "ap-southeast-1",
    "poolArn" : "ARN of the KVS Pool"
  }
}
```

メディアストリームパイプラインデータの使用

通知中のメタデータを使用して KVS ARN、フラグメント番号、フラグメントタイムスタンプを取得できます。そうした情報は、KVS ストリームで音声データを処理するのに役立ちます。

また、KVS ARN と KVS API を組み合わせてストリームからデータを読み取ることもできます。ユースケースに応じて、[GetMedia](#)および [GetMediaForFragmentList](#) APIs。通

常、GetMediaForFragmentList呼び出しの前に [ListFragments](#) API への呼び出しが続きます。詳細については、「Amazon Kinesis Video Streams のよくある質問」の「[ストリームからのデータの読み取り](#)」を参照してください。

ユースケースに応じて、ビルダーは Kinesis Video Streams パーサーライブラリを使用できます。これにより、KVS [GetMedia](#) API が使用されます。

メディアストリームパイプラインによって、次の会議および参加者のメタデータが各フラグメントに追加されます。

```
"meetingId"  
"externalMeetingId"  
"attendeeId"  
"externalUserId"  
"sampleRate"  
"channels"
```

メディアデータは MKV 形式で格納されます。MKV 音声データはすべて AAC でエンコードされます。詳細については、「Kinesis Video Streams デベロッパーガイド」の「[Kinesis Video Streams データモデル](#)」を参照してください。

メディアパイプライン用のサービスにリンクされたロールを作成する

以下のセクションでは、Amazon Chime SDK ミーティングへのアクセスをメディアパイプラインに許可するサービスにリンクされたロールの作成方法について説明します。

トピック

- [ロールのアクセス許可を設定する](#)
- [サービスリンクロールの作成](#)
- [サービスリンクロールの編集](#)
- [サービスリンクロールの削除](#)
- [サービスにリンクされたロールをサポートするリージョン](#)

ロールのアクセス許可を設定する

メディアパイプラインは、`という名前のサービスにリンクされたロール`を使用します `AWSServiceRoleForAmazonChimeSDKMediaPipelines`。このロールにより、キャプチャパイプラ

インはユーザーに代わって Amazon Chime SDK 会議にアクセスし、Amazon にメトリクスを発行 CloudWatch できます。このロールでは `mediapipelines.chime.amazonaws.com` サービスを信頼します。

ロールのアクセス許可ポリシーにより、Amazon Chime SDK はすべての AWS リソースに対して次のアクションを実行できます。

- アクション: all AWS resources 上で `cloudwatch:PutMetricData`
- アクション: all AWS resources 上で `chime:CreateAttendee`
- アクション: all AWS resources 上で `chime>DeleteAttendee`
- アクション: all AWS resources 上で `chime:GetMeeting`
- アクション: `arn:aws:kinesisvideo:*:111122223333:stream/ChimeMediaPipelines-*` 上で `kinesisvideo:CreateStream`
- アクション: `arn:aws:kinesisvideo:*:111122223333:stream/ChimeMediaPipelines-*` 上で `kinesisvideo:PutMedia`
- アクション: `arn:aws:kinesisvideo:*:111122223333:stream/ChimeMediaPipelines-*` 上で `kinesisvideo:UpdateDataRetention`
- アクション: `arn:aws:kinesisvideo:*:111122223333:stream/ChimeMediaPipelines-*` 上で `kinesisvideo:DescribeStream`
- アクション: `arn:aws:kinesisvideo:*:111122223333:stream/ChimeMediaPipelines-*` 上で `kinesisvideo:GetDataEndpoint`
- アクション: `arn:aws:kinesisvideo:*:111122223333:stream/*` 上で `kinesisvideo:ListStreams`

サービスにリンクされたロールの作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) に許可するには、アクセス許可を設定する必要があります。詳細については、「IAM ユーザーガイド」の「[Service linked role permissions](#)」を参照してください。

サービスリンクロールの作成

IAM コンソールを使用して、Amazon Chime SDK メディアパイプラインで使用するためのサービスにリンクされたロールを作成できます。これらのステップを完了するには、IAM 管理者権限が必要です。お持ちでない場合、システム管理者に相談してください。

ロールを作成するには

1. AWS マネジメントコンソールにサインインし、<https://console.aws.amazon.com/iam/> で IAM コンソールを開きます。
2. IAM コンソールのナビゲーションペインで、[ロール]、[ロールを作成] を選択します。
3. [AWS サービス] ロールタイプを選択した後、[Chime SDK メディアパイプライン] を選択します。

IAM ポリシーが表示されます。

4. ポリシーの横にあるチェックボックスを選択してから、[次へ: タグ] を選択します。
5. [Next: Review] (次へ: 確認) を選択します。
6. 必要に応じて説明を編集してから [Create role] (ロールの作成) を選択します。

AWS CLI または AWS API を使用して、mediapipelines.chime.amazonaws.com という名前のサービスにリンクされたロールを作成することもできます。AWS CLI で、次のコマンドを実行します。

```
aws iam create-service-linked-role --aws-service-name
mediapipelines.chime.amazonaws.com
```

ロールの作成の詳細については、「IAM ユーザーガイド」の「[サービスにリンクされたロールを作成する](#)」を参照してください。このロールを削除する場合、この同じプロセスを使用して、もう一度ロールを作成できます。

サービスリンクロールの編集

AWSServiceRoleForAmazonChimeSDKMediaPipelines サービスにリンクされたロールを編集することはできません。このロールは他のエンティティによって参照される場合があるため、ロールの作成後にロールの名前を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、IAM ユーザーガイドの「[サービスリンクロールの編集](#)」を参照してください。

サービスリンクロールの削除

サービスにリンクされたロールが不要な場合は、削除することを推奨します。そのためには、まず、そのロールを使用するメディアパイプラインを削除します。AWS CLI または [DeleteMediaCapturePipeline](#) API を使用してパイプラインを削除できます。

CLI を使用してパイプラインを削除する

AWS CLI でこのコマンドを使用して、アカウントのメディアパイプラインを削除します。

```
aws chime-sdk-media-pipelines delete-media-capture-pipeline --media-pipeline-id Pipeline_Id
```

API を使用してパイプラインを削除する

[DeleteMediaCapturePipeline](#) API を使用して、アカウントのメディアパイプラインを削除します。

ロールの削除

パイプラインを削除したら、IAM コンソール、AWS CLI、または AWS API を使用してロールを削除できます。ロールの削除の詳細については、「IAM ユーザーガイド」の「[サービスにリンクされたロールの削除](#)」を参照してください。

サービスにリンクされたロールをサポートするリージョン

Amazon Chime SDK は、サービスが利用可能なすべての AWS リージョンでサービスにリンクされたロールの使用をサポートしています。詳細については、「Amazon Web Services 全般のリファレンス」の「[Amazon Chime SDK エンドポイントとクォータ](#)」を参照してください。

メディアパイプラインイベントを使用する

各タイプのメディアパイプラインでライフサイクルイベントを送信し、これを使用して通知をトリガーし、ダウンストリームワークフローを開始することができます。メディアパイプラインイベントの使用例をいくつか次に示します。

- メディアパイプラインが完了した後に、キャプチャしたメディアを処理する。
- メディアパイプラインに一時的な障害が発生した場合に、会議参加者に通知する。
- メディアパイプラインに恒久的に障害が発生した場合に、会議を中止する。

Amazon、Amazon Simple Notification Service (SNS) EventBridge、および Amazon Simple Queue Service (SQS) にイベントを送信できます。詳細については、「Amazon ユーザーガイド」の「[AWS のサービスからのイベント](#)」を参照してください。 EventBridge

Amazon Chime SDK media pipeline created

Amazon Chime SDK では、メディアパイプラインが作成されたときにこのイベントを送信します。

例: イベントデータ

以下はこのイベントのサンプルデータです。

```
{
  "version": "0",
  "id": "5ee6265a-0a40-104e-d8fd-a3b4bdd78483",
  "detail-type": "Chime Media Pipeline State Change",
  "source": "aws.chime",
  "account": "111122223333",
  "time": "2021-07-28T20:20:49Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "version": "0",
    "eventType": "chime:MediaPipelineInProgress",
    "timestamp": 1627503649251,
    "meetingId": "1e6bf4f5-f4b5-4917-b8c9-bda45c340706",
    "externalMeetingId": "Meeting_Id",
    "mediaPipelineId": "e40ee45e-2ed1-408e-9156-f52b8208a491",
    "mediaRegion": "ap-southeast-1"
  }
}
```

Amazon Chime SDK media pipeline deleted

Amazon Chime SDK では、メディアパイプラインが削除されたときにこのイベントを送信します。

例: イベントデータ

以下はこのイベントのサンプルデータです。

```
{
  "version": "0",
  "id": "9e11e429-97fd-9532-5670-fac3f7abc05f",
  "detail-type": "Chime Media Pipeline State Change",
  "source": "aws.chime",
  "account": "365135496707",
  "time": "2021-07-28T20:21:50Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "version": "0",
    "eventType": "chime:MediaPipelineDeleted",
    "timestamp": 1627503710485,
    "meetingId": "1e6bf4f5-f4b5-4917-b8c9-bda45c340706",
  }
}
```

```
    "externalMeetingId": "Meeting_Id",
    "mediaPipelineId": "e40ee45e-2ed1-408e-9156-f52b8208a491",
    "mediaRegion": "ap-southeast-1"
  }
}
```

Amazon Chime SDK media pipeline has a temporary failure

Amazon Chime SDK では、メディアパイプラインに一時的な障害が発生したときにこのイベントを送信します。

例: イベントデータ

以下はこのイベントのサンプルデータです。

```
{
  "version": "0",
  "id": "abc141e1-fc2e-65e8-5f18-ab5130f1035a",
  "detail-type": "Chime Media Pipeline State Change",
  "source": "aws.chime",
  "account": "365135496707",
  "time": "2021-07-28T21:16:42Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "version": "0",
    "eventType": "chime:MediaPipelineTemporaryFailure",
    "timestamp": 1627507002882,
    "meetingId": "7a5434e3-724a-4bbb-9eb6-2fb209dc0706",
    "externalMeetingId": "Meeting_Id",
    "mediaPipelineId": "ebd62f4e-04a9-426d-bcb0-974c0f266400",
    "mediaRegion": "eu-south-1"
  }
}
```

Amazon Chime SDK メディアパイプラインが進行中

Amazon Chime SDK は、メディアパイプラインがアーティファクトのキャプチャを開始したときにこのイベントを送信します。

例: イベントデータ

以下はこのイベントのサンプルデータです。

```
{
  "version": "0",
  "id": "9e11e429-97fd-9532-5670-fac3f7abc05f",
  "detail-type": "Chime Media Pipeline State Change",
  "source": "aws.chime",
  "account": "365135496707",
  "time": "2021-07-28T20:21:50Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "version": "0",
    "eventType": "chime:MediaPipelineInProgress",
    "timestamp": 1627503710485?,
    "meetingId": "1e6bf4f5-f4b5-4917-b8c9-bda45c340706",
    "externalMeetingId": "Meeting_Id",
    "mediaPipelineId": "e40ee45e-2ed1-408e-9156-f52b8208a491",
    "mediaRegion": "ap-southeast-1"
  }
}
```

Amazon Chime SDK media pipeline permanent failure

Amazon Chime SDK では、メディアパイプラインに恒久的に障害が発生したときにこのイベントを送信します。

例: イベントデータ

以下はこのイベントのサンプルデータです。

```
{
  "version": "0",
  "id": "9e11e429-97fd-9532-5670-fac3f7abc05f",
  "detail-type": "Chime Media Pipeline State Change",
  "source": "aws.chime",
  "account": "365135496707",
  "time": "2021-07-28T20:21:50Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "version": "0",
    "eventType": "chime:MediaPipelinePermanentFailure",
    "timestamp": 1627503710485,
    "meetingId": "1e6bf4f5-f4b5-4917-b8c9-bda45c340706",
  }
}
```



```
    "externalMeetingId": "Meeting_Id",
    "mediaPipelineId": "e40ee45e-2ed1-408e-9156-f52b8208a491",
    "mediaRegion": "ap-southeast-1"
  }
}
```

Amazon S3 バケット許可をセッティングする

Amazon S3 バケットを作成していない場合は、会議をホストするアカウントおよびリージョンに必ずバケットを作成してください。また、必ずサービスに適切なアクセス許可を付与してください。Amazon S3 バケットの作成に関する詳細については、「[Amazon S3 バケットの作成](#)」を参照してください。

CloudTrail へのメディアパイプラインイベントの送信

AWS では、AWS アカウントを作成すると自動的に CloudTrail が有効になります。ユーザーがメディアパイプライン SDK でサポートされている API を呼び出すと、CloudTrail はその API のアクティビティを他の AWS イベントと共にイベント履歴に記録します。AWS アカウント内のメディアパイプラインイベントを表示、検索、ダウンロードできます。詳細については、「CloudTrail ユーザーガイド」の「[CloudTrail イベント履歴でのイベントの表示](#)」を参照してください。

メディアパイプラインイベントを継続的に記録するために、証跡を作成できます。証跡により、CloudTrail はログファイルをお客様の Amazon S3 バケットに配信できます。以下の例は、メディアパイプライン証跡を示しています。データには、API を呼び出したユーザー、API の呼び出しに使用された IAM ロール、タイムスタンプが含まれます。CloudTrail の使用に関する詳細については、「Amazon Chime SDK Administrator Guide」の「[Logging and monitoring](#)」を参照してください。

```
{
  "Records": [
    {
      "eventVersion": "1.08",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "ABCDEFGHJKLMNOPQRSTUVWXYZ:user-name",
        "arn": "arn:aws:sts::123456789101:assumed-role/role-name/user-name",
        "accountId": "109876543210",
        "accessKeyId": "ABCDEFGHJKLMNOPQRSTUVWXYZ",
        "sessionContext": {
          "sessionIssuer": {
            "type": "Role",
```

```
        "principalId": "ABCDEFGHJKLMNOPQRSTU",
        "arn": "arn:aws:iam::109876543210:role/role-name",
        "accountId": "012345678910",
        "userName": "user-name"
    },
    "webIdFederationData": {},
    "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2022-03-08T19:34:55Z"
    }
}
},
"eventTime": "2022-03-08T20:28:41Z",
"eventSource": "chime-sdk-media-pipelines.amazonaws.com",
"eventName": "CreateMediaCapturePipeline",
"awsRegion": "us-east-1",
"sourceIPAddress": "127.0.0.1",
"userAgent": "[ ]/[ ]",
"requestParameters": {
    "sourceType": "ChimeSdkMeeting",
    "sourceArn": "Hidden_For_Security_Reasons",
    "sinkType": "S3Bucket",
    "sinkArn": "Hidden_For_Security_Reasons",
    "chimeSdkMeetingConfiguration": {
        "artifactsConfiguration": {
            "audio": {
                "muxType": "AudioOnly"
            },
            "video": {
                "state": "Enabled",
                "muxType": "VideoOnly"
            },
            "content": {
                "state": "Enabled",
                "muxType": "ContentOnly"
            }
        }
    }
}
},
"responseElements": {
    "mediaCapturePipeline": {
        "mediaPipelineId": "pipeline-uuid",
        "sourceType": "ChimeSdkMeeting",
        "sourceArn": "Hidden_For_Security_Reasons",
```

```
"status": "Initializing",
"sinkType": "S3Bucket",
"sinkArn": "Hidden_For_Security_Reasons",
"createdTimestamp": "2022-03-08T20:28:41.336Z",
"updatedTimestamp": "2022-03-08T20:28:41.463Z",
"chimeSdkMeetingConfiguration": {
  "artifactsConfiguration": {
    "audio": {
      "muxType": "AudioOnly"
    },
    "video": {
      "state": "Enabled",
      "muxType": "VideoOnly"
    },
    "content": {
      "state": "Enabled",
      "muxType": "ContentOnly"
    }
  }
},
"requestID": "request-id",
"eventID": "event-id",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "112233445566",
"tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "clientProvidedHostHeader": "example.com"
},
]
```

トランスクリプトを解析する

次のコマンドを使用して、文字起こしメッセージから文字起こしの内容を解析します。このコマンドでは transcript-message.txt ファイルから全文を解析します。

```
with open('transcript-message.txt') as f:
```

```
for line in f:
    result_json = json.loads(line)["transcript"]["results"][0]
    if result_json['isPartial'] == False:
        print(result_json["alternatives"][0]["transcript"])
```

パイプラインを停止するためのベストプラクティス

メディアパイプラインを停止するためのベストプラクティスとして、[DeleteMediaPipeline](#) API を呼び出します。この API では、メディアキャプチャパイプラインとメディアライブコネクタパイプラインを削除できます。[DeleteMediaCapturePipeline](#) API を呼び出して、メディアキャプチャパイプラインを削除することもできます。会議が終了すると、すべてのメディアパイプラインが停止します。

Amazon Chime SDK ライブ文字起こしの使用

Amazon Chime SDK ライブ文字起こしを使用すると、ユーザー属性付きの会議の文字起こしがリアルタイムで生成されます。Amazon Chime SDK ライブ文字起こしは、Amazon Transcribe サービスおよび Amazon Transcribe Medical サービスと統合され、Amazon Chime SDK ミーティングの文字起こしを会議の進行中に生成します。

Amazon Chime SDK ライブ文字起こしは、各ユーザーの音声を個別に処理することで、マルチスピーカーシナリオでの精度を向上させます。Amazon Chime SDK は、アクティブトーカーアルゴリズムを使用して上位 2 人のアクティブトーカーを選択し、それらのユーザーの音声を個別のチャンネルに分離したうえで 1 つのストリームとして Amazon Transcribe に送信します。会議の参加者は、Amazon Chime SDK データメッセージを介してユーザー属性付きの文字起こしを受け取ります。文字起こしは、字幕の表示、会議の文字起こしの作成、文字起こしを使用したコンテンツ分析など、さまざまな方法で使用できます。

ライブ文字起こしでは、会議の文字起こしを行っている間、Amazon Transcribe への 1 つのストリームを使用します。Amazon Transcribe および Amazon Transcribe Medical の標準コストが適用されます。詳細については、「[Amazon Transcribe の料金](#)」を参照してください。使用量または請求に関するご質問は、AWS アカウントマネージャーにお問い合わせください。

Important

デフォルトでは、Amazon Transcribe は、サービス [AWS 条件](#) のセクション 50 で詳しく説明されているように、サービスによって処理されたオーディオコンテンツを使用して保存し、AWS AI/ML サービスを開発および改善する場合があります。Amazon Transcribe の使用は、電子通信記録または傍受に関する連邦および州の法律または規制の対象となる場合が

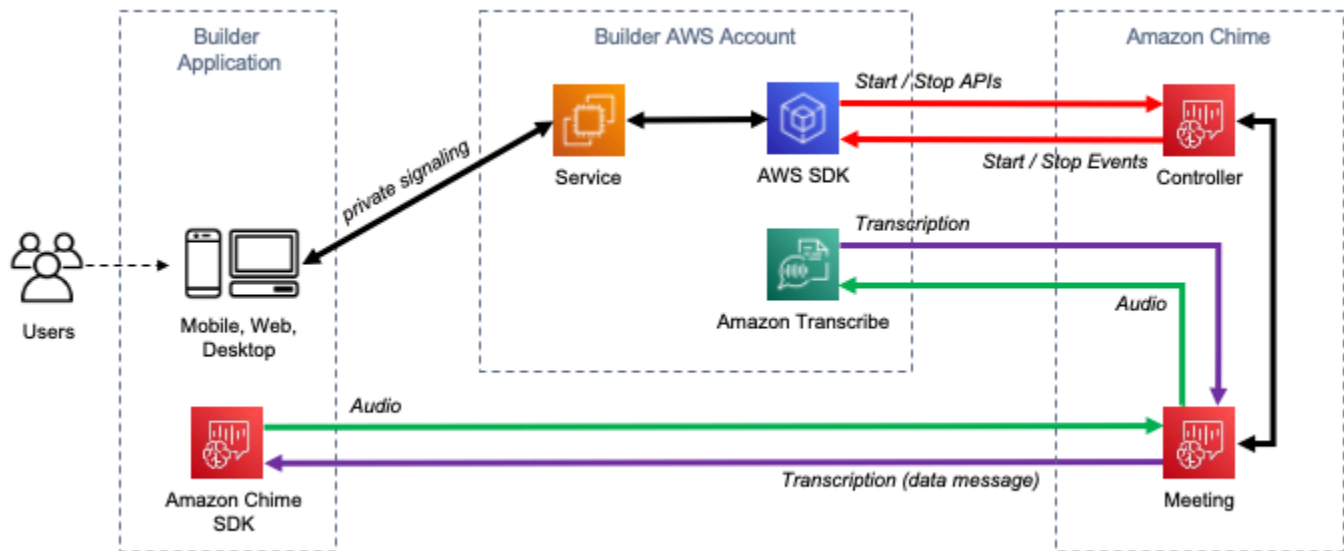
あります。録音に関するすべての適用法を遵守することは、お客様とそのエンドユーザーの責任です。これには、録音セッションまたは通信のすべての参加者にセッションまたは通信が録音されていることを適切に通知し、必要なすべての同意を得ることを含みます。AWS Organizations AWS を使用して AWS AI サービスのオプトアウトポリシーを設定することで、オーディオコンテンツを使用して AI/ML サービスを開発および改善することをオプトアウトできます。

トピック

- [システムアーキテクチャ](#)
- [請求と使用状況](#)
- [アカウントの設定](#)
- [文字起こしオプションの選択](#)
- [文字起こしの開始と停止](#)
- [文字起こしパラメータ](#)
- [文字起こしイベント](#)
- [文字起こしメッセージ](#)
- [配信例](#)

システムアーキテクチャ

Amazon Chime SDK は、Amazon Transcribe または Amazon Transcribe Medical アカウントとサービス側の統合を介して、音声は AWS ネットワークを離れることなく、リアルタイムの会議 Amazon Transcribe を作成します。精度を向上させるため、ユーザーの音声は個別に処理されてから、会議の音声に融合されます。Amazon Chime SDK は、アクティブトーカーアルゴリズムを使用して上位 2 人のアクティブトーカーを選択し、それらのユーザーの音声を個別のチャンネルに分離したうえで 1 つのストリームとして Amazon Transcribe または Amazon Transcribe Medical に送信します。レイテンシーを低減するため、ユーザー属性付きの文字起こしは、データメッセージを介してすべての会議参加者に直接送信されます。メディアパイプラインを使用して会議の音声をキャプチャする場合、会議の文字起こし情報もキャプチャされます。



請求と使用状況

ライブ文字起こしでは、会議の文字起こしを行っている間、Amazon Transcribe または Amazon Transcribe Medical への 1 つのストリームを使用します。Amazon Transcribe および Amazon Transcribe Medical の標準コストが適用されます。詳細については、「[Amazon Transcribe の料金](#)」を参照してください。使用量または請求に関するご質問は、AWS アカウントマネージャーにお問い合わせください。

アカウントの設定

Amazon Chime SDK ライブ文字起こしを使用する前に、AWS アカウントで Amazon Transcribe と Amazon Transcribe Medical を呼び出すためのアクセス許可を Amazon Chime SDK に付与する必要があります。そのためには、Chime Transcription サービスリンクロールをアカウントに追加します。ライブ文字起こし用のサービスリンクロールの作成については、「Amazon Chime SDK Administration Guide」の「[Using roles with live transcription](#)」を参照してください。IAM サービスリンクロールの詳細については、「IAM ユーザーガイド」の「[Service Linked Roles](#)」を参照してください。

文字起こしオプションの選択

Amazon Chime SDK ライブ文字起こしを使用する場合は、AWS アカウントで [Amazon Transcribe](#) または [Amazon Transcribe Medical](#) を使用します。[Amazon Transcribe がサポートするすべてのストリーミング言語](#)に加えて、[カスタム語彙](#)や[語彙フィルター](#)などの機能にもアクセスできま

す。Amazon Transcribe Medical を使用する場合、医療専門分野や会話タイプを選択することができ、オプションで任意のカスタム語彙を指定することもできます。Amazon Transcribe および Amazon Transcribe Medical の標準コストが適用されます。

文字起こしオプションを選択するプロセスでは、以下のステップを実行します。

ステップ 1: 文字起こしサービスを選択する

[Amazon Transcribe](#) または [Amazon Transcribe Medical](#) のどちらの文字起こしサービスを使用するかを決める必要があります。

医療分野の音声テキスト変換機能が必要なユースケースでは、ほとんどの場合に Amazon Transcribe Medical を使用することが推奨されます。それ以外のユースケースでは、ほとんどの場合に Amazon Transcribe を使用することが推奨されます。

StartMeetingTranscription API を呼び出すときに使用する文字起こしサービスを指定します。

- Amazon Transcribe を使用するには、TranscriptionConfiguration と EngineTranscribeSettings を指定します。
- Amazon Transcribe Medical を使用するには、TranscriptionConfiguration と EngineTranscribeMedicalSettings を指定します。

ステップ 2: 文字起こしのリージョンを選択する

文字起こしサービスの AWS リージョンを選択する必要があります。Amazon Transcribe および Amazon Transcribe Medical で利用可能な AWS リージョンについては、[AWS 「リージョンサービス」](#) 表を参照してください。

一般的に、会議のメディアリージョンと文字起こしリージョンの間のレイテンシーが低いほど、ユーザーエクスペリエンスが高くなります。レイテンシーを最低限に抑えるために、可能な限りメディアと文字起こしに同じリージョンを使用してください。ただし、リージョンを選択する際には、規制要件や、Amazon Transcribe または Amazon Transcribe Medical を設定したリージョンなど、他の要素を考慮する必要がある場合があります。

Amazon Transcribe と Amazon Transcribe Medical の機能 (カスタム語彙や語彙フィルターなど) は、リージョンに固有のものです。これらの機能のいずれかを設定する場合は、ライブ文字起こしを使用する予定のすべての AWS リージョンで同じように設定する必要があります。または、すべての会議に同じ Amazon Transcribe リージョンを使用することもできます。

文字起こしサービスで使用するリージョンを指定できます。これを行うには、StartMeetingTranscription API を呼び出すときに、文字起こしエンジン設定の Region フィールドにリージョン名を追加します。リージョンを指定しない場合、Amazon Chime SDK は会議のメディアリージョンで文字起こしサービスの使用を試みます。Amazon Chime SDK で文字起こしサービスのリージョンが自動的に選択されるようにするには、Region フィールドで auto を指定します。そうすると、Amazon Chime は、以下の表で説明されているように、会議のメディアリージョンに基づいて文字起こしサービスのリージョンを選択します。StartMeetingTranscription API の詳細については、本ガイドの「[文字起こしの開始と停止](#)」を参照してください。

Note

Amazon Chime SDK によって選択された文字起こしリージョンは、Amazon Chime SDK AWS、Amazon Transcribe、Amazon Transcribe Medical がより多くのリージョンを利用可能にするため、変更される可能性があります。

Amazon Transcribe の自動リージョン選択

Amazon Chime SDK メディアリージョン	リージョンコード	文字起こしリージョン
米国東部 (オハイオ)	us-east-2	us-east-2
米国東部 (バージニア北部)	us-east-1	us-east-1
米国西部 (北カリフォルニア)	us-west-1	us-west-2
米国西部 (オレゴン)	us-west-2	us-west-2
アフリカ (ケープタウン)**	af-south-1	eu-west-2
アジアパシフィック (ムンバイ)	ap-south-1	eu-west-2
アジアパシフィック (ソウル)	ap-northeast-2	ap-northeast-2
アジアパシフィック (シンガポール)	ap-southeast-1	ap-northeast-1

Amazon Chime SDK メディア リージョン	リージョンコード	文字起こしリージョン
アジアパシフィック (シドニー)	ap-southeast-2	ap-southeast-2
アジアパシフィック (東京)	ap-northeast-1	ap-northeast-1
カナダ (中部)	ca-central-1	ca-central-1
欧州 (フランクフルト)	eu-central-1	eu-central-1
欧州 (アイルランド)	eu-west-1	eu-west-1
欧州 (ロンドン)	eu-west-2	eu-west-2
欧州 (ミラノ)**	eu-south-1	eu-central-1
欧州 (パリ)	eu-west-3	eu-central-1
欧州 (ストックホルム)	eu-north-1	eu-central-1
南米 (サンパウロ)	sa-east-1	sa-east-1
GovCloud (米国東部)	us-gov-east-1	us-gov-west-1
GovCloud (米国西部)	us-gov-west-1	us-gov-west-1

Amazon Transcribe Medical の自動リージョン選択

Amazon Chime SDK メディア リージョン	リージョンコード	文字起こしリージョン
米国東部 (オハイオ)	us-east-2	us-east-2
米国東部 (バージニア北部)	us-east-1	us-east-1
米国西部 (北カリフォルニア)	us-west-1	us-west-2

Amazon Chime SDK メディア リージョン	リージョンコード	文字起こしリージョン
米国西部 (オレゴン)	us-west-2	us-west-2
アフリカ (ケープタウン)**	af-south-1	eu-west-1
アジアパシフィック (ムンバイ)	ap-south-1	eu-west-1
アジアパシフィック (ソウル)	ap-northeast-2	us-west-2
アジアパシフィック (シンガポール)	ap-southeast-1	ap-southeast-2
アジアパシフィック (シドニー)	ap-southeast-2	ap-southeast-2
アジアパシフィック (東京)	ap-northeast-1	us-west-2
カナダ (中部)	ca-central-1	ca-central-1
欧州 (フランクフルト)	eu-central-1	eu-west-1
欧州 (アイルランド)	eu-west-1	eu-west-1
欧州 (ロンドン)	eu-west-2	us-east-1
欧州 (ミラノ)**	eu-south-1	eu-west-1
欧州 (パリ)	eu-west-3	eu-west-1
欧州 (ストックホルム)	eu-north-1	eu-west-1
南米 (サンパウロ)	sa-east-1	us-east-1

Note

アスタリスク (*) が付いたリージョンでライブ文字起こしを使用するには、まず AWS アカウントでリージョンを有効にする必要があります。詳細については、「AWS 全般のリファレンス」の「[リージョンを有効にする](#)」を参照してください。

各サービスのリージョンとエンドポイントの詳細については、以下を参照してください。

- [Amazon Chime SDK メディアリージョン](#)
- [Amazon Transcribe エンドポイントとクォータ](#)
- [Amazon Transcribe Medical エンドポイントとクォータ](#)

ステップ 3: サービスクォータを確認する

ライブ文字起こしを使用する Amazon Chime SDK ミーティングごとに、Amazon Transcribe または Amazon Transcribe Medical への HTTP/2 ストリームが 1 つだけ必要です。どちらのサービスにも、同時 HTTP/2 ストリームの数と 1 秒あたりのストリーム開始トランザクション数に対応したリージョン別のサービスクォータがあります。クォータの詳細については、「Amazon Transcribe Developer Guide」の「[Guidelines and quotas](#)」を参照してください。クォータの引き上げについては、AWS コンソールの「Service Quotas」を参照してください。

文字起こしの開始と停止

Amazon Chime SDK [StartMeetingTranscription](#) API を使用して、会議に を適用して会議の文字起こし TranscriptionConfiguration を開始します。Amazon Chime SDK コントローラは、設定を会議に非同期的に転送します。会議の文字起こしの開始の成功または失敗は、Amazon Simple Notification Service (Amazon SNS) と Amazon 経由でメッセージを通じて通知されます EventBridge。

文字起こしを開始する

この例は、Amazon Transcribe でライブ文字起こしを開始する方法を示しています。

```
POST /meetings/meetingId/transcription?operation=start HTTP/1.1
Content-type: application/json
{
  "TranscriptionConfiguration": {
```

```

    "EngineTranscribeSettings": {
      "LanguageCode": "en-US",
      "VocabularyFilterMethod": "tag",
      "VocabularyFilterName": "profanity",
      "VocabularyName": "lingo",
      "Region": "us-east-1"
      "EnablePartialResultsStabilization": true,
      "PartialResultsStability": "high",
      "ContentIdentificationType": "PII",
      "ContentRedactionType": "PII",
      "PiiEntityTypes": "ALL",
      "LanguageModelName": "language-model"
    }
  }
}

```

この例は、Amazon Transcribe Medical でライブ文字起こしを開始する方法を示しています。

```

POST /meetings/meetingId/transcription?operation=start HTTP/1.1
Content-type: application/json
{
  "TranscriptionConfiguration": {
    "EngineTranscribeMedicalSettings": {
      "LanguageCode": "en-US",
      "Specialty": "PRIMARYCARE",
      "Type": "CONVERSATION",
      "VocabularyName": "lingo",
      "Region": "us-east-1",
      "ContentIdentificationType": "PHI",
    }
  }
}

```

StartMeetingTranscription – 会議の文字起こしを開始します。

meetingId – [CreateMeeting API](#) によって返される会議の ID。

TranscriptionConfiguration – ライブ文字起こしのパラメータをカプセル化します。設定は 1 つだけ (EngineTranscribeSettings または EngineTranscribeMedicalSettings) 指定する必要があります。

EngineTranscribeSettings – Amazon Transcribe の使用を指定し、その設定を に渡しま
す [Amazon Transcribe](#)。

LanguageCode – 必須。

VocabularyFilterMethod - オプション。

VocabularyFilterName - オプション。

VocabularyName - オプション。

Region - オプション。

EnablePartialResultsStabilization - オプション。

PartialResultsStability - オプション。

ContentIdentificationType - オプション。

ContentRedactionType - オプション。

PiiEntityTypes - オプション。

LanguageModelName - オプション。

EngineTranscribeMedicalSettings — Amazon Transcribe Medical の使用を指定し、その設定
を に渡します [Amazon Transcribe Medical](#)。

LanguageCode – 必須。

Speciality – 必須。

Type – 必須。

VocabularyName - オプション。

Region - オプション。

ContentIdentificationType - オプション。

レスポンス

Amazon Transcribe と Amazon Transcribe Medical は、以下のレスポンスを返します。

- 会議に `TranscriptionConfiguration` が正常に適用された場合、OK (200) と空の本文。

エラーメッセージ

Amazon Transcribe と Amazon Transcribe Medical は、以下のエラーメッセージを表示します。

- `BadRequestException` (400): 入力パラメータがサービスの制約と一致しません。
- `ForbiddenException` (403): クライアントはリクエストを行うことを永続的に禁止されています。
- `NotFoundException` (404): `meetingId`は存在しません。
- `ResourceLimitExceededException` (400): リクエストがリソース制限を超えています。例えば、ライブ文字起こしが有効になっている会議が多すぎます。
- `ServiceFailureException` (500): サービスで予期しないエラーが発生しました。
- `ServiceUnavailableException` (503): サービスは現在利用できません。
- `ThrottledClientException` (429): クライアントがリクエストレート制限を超えました。
- `UnauthorizedClientException` (401): クライアントは現在、リクエストを行う権限がありません。

`StartMeetingTranscription` の 2 回目の呼び出しにより、会議に適用されている `TranscriptionConfiguration` が更新されます。

文字起こしの停止

[StopMeetingTranscription](#) API を使用して、特定の `TranscriptionConfiguration` の削除 `meetingId` し、会議の文字起こしを終了します。会議を終了すると、文字起こしは自動的に停止します。

この例は、`StopMeetingTranscription` を呼び出すリクエスト構文を示しています。

```
POST/meetings/meetingId/transcription?operation=stop HTTP/1.1
```

レスポンス

Amazon Transcribe と Amazon Transcribe Medical は、以下のレスポンスを返します。

- 会議から `TranscriptionConfiguration` が正常に削除された場合、OK (200) と空の本文。

エラーメッセージ

Amazon Transcribe と Amazon Transcribe Medical は、以下のエラーメッセージを表示します。

- `BadRequestException` (400): 入力パラメータがサービスの制約と一致しません。
- `ForbiddenException` (403): クライアントはリクエストを行うことを永久に禁止されます。
- `NotFoundException` (404): `meetingId`は存在しません。
- `ServiceFailureException` (500): サービスで予期しないエラーが発生しました。
- `ServiceUnavailableException` (503): サービスは現在利用できません。
- `ThrottledClientException` (429): クライアントがリクエストレート制限を超えました。
- `UnauthorizedClientException` (401): クライアントは現在、リクエストを行う権限がありません。

文字起こしパラメータ

Amazon Transcribe および Amazon Transcribe Medical APIs は、ストリーミング文字起こしを開始するときに、[StartStreamTranscription](#) や など、さまざまなパラメータを提供します。[StartMedicalStreamTranscription](#)。Amazon Chime SDK がパラメータの値を事前に決定していない限り、`StartMeetingTranscriptionAPI` で 1 つの `Hose` パラメータを使用できます。例えば、`MediaEncoding` パラメータと `MediaSampleRateHertz` パラメータは、Amazon Chime SDK によって自動的に設定されるため使用できません。

Amazon Transcribe と Amazon Transcribe Medical はパラメータを検証します。これにより、新しいパラメータ値が利用可能になりしだい、すぐにそれらを使用できるようになります。例えば、Amazon Transcribe Medical が新しい言語のサポートを開始した場合、必要なのは、`LanguageCode` パラメータに新しい言語値を指定することだけです。

文字起こしイベント

Amazon Chime SDK は文字起こしのライフサイクルイベントを送信します。これを使用して通知をトリガーし、ダウンストリームワークフローを開始できます。文字起こしイベントを使用する例としては、以下のようなものがあります。

- Amazon Chime SDK ミーティングでのライブ文字起こしの採用状況を測定する
- 言語設定を追跡する

Amazon、Amazon Simple Notification Service EventBridge、および Amazon Simple Queue Service にイベントを送信できます。詳細については、「Amazon ユーザーガイド」の「[AWS のサービスからのイベント](#)」を参照してください。EventBridge

Amazon Chime SDK ミーティングの文字起こしが開始されました

Amazon Chime SDK は、会議の文字起こしが開始されたとき、または [TranscriptionConfiguration](#) が更新されたときに、このイベントを送信します。

例: イベントデータ

以下はこのイベントのサンプルデータです。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "id": "12345678-1234-1234-1234-111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:TranscriptionStarted",
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "externalMeetingId": "mymeeting",
    "mediaRegion": "us-west-1",
    "transcriptionRegion": "us-west-2",
    "transcriptionConfiguration": "{...}"
  }
}
```

Amazon Chime SDK ミーティングの文字起こしが停止されました

Amazon Chime SDK は、会議の文字起こしが停止されると、このイベントを送信します。

例: イベントデータ

以下はこのイベントのサンプルデータです。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "id": "12345678-1234-1234-1234-111122223333",
```



```
"region": "us-east-1",
"detail-type": "Chime Meeting State Change",
"time": "yyyy-mm-ddThh:mm:ssZ",
"resources": []
"detail": {
  "version": "0",
  "eventType": "chime:TranscriptionStopped",
  "timestamp": 12344566754,
  "meetingId": "87654321-4321-4321-1234-111122223333",
  "externalMeetingId": "mymeeting",
  "mediaRegion": "us-west-1",
  "transcriptionRegion": "us-west-2",
  "transcriptionConfiguration": "{...}"
}
}
```

Amazon Chime SDK ミーティングの文字起こしが中断されました

Amazon Chime SDK は、会議の文字起こしが中断されると、このイベントを送信します。

例: イベントデータ

以下はこのイベントのサンプルデータです。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "id": "12345678-1234-1234-1234-111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:TranscriptionInterrupted",
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "externalMeetingId": "mymeeting",
    "message": "Internal server error",
    "mediaRegion": "us-west-1",
    "transcriptionRegion": "us-west-2",
    "transcriptionConfiguration": "{...}"
  }
}
```

```
}
```

Amazon Chime SDK ミーティングの文字起こしが再開されました

Amazon Chime SDK は、会議の文字起こしが中断後に再開されると、このイベントを送信します。

例: イベントデータ

以下はこのイベントのサンプルデータです。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
  "id": "12345678-1234-1234-1234-111122223333",
  "region": "us-east-1",
  "detail-type": "Chime Meeting State Change",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "eventType": "chime:TranscriptionResumed",
    "timestamp": 12344566754,
    "meetingId": "87654321-4321-4321-1234-111122223333",
    "externalMeetingId": "mymeeting",
    "mediaRegion": "us-west-1",
    "transcriptionRegion": "us-west-2",
    "transcriptionConfiguration": "{...}"
  }
}
```

Amazon Chime SDK ミーティングの文字起こしが失敗しました

Amazon Chime SDK は、会議の文字起こしが開始に失敗するか、中断後の再開に失敗すると、このイベントを送信します。

例: イベントデータ

以下はこのイベントのサンプルデータです。

```
{
  "version": "0",
  "source": "aws.chime",
  "account": "111122223333",
```

```
"id": "12345678-1234-1234-1234-111122223333",
"region": "us-east-1",
"detail-type": "Chime Meeting State Change",
"time": "yyyy-mm-ddThh:mm:ssZ",
"resources": []
"detail": {
  "version": "0",
  "eventType": "chime:TranscriptionFailed",
  "timestamp": 12344566754,
  "meetingId": "87654321-4321-4321-1234-111122223333",
  "externalMeetingId": "mymeeting",
  "message": "Internal server error",
  "mediaRegion": "us-west-1",
  "transcriptionRegion": "us-west-2",
  "transcriptionConfiguration": "{...}"
}
}
```

文字起こしメッセージ

Amazon Chime SDK サービスは、データメッセージで TranscriptEvent オブジェクトを送信することにより、文字起こし情報を参加者と共有します。TranscriptEvent は Transcript または TranscriptionStatus を配信します。

Transcript には、タイムスタンプとユーザー属性が付いた単語および句読点を含む結果が含まれます。結果が「部分的」なものになることもあります。その場合は通常、後続の TranscriptEvent で更新されます。これにより、文字起こしをすばやく確認することができ、必要に応じて後でインラインで更新を適用することもできます。

TranscriptionStatus は、次のセクションの例にリストされている TranscriptionStatusType イベントのいずれかを配信します。

新しいバージョンの Amazon Chime SDK には、TranscriptEvent の一般的な処理に使用する追加のデータ型とヘルパー関数が含まれています。

TranscriptEvent

この例は、典型的な文字起こしイベントを示しています。

```
type TranscriptEvent = Transcript | TranscriptionStatus;

export class TranscriptEventConverter {
```

```
static from(dataMessage: DataMessage): TranscriptEvent[] {
    // convert DataMessage to TranscriptEvents
    return ...
}

export default class TranscriptionStatus {
    type: TranscriptionStatusType;
    eventTimeMs:          number;
    transcriptionRegion:  string;
    transcriptionConfiguration: string;
    message?:             string;
}

enum TranscriptionStatusType {
    STARTED      = 'started',
    INTERRUPTED  = 'interrupted',
    RESUMED      = 'resumed',
    STOPPED      = 'stopped',
    FAILED       = 'failed',
}

export default class Transcript {
    results: TranscriptResult[]; // at least one
}

export class TranscriptResult {
    resultId:      string;
    isPartial:     boolean;
    startTimeMs:   number;
    endTimeMs:     number;
    alternatives:  TranscriptAlternative[]; // most confident first
}

export default class TranscriptAlternative {
    items: TranscriptItem[]; // in start time order
    transcript: string; // concatenated transcript items
    entities?: TranscriptEntity[];
}

export default class TranscriptItem {
    type:          TranscriptItemType;
    startTimeMs:   number;
    endTimeMs:     number;
}
```

```
    attendee:      Attendee;
    content:        string;
    vocabularyFilterMatch?: boolean;
    confidence?:   number;
    stable?:       boolean;
}

enum TranscriptItemType {
    PRONUNCIATION    = 'pronunciation', // content is a word
    PUNCTUATION      = 'punctuation', // content is punctuation
}

export default class TranscriptEntity {
    category:      string;
    confidence:    number;
    content:        string;
    endTimeMs:     number;
    startTimeMs:   number;
    type?:         string;
}

// This is an existing SDK model
export default class Attendee {
    attendeeId:    string;
    externalUserId: string;
}
```

データのガイドライン

これらのガイドラインに留意しながら作業を進めてください。

1. `transcription.results` には複数の結果が含まれる場合があります。
2. `transcription.results[i].isPartial = true` の場合、結果全体が更新される可能性があります。更新される可能性は高いですが、確実に更新されるわけではありません。更新後も `transcript.result[i].resultId` は同じです。信頼度の低い文字起こしを回避したい場合は、部分的な結果を完全にスキップできます。低レイテンシーの結果が必要な場合は、いったん部分的な結果を表示してから、後で更新が到着したときに完全に上書きすることができます。
3. `transcription.results[i].alternatives` には、常に少なくとも1つのエントリが含まれます。複数のエントリが含まれる場合は、最も信頼度の高いエントリがリストの先頭に表示されます。ほとんどの場合、`transcription.results[i].alternatives` の最初のエントリを採用し、その他のエントリは無視できます。

4. `transcription.results[i].alternatives[j].items` には、各単語または句読点のエントリが含まれます。
5. `transcription.results[i].alternatives[j].items[k].content` には、発言内容が含まれます。
6. `transcription.results[i].alternatives[j].items[k].attendee` は、コンテンツのユーザー属性 (誰) です。
7. `transcription.results[i].alternatives[j].items[k].startTimeMs` は、コンテンツの「いつ」です。これにより、単語が話された順序で、異なるユーザー間でユーザー属性の文字起こしを word-by-word レンダリングできます。
8. `transcription.results[i].alternatives[j].items[k].endTimeMs` フィールドは通常は無視して構いませんが、誰がいつ何を言ったかを完全に把握できるように提供されています。
9. `transcription.results[i].alternatives[j].items[k].vocabularyFilterMatch` は、コンテンツがフィルター内の単語と一致した場合は `true`、それ以外の場合は `false` です。
10. `transcription.results[i].alternatives[j].items[k].confidence` は、0~1 の値です。これは、項目のコンテンツが発言と正しく一致しているというエンジンの信頼度を示します。0 が最も低い信頼度、1 が最も高い信頼度を示します。
11. `transcription.results[i].alternatives[j].items[k].stable` は、現在の単語が部分的な結果の更新によって将来的に変更されるかどうかを示します。この値が `true` になるのは、リクエストで `EnablePartialResultsStabilization` を `true` に設定して部分的な結果の安定化機能を有効にした場合だけです。
12. `transcription.results[i].alternatives[j].entities` には、コンテンツ識別またはリダクション機能によって検出された各エンティティのエントリが含まれます。このリストには、コンテンツ識別またはリダクションを有効にした場合にのみ値が入ります。エンティティは、個人情報や個人の健康情報などのデータである可能性があります。エンティティを使用して、文字起こし中に関心のある単語に対して、強調表示したり、アクションを実行したりすることができます。
13. `transcription.results[i].alternatives[j].entities[k].category` はエンティティのカテゴリです。これは、リクエストで提供されたコンテンツ識別またはリダクションのタイプ (「PII」や「PHI」などの) と同じです。
14. `transcription.results[i].alternatives[j].entities[k].confidence` は、特定のコンテンツが本当にエンティティであることに対するエンジンの信頼度を示します。これは、単語そのものの正確さに対するエンジンの信頼度を示す項目レベルの信頼度とは異なることに注意してください。

15.transcription.results[i].alternatives[j].entities[k].content は、エンティティを構成する実際のテキストです。これは、複数の項目 (住所など) である場合もあります。

16.transcription.results[i].alternatives[j].entities[k].startTimeMs は、当該エンティティの発言が開始された時刻をキャプチャします。

17.transcription.results[i].alternatives[j].entities[k].endTimeMs は、当該エンティティの発言が終了した時刻をキャプチャします。

18.transcription.results[i].alternatives[j].entities[k].type は、文字起こしエンジンでのみサポートされ、エンティティのサブタイプを提供します。これらは、`ADDRESS`、`CREDIT_DEBIT_NUMBER` などの値です。

のイベントハンドラーの登録 TranscriptEvents

次の例では、の Amazon Chime SDK クライアントライブラリを使用します JavaScript。ただし、パターンはすべての Amazon Chime SDK で一貫しています。

RealtimeController および RealtimeControllerFacade の TranscriptionController には、TranscriptionEvents を処理するハンドラーを追加するための特定の関数が含まれています。

```
/**
 * Returns the [[TranscriptionController]] for this real-time controller.
 */
readonly transcriptionController?: TranscriptionController;
```

TranscriptionController には、TranscriptionEvent コールバックのサブスクライブとサブスクライブ解除を管理する 2 つの関数があります。

```
import TranscriptEvent from './TranscriptEvent';

export default interface TranscriptionController {
  /**
   * Subscribe a callback to handle received transcript event
   */
  subscribeToTranscriptEvent(callback: (transcriptEvent: TranscriptEvent) => void):
  void;

  /**
   * Unsubscribe a callback from receiving transcript event
```

```
*/
unsubscribeFromTranscriptEvent(callback: (transcriptEvent: TranscriptEvent) => void):
void;
}
```

オプションの **TranscriptionController** を使用する

TranscriptionController インターフェイスのデフォルト実装 (DefaultTranscriptionController という名前) が用意されています。DefaultRealtimeController および DefaultAudioVideoFacade のデフォルト実装では、DefaultTranscriptionController オブジェクトが返されます。

```
/**
get transcriptionController(): TranscriptionController {
    return this.realtimeController.transcriptionController;
}
```

DefaultRealtimeController では、そのコンストラクターにオプションの TranscriptionController オブジェクトを指定することもできます。これにより、DefaultTranscriptionController の動作をオーバーライドできます。デベロッパーアプリケーションは、AudioVideoFacade オブジェクトの TranscriptionController オブジェクトを通じて、1 つ以上のコールバックをサブスクライブまたはサブスクライブ解除します。

```
// Subscribe
this.audioVideo.transcriptionController?.subscribeToTranscriptEvent(this.transcriptEventHandler)

// Unsubscribe
this.audioVideo.transcriptionController?.unsubscribeFromTranscriptEvent(this.transcriptEventHandler)
```

配信例

以下の例は、受け取った TranscriptEvent を処理する方法を示しています。

Note

正確な出力は、各個人の話す速度や休止のタイミングなど、いくつかの要因によって異なります。

例 1: StartMeetingTranscription

この例は、典型的な StartMeetingTranscription オペレーションを示しています。

```
meeting.StartMeetingTranscription(  
  { EngineTranscribeSettings: { Languagecode: 'en-US' } } );
```

このオペレーションでは TranscriptEvent が生成されます。

```
{  
  status: {  
    type: 'started',  
    eventTimeMs: 1620118800000,  
    transcriptionConfig: {  
      LanguageCode: 'en-US'  
    }  
  }  
}
```

例 2: 部分的な文字起こしの結果

この例では、参加者は「The quick brown fox jumps over the lazy dog」と発言します。この例では、isPartial の値は true です。メッセージを詳しく調べると、システムが「fox」という単語を「facts」として処理していることがわかります。システムは同じ resultId を使用して文字起こしを更新します。

```
{  
  transcript: {  
    results: [{  
      resultId: "1",  
      startTimeMs: 1620118800000,  
      alternatives: [{  
        items: [{  
          type: 'pronunciation',  
          startTimeMs: 1620118800000,  
          attendee: { attendeeId: "1",  
            content: "the",  
          },  
          endTimeMs: 1620118800200,  
          externalUserId: "A",  
          vocabularyFilterMatch: false  
        },  
        {  
          type: 'pronunciation',  
          startTimeMs: 1620118800200,  
          attendee: { attendeeId: "1",  
            externalUserId: "A" },  
          endTimeMs: 1620118800400,  
        }  
      ]  
    }  
  }  
}
```

```
        content:"quick",                                vocabularyFilterMatch: false
      },
      {
        type:'pronunciation',
        startTimeMs: 1620118800400,                     endTimeMs: 1620118800750,
        attendee: { attendeeId: "1",                    externalUserId: "A" },
        content:"brown",                                vocabularyFilterMatch: false
      },
      {
        type:'pronunciation',
        startTimeMs: 1620118800750,                     endTimeMs: 1620118801000,
        attendee:{ attendeeId: "1",                      externalUserId: "A" },
        content:"facts",                                vocabularyFilterMatch: false
      },
      {
        type:'punctuation',
        startTimeMs: 1620118801000,                     endTimeMs: 1620118801500,
        attendee:{ attendeeId: "1",                      externalUserId: "A" },
        content:  ", ",                                 vocabularyFilterMatch: false
      }
    ]
  ]
}
}
```

例 3: 最終的な文字起こしの結果

部分的な文字起こしが発生した場合、システムはフレーズを再度処理します。この例の値では、`isPartial` の値は `false` で、メッセージには「facts」の代わりに「fox」が含まれています。システムは同じ ID を使用してメッセージを再発行します。

```
{
  transcript: {
    results: [{
      resultId:"1",                                    isPartial: false,
      startTimeMs: 1620118800000,                      endTimeMs: 1620118801000,
      alternatives: [{
        items:[{
          type:      'pronunciation',
          startTimeMs: 1620118800000,                  endTimeMs: 1620118800200,
          attendee: { attendeeId: "1",                  externalUserId: "A"},
          content:  "the",                               vocabularyFilterMatch: false
        },

```

```

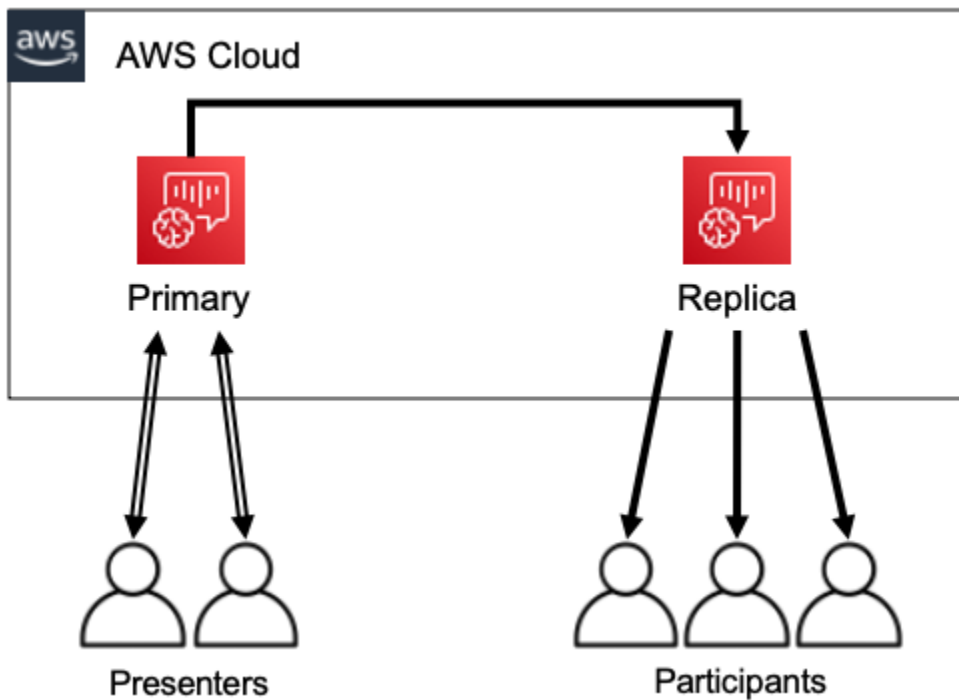
    {
      type: 'pronunciation',
      startTimeMs: 1620118800200,
      attendee: { attendeeId: "1",
      content:"quick",
      endTimeMs: 1620118800400,
      externalUserId: "A" },
      vocabularyFilterMatch: false
    },
    {
      type:'pronunciation',
      startTimeMs: 1620118800400,
      attendee: { attendeeId: "1",
      content:"brown",
      endTimeMs: 1620118800750,
      externalUserId: "A" },
      vocabularyFilterMatch: false
    },
    {
      type:'pronunciation',
      startTimeMs: 1620118800750,
      attendee: { attendeeId: "1",
      content:"fox",
      endTimeMs: 1620118801000,
      externalUserId: "A" },
      vocabularyFilterMatch: false
    },
    {
      type:'punctuation',
      startTimeMs: 1620118801000,
      attendee: { attendeeId: "1",
      content:  ", ",
      endTimeMs: 1620118801500,
      externalUserId: "A" },
      vocabularyFilterMatch: false
    }
  ]
}

```

メディアレプリケーションの使用

メディアレプリケーションを使用すると、プライマリ WebRTC セッションを複数のレプリカセッションにリンクして、より多くの視聴者に届けることができます。各 WebRTC メディアセッションがサポートする接続数は 250 で、プライマリセッションを複数のレプリカセッションにレプリケートできます。レプリカセッションに接続している参加者は、プライマリセッションに接続しているプレゼンターの音声と動画のみを受信します。レプリケートされたセッションに接続している参加者に関する情報は取得されないため、メディアレプリケーションはウェビナーやプライベートが求められるその他のユースケースに最適です。

次の図は、プレゼンターが音声と動画を共有するプライマリセッションと、参加者がメディアを利用するレプリカセッションの間のメディアレプリケーションを示しています。



Note

サービスクォータの「Chime SDK 会議 - プライマリ会議あたりのレプリカ会議」のデフォルト値は 4 で、リクエストに応じてこの制限を増やすことができます。クォータの詳細については、「AWS 全般リファレンス」の「[AWS サービスクォータ](#)」を参照してください。

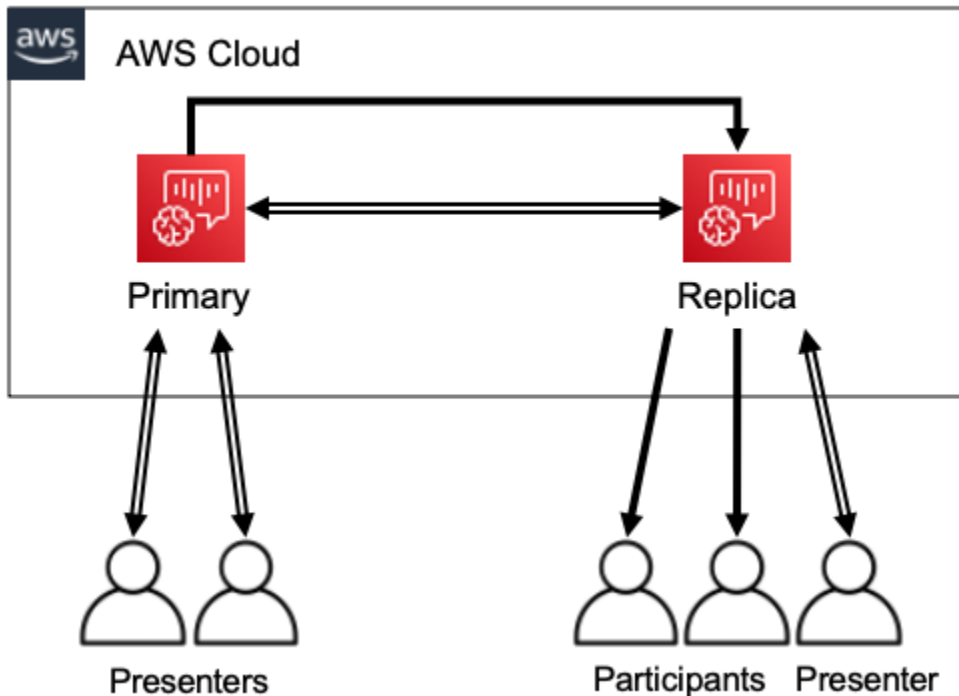
トピック

- [インタラクティブな参加者](#)
- [グローバル参加者](#)
- [セッションライフサイクル](#)

インタラクティブな参加者

レプリカセッションに接続している参加者に、プライマリセッションに参加するためのアクセス権を付与することができます。全員が WebRTC 接続を使用しているため、プレゼンターと参加者はトランスコーディングの遅延を経験しません。参加者がプライマリセッションとレプリケートされたセッションを切り替える際には WebRTC 接続が再利用されるため、切り替えは非常に高速です。これにより、参加者はコンテンツを見逃すことなくライブ会話に参加できます。

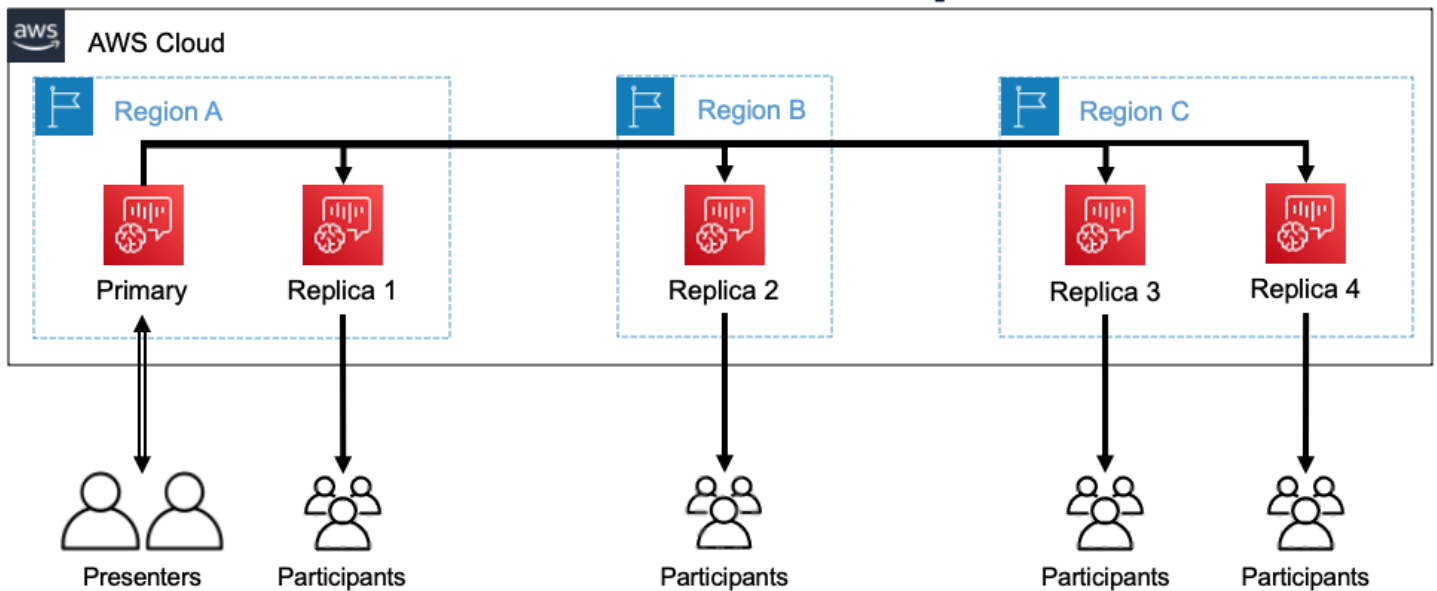
次の図は、レプリカセッションの参加者が WebRTC 接続を使用してプライマリセッションに切り替えている様子を示しています。



グローバル参加者

WebRTC メディアセッションごとに AWS リージョンを選択できます。これにより、プライマリセッションのリージョンよりも参加者に近いリージョンにレプリカセッションを作成できます。これを行うと、メディアは AWS ネットワーク経由でプライマリセッションからレプリカセッションに送られた後、インターネット経由でレプリカセッションから参加者に送られます。世界中の視聴者にプレゼンテーションを行う場合、参加者の近くにレプリカセッションを用意することで、メディアがインターネットではなく AWS ネットワーク上で世界中を移動できるようになり、会議体験が向上します。

次の図は、プライマリセッションと、別のリージョンにレプリケートされたセッションを示しています。



セッションライフサイクル

セッションの作成

[CreateMeeting](#) または [CreateMeetingWithAttendees](#) APIs を使用して WebRTC メディアセッションを作成します。レプリカセッションを明示的に作成しない限り、これらの API はデフォルトでプライマリセッションを作成します。

レプリカセッションを作成するには、[CreateMeeting](#) または [CreateMeetingWithAttendees](#) API コールで `PrimaryMeetingId` としてプライマリセッションの `MeetingId` を指定します。

Note

`PrimaryMeetingId` としてレプリカセッションの `MeetingId` を指定すると、API コールは失敗します。

参加者の作成

WebRTC メディアセッションへの参加に必要な参加者の認証情報を作成するには、[CreateMeetingWithAttendeesBatchCreateAttendee](#)、または [CreateAttendee](#) APIs を使用できます。

Note

多数の参加者を対象としたセッションを作成する場合は、`CreateMeetingWithAttendees` または `BatchCreateAttendee` を使用して、必要な API コールの回数を最小限に抑えます。

参加者の削除

[DeleteAttendee](#) API を使用して、WebRTC メディアセッションの参加者の認証情報を取り消します。参加者がセッションに接続している場合、接続は切断され、再び参加することはできません。

[DeleteMeeting](#) API を使用して WebRTC メディアセッションを削除すると、API はすべての参加者を自動的に削除するため、`DeleteAttendee` を呼び出す必要はありません。

セッションの切り替え

参加者がレプリカセッションからプライマリセッションに切り替えられるようにするには、プライマリ会議で、対象の参加者用の認証情報を作成する必要があります。このリストの前半の「参加者の作成」を参照してください。プライマリセッションに切り替えるには、Amazon Chime SDK クライアントライブラリの `promoteToPrimaryMeeting` メソッドでこれらの認証情報を使用します。

参加者をレプリカセッションに戻すには、Amazon Chime SDK クライアントライブラリの `demoteFromPrimaryMeeting` メソッドを使用するか、[DeleteAttendee](#) API を使用してプライマリセッション認証情報を無効にします。

Note

プライマリセッションに直接接続するプレゼンターは、レプリカセッションに切り替えることはできません。

セッションの切り替えの詳細については、クライアントライブラリのドキュメントを参照してください。

- [での Android 用 Amazon Chime SDK GitHub](#)。
- [Amazon Chime SDK for iOS on GitHub](#)。
- [上の 用の Amazon Chime SDK クライアントライブラリ JavaScript GitHub](#)。

セッションの削除

[DeleteMeeting](#) API を使用して WebRTC メディアセッションを削除します。

プライマリセッションを削除すると、DeleteMeeting API によって、アタッチされているすべてのレプリカセッションが自動的に削除されます。そのため、すべてのセッションを削除するには、プライマリを削除するだけです。

接続している参加者が 5 分間連続して存在しない場合、サービスによってプライマリセッションが自動的に削除されます。このサービスは、プライマリセッションを削除する場合にのみレプリカセッションを自動的に削除します。つまり、プライマリセッションの作成時にレプリカセッションを作成することができ、プライマリセッションの存続期間中はそのレプリカを使用できます。

Amazon Chime SDK ミーティングのトラブルシューティングとデバッグ

以下のトピックは、Amazon Chime SDK の使用時に発生する問題の診断やトラブルシューティングに役立ちます。

トピック

- [システム要件の理解](#)
- [ログ記録とモニタリングのセットアップ](#)
- [セルフトラブルシューティング](#)
- [一般的な問題](#)

システム要件の理解

トラブルシューティングの一環として、サポートされているブラウザ用にコーディングしていることを確認してください。サポートされているブラウザ、バージョン、オペレーティングシステムの最新リストについては、「[Amazon Chime SDK のシステム要件](#)」を参照してください。[Github の開発者ガイドと FAQ](#) では、ブラウザおよびその他の互換性の問題に関する情報が提供されています。また、[の既知のブラウザの問題](#) GitHubと回避策を理解してください。

Amazon Chime SDK ミーティングを開始したばかりの場合、[Amazon Chime SDK の Builder Journey](#) には、Amazon Chime SDK で構築するための step-by-step ガイドと、トラブルシューティングに必要なツールが用意されています。

ログ記録とモニタリングのセットアップ

ログ記録は、サーバー側の会議イベントやクライアント側のブラウザコンソールログなどの情報を収集するのに役立ちます。

Amazon Chime SDK は、Amazon EventBridge および Amazon CloudWatch Events ログに送信できるサーバー側の会議イベントを提供します。CloudWatch メトリクスとインサイトを作成し、ダッシュボードでモニタリングに使用できます。[「Amazon Chime SDK イベントのサーバー側のログ記録とモニタリング」](#) ブログ記事では、CloudWatch メトリクス、インサイト、ダッシュボードを有効にする方法について説明します。

Amazon Chime SDK は、音声と動画の品質、ネットワーク帯域幅、接続の問題に関するクライアント側イベントを提供します。[「Amazon Chime SDK ミーティングイベントによるモニタリングとトラブルシューティング」](#) ブログ記事では、結合失敗、オーディオ品質の問題、マイクとカメラのセットアップ失敗に対して CloudWatch メトリクス、インサイト、ダッシュボードを有効にする方法について説明します。会議イベントの詳細については、Github の [「Meeting Events」](#) を参照してください。

トラブルシューティングメトリクスのオプション

トラブルシューティングイベントの収集では、以下のオプションがあります。

- イベントごとにメトリクスを送信する
- N 秒ごとにイベントをバッチ処理する
- 会議の終了時にメトリクスを送信する
- ブラウザコンソールログのログ記録レベル

推奨メトリクス

少なくとも、以下のメトリクス収集してログ記録する必要があります。

- SDK プラットフォームとバージョン
- ブラウザとバージョン
- オペレーティングシステム
- 論理コア
- 開始された会議

- 終了した会議
- 参加した参加者
- 退出した参加者
- 切断された参加者

さらに、直面している問題に応じて、以下のメトリクスから接続、帯域幅、品質の問題に関する情報が得ることができます。これらのメトリックが発生するたびに記録することも、カウントだけすることもできます。カウントすることで、根本的な問題の概要を把握できます。

- connectionDidSuggestStopVideo
- connectionDidBecome良い
- connectionDidBecome不良
- 参加者の参加時間 > t 秒
- MeetingStartFailed
- MeetingFailed

クライアント側のログ記録を有効にする

LogLevel.INFO を ConsoleLogger オブジェクトに渡すことで、INFO レベルのブラウザログを有効にできます。

```
const logger = new ConsoleLogger('MyLogger', LogLevel.INFO);const meetingSession = new DefaultMeetingSession(configuration, logger, deviceController);
```

Amazon Chime SDK for の POSTLogger コンポーネントを使用して、Amazon CloudWatch Logs などのブラウザログをバックエンドで JavaScript キャプチャすることもできます。

POSTLogger は、[POSTLogger コンストラクタ 内の特定の URL](#) にブラウザログをアップロードするように HTTP POST リクエストします。例えば、の [Amazon Chime SDK サーバーレスデモ GitHub](#) では、を使用してブラウザログを Amazon CloudWatch Logs に送信 POSTLogger し、今後の調査を行います。

サーバー側のログ記録を有効にする

の Amazon Chime SDK JavaScript は、MeetingStartFailed や などの主要な会議イベントを使用して eventDidReceive オブザーバーメソッドも呼び出します MeetingFailed。多くの場合、会

議イベントには失敗の具体的な理由が含まれます。例えば、多数のお客様で障害が発生したとします。ウェブアプリケーションはこれらの会議イベントを収集し、根本原因をトラブルシューティングするためにその情報を共有できます。会議イベントの詳細については、[「」の会議イベントガイド](#) [GitHub](#)、および [「Amazon Chime SDK 会議イベントのモニタリングとトラブルシューティング」](#) ブログ記事を参照してください。

セルフトラブルシューティング

このトピックのセクションでは、Amazon Chime SDK ミーティングをセルフトラブルシューティングするいくつかの方法について説明します。

トピック

- [FAQ と既知の問題を確認する](#)
- [ネットワークアクセスを検証する](#)

FAQ と既知の問題を確認する

トラブルシューティングとデバッグのアドバイス [GitHub](#) については、FAQsと既知の問題のリストを確認してください。

- [Amazon Chime SDK – JavaScript 会議](#)
- [Amazon Chime SDK – JavaScript メディア](#)
- [Amazon Chime SDK – JavaScript ネットワーク](#)
- [Amazon Chime SDK – 音声と動画](#)

ネットワークアクセスを検証する

多くの場合、企業には、特定のポートへのアクセスや企業ネットワーク外の IP アドレス範囲への接続を制限するネットワークファイアウォールがあります。以下のセクションでは、ネットワークアクセスを検証できるいくつかの方法について説明します。

トピック

- [AWS SDK と Amazon Chime SDK のサブネットとポートの検証](#)
- [デモアプリを使用して問題を再現する](#)
- [会議準備状況チェッカーを使用する](#)

AWS SDK と Amazon Chime SDK のサブネットとポートの検証

Amazon Chime SDK を使用するアプリケーションは、2 つの階層 (サーバーとクライアント) を使用します。サーバー階層は AWS SDK を使用し、サーバー側の会議ハンドラーがあります。クライアント層はクライアント SDK を使用します。

AWS SDK は、などのサーバー APIs 呼び出すために使用されます [CreateMeeting](#)。このような APIs、us-east-1、us-west-2、ap-southeast-1、eu-central-1、us-gov-east-1 および us-gov-west-1 リージョン AWS のグローバルサービスエンドポイントに接続します。「AWS 全般リファレンス」の「[AWS IP アドレスの範囲](#)」ページに、各リージョンの IP アドレス範囲がリストされています。サービスエンドポイントとクォータについては、「[Amazon Chime SDK エンドポイントとクォータ](#)」を参照してください。

Amazon Chime SDKs などのクライアント SDK は JavaScript、*.chime.aws ドメイン内のサービスエンドポイントに接続します。

以下の検証を使用して、ネットワークアクセス許可があることを確認します。

- [Amazon Chime SDK ミーティング準備チェッカー](#) を実行して GitHub、ネットワークとポートにアクセスできることを確認します。
- 自分のネットワークまたはエンドユーザーのネットワークから *.chime.aws ドメインを解決できることを確認します。
- ファイアウォールで、制御コマンドの場合は TCP ポート 443 経由で、メディアの場合は UDP ポート 3478 経由で AWS IP 範囲への接続が許可されていることを確認します。

デモアプリを使用して問題を再現する

ベストプラクティスとして、いずれかのデモアプリで問題を再現していただくことからデバッグプロセスを開始してください。これにより、サービスチームが問題の所在を見つけることができます。デモアプリで問題を再現できない場合は、アプリのコードを確認して、関連するユースケースがどのように実装されているかを確認できます。

Amazon Chime SDK	機能	デモアプリのリソース
JavaScript SDK	会議	デモ手順 、 ソースコード
React コンポーネント	会議	デモ手順 、 ソースコード

Amazon Chime SDK	機能	デモアプリのリソース
会議チャット	メッセージング	ブログ投稿 、 デモ手順 、 ソースコード
iOS/Android	会議	(ブログ投稿) Building a Meeting Application on Android using the Amazon Chime SDK (ブログ投稿) Building a Meeting Application on iOS using the Amazon Chime SDK
PSTN 音声	受信通話	ブログ投稿 、 ソースコード

会議準備状況チェッカーを使用する

で [Amazon Chime SDK ミーティング準備チェッカー](#) を使用します GitHub。このチェッカーは、音声デバイス、動画デバイス、ユーザー接続の検証に役立ちます。問題の根本原因を明らかにする合格/不合格のステータスを使用して、結果をエンドユーザーに提示できます。

一般的な問題

以下のセクションでは、会議の一般的な問題のトラブルシューティング方法について説明します。

トピック

- [接続の問題](#)
- [音声と動画の品質に関する問題](#)
- [SDK クォータと API スロットリングの検証](#)
- [サポートケースを開く](#)

接続の問題

接続の問題については、「[ネットワークアクセスを検証する](#)」を参照してください。

音声と動画の品質に関する問題

音声と動画の品質に関する問題には、いくつかの原因が考えられます。音声/動画の品質が最適ではない主な理由は、ネットワーク帯域幅とデバイスのパフォーマンスの2つです。さまざまな課題と、それらがオーディオ/ビデオの品質にどのように影響するかの詳細については、「品質、帯域幅、接続性 (https://aws.github.io/amazon-chime-sdk-js/modules/qualitybandwidth_connectivity.html)」を参照してください。この記事では、帯域幅の問題を検出するためにモニタリングできるさまざまなイベントやメトリクスと、実施可能な緩和策について説明しています。

対象の会議セッションの視聴者により近いメディアリージョンを選択できます。最適なメディアリージョンを選択する方法については、「会議リージョンの使用 (<https://docs.aws.amazon.com/chime-sdk/latest/dg/chime-sdk-meetings-regions.html>)」を参照してください。

Amazon Chime SDK は、会議の参加者が利用できる帯域幅に応じて、受信/アップロードされる動画の品質を調整します。さまざまなビデオレイアウトのビデオ品質を制御する方法については、「さまざまなビデオレイアウトのビデオ品質の管理 (<https://aws.github.io/amazon-chime-sdk-js/modules/videolayout.html>)」を参照してください。この記事では、動画のライフサイクル管理とアップリンク/ダウンリンクのポリシーについて説明しています。

動画の解像度に関する考慮事項

- 動画をアップロードする場合のデフォルトの解像度は 540p、フレームレートは 15fps、ビットレートは 1400 kbps です。帯域幅に応じて、解像度とフレームレートを下げることができます。
- 受信側の利用可能な帯域幅に基づいて、表示する動画タイルの数を決定します。すべての動画タイルとコンテンツ共有で 6 Mbps を超えないようにしてください。エンドユーザーに十分な帯域幅がない場合、そのユーザーには黒い動画タイルが表示されます。

動画のアップリンクとダウンリンクの帯域幅ポリシーを使用する

Amazon Chime SDK には、以下の帯域幅ポリシーが用意されています。

- N ScaleVideoUplinkBandwidthPolicy - デスクトップ、ウェブ、モバイルクライアントで使用されるパラメータとほぼ同じキャプチャおよびエンコーディングパラメータを実装します。
- AllHighestVideoBandwidthPolicy - 常に最高品質のビデオストリームにサブスクライブします。
- NoVideoDownlinkBandwidthPolicy - 帯域幅が特定のしきい値を下回ると、ビデオを無効にします。
- VideoPriorityBasedPolicy - 低帯域幅の場合、ビデオよりもオーディオを優先します。

- VideoAdaptiveProbePolicy

SDK クォータと API スロットリングの検証

「[Amazon Chime SDK エンドポイントとクォータ](#)」ページには、サービスクォータ、API レート、およびそれらを調整できるかどうかのリストされています。[AWS コンソールのサービスクォータ](#)ページを使用して、クォータの調整をリクエストします。

API レートの微調整

API レートを越えたアプリケーションは、HTTP ステータスコード 429 と `ThrottledClientException` メッセージを受け取ります。API レートは調整できますが、それを行う前に、そのレートを使い果たす可能性のあるバグがアプリケーションにないかを確認してください。例えば、ループ内で会議を作成していたり、会議を作成してクリーンアップを行わなかったりしている場合があります。

会議の作成方法によっては、コードを変更する必要がある場合があります。例えば、`CreateMeeting` と `CreateAttendee` を以下に置き換えることができます。

- [CreateMeetingWithAttendees](#) – 会議ごとに最大 10 人の参加者を作成します。
- [BatchCreateAttendee](#) – 会議ごとに最大 100 人の参加者を作成します。

作成した参加者をデータベースに保存し、招待者が会議に参加したときに参加者情報をプルして、事前に作成した参加者と関連付けることができます。

サポートケースを開く

さらに質問がある場合、またはビジネスに関するサポートが必要な場合は、[AWS カスタマーサポート](#)にお問い合わせください。サポートプランの詳細については、「[Compare support plans](#)」を参照してください。サポートケースを作成する場合は、必ず問題のあるアカウントで開いてください。コンソールブラウザログ、会議 ID と参加者 IDs、および関連するサポートケースや GitHub 問題を含めます。

Amazon Chime SDK メッセージングを使用する

「Amazon Chime SDK Developer Guide」のこのセクションは、Amazon Chime SDK サービスで実行されるメッセージングアプリケーションの作成に役立ちます。この SDK は、基本的なメッセージングアプリケーションを作成するのに必要な概念的かつ実用的な情報を提供します。

トピック

- [Amazon Chime SDK Identity 名前空間への移行](#)
- [Amazon Chime SDK Messaging 名前空間への移行](#)
- [メッセージングの前提条件](#)
- [メッセージングの概念](#)
- [メッセージングアーキテクチャ](#)
- [メッセージタイプ](#)
- [開始](#)
- [システムメッセージについて](#)
- [IAM ロールの例](#)
- [ロールごとの権限について](#)
- [メッセージングデータのストリーミング](#)
- [Elastic チャンnelを使用してライブイベントをホストする](#)
- [モバイルプッシュ通知を使用してメッセージを受信する](#)
- [サービスリンクロールの使用](#)
- [チャンネルフローを使用したメッセージの処理](#)
- [をインテリジェントチャンネルエージェント AppInstanceBots として使用する](#)
- [メッセージ保持の管理](#)
- [メッセージングのユーザーインターフェイスコンポーネント](#)
- [クライアントライブラリとの統合](#)
- [での Amazon Chime SDK メッセージングの使用 JavaScript](#)

Amazon Chime SDK Identity 名前空間への移行

[Amazon Chime SDK Identity](#) 名前空間は、AppInstances や などの Amazon Chime SDK ID リソースの作成と管理に使用される APIs 専用の場所です AppInstanceUsers。名前空間を使用して、Amazon

Chime SDK ID API エンドポイントが利用可能な任意の AWS リージョンのエンドポイントに対処します。Amazon Chime SDK を使い始めたばかりの場合は、この名前空間を使用してください。リージョンの詳細については、このガイドの「[利用できるリージョン](#)」を参照してください。

[Amazon Chime](#) 名前空間を使用する既存のアプリケーションでは、専用の名前空間への移行を計画する必要があります。

トピック

- [移行すべき理由](#)
- [移行する前に](#)
- [名前空間の相違点](#)

移行すべき理由

以下の理由から、[Amazon Chime SDK Identity](#) 名前空間に移行することをお勧めします。

API エンドポイントの選択

Amazon Chime SDK Identity 名前空間は、[API エンドポイントを利用できる任意のリージョン](#)の API エンドポイントを使用できる唯一の API 名前空間です。us-east-1 以外の API エンドポイントを使用する場合は、Amazon Chime SDK Identity 名前空間を使用する必要があります。現在のエンドポイントの詳細については、このガイドの「[API マッピング](#)」を参照してください。

メッセージング API の更新と新規追加

Amazon Chime SDK Identity 名前空間の ID API のみ追加または更新されます。

移行する前に

移行する前に、名前空間の相違点に注意してください。以下の表では、名前空間の一覧と説明を示しています。

	Amazon Chime SDK Identity 名前空間	Amazon Chime 名前空間
AWS SDK 名前空間	ChimeSDKIdentity	Chime
リージョン	複数	us-east-1 のみ

	Amazon Chime SDK Identity 名前空間	Amazon Chime 名前空間
サービスプリンシパル	https://identity.chime.amazonaws.com	https://chime.amazonaws.com
API	ID 用の API のみ	ID 用の API と Amazon Chime のその他の部分
ユーザー有効期限	利用可能	利用不可
ポット	利用可能	利用不可

名前空間の相違点

以下のセクションでは、Chime 名前空間と ChimeSDKIdentity 名前空間の相違点について説明します。

AWS SDK 名前空間

Amazon Chime SDK 名前空間では Chime という正式名を使用します。Amazon Chime SDK Identity 名前空間では ChimeSDKIdentity という正式名を使用します。名前の正確な形式はプラットフォームによって異なります。

例えば、Node.js の AWS SDK を使用して ID を作成する場合は、コード行を使用して名前空間に対処します。

```
const chimeIdentity = AWS.Chime();
```

ChimeSDKIdentity 名前空間に移行するには、このコード行を新しい名前空間とエンドポイントリージョンで更新します。

```
const chimeIdentity = AWS.ChimeSDKIdentity({ region: "eu-central-1" });
```

リージョン

[Amazon Chime](#) 名前空間は、us-east-1 リージョンの API エンドポイントのみをアドレス指定します。[Amazon Chime SDK Identity](#) 名前空間は、利用可能な任意のリージョンで Amazon Chime SDK

Identity API エンドポイントをアドレス指定できます。現在のエンドポイントリージョンのリストについては、本ガイドの「[利用できるリージョン](#)」を参照してください。

エンドポイント

[Amazon Chime SDK Identity](#) 名前空間は、[Amazon Chime](#) 名前空間とは異なる API エンドポイントを使用します。

ID リソースを更新できるのは、ID リソースの作成に使用したエンドポイントだけです。つまり、のエンドポイントを介して ApplInstance 作成された eu-central-1 は、を介してのみ変更できます eu-central-1。また、の ChimeSDKIdentity 名前空間を使用して Chime 名前空間を介して ApplInstance 作成された をアドレス指定したり us-east-1、 ApplInstance および ApplInstanceUser メンバーが作成されたリージョン以外のリージョンにチャンネルを作成したりすることもできません。現在のエンドポイントの詳細については、このガイドの「[API マッピング](#)」を参照してください。

サービスプリンシパル

[Amazon Chime SDK Identity](#) 名前空間は、新しいサービスプリンシパル

Identity.chime.amazonaws.com を使用します。サービスへのアクセスを許可する SQS、SNS またはその他の IAM アクセスポリシーがある場合は、それらのポリシーを更新して新しいサービスプリンシパルへのアクセスを許可する必要があります。

API

[Amazon Chime SDK Identity](#) 名前空間には、メッセージングリソースの作成と管理、およびメッセージの送受信のための API のみが含まれています。[Amazon Chime](#) 名前空間には、メッセージング用および Amazon Chime サービスのその他の部分用の API が含まれています。

ユーザー有効期限

作成時の有効期限設定 ApplInstanceUsers により、一時ユーザーを作成できます。例えば、大規模なブロードキャストの間だけ存在するチャットユーザーを作成できます。ID 名前空間のみが の有効期限設定をサポートします ApplInstanceUsers。

ボット

[ApplInstanceBot](#) API を使用して、Amazon Lex V2 を搭載したチャットボットをアプリケーションに追加します。V2 は、ApplInstanceBots アイデンティティ名前空間でのみ使用できます。ボットの詳細については、このガイドの「[をインテリジェントチャンネルエージェント ApplInstanceBots として使用する](#)」を参照してください。

その他の API

Identity 名前空間には、Chime 名前空間にはないその他の API が増え続けています。Amazon Chime SDK の使用を開始する場合は、Identity 名前空間を使用してすべての最新機能にアクセスできるようにしてください。現在の API の詳細については、「[Amazon Chime SDK API Reference](#)」の「[Amazon Chime SDK Identity](#)」を参照してください。

Amazon Chime SDK Messaging 名前空間への移行

[Amazon Chime SDK Messaging](#) 名前空間は、Amazon Chime SDK メッセージングリソースを作成および管理する API のための専用の場所です。名前空間を使用して、Amazon Chime SDK メッセージング API エンドポイントが利用可能な任意の AWS リージョンのエンドポイントに対処します。Amazon Chime SDK を使い始めたばかりの場合は、この名前空間を使用してください。リージョンの詳細については、このガイドの「[利用できるリージョン](#)」を参照してください。

[Amazon Chime](#) 名前空間を使用する既存のアプリケーションでは、専用の名前空間への移行を計画する必要があります。

トピック

- [移行すべき理由](#)
- [移行する前に](#)
- [名前空間の相違点](#)

移行すべき理由

以下の理由から、[Amazon Chime SDK Messaging](#) 名前空間に移行することをお勧めします。

API エンドポイントの選択

Amazon Chime SDK Messaging 名前空間は、[利用できる任意のリージョンで API エンドポイントを使用できる唯一の API 名前空間です](#)。米国東部 (バージニア北部) 以外の API エンドポイントを使用する場合は、Amazon Chime SDK Messaging 名前空間を使用する必要があります。

Amazon Chime SDK メッセージングが AWS リージョンを使用する方法の詳細については、このガイドの「[利用可能なリージョン](#)」を参照してください。

メッセージング API の更新と新規追加

Amazon Chime SDK Messaging 名前空間のメッセージング API のみを追加または更新します。

移行する前に

移行する前に、名前空間の相違点に注意してください。以下の表では、名前空間の一覧と説明を示しています。

	Amazon Chime SDK Messaging 名前空間	Amazon Chime 名前空間
AWS SDK 名前空間	ChimeSDKMessaging	Chime
リージョン	複数	米国東部 (バージニア北部) のみ
API	メッセージング用の API のみ	メッセージング用の API と Amazon Chime のその他の部分
フロー	利用可能	利用不可
Elastic チャネル	利用可能	利用不可

名前空間の相違点

以下のセクションでは、Amazon Chime 名前空間と Amazon Chime SDK Messaging 名前空間の相違点について説明します。

AWS SDK 名前空間

Amazon Chime SDK 名前空間では Chime という正式名を使用します。Amazon Chime SDK Messaging 名前空間では ChimeSDKMessaging という正式名を使用します。名前の正確な形式はプラットフォームによって異なります。

例えば、Node.js の AWS SDK を使用してメッセージングを作成する場合は、コード行を使用して名前空間に対処します。

```
const chimeMessaging = AWS.Chime();
```

Amazon Chime Messaging SDK に移行するには、このコード行を新しい名前空間とエンドポイントリージョンで更新します。

```
const chimeMessaging = AWS.ChimeSDKMessaging({ region: "Europe (Frankfurt)" });
```

リージョン

[Amazon Chime](#) 名前空間は、US East (N. Virginia) リージョンの API エンドポイントのみをアドレス指定します。[Amazon Chime SDK Messaging](#) 名前空間は、利用可能な任意のリージョンで Amazon Chime SDK メッセージング API エンドポイントをアドレス指定できます。メッセージングリージョンの最新リストについては、本ガイドの「[利用できるリージョン](#)」を参照してください。

エンドポイント

[Amazon Chime SDK Messaging](#) 名前空間は、[Amazon Chime](#) 名前空間とは異なる API エンドポイントを使用します。

メッセージングリソースを変更するために使用できるのは、メッセージングリソースの作成に使用したエンドポイントだけです。つまり、Europe (Frankfurt) のエンドポイントを介して作成されたメッセージングリソースは、Europe (Frankfurt) 経由でしか変更できないということです。つまり、欧州 (フランクフルト) のエンドポイントを介して作成されたチャンネルは、欧州 (フランクフルト) 経由でのみ変更できます。また、米国東部 (バージニア北部) の Chime 名前空間を使用して ChimeSDKMessaging 名前空間を介して作成されたチャンネルに対応できないことも意味します。現在のエンドポイントの詳細については、このガイドの「[API マッピング](#)」を参照してください。

サービスプリンシパル

[Amazon Chime SDK Messaging](#) 名前空間は、新しいサービスプリンシパル `messaging.chime.amazonaws.com` を使用します。サービスへのアクセスを許可する SQS、SNS またはその他の IAM アクセスポリシーがある場合は、それらのポリシーを更新して新しいサービスプリンシパルへのアクセスを許可する必要があります。

API

[Amazon Chime SDK Messaging](#) 名前空間には、メッセージングリソースの作成と管理、およびメッセージの送受信のための API のみが含まれています。[Amazon Chime](#) 名前空間には、メッセージング用および Amazon Chime サービスのその他の部分用の API が含まれています。

チャンネルフロー

チャンネルフローを使用すると、送信中のメッセージがメッセージングチャンネルのメンバーに配信される前に、そのメッセージに対してビジネスロジックを実行できます。例えば、行政 ID 番号、電話番

号、または冒法的な表現などの機密データをメッセージが配信される前に削除するフローを作成できます。これは、企業のコミュニケーションポリシーやその他のコミュニケーションガイドラインを実施するのに役立ちます。

投票アンケートへの回答を集約した後で結果を参加者に返送する、メッセージを SMS で送信するなどの機能を実行するためにチャネルフローを使用することもできます。

チャネルフローは ChimeSDKMessaging 名前空間でのみ使用できます。詳細については、このガイドの「[チャネルフローを使用したメッセージの処理](#)」を参照してください。

Elastic チャネル

Elastic チャネルは、所定の数のサブチャネル間で最大 100 万人のチャットユーザーを自動的にバランスよく分散する、大規模なチャット体験をサポートします。Elastic チャネルは ChimeSDKMessaging エンドポイントでのみ使用できます。Elastic チャネルの詳細については、このガイドの「[Elastic チャネルを使用してライブイベントをホストする](#)」を参照してください。

その他の API

Messaging 名前空間には、Chime 名前空間にはない API が増え続けています。Amazon Chime SDK の使用を開始する場合は、メッセージング名前空間を使用してすべての最新機能にアクセスできるようにしてください。現在の API の詳細については、「Amazon Chime SDK API Reference」の「[Amazon Chime SDK Messaging](#)」を参照してください。

メッセージングの前提条件

Amazon Chime SDK メッセージングを使用するには、以下が必要です。

- プログラムする機能。
- AWS アカウント。
- Amazon Chime SDK メッセージングを使用してアプリケーションの IAM ロールを設定するアクセス許可。

ほとんどの場合、以下も必要です。

- クライアントアプリケーション — メッセージング UI を表示し、Amazon Chime SDK を使用してウェブソケットに接続し、状態を管理します。
- サーバーアプリケーション — アイデンティティおよびユーザーを管理します。

メッセージングの概念

Amazon Chime SDK メッセージングを効果的に使用するには、以下の用語および概念を理解する必要があります。

AppInstance

Amazon Chime SDK メッセージングを使用するには、まず `AppInstance` を作成する必要があります。`AppInstance` には、`AppInstanceUsers` および `Channel` が含まれています。通常、アプリケーション `AppInstance` 用に 1 つの `AppInstance` を作成します。AWS アカウントには複数の `AppInstance` を含めることができます。`AppInstance` は、メッセージ保持やストリーミング設定などのアプリレベルの設定は、`AppInstance` レベルで行います。`AppInstance` は、`arn:aws:chime:region:aws_account_id:app-instance/app_instance_id` という形式の一意的な ARN によって識別されます。

AppInstanceUser

`AppInstanceUsers` は、メッセージの送信、`Channel` の作成、`Channel` の結合などのエンティティです。通常、アプリケーションのユーザー `AppInstanceUser` への `one-to-one` マッピングを作成します。また、`AppInstanceUser` を作成してバックエンドサービス `AppInstanceUser` に接続することもできます。これにより、ユーザーはバックエンドサービスからのメッセージを識別できます。`AppInstanceUsers` は、`arn:aws:chime:region:aws_account_id:app-instance/app_instance_id/user/app_instance_user_id` などの ARN によって識別されます。`app_instance_user_id` を制御し、ベストプラクティスとして、アプリケーションが既に持っている ID を再利用します。

Channel

`Channel` に `AppInstanceUser` を追加すると、そのユーザーはメンバーになり、メッセージを送受信できるようになります。`Channel` は、すべてのユーザーが自分自身をメンバーとして追加できるパブリックにすることも、`Channel` のモデレーターだけがメンバーを追加できるプライベートにすることもできます。`Channel` メンバーを非表示にすることもできます。非表示になっているメンバーは会話を観察できますが、メッセージを送信することはできず、`Channel` メンバーシップには追加されません。

SubChannel

Elastic `Channel` のメンバーは、`SubChannels` という論理コンテナに分割されます。Elastic `Channel` `AppInstanceUser` に `SubChannel` を追加すると、ユーザーは `SubChannel` のメンバーになり、その特定の `SubChannel` のメッセージを送受信できます。`SubChannel` メンバーシップとメッセージは `SubChannel`、ある `SubChannel` のメンバーによって送信されたメッセージが別の `SubChannel` のメンバーによって受信

SubChannel されないことを意味します SubChannel。メンバーは、チャンネルの伸縮自在な性質をサポートし、エンゲージメントを促進する SubChannels ために、異なる に転送されます。

UserMessage

チャンネルに属している AppInstanceUser はユーザーメッセージを送受信できます。AppInstanceUser はメッセージを STANDARD または CONTROL メッセージを送信できます。STANDARD メッセージには 4 KB のデータおよび 1 KB のメタデータを含めることができます。CONTROL メッセージには 30 バイトのデータしか含めることができません。メッセージは PERSISTENT または NON_PERSISTENT にすることができます。チャンネル履歴から PERSISTENT メッセージを取得できます。NON_PERSISTENT メッセージは、現在 Amazon Chime SDK メッセージングに接続しているチャンネルメンバーのみに表示されます。

システムメッセージ

Amazon Chime SDK は、メンバーがチャンネルに参加または退出するなどのイベントに応じてシステムメッセージを生成します。

メッセージングアーキテクチャ

Amazon Chime SDK メッセージングは、サーバー側およびクライアント側の SDK として使用できます。サーバー側 API は AppInstance および AppInstanceUser を作成します。さまざまなフックおよび設定を使用して、アプリケーション固有のビジネスロジックおよび検証を追加できます。これを行う方法については、「[メッセージングデータのストリーミング](#)」を参照してください。さらに、サーバー側のプロセスが AppInstanceUser に代わって API を呼び出したり、バックエンドプロセスを表す専用 AppInstanceUser を制御したりできます。

AppInstanceUser として表されるクライアント側アプリケーションは、Amazon Chime SDK メッセージング API を直接呼び出すことができます。クライアント側のアプリケーションは、オンライン時に WebSocket プロトコルを使用してメッセージング SDK に接続します。接続すると、参加しているすべてのチャンネルからリアルタイムのメッセージを受信します。接続を解除しても、AppInstanceUser は追加先のチャンネルに属し、SDK の HTTP ベースの API を使用してそれらのチャンネルのメッセージ履歴を読み込むことができます。

クライアント側のアプリケーションには、単一の AppInstanceUser として API コールを行うアクセス許可があります。IAM 認証情報の範囲を単一の AppInstanceUser に限定するには、クライアント側のアプリケーションが AWS Cognito アイデンティティプールまたは小規模なセルフホスト型バックエンド API を介してパラメータ化された IAM ロールを引き受けます。認証の詳細について

は、「[エンドユーザークライアントアプリケーションの認証](#)」を参照してください。対照的に、サーバー側のアプリケーションには通常、管理者権限を持つユーザーなど、単一のアプリケーションインスタンスユーザーに紐付けられたアクセス許可や、すべてのアプリインスタンスユーザーに代わって API コールを行うアクセス許可があります。

メッセージタイプ

メッセージはチャンネルを通じて送信します。STANDARD、CONTROL、または SYSTEM メッセージを送信できます。

- STANDARD メッセージのサイズは最大 4 KB で、メタデータを含めることができます。メタデータは任意で、アタッチメントへのリンクを含めるなど、さまざまな方法で使用できます。
- CONTROL メッセージは 30 バイトに制限されており、メタデータは含まれていません。
- STANDARD および CONTROL メッセージは永続的または非永続的にすることができます。永続的なメッセージはチャンネルの履歴に保存され、ListChannelMessages API コールを使用して表示します。非永続メッセージは、を介してAppInstanceUser接続されているすべてのに送信されず WebSocket。
- Amazon Chime SDK は、メンバーがチャンネルに参加または退会するなどのイベントが発生すると、自動 SYSTEM メッセージを送信します。

開始

このセクションのトピックでは、Amazon Chime SDK メッセージングアプリケーションの構築を開始する方法について説明します。

トピック

- [AppInstance の作成](#)
- [バックエンドサービスから SDK 呼び出しを行う](#)
- [エンドユーザークライアントアプリケーションの認証](#)
- [チャンネルの作成](#)
- [メッセージの送信](#)
- [の使用 ExpirationSettings](#)
- [WebSockets を使用したメッセージの受信](#)
- [アタッチメントの設定](#)

AppInstance の作成

Amazon Chime SDK メッセージングを使用するには、まず AppInstance AWS アカウントに Amazon Chime SDK を作成する必要があります。

トピック

- [の構築 AppInstance](#)
- [の作成 AppInstanceUser](#)

の構築 AppInstance

メッセージング用の **AppInstance** を作成するには

1. CLI で、以下を実行します: `aws chime-sdk-identity create-app-instance --name NameOfAppInstance`.
2. 作成レスポンスで、以下を書き留めます: AppInstanceArn および `arn:aws:chime:region:aws_account_id:app-instance/app_instance_id`.

の作成 AppInstanceUser

AppInstance を作成したら、その AppInstance 内に AppInstanceUser を作成します。これは通常、ユーザーがアプリケーションに初めて登録またはログインするときに行います。バックエンドサービスに代わって動作する AppInstanceUser を作成することもできます。

以下の例では、バックエンド AppInstanceUser を作成する方法を示します。

```
aws chime-sdk-identity create-app-instance-user \  
  --app-instance-arn "app_instance_arn" \  
  --app-instance-user-id "back-end-worker" \  
  --name "back-end-worker"
```

作成レスポンスの AppInstanceUserArn を書き留めます。以下の書式が使用されます:

`arn:aws:chime:region:aws_account_id:app-instance/app_instance_id/user/app_instance_user_id`。この例では、`app_instance_user_id` は「back-end-worker」です。

Note

ベストプラクティスとして、クライアントアプリケーションの AppInstanceUser を作成するときは、AppInstanceUserId をそのユーザーの既存の一意の ID (アイデンティティプロバイダーの sub など) と一致させます。名前は、メッセージ送信者などの一部の API エンティティに添付されるオプションのプレースホルダーです。これにより、メッセージの送信者としても添付されている AppInstanceUser ARN から検索する必要がなく、ユーザーの表示名を 1 か所で制御できます。

バックエンドサービスから SDK 呼び出しを行う

バックエンドサービスを表すユーザーを作成したら、チャンネルを作成し、そのチャンネルにメッセージを送信し、そのチャンネルからのメッセージを読み取ります。

以下の CLI コマンドを実行して、パブリックチャンネルを作成します。

```
aws chime-sdk-messaging create-channel \  
  --chime-bearer "app_instance_user_arn" \  
  --app-instance-arn "app_instance_arn" \  
  --name "firstChannel"
```

このコマンドは、以下の形式の ARN を生成します:

```
arn:aws:chime:region:aws_account_id:app-instance/app_instance_id/  
channel/channel_id.
```

トピック

- [バックエンドサービスの IAM 認証の仕組み](#)
- [暗黙的な API 認証について](#)
- [チャンネルメッセージの送信と一覧表示](#)

バックエンドサービスの IAM 認証の仕組み

前のセクションの CLI コマンドの chime-bearer パラメータを書き留めます。チャンネルやメッセージなどのリソースを作成したり操作したりするユーザーを識別します。ほぼすべての Amazon Chime SDK メッセージング API は、パラメータとして chime-bearer を受け取ります。ただし、CreateAppInstance などの開発者だけが呼び出すための API は例外です。

Amazon Chime SDK メッセージング API の IAM アクセス許可には、`chime-bearer` パラメータと一致する `app-instance-user-arn` が必要です。API によっては、追加の ARN (通常はチャンネル ARN) が必要になる場合があります。これにより、上記の例のようなバックエンドサービスの場合、IAM ポリシーは以下の例のようになります。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "chime:SendMessage",
      "chime:ListChannelMessages",
      "chime:CreateChannelMembership",
      "chime:ListChannelMemberships",
      "chime>DeleteChannelMembership",
      "chime:CreateChannel",
      "chime:ListChannels",
      "chime>DeleteChannel",
      ...
    ],
    "Resource": [
      "arn:aws:chime:region:aws_account_id:app-instance/app_instance_id/user/back-end-worker",
      "arn:aws:chime:region:aws_account_id:app-instance/app_instance_id/channel/*"
    ]
  }
}
```

Resource セクションの `AppInstanceUser` ARN とチャンネル ARN を書き留めます。この IAM ポリシーの例では、「back-end-worker」の ID を持つユーザーとして API コールを行うアクセス許可をバックエンドサービスに付与します。バックエンドサービスがアプリケーションを使用するユーザーを呼び出せるようにするには、`app_instance_user_arn` を `arn:aws:chime:region:aws_account_id:app-instance/app_instance_id/user/*` に変更します。

暗黙的な API 認証について

IAM ポリシーに加えて、Amazon Chime SDK メッセージング API には暗黙的なアクセス許可があります。例えば、`AppInstanceUser` はメッセージを送信したり、ユーザーが属するチャンネルのチャンネルメンバーシップを一覧表示することしかできません。ただし、`AppInstanceAdmin` に昇格した `AppInstanceUser` は例外です。デフォルトでは、管理者はアプリケーション内のすべてのチャネ

ルへのアクセス許可を持っています。ほとんどのユースケースでは、重要なビジネスロジックを含むバックエンドサービスにのみこれが必要です。

以下の CLI コマンドは、バックエンドユーザーを管理者に昇格させます。

```
aws chime-sdk-identity create-app-instance-admin \  
  --app-instance-admin-arn "app_instance_user_arn" \  
  --app-instance-arn "app_instance_arn"
```

チャンネルメッセージの送信と一覧表示

以下の CLI コマンドは、チャンネルメッセージを送信します。

```
aws chime-sdk-messaging send-channel-message \  
  --chime-bearer "app_instance_user_arn" \  
  --channel-arn "channel_arn" \  
  --content "hello world" \  
  --type STANDARD \  
  --persistence PERSISTENT
```

以下の CLI コマンドは、チャンネルメッセージを新しい順に一覧表示します。

- `aws chime list-channel-messages`
- `aws chime-sdk-messaging list-channel-messages`

```
aws chime-sdk-messaging list-channel-messages \  
  --chime-bearer "app_instance_user_arn" \  
  --channel-arn "channel_arn"
```

エンドユーザークライアントアプリケーションの認証

エンドユーザークライアントアプリケーションから Amazon Chime SDK メッセージングを実行することもできます。 [バックエンドサービスから SDK 呼び出しを行う](#)では、`create-channel`、`send-channel-message`、などの API コールを行う方法について説明します `list-channel-messages`。ブラウザおよびモバイルアプリケーションなどのエンドユーザークライアントアプリケーションがこれらの同じ API コールを行います。クライアントアプリケーションは、経由で接続 WebSocket して、

メンバーであるチャンネルへのメッセージやイベントのリアルタイムの更新を受信することもできます。このセクションでは、特定のアプリケーションインスタンスユーザーを対象とするクライアントアプリケーションに IAM 認証情報を付与する方法について説明します。エンドユーザーがこれらの認証情報を取得すると、[バックエンドサービスから SDK 呼び出しを行う](#) に示す API コールを行うことができます。クライアントアプリケーションの完全なデモを見るには、<https://github.com/aws-samples/amazon-chime-sdk/tree/main/apps/chat> を参照してください。クライアントアプリケーションが属するチャンネルからリアルタイムメッセージを受信する方法については、「[WebSockets を使用したメッセージの受信](#)」を参照してください。

エンドユーザーへの IAM 認証情報の付与

Amazon Chime SDK メッセージングは、AWS Identity and Access Management (IAM) ポリシーとネイティブに統合され、受信リクエストを認証します。IAM ポリシーは、個々のユーザーができることを定義します。IAM ポリシーは、ユースケースに合わせてスコープダウンされ限定された認証情報を提供するように作成できます。Amazon Chime SDK メッセージングユーザーに対するポリシー作成の詳細については、「[IAM ロールの例](#)」を参照してください。

既存のアイデンティティプロバイダーをお持ちの場合、既存のアイデンティティを Amazon Chime SDK メッセージングと統合するための以下のオプションがあります。

- 既存の ID プロバイダーを使用してユーザーを認証し、認証サービスを AWS Security Token Service (STS) と統合して、クライアント用の独自の認証情報供給サービスを作成できます。STS には IAM ロールを引き受けるための API が用意されています。
- SAML または OpenID 互換の ID プロバイダーがすでにある場合は、AWS STS [AssumeRoleWithSAML](#) および [AssumeRoleWithWebIdentity](#) への呼び出しを抽象化する Amazon [Cognito ID プール](#) を使用することをお勧めします。Amazon Cognito は、OpenID や SAML のほか、Facebook、Login with Amazon、Google、Sign in with Apple などのパブリックアイデンティティプロバイダーと統合します。

アイデンティティプロバイダーをお持ちでない場合は、Amazon Cognito ユーザープールの使用を開始できます。Amazon Cognito を Amazon Chime SDK メッセージング機能と共に使用する方法的例については、「[Build chat features into your application with Amazon Chime SDK messaging](#)」を参照してください。

あるいは、[AWS STS](#) を使用して独自の認証情報供給サービスを作成したり、独自のアイデンティティプロバイダーを構築したりすることもできます。

STS を使用して認証情報を供給する

ActiveDirectory LDAP などの IDP がすでにあり、カスタム認証情報供給サービスを実装する場合、または認証されていない会議の参加者にチャットへのアクセスを許可する場合は、[AWS STS AssumeRole API](#) を使用できます。これを行うには、まず Amazon Chime SDK メッセージング SDK ロールを作成します。このロールの作成の詳細については、「[IAM ユーザーにアクセス許可を委任するロールの作成](#)」を参照してください。

IAM ロールには、次のような、アプリケーションが使用する Amazon Chime SDK メッセージングアクションに対するアクセス許可があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "chime:GetMessagingSessionEndpoint"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "chime:SendChannelMessage",
        "chime:ListChannelMessages",
        "chime:CreateChannelMembership",
        "chime:ListChannelMemberships",
        "chime>DeleteChannelMembership",
        "chime:CreateChannelModerator",
        "chime:ListChannelModerators",
        "chime:DescribeChannelModerator",
        "chime:CreateChannel",
        "chime:DescribeChannel",
        "chime:ListChannels",
        "chime>DeleteChannel",
        "chime:RedactChannelMessage",
        "chime:UpdateChannelMessage",
        "chime:Connect",
        "chime:ListChannelBans",
        "chime:CreateChannelBan",
        "chime>DeleteChannelBan",
        "chime:ListChannelMembershipsForAppInstanceUser"
      ]
    }
  ]
}
```



```

        "chime:AssociateChannelFlow",
        "chime:DisassociateChannelFlow",
        "chime:GetChannelMessageStatus"
    ],
    "Resource": [
        "${chime_app_instance_arn}/user/
${aws:PrincipalTag/my_applications_user_id}",
        "${chime_app_instance_arn}/channel/*"
    ]
}
]
}
}

```

この例では、このロールを と呼び出しますChimeMessagingSampleAppUserRole。

ユーザー ARN リソースのChimeMessagingSampleAppUserRoleポリ

シー`-${my_application_user_id}`のセッションタグを書き留めます。このセッションタグは [AssumeRole](#) API コールでパラメータ化され、返される認証情報を 1 人のユーザーのアクセス許可に制限します。

[AssumeRole](#) および [TagSession](#) APIs は、IAM ユーザーなど、既に認証情報が付与されている IAM エンティティを使用して呼び出されます。APIs は、実行ロールなどの別の IAM [AWS Lambda ロール](#)によって呼び出すこともできます。その IAM アイデンティティには、[TagSession](#)で [AssumeRole](#)および [TagSession](#)を呼び出すためのアクセス許可が必要ですChimeMessagingSampleAppUserRole。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ],
      "Resource":
        "arn:aws:iam::my_aws_account_id:role/ChimeMessagingSampleAppUserRole"
    }
  ]
}

```

この例では、このロールをChimeSampleAppServerロール と呼び出します。

が [STS AssumeRole API](#) をChimeMessagingSampleAppServerRole呼び出すことを許可する信頼ポリシーChimeMessagingSampleAppUserRoleを使用してを設定する必要があります。IAM ロールでの信頼ポリシーの使用の詳細については、「[How to use trust policies with IAM roles](#)」を参照してください。AWS IAM ロールコンソールを使用して、このポリシーを に追加できますChimeMessagingSampleAppUserRole。以下の例は、標準的な信頼関係を示しています。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::my_aws_account_id:role/ChimeMessagingSampleAppServerRole"
      }
      "Action": "sts:AssumeRole"
    }
  ]
}
```

サンプルデプロイでは、[Amazon EC2](#) インスタンス、または AWS Lambda が で起動されますChimeMessagingSampleAppServerRole。その後、サーバーは以下を実行します。

1. クライアントの認証情報受信リクエストに対して、アプリケーション固有の認証を実行します。
2. `${aws:PrincipalTag/my_applications_user_id}` をパラメータ化するタグを使用して、ChimeMessagingSampleAppUserRole で STS AssumeRole を呼び出します。
3. AssumeRole 呼び出しで返された認証情報をユーザーに転送します。

以下の例は、ステップ 2 のロールを引き受けるための CLI コマンドを示しています。

```
aws sts assume-role --role-arn
arn:aws:iam::my_aws_account_id:role/ChimeMessagingSampleAppUserRole --role-
session-name demo --tags Key=my_applications_user_id,Value=123456789
```

チャネルの作成

ユーザーおよびエンドユーザーはチャネルを作成できます。作成したら、ユーザーまたはエンドユーザーもチャネルにメンバーを追加する必要があります。チャネルを作成するためのサンプルコードは、[のサンプルアプリケーションで GitHub](#)入手できます。

チャンネルの作成およびメンバーの追加の詳細については、以下を参照してください。

- [CreateChannel](#)
- [CreateChannelMembership](#)

メッセージの送信

[SendChannelMessage](#) API を使用してメッセージを送信します。サンプルコードは、[のサンプルアプリケーションで GitHub](#) 使用できます。

の使用 ExpirationSettings

AppInstanceUser または を作成する場合Channel、 を使用してこれらのリソースExpirationSettingsを自動削除するように設定できます。ExpirationSettingsは、ストレージコストを削減し、問題を防ぐ resource-limit-exceededの役に立ちます。例えば、未使用のチャンネルを7日後に削除したり、テスト目的でのみ呼び出された AppInstanceUser を削除したりできます。

AppInstanceUser では、ユーザーの作成時刻に基づいて有効期限を指定します。Channel では、チャンネルの作成時刻または最終メッセージ時刻に基づいて有効期限を指定します。後者の場合、メッセージアクティビティを使用して自動削除をカスタマイズできます。

Important

リソースの有効期限が切れるとすぐに、ExpirationSettings はそのリソースを削除するバックグラウンドプロセスを開始します。このプロセスには通常6時間かかりますが、この時間は変わる場合があります。

期限切れとなってもまだ削除されていない AppInstanceUsers および Channels は、引き続き有効かつアクティブと表示されます。有効期限の設定は更新または削除でき、変更内容はシステムによって反映されます。

トピック

- [の設定 ExpirationSettings](#)
- [AWS CloudTrail 期限切れのリソース削除の イベント](#)

の設定 ExpirationSettings

以下のセクションでは、AppInstanceUser または Channel の ExpirationSettings の設定方法について説明します。

リソースの作成ExpirationSettings時の の設定

[CreateAppInstanceUser](#) または [CreateChannel](#) APIs を実行するExpirationSettingsときに を設定できます。ExpirationSettings パラメータを設定する場合は、次の IAM アクセス許可を付与する必要があります。

- AppInstanceUser を作成する場合は `chime:PutAppInstanceUserExpirationSettings`。
- Channel を作成する場合は `chime:PutChannelExpirationSettings`。

次の例では、AWS CLI を使用して、1 日後に期限切れAppInstanceUserになる を作成します。

```
aws chime-sdk-identity create-app-instance-user \  
--app-instance-arn "app_instance_arn" \  
--app-instance-user-id "backend-worker" \  
--name "backend-worker" \  
--expiration-settings '{  
    "ExpirationDays": 1,  
    "ExpirationCriterion": "CREATED_TIMESTAMP"  
}'
```

次の例では、AWS CLI を使用して、最後にメッセージを受信してから 1 日後に期限切れChannelになる を作成します。

```
aws chime-sdk-messaging create-channel \  
--chime-bearer "app_instance_user_arn" \  
--app-instance-arn "app_instance_arn" \  
--name "firstChannel" \  
--expiration-settings '{  
    "ExpirationDays": 1,  
    "ExpirationCriterion": "LAST_MESSAGE_TIMESTAMP"  
}'
```

Put APIs を使用して を設定する ExpirationSettings

[PutAppInstanceUserExpirationSettings](#) および [PutChannelExpirationSettings](#) APIs を使用して、 を作成、更新、削除することもできますExpirationSettings。

次の例は、AWS CLI を使用して の を更新するAppInstanceUser方法を示していませんExpirationSettings。

```
aws chime-sdk-identity put-app-instance-user-expiration-settings \  
--app-instance-user-arn "app_instance_user_arn" \  
--expiration-settings '{  
    "ExpirationDays": 30,  
    "ExpirationCriterion": "CREATED_TIMESTAMP"  
}'
```

次の例は、AWS CLI を使用してチャンネルの を削除する方法を示していませんExpirationSettings。

```
aws chime-sdk-messaging put-channel-expiration-settings \  
--chime-bearer "app_instance_user_arn" \  
--channel-arn "channel_arn"
```

AWS CloudTrail 期限切れのリソース削除の イベント

期限切れのリソースを削除すると、ExpireAppInstanceUserまたは ExpireChannelイベントがに送信されます AWS CloudTrail。イベントの種類は、削除されたアセットの種類によって異なります。

次の例は、AppInstanceUser イベントを示しています。

```
{  
  "eventVersion": "1.08",  
  "userIdentity": {  
    "accountId": "123456789012",  
    "invokedBy": "chime.amazonaws.com"  
  },  
  "eventTime": "2023-03-15T00:00:00Z",  
  "eventSource": "chime.amazonaws.com",  
  "eventName": "ExpireAppInstanceUser",  
  "awsRegion": "us-east-1",  
  "sourceIPAddress": "chime.amazonaws.com",  
  "userAgent": "chime.amazonaws.com",  
  "requestParameters": null,  
  "responseElements": null,  
  "eventID": "12345678-1234-1234-1234-123456789012",  
  "readOnly": false,  
}
```

```
"resources": [
  {
    "accountId": "123456789012",
    "type": "AWS::Chime::AppInstanceUser",
    "ARN": "arn:aws:chime:us-east-1:123456789012:app-instance/app-instance-id/
user/user-id"
  }
],
"eventType": "AwsServiceEvent",
"managementEvent": true,
"recipientAccountId": "123456789012",
"serviceEventDetails": {
  "reason": "AppInstanceUser deleted due to expiration settings."
},
"eventCategory": "Management"
}
```

WebSockets を使用したメッセージの受信

[Amazon Chime JS SDK](#) を使用して を使用してメッセージを受信するか WebSockets、任意の WebSocket クライアントライブラリを使用できます。

の使用を開始するには、リストされている順序で以下のトピックに従ってください WebSockets。

トピック

- [IAM ポリシーの定義](#)
- [エンドポイントの取得](#)
- [接続の確立](#)
- [プリフェッチを使用してチャネルの詳細を配信する](#)
- [イベントの処理](#)

IAM ポリシーの定義

開始するには、WebSocket 接続を確立するアクセス許可を付与する IAM ポリシーを定義します。次のポリシー例では、接続を確立するAppInstanceUserアクセス許可を に付与します WebSocket。

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

```
"Effect": "Allow",
"Action": [
  "chime:Connect"
],
"Resource": [
  "arn:aws:chime:region:{aws_account_id}:app-instance/{app_instance_id}/user/
{app_instance_user_id}"
]
},
{
  "Effect": "Allow",
  "Action": [
    "chime:GetMessagingSessionEndpoint"
  ],
  "Resource": [
    "*"
  ]
}
]
```

エンドポイントの取得

次の手順では、WebSocket 接続で使用されるエンドポイントを取得する方法について説明します。

1. [GetMessagingSessionEndpoint](#) API を使用して WebSocket エンドポイントを取得します。
2. [GetMessagingSessionEndpoint](#) API によって返される URL を使用して、署名バージョン 4 の署名付き WebSocket URL を作成します。その際にヘルプが必要な場合は、[接続の確立](#) の指示に従ってください。

Note

WebSocket URLsの形式は次のとおりです。 *id.region*.ws-messaging.chime.aws

接続の確立

エンドポイントを取得したら、Connect API を使用して Amazon Chime SDK バックエンドサーバー WebSocket への接続を確立し、 のメッセージを受信しますAppInstanceUser。リクエストに署名するには、AWS 署名バージョン 4 を使用する必要があります。リクエストの署名の詳細については、「署名 [バージョン 4 によるリクエストの署名 AWS](#)」を参照してください。

Note

エンドポイントを取得するには、[GetMessagingSessionEndpoint](#) API を呼び出します。選択した WebSocket クライアントライブラリを使用してエンドポイントに接続できます。

リクエストの構文

```
GET /connect
?X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIARALLEXAMPLE%2F20201214%2Fregion%2Fchime%2Faws4_request
&X-Amz-Date=20201214T171359Z
&X-Amz-Expires=10
&X-Amz-SignedHeaders=host
&sessionId={sessionId}
&userArn={appInstanceUserArn}
&X-Amz-Signature=db75397d79583EXAMPLE
```

URI リクエストパラメータ

URI リクエストクエリパラメータはすべて URL でエンコードされている必要があります。

X-Amz-Algorithm

AWS 署名のバージョンと、署名の計算に使用したアルゴリズムを識別します。Amazon Chime SDK は AWS 署名バージョン 4 認証のみをサポートしているため、この値は AWS4-HMAC-SHA256 です。

X-Amz-Credential

このパラメータは、アクセスキー ID に加えて、署名が有効な AWS リージョンとサービス、つまりスコープも提供します。この値は、署名の計算で使用するスコープと一致する必要があります。このパラメータ値の一般的な形式は次のとおりです。

```
<yourAccessKeyId>/<date>/<awsRegion>/<awsService >/aws4_request
```

例:

```
AKIAIOSFODNN7EXAMPLE/20201214/us-east-1/chime/aws4_request
```

X-Amz-Date

日付と時刻の形式は ISO 8601 規格に準拠している必要があるため、yyyyMMdTTHHmssZ という形式にする必要があります。例えば、2020 年 8 月 1 日 15:32:41.982-700 を協定世界時 (UTC) に変換し、20200801T083241Z として送信する必要があります。

X-Amz-Signed-Headers

署名の計算に使用したヘッダーを一覧表示します。署名計算には次のヘッダーが必要です。

- HTTP ホストヘッダー。
- リクエストに追加する予定のすべての x-amz-* ヘッダー。

Note

セキュリティを強化するため、リクエストに含める予定のすべてのリクエストヘッダーに署名します。

X-Amz-Signatures

リクエストを認証するための署名を指定します。この署名は、Amazon Chime SDK が計算する署名と一致する必要があります。一致しない場合、Amazon Chime SDK はリクエストを拒否します。例えば 733255ef022bec3f2a8701cd61d4b371f3f28c9f19EXAMPLEd48d5193d7 です。

X-Amz-Security-Token

Security Token Service から取得した認証情報を使用する場合のオプションの認証情報パラメータ。このサービスの詳細については、「<https://docs.aws.amazon.com/STS/latest/APIReference/>」を参照してください。

SessionId

確立されている WebSocket 接続の一意の ID を示します。

UserArn

接続を確立しようとしている AppInstanceUser の ID を示します。値は、AppInstanceUser の ARN でなければなりません。例えば、次のようになります: `arn:aws:chime:us%2Deast%2D1:123456789012:app%2Dinstance/694d2099%2Dcb1e%2D463e%2D9d64%2D697ff5b8950e/user/johndoe`

プリフェッチを使用してチャンネルの詳細を配信する

WebSocket 接続を確立するときに、クエリパラメータ `prefetch-on=connect` を指定して `CHANNEL_DETAILS` イベントを配信できます。プリフェッチ機能は接続 API に付属しており、この機能によりユーザーは API を余分に呼び出さなくても充実したチャットビューを表示できます。ユーザーは次の操作を実行できます。

- 直近のチャンネルメッセージのプレビューとそのタイムスタンプを表示する。
- チャンネルのメンバーを表示する。
- チャンネルの未読マーカを表示する。

ユーザーがプリフェッチパラメータを指定して接続すると、接続が確立されたことを示すセッション確立イベントがユーザーに届きます。その後、ユーザーは最大 50 件の `CHANNEL_DETAILS` イベントを受信します。ユーザーのチャンネル数が 50 個未満の場合、接続 API は `CHANNEL_DETAILS` イベントを介してすべてのチャンネルをプリフェッチします。ユーザーのチャンネルが 50 個を超える場合、API は未読メッセージと最新の `LastMessageTimestamp` 値を含む上位 50 個のチャンネルをプリフェッチします。`CHANNEL_DETAILS` イベントはランダムな順序で到着し、ユーザーは 50 個のチャンネルすべてのイベントを受信します。

また、プリフェッチは `ChannelMessages` と `ChannelMemberships` に対して以下を返します。

- `ChannelMessages` - [ChannelMessageSummary](#) オブジェクトのリスト。降順 `CreatedTimestamp` で並べられます。ユーザーに表示されている最新 20 件のメッセージのみが含まれます。現在のユーザーには表示されないターゲットを絞ったメッセージがチャンネルにある場合、返されるメッセージは 20 件未満になる可能性があります。`ChannelMessagesHasMore` ブール値は `true` に設定され、メッセージが他にもあることを示します。AWS アカウントレベルで調整可能なソフト制限。
- `ChannelMemberships` - [ChannelMembershipSummary](#) オブジェクトのリスト。最大 30 人のチャンネルメンバーが含まれます。ソフト制限、AWS アカウントレベルで調整可能。

次の例では、`prefetch-on=connect` の使用方法を示します。

```
GET /connect
?X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIARALLEEXAMPLE%2F20201214%2Fregion%2Fchime%2Faws4_request
&X-Amz-Date=20201214T171359Z
&X-Amz-Expires=10
&X-Amz-SignedHeaders=host
```

```
&sessionId=sessionId
&prefetch-on=connect
&userArn=appInstanceUserArn
&X-Amz-Signature=db75397d79583EXAMPLE
```

この例は、1つのチャンネルのレスポンスを示しています。50個のチャンネルすべてについてのレスポンスが届きます。

```
{
  "Headers": {
    "x-amz-chime-event-type": "CHANNEL_DETAILS",
    "x-amz-chime-message-type": "SYSTEM"
  },
  "Payload": JSON.stringify({
    Channel: ChannelSummary
    ChannelMessages: List of ChannelMessageSummary
    ChannelMemberships: List of ChannelMembershipSummary
    ReadMarkerTimestamp: Timestamp
    ChannelMessagesHasMore: Boolean
  })
}
```

イベントの処理

AppInstanceUser が接続を確立した後にメッセージを受信するには、メッセージをチャンネルに追加する必要があります。そのためには、[CreateChannelMembership](#) API を使用します。

Note

AppInstanceUser は、自分が属しているすべてのチャンネルのメッセージを常に受信します。AppInstance ユーザーが接続を切断すると、メッセージングは停止します。

[CreateChannelMembership](#) API を使用して明示的に追加しない限り、AppInstanceAdminとはチャンネルでメッセージを受信ChannelModeratorしません。

以下のトピックでは、イベントを処理する方法について説明します。

トピック

- [メッセージの構造について](#)
- [切断への対応](#)

メッセージの構造について

受信するすべての WebSocket メッセージは、次の形式に従います。

```
{
  "Headers": {"key": "value"},
  "Payload": "{\"key\": \"value\"}"
}
```

ヘッダー

Amazon Chime SDK メッセージングでは、以下のヘッダーキーを使用します。

- x-amz-chime-event-type
- x-amz-chime-message-type
- x-amz-chime-event-reason

次のセクションでは、ヘッダーで有効な値およびペイロードを示して説明します。

ペイロード

Websocket メッセージは JSON 文字列を返します。JSON 文字列の構造は x-amz-event-type ヘッダーによって異なります。以下の表に、有効な x-amz-chime-event-type 値とペイロードを示します。

EventType	ペイロード形式	
SESSION_ESTABLISHED	該当なし。このメッセージは、ユーザーが に接続した後に 1 回送信されます WebSocket。これは、ユーザーがメッセージを受信した後に到着するチャンネル上の SESSION_ESTABLISHED メッセージまたはイベントが、 が WebSocket 開いている限り、ユーザーに配信され	

EventType	ペイロード形式	
	<p>ることが保証されていることを示します。</p>	
CREATE_CHANNEL_MESSAGE	ChannelMessage	
REDACT_CHANNEL_MESSAGE		
UPDATE_CHANNEL_MESSAGE		
DELETE_CHANNEL_MESSAGE		
PENDING_CREATE_CHANNEL_MESSAGE		
PENDING_UPDATE_CHANNEL_MESSAGE		
FAILED_CREATE_CHANNEL_MESSAGE		
FAILED_UPDATE_CHANNEL_MESSAGE		
DENIED_CREATE_CHANNEL_MESSAGE		
DENIED_UPDATE_CHANNEL_MESSAGE		

EventType	ペイロード形式	
CHANNEL_DETAILS	<p>Channel</p> <p>ChannelSummary オブジェクト。</p> <p>ChannelMessages</p> <p>ChannelMessageSummary オブジェクトのリスト。降順CreatedTimestamp で並べられます。最新の 20 件のメッセージが含まれますが、その制限は AWS アカウントレベルで調整できます。</p> <p>ChannelMemberships</p> <p>ChannelMembershipSummary オブジェクトのリスト 最大 30 のチャネルメンバーを返しますが、その制限は AWS アカウントレベルで調整できます。</p> <p>ReadMarkerタイムスタンプ</p> <p>AppInstanceUser が最後にチャネルを既読としてマークした時刻。</p>	
UPDATE_CHANNEL	Channel	
DELETE_CHANNEL		
BATCH_CREATE_CHANNEL_MEMBERSHIP	BatchChannelMembership	

EventType	ペイロード形式
CREATE_CHANNEL_MEMBERSHIP	ChannelMembership
DELETE_CHANNEL_MEMBERSHIP	
UPDATE_CHANNEL_MEMBERSHIP	

x-amz-chime-message-type

以下の表に x-amz-chime-message-type メッセージタイプを示します。

メッセージの種類	説明
STANDARD	WebSocket が STANDARD チャネルメッセージを受信すると送信されます。
CONTROL	が「コントロールチャネル」メッセージ WebSocket を受信すると送信されます。
SYSTEM	Amazon Chime SDK メッセージングによって送信されるその他すべての WebSocket メッセージ。

x-amz-chime-event-reason

これは特定のユースケースでサポートされるオプションのヘッダーです。ヘッダーには、特定のイベントを受信した理由に関する情報が表示されます。

イベント理由	説明
subchannel_DELETED	Elastic チャネルモデレーターが受信した DELETE_CHANNEL_MEMBERSHIP イベント。メンバーシップバランシングにより属して

イベント理由	説明
	いたサブチャンネルが削除された後、モデレーターにのみ表示されます。

切断への対応

WebSocket は、ネットワーク接続の変化や認証情報の有効期限が切れると切断されることがあります。を開くと WebSocket、Amazon Chime SDK はメッセージングクライアントに通常の ping を送信して、まだ接続されていることを確認します。接続が閉じると、クライアントは WebSocket クローズコードを受け取ります。クローズコードによっては、クライアントが再接続できる場合とできない場合があります。次の表は、クライアントが再接続に使用できるクローズコードを示しています。

1000~4000 のクローズコードについては、次のメッセージの場合にのみ再接続してください。

クローズコード	再接続可能	理由
1001	はい	正常なクローズ
1006	はい	異常なクローズ
1011	はい	内部サーバーエラー
1012	はい	サービスの再起動
1013	はい	後でもう一度試してみてください
1014	はい	サーバーは、ゲートウェイまたはプロキシとして機能しており、上流サーバーから無効な応答を受信しました。これは 502 の HTTP ステータスコードに似ています。

4XXX コードの場合は、常に再接続できます。ただし、次のメッセージを除きます。

クローズコード	再接続可能	理由
4002	いいえ	クライアント自身によるクローズ
4003	いいえ	Forbidden
4401	いいえ	権限がありません

アプリケーションが再接続にクローズコードを使用する場合、アプリケーションには、次の動作が求められます。

1. [GetMessagingSessionEndpoint](#) API を再度呼び出して、新しいベース URL を取得します。
2. IAM 認証情報の有効期限が切れている場合は更新する。
3. 経由で接続します WebSocket。

amazon-chime-sdk-js ライブラリを使用する場合、[needsRefresh \(\)](#) プロパティと [refresh\(\)](#) メソッドを実装すると、これが処理されます。実際の例については、<https://github.com/aws-samples/amazon-chime-sdk/blob/dc11c4c76c78d28f618577706bba2087919a5635/apps/chat/src/providers/AuthProvider.jsx#L93-L101> を参照してください。

アタッチメントの設定

Amazon Chime SDK では、メッセージのアタッチメントに独自のストレージを使用し、メッセージのメタデータとして含めることができます。Amazon Simple Storage Service (S3) は、アタッチメントの使用を開始する最も簡単な方法です。

S3 をアタッチメントに使用するには

1. アタッチメントを保存する S3 バケットを作成します。
2. Amazon Chime SDK ユーザーが S3 バケットからアタッチメントをアップロード、ダウンロード、削除できるようにするバケットの IAM ポリシーを作成します。
3. アイデンティティプロバイダーが使用する IAM ロールを作成して、アタッチメント用の認証情報をユーザーに供給します。

[サンプルアプリケーション](#)には、Amazon S3、Amazon Cognito、および Amazon Chime SDK を使用してこれを行う方法の例が示されています。

システムメッセージについて

Amazon Chime SDK は、チャンネル内で発生するイベントについて、接続されているすべてのクライアントにシステムメッセージを送信します。イベントは以下のとおりです。

- UPDATE_CHANNEL — このイベントは、名前やメタデータなどのチャンネルの詳細が更新されたことを示します。
- DELETE_CHANNEL — このイベントは、メッセージ、メンバーシップ、モデレーター、アクセス禁止など、チャンネルおよびそのすべてのデータが削除されることを意味します。
- CREATE_CHANNEL_MEMBERSHIP — このイベントは、特定の AppInstanceUser がチャンネルにメンバーとして追加されたことを示します。イベントには新しい AppInstanceUser の詳細も含まれています。
- DELETE_CHANNEL_MEMBERSHIP — このイベントは、AppInstanceUser がチャンネルから削除されたことを示します。イベントには削除された AppInstanceUser 詳細も含まれています。
- UPDATE_CHANNEL_MEMBERSHIP — このイベントは Elastic チャンネルにのみ適用されます。このイベントは、メンバーシップバランシングが AppInstanceUser をあるサブチャンネルから別のサブチャンネルに移管したことを意味します。このイベントには、AppInstanceUser の詳細に加えて、AppInstanceUser の移管先のサブチャンネルに関する情報も含まれています。

IAM ロールの例

ユーザーが Amazon Chime SDK のメッセージング機能にアクセスするには、サインイン時にユーザーに認証情報を提供するための IAM ロールおよびポリシーを定義する必要があります。IAM ポリシーは、ユーザーがアクセスできるリソースを定義します。

このセクションの例では、ニーズに合わせて調整できる基本的なポリシーを示しています。ポリシーの仕組みの詳細については、「[バックエンドサービスから SDK 呼び出しを行う](#)」を参照してください。

この例は、Amazon Chime SDK メッセージングを使用してアプリケーションを構築する開発者向けのポリシーを示しています。

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

```
"Statement": [
  {
    "Action": [
      "chime:CreateAppInstance",
      "chime:DescribeAppInstance",
      "chime:ListAppInstances",
      "chime:UpdateAppInstance",
      "chime>DeleteAppInstance",
      "chime:CreateAppInstanceUser",
      "chime>DeleteAppInstanceUser",
      "chime:ListAppInstanceUsers",
      "chime:UpdateAppInstanceUser",
      "chime:DescribeAppInstanceUser",
      "chime:CreateAppInstanceAdmin",
      "chime:DescribeAppInstanceAdmin",
      "chime:ListAppInstanceAdmins",
      "chime>DeleteAppInstanceAdmin",
      "chime:PutAppInstanceRetentionSettings",
      "chime:GetAppInstanceRetentionSettings",
      "chime:PutAppInstanceStreamingConfigurations",
      "chime:GetAppInstanceStreamingConfigurations",
      "chime>DeleteAppInstanceStreamingConfigurations",
      "chime:TagResource",
      "chime:UntagResource",
      "chime:ListTagsForResource",
      "chime:CreateChannelFlow",
      "chime:UpdateChannelFlow",
      "chime:DescribeChannelFlow",
      "chime>DeleteChannelFlow",
      "chime:ListChannelFlows",
      "chime:ListChannelsAssociatedWithChannelFlow",
      "chime:ChannelFlowCallback",
    ],
    "Effect": "Allow",
    "Resource": "*"
  }
]
```

この例は、Amazon Chime SDK のユーザーアクションへのアクセスをユーザーに許可するポリシーを示しています。

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Action": "chime:GetMessagingSessionEndpoint",
    "Effect": "Allow",
    "Resource": "*"
  },
  {
    "Action": [
      "chime:CreateChannel",
      "chime:DescribeChannel",
      "chime>DeleteChannel",
      "chime:UpdateChannel",
      "chime:ListChannels",
      "chime:Listsubchannels",
      "chime:ListChannelMembershipsForAppInstanceUser",
      "chime:DescribeChannelMembershipForAppInstanceUser",
      "chime:ListChannelsModeratedByAppInstanceUser",
      "chime:DescribeChannelModeratedByAppInstanceUser",
      "chime:UpdateChannelReadMarker",
      "chime:CreateChannelModerator",
      "chime:DescribeChannelModerator",
      "chime:ListChannelModerators",
      "chime>DeleteChannelModerator",
      "chime:SendChannelMessage",
      "chime:GetChannelMessage",
      "chime>DeleteChannelMessage",
      "chime:UpdateChannelMessage",
      "chime:RedactChannelMessage",
      "chime:ListChannelMessages",
      "chime:CreateChannelMembership",
      "chime:DescribeChannelMembership",
      "chime>DeleteChannelMembership",
      "chime:ListChannelMemberships",
      "chime:CreateChannelBan",
      "chime>DeleteChannelBan",
      "chime:ListChannelBans",
      "chime:DescribeChannelBan",
      "chime:Connect",
      "chime:AssociateChannelFlow",
      "chime:DisassociateChannelFlow",
      "chime:GetChannelMessageStatus"
    ],
    "Effect": "Allow",
```

```

    "Resource": [
      "arn:aws:chime:region:{aws_account_id}:app-instance/{app_instance_id}/
user/{app_instance_user_id}",
      "arn:aws:chime:region:{aws_account_id}:app-instance/{app_instance_id}/
channel/*"
    ]
  }
]
}

```

この例は、Amazon Chime SDK のユーザーアクションへの最小限のアクセスをユーザーに与えるポリシーを示しています。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "chime:GetMessagingSessionEndpoint",
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "chime:ListChannels",
        "chime:DescribeChannel",
        "chime:ListChannelMembershipsForAppInstanceUser",
        "chime:DescribeChannelMembershipForAppInstanceUser",
        "chime:ListChannelsModeratedByAppInstanceUser",
        "chime:DescribeChannelModeratedByAppInstanceUser",
        "chime:SendChannelMessage",
        "chime:GetChannelMessage",
        "chime:ListChannelMessages",
        "chime:Connect"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:chime:region:{aws_account_id}:app-instance/{app_instance_id}/
user/{app_instance_user_id}",
        "arn:aws:chime:region:{aws_account_id}:app-instance/{app_instance_id}/
channel/*"
      ]
    }
  ]
}

```

```
}
```

この例は、WebSocket の接続を確立するためのポリシーを示していますAppInstanceUser。WebSocket 接続の詳細については、「」を参照してください [WebSockets を使用したメッセージの受信](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "chime:Connect"
      ],
      "Resource": [
        "arn:aws:chime:region:{aws_account_id}:app-instance/{app_instance_id}/
user/{app_instance_user_id}"
      ]
    }
  ]
}
```

ロールごとの権限について

このトピックの表は、アプリケーションインスタンスのユーザーがロールに応じて実行できるアクションを示しています。

凡例

- [許可] — IAM ポリシーに正しいアクション/リソースコンテキストが指定されていれば、正常に実行できます。
- [制限付きで許可] — IAM ポリシーに正しいアクション/リソースコンテキストが指定されている場合、特定の条件を満たしていればアクションを正常に実行することができます。
- [拒否] — IAM ポリシーで正しいアクション/リソースコンテキストが指定されていても、バックエンドによってブロックされます。

トピック

- [AppInstanceAdmin](#)

- [ChannelModerator](#)
- [メンバー](#)
- [非メンバー](#)

AppInstanceAdmin

アプリケーションインスタンスの管理者は、自分が管理者であるアプリケーションインスタンス内のチャンネルに対してアクションを実行できます。

API 名	許可または拒否	メモ
UpdateChannel	制限付きで許可	一度設定 ElasticChannelConfiguration すると更新できない
DeleteChannel	許可	
DescribeChannel	許可	
ListChannel	許可	
ListChannelMembershipsForAppInstanceUser	許可	AppInstanceUserArn に別のを入力することもできます AppInstanceUser 。
DescribeChannelMembershipForAppInstanceUser	許可	AppInstanceUserArn 別のを入力することもできます AppInstanceUser 。
ListChannelsModeratedByAppInstanceUser	許可	AppInstanceUserArn 別のを入力することもできます AppInstanceUser。
DescribeChannelModeratedByAppInstanceUser	許可	別の ppInstanceUser A AppInstanceUserArn を入力することもできます。エラスティックチャンネルには使用できません。

API 名	許可または拒否	メモ
CreateChannelMembership	許可	
DescribeChannelMembership	許可	
ListChannelMemberships	許可	
DeleteChannelMembership	許可	
SendChannelMessage	制限付きで許可	まず、 CreateChannelMembership を使用して自分のメンバーシップを作成し、次にAPIを呼び出す必要があります。
GetChannelMessage	許可	
ListChannelMessage	許可	
DeleteChannelMessage	許可	
RedactChannelMessage	許可	
UpdateChannelMessage	制限付きで許可	自分のメッセージのみを編集できます。
CreateChannelModerator	許可	
DeleteChannelModerator	許可	
DescribeChannelModerator	許可	

API 名	許可または拒否	メモ
ListChannelModerator	許可	
CreateChannelBan	制限付きで許可	禁止する AppInstanceUser は、そのチャンネルの AppInstanceAdmin またはモデレーターであってはなりません。
DeleteChannelBan	制限付きで許可	
DescribeChannelBan	許可	
ListChannelBan	許可	
UpdateChannelReadMarker	制限付きで許可	非エラスティックチャンネルの場合は、まず CreateChannelMembership API を使用して自分のメンバーシップを作成し、次に API を呼び出す必要があります。 Elastic チャンネルでは許可されません。
GetChannelMessage	制限付きで許可	送信済みメッセージにのみ許可されます。メッセージの送信者でない限り、チャンネルフローで処理中のメッセージでは許可されません。
ListChannelMessages	許可	
DeleteChannelMessage	制限付きで許可	送信済みメッセージにのみ許可されます。
RedactChannelMessage	制限付きで許可	送信済みメッセージにのみ許可されます。

API 名	許可または拒否	メモ
UpdateChannelMessage	制限付きで許可	自分の送信したメッセージのみを編集できます。
AssociateChannelFlow	許可	
DisassociateChannelFlow	許可	
GetChannelMessageStatus	制限付きで許可	自分のメッセージのメッセージステータスのみを取得できます。
ListSubChannels	許可	

ChannelModerator

チャンネルモデレーターは、自分がモデレーターの役割を担っているチャンネルでのみアクションを実行できます。

Note

AppInstanceAdmin であるモデレーターは、そのロールで許可されているチャンネルでアクションを実行できます。

API 名	許可または拒否	メモ
UpdateChannel	許可	一度設定 ElasticChannelConfiguration すると更新できない
DeleteChannel	許可	
DescribeChannel	制限付きで許可	パブリックチャンネルの詳細のみを取得できます。

API 名	許可または拒否	メモ
ListChannel	制限付きで許可	パブリックチャンネルの詳細のみを取得できます。
ListChannelMembershipsForAppInstanceUser	制限付きで許可	ARN は AppInstanceUserArn 値としてのみ使用できます。
DescribeChannelMembershipForAppInstanceUser	制限付きで許可	ARN は AppInstanceUserArn 値としてのみ使用できます。
ListChannelsModeratedByAppInstanceUser	制限付きで許可	ARN は AppInstanceUserArn 値としてのみ使用できます。
DescribeChannelModeratedByAppInstanceUser	制限付きで許可	AppInstanceUserArn に別のを入力することもできます AppInstanceUser。
CreateChannelMembership	許可	
DescribeChannelMembership	許可	
ListChannelMembership	許可	
DeleteChannelMembership	許可	
SendChannelMessage	制限付きで許可	CreateChannelMembership API を使用して、まず自分のメンバーシップを作成し、次に SendChannelMessage API を呼び出す必要があります。

API 名	許可または拒否	メモ
GetChannelMessage	許可	
ListChannelMessage	許可	
DeleteChannelMessage	拒否	
RedactChannelMessage	許可	
UpdateChannelMessage	制限付きで許可	自分のメッセージのみを更新できます。
CreateChannelModerator	許可	まず CreateChannelMembership API を使用して自分のメンバーシップを作成し、次に CreateChannelModerator API を呼び出す必要があります。
DeleteChannelModerator	許可	
DescribeChannelModerator	許可	
ListChannelModerator	許可	
CreateChannelBan	制限付きで許可	禁止する AppInstanceUser は、そのチャンネルの AppInstanceAdmin またはモデレーターであってはなりません。
DeleteChannelBan	制限付きで許可	
DescribeChannelBan	許可	
ListChannelBan	許可	

API 名	許可または拒否	メモ
UpdateChannelReadMarker	制限付きで許可	非エラスティックチャネルの場合は、まず CreateChannelMembership を使用して自分のメンバーシップを作成し、次に UpdateChannelReadMarker API を呼び出す必要があります。 Elastic チャネルでは許可されません。
GetChannelMessage	制限付きで許可	送信済みメッセージにのみ許可されます。メッセージの送信者でない限り、チャネルフローで処理中のメッセージでは許可されません。
ListChannelMessages	許可	
DeleteChannelMessage	拒否	
RedactChannelMessage	制限付きで許可	送信済みメッセージにのみ許可されます。
UpdateChannelMessage	制限付きで許可	自分の送信したメッセージのみを編集できます。
AssociateChannelFlow	許可	
DisassociateChannelFlow	許可	
GetChannelMessageStatus	制限付きで許可	自分のメッセージのメッセージステータスのみを取得できます。
ListSubChannels	許可	

メンバー

[CreateChannelMembership](#) API を介してチャンネルに追加されると、はチャンネルのメンバーAppInstanceUserになります。

メンバーは自分が属するチャンネルでのみアクションを実行できます。

Note

AppInstanceAdmin または ChannelModerator であるメンバーは、これら 2 つのロールで許可されているチャンネルでアクションを実行できます。

API 名	許可または拒否	メモ
UpdateChannel	拒否	
DeleteChannel	拒否	
DescribeChannel	制限付きで許可	パブリックチャンネルの詳細のみを取得できます。
ListChannel	制限付きで許可	パブリックチャンネルの詳細のみを取得できます。
ListChannelMembershipsForAppInstanceUser	制限付きで許可	ARN は AppInstanceUserArn 値としてのみ使用できます。
DescribeChannelMembershipForAppInstanceUser	制限付きで許可	ARN は AppInstanceUserArn 値としてのみ使用できます。
ListChannelsModeratedByAppInstanceUser	制限付きで許可	ARN は AppInstanceUserArn 値としてのみ使用できます。

API 名	許可または拒否	メモ
DescribeChannelModeratedByAppInstanceUser	制限付きで許可	AppInstanceUserArn に別のを入力することもできます AppInstanceUser。 Elastic チャンネルでは許可されません。
CreateChannelMembership	制限付きで許可	UNRESTRICTED チャンネルには他のメンバーのみを追加できます。
DescribeChannelMembership	許可	
ListChannelMemberships	許可	
DeleteChannelMembership	許可	
SendChannelMessage	許可	
GetChannelMessage	許可	
ListChannelMessage	許可	
DeleteChannelMessage	拒否	
RedactChannelMessage	制限付きで許可	自分のメッセージのみを編集できます。
UpdateChannelMessage	制限付きで許可	自分のメッセージのみを更新できます。
CreateChannelModerator	拒否	

API 名	許可または拒否	メモ
DeleteChannelModerator	拒否	
DescribeChannelModerator	拒否	
ListChannelModerator	拒否	
CreateChannelBan	拒否	
DeleteChannelBan	拒否	
DescribeChannelBan	拒否	
ListChannelBan	拒否	
UpdateChannelReadMarker	制限付きで許可	Elastic チャンネルでは許可されません。
GetChannelMessage	制限付きで許可	送信済みメッセージにのみ許可されます。メッセージの送信者でない限り、チャンネルフローで処理中のメッセージでは許可されません。
ListChannelMessages	許可	
DeleteChannelMessage	制限付きで許可	送信済みメッセージにのみ許可されます。
RedactChannelMessage	制限付きで許可	送信済みメッセージにのみ許可されます。
UpdateChannelMessage	制限付きで許可	自分の送信したメッセージのみを編集できます。
AssociateChannelFlow	拒否	

API 名	許可または拒否	メモ
DisassociateChannelFlow	拒否	
GetChannelMessageStatus	制限付きで許可	自分のメッセージのメッセージステータスのみを取得できます。
Listsubchannels	拒否	

非メンバー

非メンバーは通常のメンバーAppInstanceUserであり、[CreateChannelMembership](#)API を使用して追加しない限り、チャンネル関連のアクションを実行することはできません。

Note

AppInstanceAdmin または ChannelModerator である非メンバーは、これら 2 つのロールで許可されているチャンネル関連のアクションを実行できます。

API 名	許可または拒否	メモ
UpdateChannel	拒否	
DeleteChannel	拒否	
DescribeChannel	制限付きで許可	パブリックチャンネルの詳細のみを取得できます。
ListChannel	制限付きで許可	パブリックチャンネルの詳細のみを取得できます。
ListChannelMembershipsForAppInstanceUser	制限付きで許可	ARN は AppInstanceUserArn 値としてのみ使用できます。

API 名	許可または拒否	メモ
DescribeChannelMembershipForAppInstanceUser	制限付きで許可	AppInstanceArn に別のを入力することもできますAppInstanceUser。 Elastic チャンネルでは許可されません。
ListChannelsModeratedByAppInstanceUser	制限付きで許可	ARN は AppInstanceUserArn 値としてのみ使用できます。
DescribeChannelModeratedByAppInstanceUser	制限付きで許可	ARN は AppInstanceUserArn 値としてのみ使用できます。
CreateChannelMembership	拒否	
DescribeChannelMembership	制限付きで許可	パブリックチャンネルの詳細のみを取得できます。
ListChannelMembership	制限付きで許可	パブリックチャンネルの詳細のみを取得できます。
DeleteChannelMembership	拒否	
SendChannelMessage	拒否	
GetChannelMessage	制限付きで許可	パブリックチャンネルの詳細のみを取得できます。
ListChannelMessage	制限付きで許可	パブリックチャンネルの詳細のみを取得できます。
DeleteChannelMessage	拒否	
RedactChannelMessage	拒否	

API 名	許可または拒否	メモ
UpdateChannelMessage	拒否	
CreateChannelModerator	拒否	
DeleteChannelModerator	拒否	
DescribeChannelModerator	拒否	
ListChannelModerator	拒否	
CreateChannelBan	拒否	
DeleteChannelBan	拒否	
DescribeChannelBan	拒否	
ListChannelBan	拒否	
UpdateChannelReadMarker	拒否	
GetChannelMessage	制限付きで許可	送信済みメッセージにのみ許可されます。メッセージの送信者でない限り、チャネルフローで処理中のメッセージでは許可されません。
ListChannelMessages	制限付きで許可	
DeleteChannelMessage	拒否	拒否
RedactChannelMessage	拒否	
UpdateChannelMessage	拒否	
AssociateChannelFlow	拒否	

API 名	許可または拒否	メモ
DisassociateChannelFlow	拒否	
GetChannelMessageStatus	制限付きで許可	自分のメッセージのメッセージステータスのみを取得できます。

メッセージングデータのストリーミング

メッセージおよびチャネルイベントなどのデータをストリーム形式で受信するように AppInstance を設定できます。その後、そのデータにリアルタイムで対応することができます。現在、Amazon Chime SDK メッセージングは Kinesis ストリームのみをストリームの送信先として受け入れます。この機能で Kinesis ストリームを使用するには、以下の前提条件を満たす必要があります。

- Kinesis ストリームは、と同じ AWS アカウントに存在する必要があります AppInstance。
- ストリームは AppInstance と同じリージョンに存在する必要があります。
- ストリーム名には `chime-messaging-` で始まるプレフィックスが付きます。
- 少なくとも 2 つのシャードを設定する必要があります。各シャードは 1 秒あたり最大 1 MB のデータを受信できるため、それに応じてストリームをスケールしてください。
- サーバー側の暗号化 (SSE) を有効にします。

Kinesis ストリームを設定するには

1. 前のセクションの前提条件を使用して 1 つ以上の Kinesis ストリームを作成し、ARN を取得します。Amazon Chime のアクセス許可に加えて、発信者に Kinesis アクセス許可があることを確認します。

次の例は、AWS CLI を使用して 2 つのシャードを持つ Kinesis ストリームを作成する方法と、SSE を有効にする方法を示しています。

```
aws kinesis create-stream --stream-name chime-messaging-unique-name --shard-count 2
```

```
aws kinesis start-stream-encryption --stream-name chime-messaging-unique-name --encryption-type KMS --key-id "alias/aws/kinesis"
```

2. [PutMessagingStreamingConfigurations](#) API を呼び出してストリーミングを設定します。

2つのデータタイプの一方または両方を設定でき、同じストリームまたは別々のストリームを選択できます。

次の例は、AWS CLI を使用して および ChannelMessage Channel データ型をストリーミング appinstance するように を設定する方法を示しています。

```
aws chime-sdk-messaging put-messaging-streaming-configurations --app-instance-arn app_instance_arn \
--streaming-configurations
  DataType=ChannelMessage,ResourceArn=kinesis_data_stream_arn
```

```
aws chime-sdk-messaging put-messaging-streaming-configurations --app-instance-arn app_instance_arn \
--streaming-configurations DataType=Channel,ResourceArn=kinesis_data_stream_arn
```

データタイプには以下のスコープがあります。

DataType	生成されたイベントタイプ
ChannelMessage	CREATE_CHANNEL_MESSAGE
	REDACT_CHANNEL_MESSAGE
	UPDATE_CHANNEL_MESSAGE
	DELETE_CHANNEL_MESSAGE
Channel	CREATE_CHANNEL
	CREATE_SUB_CHANNEL

DataType	生成されたイベントタイプ
	UPDATE_CHANNEL
	DELETE_CHANNEL
	UPDATE_CHANNEL_EXPIRATION_SETTINGS
	DELETE_SUB_CHANNEL
	CREATE_CHANNEL_MEMBERSHIP
	DELETE_CHANNEL_MEMBERSHIP
	CREATE_CHANNEL_BAN
	DELETE_CHANNEL_BAN
	CREATE_CHANNEL_MODERATOR
	DELETE_CHANNEL_MODERATOR

3. 設定した Kinesis ストリームからデータの読み取りを開始します。

Note

ストリーミングを設定する前に送信されたイベントは、Kinesis ストリームには送信されません。

データ形式

Kinesis は、EventType および Payload のフィールドを含む JSON 形式のレコードを出力します。ペイロード形式は EventType によって異なります。以下の表は、イベントタイプおよびそれに対応するペイロード形式を示しています。

EventType	ペイロード形式	
CREATE_CHANNEL_MESSAGE	Channel message	
REDACT_CHANNEL_MESSAGE		
UPDATE_CHANNEL_MESSAGE		
DELETE_CHANNEL_MESSAGE		
CREATE_CHANNEL	Channel	
UPDATE_CHANNEL		
DELETE_CHANNEL		
UPDATE_CHANNEL_EXPIRATION_SETTINGS		
CREATE_CHANNEL_MEMBERSHIP	ChannelMembership	
DELETE_CHANNEL_MEMBERSHIP		
CREATE_CHANNEL_BAN	ChannelBan	
DELETE_CHANNEL_BAN		

EventType	ペイロード形式
CREATE_CHANNEL_MODERATOR	ChannelModerator
DELETE_CHANNEL_MODERATOR	
CREATE_SUB_CHANNEL	channelARN
DELETE_SUB_CHANNEL	SubChannelId

Elastic チャンネルを使用してライブイベントをホストする

Elastic チャンネルは、最大 100 万人が参加する大規模なチャットエクスペリエンスをサポートします。代表的な用途として、スポーツイベントや政治イベントのウォッチパーティーなどがあります。Elastic チャンネルは米国東部 (バージニア北部) リージョンでのみ使用できます。

Elastic チャンネルは、共通の設定の 1 つのチャンネルと、さまざまな (つまり伸縮自在な) 数のサブチャンネルで構成されます。この設定には、サブチャンネル内のメンバー数の下限および上限も含まれています。

例えば、100 のサブチャンネルを含む Elastic チャンネルを作成し、そのサブチャンネルに対してメンバー数の下限を 500 人、上限を 10,000 人に設定したとします。このサンプルチャンネルにユーザーが参加すると、メンバー数が 10,000 人を超えるまで、システムはユーザーを 1 つのサブチャンネルに自動的に割り当てます。メンバー数が 10,000 人を超えると、システムは新しいサブチャンネルを作成し、そこに新しいメンバーを追加します。ユーザーが退室すると、システムはサブチャンネルを削除し、残りのサブチャンネルにメンバーを配分します。

視聴者をサブチャンネルに分割すると、参加者が会話をフォローしやすくなります。モデレーターは一部のサブチャンネルを見るだけで済むため、ワークロードも軽減されます。さらに、モデレーターは Elastic チャンネルが提供する組み込みツールを使用できます。例えば、モデレーターは [ユーザーによるチャンネルへのアクセスを禁止](#) したり、[モデレーターを作成](#) したり、[チャンネルフロー](#) を使用してチャンネル内のすべてのメッセージを自動的にモデレートしたりできます。

Amazon Chime SDK メッセージングクォータの詳細については、「Amazon Chime SDK General Reference」の「[Messaging Quotas](#)」を参照してください。

トピック

- [前提条件](#)
- [Elastic チャネルの概念](#)
- [サポートされる追加の機能](#)
- [Elastic チャネルの作成](#)
- [Elastic チャネルメンバーの管理](#)
- [Elastic チャネルメッセージの送信](#)
- [Elastic チャネルの WebSocket システムメッセージについて](#)
- [Kinesis ストリームを使用してシステムメッセージを受信する](#)
- [デモアプリケーションでの Elastic チャネルのテスト](#)

前提条件

Elastic チャネルを使用するには、以下が必要です。

- Amazon Chime SDK メッセージング機能 (チャネルの管理、メッセージの送受信など) に関する知識。
- Amazon Chime SDK メッセージング API を呼び出す機能。

Elastic チャネルの概念

Elastic チャネルを効果的に使用するには、これらの概念を理解する必要があります。

サブチャネル

Elastic チャネルは、メンバーをサブチャネルと呼ばれる論理コンテナに分割します。Elastic チャネルに AppInstanceUser を追加すると、ユーザーはサブチャネルのメンバーになります。そのユーザーはメッセージを送受信できますが、そのサブチャネルの他のメンバーとのみメッセージを送受信できます。システムは、あるサブチャネルからのメッセージを他のサブチャネルに表示することを許可しません。

スケーリング

ユーザーエンゲージメントをサポートするには、すべてのサブチャネルがメンバーシップの最低要件を満たしている必要があります。Elastic チャネルを作成するときに、その値を指定します。ユーザーがイベントに参加したりイベントから退出したりすると、システムはメンバーをさま

さまざまなサブチャンネルに移管します。これにより、チャンネル全体が「伸縮自在」になります。サブチャンネルは以下のスケーリングアクションを実行します。

- `SCALE_OUT` — 新しい Elastic チャンネルメンバーシップリクエストが受信され、すべてのサブチャンネルが満杯になると、システムは新しいサブチャンネルを作成し、既存のサブチャンネルから新しいサブチャンネルにメンバーシップを移管することによってスケールアウトします。
- `SCALE_IN` — サブチャンネルのメンバーシップ数が最小要件を下回り、別のサブチャンネルに最初のサブチャンネルのメンバー全員を収容できる容量がある場合、`SCALE_IN` イベントはそれらのメンバーシップを移管し、サブチャンネルおよびすべてのメッセージを削除します。

Note

削除されたチャンネルのメッセージにアクセスする必要がある場合は、まずメッセージストリーミングを有効にする必要があります。詳細については、「[メッセージングデータのストリーミング](#)」を参照してください。

メンバーの移管

メンバーの移管は、メンバーシップバランシングによって `AppInstanceUser` があるサブチャンネルから別のサブチャンネルに移動する際に発生します。`AppInstanceUser` は移管後も Elastic チャンネルに属します。ただし、新しいサブチャンネルには異なるメンバーシップおよびメッセージが含まれているため、移管後に `AppInstanceUser` によって送信されたメッセージはそれらの異なるメンバーに送信されます。メンバーシップバランシングはモデレーターメンバーシップには影響しません。

Note

Elastic チャンネルは、非表示のメンバーシップ、メンバーシップ設定、既読メッセージのタイムスタンプをサポートしていません。

サポートされる追加の機能

Elastic チャンネルはこれらのメッセージング機能もサポートしています。

- [プリフェッチ](#)
- [チャンネルフロー](#)

Elastic チャンネルの作成

[CreateChannel](#) API の `ElasticChannelConfiguration` フィールドを使用して Elastic チャンネルを作成します。Elastic チャンネルを作成したら、チャンネルメンバーシップを作成します。

Note

- 非 Elastic チャンネルの場合、チャンネルを作成した `AppInstanceUser` がメンバーおよびモデレーターとしてそのチャンネルに自動的に追加されます。Elastic チャンネルの場合、チャンネル作成者はモデレーターとしてのみ追加されます。
- いったん設定した `ElasticChannelConfiguration` を更新することはできません。
- チャンネルを Elastic から非 Elastic に、またはその逆に更新することはできません。
- [CreateChannel](#) API リクエストにメンバー ARNs のリストを含めることはできません。ただし、モデレーター ARN のリストを含めることはできます。
- UNRESTRICTED タイプの Elastic チャンネルは作成できません。

Elastic チャンネルメンバーの管理

Elastic チャンネルのメンバーを管理するに

は、[CreateChannelMembership](#)、[CreateChannelModerator](#)、および [CreateChannelBan](#) APIs を使用します。以下の情報は、これらの使用方法を説明しています。

チャンネルメンバーシップ

`CreateChannelMembership` API はサブチャンネルレベルでメンバーシップを作成します。サブチャンネルにはモデレーターおよび一般メンバーを含めることができます。

- モデレーター — モデレーターを複数のサブチャンネルに追加できます。これにより、モデレーターは自分が属する各サブチャンネルでメッセージを送信できます。サブチャンネルにモデレーターを追加するときは、`SubChannelId` を指定する必要があります。

モデレーターを新しいサブチャンネルに自動的に割り当てる場合は、[メッセージストリーミングを有効](#)にし、サブチャンネル作成イベントをリッスンして、それらのイベントに応じてモデレーターメンバーシップを作成できます。

最後に、特定のサブチャンネルまたはすべてのサブチャンネルからモデレーターを削除できます。どちらの場合も [DeleteChannelMembership](#) API を使用します。特定のサブチャンネルから

モデレーターを削除するには、`SubChannelId` を指定します。サブチャネルの ID を指定しない場合、システムはそのモデレーターをすべてのサブチャネルから削除します。最後に、[ListSubChannels](#) API を使用して、各サブチャネルとメンバーの数を一覧表示できます。

- 一般メンバー — 一般メンバーはチャネルメンバーシップの大半を占めます。一般メンバーは1つのサブチャネルにのみ追加できます。また、どのサブチャネルでメンバーシップが作成されるかはシステムが制御するため、チャネルメンバーシップを作成または削除するときに `SubChannelId` を渡すことはできません。

チャネルモデレーター

`CreateChannelModerator` API は Elastic チャネルレベルでモデレーターを作成します。モデレーターはすべてのサブチャネルのすべてのメッセージを表示できます。一般メンバーをチャネルモデレーターに昇格させると、システムはそのメンバーの既存のチャネルメンバーシップをすべて削除します。モデレーターを降格させた場合も同様です。

チャネルへのアクセス禁止

`CreateChannelBan` API は Elastic チャネルレベルでアクセス禁止を作成します。アクセスを禁止された `AppInstanceUser` はどのサブチャネルにも属することができません。メンバーをアクセス禁止にすると、システムはそのメンバーのチャネルメンバーシップをすべて削除します。

Elastic チャネルメッセージの送信

[SendChannelMessage](#) API は、サブチャネルレベルでメッセージを作成します。メッセージを送信するには、`subChannelId` が必要です。[UpdateChannelMessage](#)、および [RedactChannelMessage](#) APIs を使用してメッセージを編集および削除することもできますが、いずれの場合も `subChannelId` が必要です。

Note

メッセージ送信者がメッセージを編集または秘匿化できるのは、メッセージの送信先のサブチャネルに属している場合のみです。メンバーシップバランシングがメンバーを別のサブチャネルに移管した場合、そのメンバーは、その新しいサブチャネルで送信したメッセージのみを編集または秘匿化できます。

Elastic チャネルの WebSocket システムメッセージについて

Amazon Chime SDK は、チャネル内で発生するイベントのシステムメッセージを接続しているすべてのクライアントに送信します。以下のリストは、Elastic チャネルのシステムメッセージについて説明しています。

メッセージイベント

Elastic チャネルのイベントペイロードには `subChannelId` フィールドが含まれます。非 Elastic チャネルのペイロードは変わりません。

メンバーシップイベント

CREATE_CHANNEL_MEMBERSHIP および DELETE_CHANNEL_MEMBERSHIP イベントのペイロードに `subChannelId` フィールドが含まれるようになりました。

Elastic チャネルは BATCH_CREATE_CHANNEL_MEMBERHSIP イベントをサポートしていません。[BatchCreateChannelMembership](#) API を呼び出すと、システムは個々の CREATE_CHANNEL_MEMBERSHIP イベントを送信します。

これで、UPDATE_CHANNEL_MEMBERSHIP イベントタイプを使用してメンバーシップ情報の変更を通知できるようになりました。例えば、あるサブチャネルから別のサブチャネルへのメンバー移管中に、システムが `SubChannelId` ペイロードに新しい内容を含む UPDATE_CHANNEL_MEMBERSHIP イベントを送信して、メンバーが移管されたことを示します。

Note

システムは移管されたメンバーにのみ UPDATE_CHANNEL_MEMBERSHIP イベントを送信し、サブチャネルの他のメンバーには送信しません。このため、チャネルメンバーシップ名簿に入力 WebSockets する代わりに [ListChannelMemberships](#) API を使用することをお勧めします。詳細については、「[WebSockets を使用したメッセージの受信](#)」を参照してください。

Kinesis ストリームを使用してシステムメッセージを受信する

データをストリーム形式で受信するように AppInstance を設定できます。例えば、ストリームにはメッセージ、サブチャネルイベント、およびチャネルイベントを含めることができます。

その一環として、CREATE_SUB_CHANNEL および DELETE_SUB_CHANNEL イベントもサポートしています。メンバーシップバランシングの一環として、サブチャネルがいつ作成または削除されたかを

示します。データストリームの受信について詳しくは、「[メッセージングデータのストリーミング](#)」を参照してください。

デモアプリケーションでの Elastic チャネルのテスト

<https://github.com/aws-samples/amazon-chime-sdk/tree/main/apps/chat> GitHub で、すべての Amazon Chime SDK メッセージング機能をテストできます。

モバイルプッシュ通知を使用してメッセージを受信する

チャネルメッセージをモバイルプッシュ通知チャネルに送信するように Amazon Chime SDK Messaging を設定できます。Amazon Chime SDK には、プッシュ通知用に設定された Amazon Pinpoint アプリケーションが必要です。Amazon Pinpoint アプリケーションは以下の前提条件を満たす必要があります。

- Amazon Pinpoint アプリケーションには、少なくとも FCM または APNS チャネルが設定され、有効になっている必要があります。
- Amazon Pinpoint アプリケーションは、Amazon Chime SDK アプリケーションインスタンスと同じ AWS アカウントとリージョンに存在する必要があります。

Note

デフォルトでは、メッセージ送信者を含むプッシュ通知チャネルのすべてのメンバーがプッシュ通知を受信します。ただし、メッセージが送信者に送信されないようにするフィルタールールを設定できます。詳細については、このセクションの後半の「[フィルタールールを使用してメッセージをフィルタリングする](#)」を参照してください。

トピック

- [Amazon Pinpoint アプリケーションを作成する](#)
- [サービスロールの作成](#)
- [モバイルデバイスのエンドポイントをアプリケーションインスタンスユーザーとして登録する](#)
- [通知を有効にした状態でチャネルメッセージを送信する](#)
- [プッシュ通知の受信](#)
- [プッシュ通知エラーのデバッグ](#)

- [フィルタルールを使用してメッセージをフィルタリングする](#)

Amazon Pinpoint アプリケーションを作成する

プッシュ通知を送信するには、Amazon Chime SDK で、モバイルアプリケーションにプッシュを送信するように設定された Amazon Pinpoint アプリケーションが必要です。次の手順では、AWS コンソールを使用して Pinpoint アプリケーションを作成する方法について説明します。

Amazon Pinpoint アプリケーションを作成するには

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

Amazon Pinpoint を初めて使用すると、サービスの特徴を紹介するページが表示されます。

2. [使用開始] セクションで、プロジェクトの名前を入力し、[Create a project] を選択します。
3. [機能の設定] ページで、[プッシュ通知] の横にある [設定] を選択します。
4. [プッシュ通知のセットアップ] ページで、[Apple プッシュ通知サービス (APNs)]、[Firebase クラウドメッセージング (FCM)]、またはその両方を切り替え、必須フィールドに入力します。

Important

Amazon Chime SDK は現在、APN および FCM へのプッシュ通知の送信のみをサポートしています。

5. 完了したら、[Save] (保存) を選択します。
6. Amazon Pinpoint コンソール (<https://console.aws.amazon.com/pinpoint/>) に戻り、[プロジェクト ID] の値を書き留めます。これを Amazon Pinpoint アプリケーションの ARN として使用します。

サービスロールの作成

AWS はサービスロールを使用して、サービスが AWS リソースにアクセスできるように AWS サービスにアクセス許可を付与します。サービスロールにアタッチするポリシーによって、どのリソースにサービスがアクセスできるか、およびそれらのリソースで何ができるかが決まります。Amazon Chime SDK 用に作成したサービスロールは、Amazon Pinpoint アプリケーションに対して SendMessages 呼び出しを行うためのアクセス許可をサービスに付与します。

サービスロールを作成する

1. AWS マネジメントコンソールにサインインし、<https://console.aws.amazon.com/iam/> で IAM コンソールを開きます。
2. ナビゲーションペインで、Policies を選択し、Create Policy を選択します。
3. [JSON] タブを選択し、以下のポリシーをテキストボックスにコピーします。を前のステップで作成した Amazon Pinpoint アプリケーションの ID `project_id`に置き換え、を AWS アカウント ID `aws_account_id` に置き換えてください。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Action": "mobiletargeting:SendMessages",
    "Resource": "arn:aws:mobiletargeting:region:aws_account_id:apps/project_id/messages",
    "Effect": "Allow"
  }
}
```

4. [次へ: タグ] を選択します。
5. [次へ: 確認] を選択し、[名前] フィールドに **AmazonChimePushNotificationPolicy** を入力して [ポリシーの作成] を選択します。
6. ナビゲーションペインで **ロール** を選択してから、**ロールを作成する** を選択します。
7. [ロールの作成] ページで [AWS サービス] を選択し、[ユーザーケースの選択] リストを開いて [EC2] を選択します。
8. [次へ: アクセス許可] を選択し、検索ボックスに **AmazonChimePushNotificationPolicy** を入力して、ポリシーの横にあるチェックボックスをオンにします。
9. [次へ: タグ] を選択します。
10. [次へ: 確認] を選択し、[名前] フィールドに **ServiceRoleForAmazonChimePushNotification** を入力します。

Important

上記の名前を使用する必要があります。Amazon Chime SDK は、その特定の名前だけを受け入れます。

11. [ロールの作成] を選択し、[ロール] ページで検索ボックスに **ServiceRoleForAmazonChimePushNotification** を入力して、一致するロールを選択します。
12. [信頼関係] タブを選択し、[信頼関係の編集] を選択して、既存のポリシーを以下のポリシーに置き換えます。

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "messaging.chime.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

13. 信頼ポリシーの更新 を選択します。

Important

名前、アクセス許可ポリシー、または信頼ポリシーを変更してロールを変更すると、プッシュ通知機能が無効になる可能性があります。

モバイルデバイスのエンドポイントをアプリケーションインスタンスユーザーとして登録する

プッシュ通知を受信するには、アプリケーションインスタンスのユーザーはまず [RegisterAppInstanceUserEndpoint](#) API を使用してモバイルデバイスを登録する必要があります。デバイスのオペレーティングシステムのデバイストークンにアクセスできるモバイルアプリケーションから登録する必要があります。

アプリケーションインスタンスユーザーが ARN にリストされている Amazon Pinpoint アプリケーションにアクセスできるようにするには、ユーザーに Amazon Pinpoint ARN で `mobiletargeting:GetApp` を呼び出すアクセス許可が必要です。それ以外

の場合、Amazon Chime SDK を呼び出すときに 403 Forbidden エラーをスローします [RegisterAppInstanceUserEndpoint](#)。

この例は、エンドポイントを登録するのに必要なポリシーを示しています。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PermissionToRegisterEndpoint",
      "Effect": "Allow",
      "Action": "chime:RegisterAppInstanceUserEndpoint",
      "Resource": "arn:aws:chime:region:aws_account_id:app-
instance/app_instance_id/user/app_instance_user_id"
    },
    {
      "Sid": "PermissionToGetAppOnPinpoint",
      "Effect": "Allow",
      "Action": "mobiletargeting:GetApp",
      "Resource": "arn:aws:mobiletargeting:region:aws_account_id:apps/project_id"
    }
  ]
}
```

エンドポイントを登録するには

- Amazon Pinpoint ARN とデバイストークンを使用して [RegisterAppInstanceUserEndpoint](#) API を呼び出します。

通知を有効にした状態でチャネルメッセージを送信する

[SendChannelMessage](#) API には、Amazon Chime SDK が Amazon Pinpoint に送信するプッシュ通知を構築するために使用するオプションの `PushNotification` 属性があります。現在、Amazon Chime SDK は通知タイトルおよび本文フィールドのみをサポートしています。

Amazon Chime SDK は APN の VoIP プッシュもサポートしています。プッシュ通知を APN VoIP プッシュとして送信するには、`PushNotification` 属性のタイプを `VOIP` に設定します。

プッシュ通知の受信

Amazon Chime SDK には、チャンネルメッセージのプッシュ通知のタイトルおよび本文に加えて、チャンネルメッセージ ID およびチャンネル ARN もデータペイロードに含まれます。この情報を使用して、チャンネルメッセージ全体をロードします。

以下の例は、標準的なプッシュ通知ペイロードを示しています。

```
{
  "pinpoint.openApp=true",
  "pinpoint.notification.title=PushNotificationTitle",
  "pinpoint.notification.body=PushNotificationBody",
  "pinpoint.campaign.campaign_id=_DIRECT",
  "pinpoint.notification.silentPush=0",
  "pinpoint.jsonBody"={
    "chime.message_id": "ChannelMessageId",
    "chime.channel_arn": "ChannelARN"
  }
}
```

プッシュ通知受信の無効化またはフィルタリング

Amazon Chime SDK には、アプリケーションインスタンスユーザーがプッシュ通知を受け取るかどうかを制御できる複数のオプションが用意されています。

すべてのプッシュ通知を無効にする

アプリケーションインスタンスのユーザーは、 を呼び出し [UpdateAppInstanceUserEndpoint](#) で AllowMessages 属性を に設定することで、プッシュ通知を完全に無効にできますNONE。

チャンネルのプッシュ通知を無効にする

アプリケーションインスタンスのユーザーは、 PushNotification 設定 フィールドNONEで [PutChannelMembershipPreferences](#) を呼び出すことで、特定のチャンネルのプッシュ通知を無効にできます。

チャンネルのプッシュ通知をフィルタリングする

App Instance ユーザーは、 [PutChannelMembershipPreferences](#) API を使用して特定のプッシュ通知のみを受信するようにフィルタールールを設定できます。詳細については、「[フィルタールールを使用してメッセージをフィルタリングする](#)」を参照してください。

プッシュ通知エラーのデバッグ

Amazon Chime SDK は Amazon と統合 EventBridge され、プッシュメッセージ配信の失敗を通知します。障害をさらにデバッグするには、Amazon Pinpoint が障害に対して送信する [CloudWatch メトリクス](#) を調べることもできます。

以下の表に、配信エラーメッセージの一覧および説明を示します。

メッセージ	説明
リクエストの処理が、不明なエラー、例外、または障害により実行できませんでした。	内部エラーが発生しました。もう一度試してください。
指定されたリソースが見つかりませんでした。AppInstanceUserEndpoint は非アクティブ化されます。	Amazon Pinpoint アプリケーションは存在しません。
Amazon Pinpoint に送信されたリクエストが多すぎます。	Amazon Pinpoint は送信メッセージをスロットリングしました。
メッセージを送信できません。で IAM アクセス許可ポリシーを確認してください ServiceRoleForAmazonChimePushNotification。	Amazon Chime SDK 用に作成されたロールには、mobiletargeting:SendMessage を呼び出すアクセス許可がありません。ロールの IAM ポリシーを確認してください。
メッセージを送信できません。で IAM 信頼関係を確認してください ServiceRoleForAmazonChimePushNotification。	Amazon Chime SDK には、プッシュ通知用のロールへのアクセス許可がありません。 IAM ロールの信頼ポリシーにサービスプリンシパル (messaging.chime.amazonaws.com) が含まれていることを確認してください。

フィルタルールを使用してメッセージをフィルタリングする

Amazon Chime SDK を使用すると、アプリケーションインスタンスユーザーのチャンネルメンバーシップにフィルタルールを設定して、受信メッセージを制限できます。フィルタルールはチャンネルメンバーシップに設定され、メッセージ属性マップに対して実行されます。メッセージ属性マップで

は、文字列キーから文字列値にマッピングさせる必要があります。フィルタルールで文字列を正確にマッチングさせることで、包含と除外を行えます。

Important

- Amazon Chime SDK のフィルタルールは、エスケープした JSON 文字列にのみ対応しています。
- 通知チャンネルのすべてのメンバーは、メッセージ送信者を含むプッシュ通知を受け取ります。これを防ぐには、以下の最初のルール例を参照してください。

チャンネルメンバーシップにフィルタルールを設定するには、[PutChannelMembershipPreferences](#) API を使用します。[SendChannelMessage](#) API コールの一部として、チャンネルメッセージにメッセージ属性を含めることができます。

トピック

- [フィルタルールタイプ](#)
- [フィルタルールの制限](#)
- [フィルタルール例](#)

フィルタルールタイプ

Amazon Chime SDK が対応しているフィルタルールタイプを次に示します。

- 包括的完全一致文字列マッチング
- 排他的完全一致文字列マッチング
- AND または OR を使用する複数のフィルタルール

フィルタルールの制限

Amazon Chime SDK のフィルタルールには、次のような制限があります。

- 文字列の完全一致にのみ対応しています。
- フィルタルールの合計サイズは 2 KB です。
- メッセージ属性の合計サイズは 1 KB です。

- OR フィルタルール内には、最大 5 つの制限を個別に設定できます。
- フィルタルール全体の最大複雑度は 20 です。複雑度とは、フィルタルール内のキーおよび値の数を合計したものです。

例えば、次のフィルタルールの複雑度は 4 です。

```
"FilterRule": "{\\"type\\": [\\"anything-but\\": [\\"Room\\"]],\\"mention\\": [\\"Bob\\"]}
```

この値は、次のように計算します。

```
Keys = "type" and "mention" - Complexity 2
Values = "Room" and "Bob" - Complexity 2

Total complexity = 4
```

フィルタルールの例

次の例は、チャンネルメンバーシップ設定とフィルタルールの使用方法を示しています。

送信者へのメッセージ送信の防止

このフィルタルールは、メッセージ送信者を除くすべてのチャンネルメンバーにメッセージを送信します。

```
{
  "Preferences": {
    "PushNotifications": {
      "FilterRule": "{\\"type\\": [\\"anything-but\\": [\\"USER_ARN\\"]]}",
      "AllowNotifications": "FILTERED"
    }
  }
}
```

上記の設定を持つアプリケーションインスタンスユーザーは、次の属性を含むチャンネルメッセージを受け取ります。

```
"MessageAttributes": {
  "senderId": {
    "StringValues": [\\"USER_ARN\\"]
```

```
}  
}
```

包括的文字列マッチング

このフィルタルールでは、メッセージ属性キーが「mention」で値が「Bob」のメッセージをすべて許可します。

```
{  
  "Preferences": {  
    "PushNotifications": {  
      "FilterRule": "{\"mention\": [\"Bob\"]}",  
      "AllowNotifications": "FILTERED"  
    }  
  }  
}
```

上記の設定を持つアプリケーションインスタンスユーザーは、次のメッセージ属性を含むチャンネルメッセージを受信します。

```
"MessageAttributes": {  
  "mention": {  
    "StringValues": ["Bob", "Alice"]  
  }  
}
```

ただし、このアプリインスタンスユーザーは、次の属性を持つチャンネルメッセージは受信しません。

```
"MessageAttributes": {  
  "mention": {  
    "StringValues": ["Tom"]  
  }  
}
```

排他的文字列マッチング

このフィルタルールでは、属性キー「type」と「Room」という値を持つメッセージ以外のメッセージをすべて許可します。

```
{
```

```
"Preferences": {
  "PushNotifications": {
    "FilterRule": "{\"type\": [{\"anything-but\": [\"Room\"]}]}",
    "AllowNotifications": "FILTERED"
  }
}
```

これらが設定されたアプリインスタンスユーザーは、次のメッセージ属性を持つチャンネルメッセージを受信します。

```
"MessageAttributes": {
  "type": {
    "StringValues": ["Conversation"]
  }
}
```

ただし、このアプリインスタンスユーザーには、次の属性を持つチャンネルメッセージは表示されません。

```
"MessageAttributes": {
  "type": {
    "StringValues": ["Room"]
  }
}
```

AND ロジックを使用したマルチフィルタルール

フィルタルールと AND ロジックを組み合わせる場合、メッセージは、適用されるフィルタ条件をすべて満たしている必要があります。

```
{
  "Preferences": {
    "PushNotifications": {
      "FilterRule": "{\"type\": [{\"anything-but\": [\"Room\"]}], \"mention\": [\"Bob\"]}",
      "AllowNotifications": "FILTERED"
    }
  }
}
```


上記が設定されたアプリインスタンスユーザーは、次のメッセージ属性を持つチャンネルメッセージを受信します。

```
"MessageAttributes": {
  "mention": {
    "StringValues": ["Bob"]
  },
  "type": {
    "StringValues": ["Conversation"]
  }
}
```

OR ロジックを使用したマルチフィルタールール

フィルタールールと OR ロジックを組み合わせるには、\$or を使用します。OR ロジックを使用する場合、メッセージは、適用されるフィルタ条件の 1 つを満たす必要があります。

```
{
  "Preferences": {
    "PushNotifications": {
      "FilterRule": "{\"$or\": [{\"mention\": [\"Bob\"]}, {\"type\": [\"anything-but-\\\": [\"Room\"]}]}\"",
      "AllowNotifications": "FILTERED"
    }
  }
}
```

上記が設定されたアプリインスタンスユーザーは、次のメッセージ属性を持つチャンネルメッセージを受信します。

```
"MessageAttributes": {
  "mention": {
    "StringValues": ["Bob"]
  }
}
```

上記が設定されたアプリインスタンスユーザーは、次のメッセージ属性を持つチャンネルメッセージを受信します。

```
"MessageAttributes": {
```

```
"type": {
  "StringValues": ["Conversation"]
}
}
```

サービスリンクロールの使用

Amazon Chime SDK は AWS Identity and Access Management、(IAM) [サービスにリンクされたロール](#) を使用します。サービスにリンクされたロールは、Amazon Chime SDK に直接リンクされた一意のタイプの IAM ロールです。Amazon Chime SDK は、サービスにリンクされたロールを事前に定義し、ユーザーに代わってサービスから他の AWS のサービスを呼び出すために必要なすべてのアクセス許可を含めます。

サービスにリンクされたロールでは、必要な許可を手動で追加する必要がないので、Amazon Chime SDK を効率的に設定できます。サービスにリンクされたロールの許可は Amazon Chime SDK が定義し、別段の定義がない限り、Amazon Chime SDK のみがそのロールを引き受けることができます。定義された許可には、信頼ポリシーとアクセス許可ポリシーが含まれます。アクセス許可ポリシーを他の IAM エンティティにアタッチすることはできません。

サービスリンクロールを削除するには、まずその関連リソースを削除します。これは、リソースにアクセスするための許可を誤って削除できないため、Amazon Chime SDK リソースを保護します。

サービスにリンクされたロールをサポートするその他のサービスの詳細については、「[IAM と連携するAWS のサービス](#)」を参照してください。サービスにリンクされたロール列が「はい」になっているサービスを見つけます。そのサービスに関するドキュメントを表示するには、[はい] リンクを選択します。

トピック

- [データストリーミングでのサービスにリンクされたロールの使用](#)

データストリーミングでのサービスにリンクされたロールの使用

以下のセクションでは、データストリーミングでサービスにリンクされたロールを管理する方法について説明します。

このセクションのトピック

- [サービスにリンクされたロールのアクセス許可](#)

- [サービスにリンクされたロールの作成](#)
- [サービスにリンクされたロールの編集](#)
- [サービスにリンクされたロールによって使用されるリソースの削除](#)
- [サービスにリンクされたロールの削除](#)

サービスにリンクされたロールのアクセス許可

Amazon Chime SDK は、 という名前のサービスにリンクされたロールを使用します `AWSServiceRoleForChimeSDKMessaging`。このロールは、データストリーミングに使用される Kinesis ストリームなど、Amazon Chime SDK が使用または管理する AWS サービスとリソースへのアクセスを許可します。

`AWSServiceRoleForChimeSDKMessaging` サービスにリンクされたロールは、以下のサービスを信頼して、それらのサービスがロールを引き受けられるようにします。

- `messaging.chime.amazonaws.com`

ロールアクセス許可ポリシーは、指定したリソースに対して以下のアクションを実行することを Amazon Chime SDK に許可します。

- `kms:GenerateDataKey` は、`kinesis.*.amazonaws.com` を使用してリクエストが行われた場合のみ。
- `kinesis:PutRecord`、`kinesis:PutRecords`、または `kinesis:DescribeStream` は、以下の形式のストリームのみ: `arn:aws:kinesis:*:*:stream/chime-messaging-*`。

次の例はポリシーを示しています。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
```

```
    "kms:ViaService": [
      "kinesis.*.amazonaws.com"
    ]
  }
},
{
  "Effect": "Allow",
  "Action": [
    "kinesis:PutRecord",
    "kinesis:PutRecords",
    "kinesis:DescribeStream"
  ],
  "Resource": [
    "arn:aws:kinesis:*:*:stream/chime-messaging-*"
  ]
}
```

サービスにリンクされたロールの作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) に許可するには、アクセス許可を設定する必要があります。詳細については、「IAM ユーザーガイド」の「[サービスにリンクされたロールのアクセス許可](#)」を参照してください。

サービスにリンクされたロールの作成

サービスリンクロールを手動で作成する必要はありません。[PutMessagingStreamingConfigurations](#) API を使用してデータストリーミング設定を作成すると、Amazon Chime SDK によってサービスにリンクされたロールが作成されます。

IAM コンソールを使用して、Amazon Chime SDK ユースケースでサービスにリンクされたロールを作成することもできます。AWS CLI または AWS API で、サービス名を使用して `messaging.chime.amazonaws.com` サービスにリンクされたロールを作成します。詳細については、「IAM ユーザーガイド」の「[サービスにリンクされたロールの作成](#)」を参照してください。このロールを削除しても、このプロセスを繰り返して再度作成することができます。

サービスにリンクされたロールの編集

サービスにリンクされたロールの作成後は、その説明のみ編集できます。編集は IAM を使用して行います。詳細については、「IAM ユーザーガイド」の「[サービスにリンクされたロールの編集](#)」を参照してください。

サービスにリンクされたロールによって使用されるリソースの削除

IAM を使用してサービスリンクロールを削除するには、最初に、そのロールで使用されているリソースをすべて削除する必要があります。

Note

Amazon Chime SDK がリソースを使用しているときにリソースを削除しようとする、削除が失敗することがあります。削除が失敗した場合は、数分待ってから操作を再試行してください。

AmazonChimeServiceChatStreamingAccess ロールが使用するリソースを削除するには

次の CLI コマンドを実行して、アプリケーションインスタンスのデータストリーミングをオフにします。

- `aws chime-sdk-messaging delete-messaging-streaming-configurations --app-instance-arn app_instance_arn`

このアクションにより、アプリインスタンスのストリーミング設定がすべて削除されます。

サービスにリンクされたロールの削除

サービスにリンクされたロールが必要な機能またはサービスが不要になった場合は、そのロールを削除することをお勧めします。そうしないと、アクティブにモニタリングもメンテナンスもされない不使用のエンティティが存在することになります。ただし、ロールを手動で削除する前に、サービスにリンクされたロールによって使用されるリソースを削除する必要があります。

IAM コンソール、または AWS API を使用して AWS

CLI、AmazonChimeServiceRoleForChimeSDKMessaging サービスにリンクされたロールを削除できます。詳細については、「IAM ユーザーガイド」の「[サービスにリンクされたロールの削除](#)」を参照してください。

チャネルフローを使用したメッセージの処理

チャネルフローを使用すると、送信中のメッセージがメッセージングチャネルの受信者に配信される前に、そのメッセージに対してビジネスロジックを実行できます。チャネルフローでは、行政 ID 番

号、電話番号、または冒涇的な表現をメッセージから削除するなどのアクションを実行できます。投票アンケートへの回答を集約した後で結果を参加者に返送するなどの機能を実行するためにチャンネルフローを使用することもできます。

前提条件

- Amazon Chime SDK の基本機能 (チャンネルの管理、メッセージの送受信など) に関する知識。
- Amazon Chime SDK メッセージング API を呼び出す機能。

チャンネルフローの概念

チャンネルフローを効果的に使用するには、以下の概念を理解する必要があります。

チャンネルプロセッサ

チャンネルメッセージで前処理ロジックを実行する AWS Lambda 関数。チャンネルをチャンネルフローに関連付けると、フロー内のプロセッサがチャンネル内のメッセージごとに呼び出されます。レイテンシーを低減するため、ほとんどのユースケースではシングルプロセッサが最適です。最後に、処理が完了すると、各プロセッサは Amazon Chime SDK サービスにコールバックする必要があります。

Note

現在、チャンネルフローごとにサポートされるプロセッサは 1 つだけです。複数のプロセッサが必要な場合は、サポートチケットを送信してプロセッサを増やしてください。

チャンネルフロー

チャンネルフローは、最大 3 つのチャンネルプロセッサと実行シーケンスを格納するコンテナです。フローをチャンネルに関連付けると、プロセッサはそのチャンネルに送信されるすべてのメッセージを処理します。

チャンネルフローの呼び出し

以下のアイテムによってチャンネルフローが呼び出されます。

- 新しい永続的な標準メッセージ
- 新しい非永続的な標準メッセージ

- 更新された永続的な標準メッセージ

Note

チャンネルフローはコントロールメッセージやシステムメッセージを処理しません。Amazon Chime SDK メッセージングによって提供されるメッセージタイプの詳細については、「[メッセージタイプ](#)」を参照してください。

トピック

- [チャンネルプロセッサの設定](#)
- [チャンネルフローの作成](#)
- [チャンネルフローの関連付けと関連付け解除](#)
- [メッセージの送信](#)
- [による自動化による障害アラートの作成 EventBridge](#)

チャンネルプロセッサの設定

チャンネルフローの使用を開始するには、まず、ユースケースの前処理を行うプロセッサ Lambda 関数を作成します。例えば、メッセージの内容やメタデータを更新したり、メッセージを拒否して送信されないようにしたり、元のメッセージを通過させたりすることができます。

前提条件

- Lambda 関数は、と同じ AWS アカウントと同じ AWS リージョンに存在する必要があります ApplInstance。

呼び出しアクセス許可の付与

Amazon Chime SDK メッセージングサービスに、Lambda リソースを呼び出すためのアクセス許可を与える必要があります。アクセス許可の詳細については、「[AWS Lambdaでのリソースベースのポリシーの使用](#)」を参照してください。例:

プリンシパル: 「messaging.chime.amazonaws.com」

アクション : Lambda:InvokeFunction

[効果]: [許可]

AWS:SourceAccount: # *AWS AccountId*

AWS:SourceArn: "arn:aws:chime:*region*:*AWS AccountId*: *appInstance*/"

Note

特定のアプリケーションインスタンス ID を指定してプロセッサを呼び出すことも、ワイルドカードを使用してアカウント内のすべての Amazon Chime SDK アプリケーションインスタンスにプロセッサの呼び出しを許可することもできます。

コールバックのアクセス許可の付与

プロセッサ Lambda 関数が ChannelFlowCallback API を呼び出せるようにする必要もあります。その方法については、「AWS Lambda 開発者ガイド」の「[AWS Lambda 実行ロール](#)」を参照してください。

Lambda 関数の実行ロールにインラインポリシーを追加できます。この例では、ChannelFlowCallback API の呼び出しをプロセッサに許可します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "chime:ChannelFlowCallback"
      ],
      "Resource": [
        "arn:aws:chime:Region:AwsAccountId:appInstance/*"
      ]
    }
  ]
}
```

Note

Lambda 関数のベストプラクティスに従ってください。詳細については、次のトピックを参照してください。

- [「Performance Efficiency Best Practices」](#)
- [の使用に関するベストプラクティス AWS Lambda](#)
- [予約済同時実行数の設定](#)
- [非同期呼び出し](#)

プロセッサ Lambda 関数の呼び出し

ユーザーがメッセージを送信すると、次の入力リクエストによってプロセッサ Lambda 関数が呼び出されます。

```
{
  "EventType": "string"
  "CallbackId": "string"
  "ChannelMessage": {
    "MessageId": "string",
    "ChannelArn": "string",
    "Content": "string",
    "Metadata": "string",
    "Sender": {
      "Arn": "string",
      "Name": "string"
    },
    "Persistence": "string",
    "LastEditedTimestamp": "string",
    "Type": "string",
    "CreatedTimestamp": "string",
  }
}
```

EventType

プロセッサに送信されるイベント。値は CHANNEL_MESSAGE_EVENT 定数です。

CallbackId

プロセッサから ChannelFlowCallback API を呼び出す際に使用されるトークン。

ChannelMessage

ChannelArn チャンネルの ARN

Content - 処理対象のメッセージコンテンツ

CreatedTimestamp メッセージが作成された時刻

LastEditedTimestamp メッセージが編集された時刻

MessageId メッセージ識別子

Metadata - 処理対象のメッセージメタデータ

Persistence - メッセージをバックエンドで永続化するかどうかを制御するブール値。有効な値: PERSISTENT | NON_PERSISTENT

Sender - メッセージの送信者。タイプ: [identity オブジェクト](#)。

タイプ メッセージタイプ。はSTANDARDメッセージタイプChannelFlowのみをサポートします。有効な値: STANDARD

プロセッサ関数は各メッセージについて以下を決定します。

- メッセージコンテンツ、メタデータ、またはその両方を更新するかどうか
- メッセージを拒否するかどうか
- メッセージを変更しないで残すかどうか

処理が終了すると、プロセッサ Lambda 関数は結果を Amazon Chime SDK メッセージングサービスに送り返し、メッセージをすべての受信者に送信できるようにします。メッセージステータスは、プロセッサ Lambda 関数が結果を返送するまで PENDING とマークされます。プロセッサ Lambda 関数は 48 時間以内に結果を送り返します。それ以降のメッセージ配信は保証されず、[ChannelFlowCallback](#) API は Forbidden Exception のエラーメッセージをスローします。結果を返すには、ChannelFlowCallback API を呼び出します。

チャネルフローの作成

プロセッサをセットアップしたら、Amazon Chime SDK メッセージング API を使用してチャネルフローを作成します。Fallback アクションを使用して、チャネルフローがプロセッサ Lambda 関数に接続できない場合に処理を停止するか続行するかを定義できます。プロセッサのフォールバックアクションが ABORT の場合、プロセッサはメッセージステータスを FAILED に設定し、メッセージは送信しません。チャネルフローシーケンスの最後のプロセッサのフォールバックアクションが CONTINUE の場合、メッセージは処理されたと見なされ、チャネル内の受信者に送信されることに

注意してください。チャンネルフローを作成したら、それを個々のチャンネルに関連付けることができます。詳細については、[CreateChannelFlow](#) API ドキュメントを参照してください。

チャンネルフローの関連付けと関連付け解除

チャンネルをチャンネルフローに関連付けると、チャンネルフロー内のプロセッサは、チャンネルに送信されたすべてのメッセージを前処理します。チャンネルフローの関連付けおよび関連付け解除 API を呼び出すには、チャンネルモデレーターまたは管理者である必要があります。作業中は以下の事柄に注意してください。

- 1つのチャンネルには最大1つのチャンネルフローを関連付けることができます。チャンネルフローを関連付けるには、[AssociateChannelFlow](#) API を呼び出します。
- チャンネルフローの関連付けを解除し、チャンネルメッセージの前処理を停止するには、[DisassociateChannelFlow](#) API を呼び出します。

メッセージの送信

SendChannelMessage API を使用してチャンネルにメッセージを送信します。チャンネルフローに関連付けられたチャンネルには、プロセッサが以下のステータス値のいずれかを割り当てます。

メッセージのステータス	説明
SENT	メッセージは正常に処理されました。
PENDING	処理中です。
FAILED	プロセッサ Lambda 関数にアクセスできないため、処理に失敗しました。
DENIED	メッセージは送信されません。

中間ステータスイベントの受信

Websocket イベント

Websocket イベントは、接続が正常に確立された後にチャンネルに送信されます。詳細については、「[WebSockets を使用したメッセージの受信](#)」を参照してください。

イベントタイプ	ステータス	受取人	メモ
CREATE_CHANNEL_MESSAGE	SENT	チャンネルメンバー全員	SendChannelMessage 前処理が成功した API
UPDATE_CHANNEL_MESSAGE	SENT	チャンネルメンバー全員	UpdateChannelMessage 前処理が成功した API
PENDING_CREATE_CHANNEL_MESSAGE	PENDING	メッセージ送信者のみ	SendChannelMessage 前処理が進行中の API
PENDING_UPDATE_CHANNEL_MESSAGE	PENDING	メッセージ送信者のみ	UpdateChannelMessage 前処理が進行中の API
FAILED_CREATE_CHANNEL_MESSAGE	FAILED	メッセージ送信者のみ	SendChannelMessage 前処理に失敗した API
FAILED_UPDATE_CHANNEL_MESSAGE	FAILED	メッセージ送信者のみ	UpdateChannelMessage 前処理に失敗した API
DENIED_CREATE_CHANNEL_MESSAGE	DENIED	メッセージ送信者のみ	SendChannelMessage プロセッサがメッセージを拒否する API
DENIED_UPDATE_CHANNEL_MESSAGE	DENIED	メッセージ送信者のみ	UpdateChannelMessage プロセッサがメッセージを拒否する API

GetChannelMessageStatus API

この API は、Websocket 接続に問題があるためにイベントが受信されなかった場合に、メッセージステータスを取得する代替方法を提供します。詳細については、[GetChannelMessageStatus](#) API ドキュメントを参照してください。

Note

この API では、拒否されたメッセージのステータスは保存されないため、返されません。

による自動化による障害アラートの作成 EventBridge

Amazon Chime SDK は、プロセッサ Lambda 関数の呼び出しでエラーが発生した場合にイベントを配信します。イベントは、チャネルフローの作成時にプロセッサに指定された Fallback アクションに関係なく送信されます。単純なルールを記述して、これらのイベントと、イベントのいずれかがルールに一致した場合に実行する自動アクションを指定できます。詳細については、「[Amazon ユーザーガイド EventBridge](#)」を参照してください。このようなエラーが発生すると、設定した Fallback アクションによっては、チャネルのメンバーがメッセージを送信できなくなったり、メッセージが処理されずにチャネルを通過したりします。Fallback アクションの詳細については、「[Amazon Chime SDK API リファレンス Processor](#)」の「」を参照してください。

この例は、一般的な障害イベントを示しています。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-111122223333",
  "detail-type": "Chime ChannelFlow Processing Status",
  "source": "aws.chime",
  "account": "111122223333",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "region",
  "resources": [],
  "detail": {
    "eventType": "ProcessorInvocationFailure",
    "appInstanceArn": "arn:aws:chime:region:AWSAccountId:app-instance/AppInstanceId",
    "channelArn": "arn:aws:chime:region:AWSAccountId:app-instance/AppInstanceId/channel/ChannelId",
    "messageId":
      "298efac7298efac7298efac7298efac7298efac7298efac7298efac7298efac7",
    "processorResourceArn":
      "arn:aws:lambda:region:AWSAccountId:function:ChannelFlowLambda",
```

```
"failureReason": "User is not authorized to perform: lambda:InvokeFunction on resource: arn:aws:lambda:region:AppInstanceId:function:ChannelFlowLambda because no resource-based policy allows the lambda:InvokeFunction action"
    }
}
```

をインテリジェントチャネルエージェント AppInstanceBots として使用する

AppInstanceBots はインテリジェントなチャネルエージェントとして使用できます。エージェントは、チャネルメンバーによって ChannelMessages から送信されたキーフレーズを認識します。ボットの自然言語理解モデルがメッセージを解決します。これにより、1人以上のチャネルメンバーが、ボットのモデルによって定義された自然言語ダイアログに参加できるようになります。ボットを提供することで、対話の深さとエンタープライズシステムとの統合を制御できます。

前提条件

- Amazon Chime SDK の基本機能 (AppInstanceUsers の作成、チャネルの管理、メッセージの送受信など) に関する知識。
- Amazon Chime SDK メッセージング API を呼び出す機能。
- Amazon Lex V2 ボットの作成、インテントとスロットのモデリング、ボットのバージョン、エイリアスの作成、セッション状態の使用、Lambda フックの統合など、Amazon Lex V2 の基本的な機能に関する知識。

Important

Amazon Lex V2 の使用には、AWS 機械学習および人工知能サービスに固有の条件を含む [AWS サービス条件](#)が適用されます。

トピック

- [Amazon Lex V2 ボットの作成](#)
- [AppInstance ボットのセットアップ](#)
- [のチャネルメンバーシップの作成 AppInstanceBot](#)
- [へのメッセージの送信 AppInstanceBot](#)

- [Amazon Lex からのメッセージの処理](#)
- [からのレスポンスの処理 AppInstanceBot](#)
- [ルールを使用して Amazon にイベントを送信する EventBridge](#)
- [Amazon Lex V2 ボットで AppInstanceBots 設定されたトラブルシューティング V2](#)

Amazon Lex V2 ボットの作成

AppInstance ボットをエージェントとして使用するには、まず Amazon Lex V2 ボットを作成して、インテリジェントエージェントシナリオのダイアログインタラクションを管理する必要があります。Amazon Lex V2 ボットの構築を開始するには、「Amazon Lex V2 開発者ガイド」の「[Amazon Lex V2 の開始方法](#)」を参照してください。Amazon Lex V1 ボットから Amazon Lex V2 への移行については、「[Amazon Lex V1 から V2 への移行ガイド](#)」を参照してください。

トピック

- [前提条件](#)
- [呼び出しアクセス許可の付与](#)
- [挨拶のインテントの作成](#)
- [Amazon Lex V2 ボットバージョンの作成](#)
- [Amazon Lex V2 ボットのエイリアスの作成](#)

前提条件

Amazon Lex V2 ボットは以下の前提条件を満たしている必要があります。

- Amazon Lex V2 ランタイムエンドポイントをサポートする AWS リージョンでボットを作成する必要があります。
- ボットは、およびと同じ AWS アカウントとリージョンに作成する必要がありますAppInstanceAppInstanceBot。
- ボットは、リソースベースのポリシーを介して messaging.chime.amazonaws.com サービスプリンシパルに呼び出し許可を付与する必要があります。
- ボットは挨拶のインテントをモデル化できます。これにより、AppInstanceBot はチャンネルのメンバーシップの取得時に自身とその機能をアナウンスできます。
- AppInstanceBot を設定するには、ボットの製品版とエイリアスが必要です。

- ボットはサポートされている言語とロケールを使用する必要があります。言語とロケールの詳細については、「Amazon Lex V2 開発者ガイド」の「[Amazon Lex V2 でサポートされている言語とロケール](#)」を参照してください。

呼び出しアクセス許可の付与

AppInstanceBot が Amazon Lex V2 ボットを呼び出すには、Amazon Chime SDK メッセージングサービスのプリンシパルに Amazon Lex ボットリソースを呼び出すアクセス許可が付与されている必要があります。Amazon Lex V2 のリソースベースのポリシーアクセス許可の詳細については、「Amazon Lex V2 開発者ガイド」の「[Amazon Lex V2 のリソースベースのポリシーの例](#)」を参照してください。

以下に、リソースベースのポリシーの例を示します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "messaging.chime.amazonaws.com"
      },
      "Action": [
        "lex:PutSession",
        "lex>DeleteSession",
        "lex:RecognizeText"
      ],
      "Resource": "arn:aws:lex:region:aws-account-id:bot-alias/lex-bot-id/lex-bot-alias-id",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "aws-account-id"
        },
        "ArnEquals": {
          "AWS:SourceArn": "arn:aws:chime:region:aws-account-id:app-instance/app-instance-id/bot/app-instance-bot-id"
        }
      }
    }
  ]
}
```


Note

Amazon Lex V2 ボットAppInstanceBotの呼び出しを許可するには、AppInstanceBotの ID を使用します。AppInstance 内のすべての AppInstanceBots が Amazon Lex V2 ボットを呼び出せるようにするには、ワイルドカードを使用します。例:

```
arn:aws:chime:region:aws-account-id:app-instance/app-instance-id/bot/*
```

挨拶のインテントの作成

Amazon Lex V2 ボットモデルに任意の挨拶のインテントを追加すると、AppInstanceBot がチャネルに参加したときに自身とその機能をアナウンスできます。挨拶のインテントはメッセージを表示したり、チャネルメンバーとの対話を開始したりできます。ウェルカムインテントの名前はさまざまで、AppInstanceBotの設定で定義します。

インテントの詳細については、「Amazon Lex V2 開発者ガイド」の「[インテントの追加](#)」を参照してください。

Amazon Lex V2 ボットバージョンの作成

Amazon Lex V2 ボットを作成する場合、作成するのはドラフトバージョンのみです。ドラフトはボットの作業用コピーで、更新できます。デフォルトでは、ドラフトバージョンには TestBotAlias と呼ばれるエイリアスが関連付けられているため、ドラフトボットは手動テストのみ使用してください。

ダイアログのモデリングとドラフトボットの構築が完了したら、ドラフト Lex ボットの 1 つ以上のバージョン、つまり番号付きのスナップショットを作成します。バージョンを使用すると、クライアントアプリケーションで使用される実装を制御できます。例えば、開発、ベータデプロイ、本稼働など、ワークフローの異なる段階で使用するためにバージョンを発行できます。

Lex ボットのバージョンングの詳細については、「Amazon Lex V2 開発者ガイド」の「[バージョンの作成](#)」を参照してください。

Amazon Lex V2 ボットのエイリアスの作成

Amazon Lex V2 ボットのバージョンを 1 つ以上作成したら、エイリアスを作成します。エイリアスは Amazon Lex V2 ボットのバージョンへの名前付きポインタとして機能します。例えば、エイリアスには、一度に 1 つのバージョンのみを関連付けることができます。

Lex ボットのエイリアスの詳細については、「Lex V2 開発者ガイド」の「[エイリアスの作成](#)」を参照してください。

AppInstance ボットのセットアップ

モデル、バージョン、エイリアスを持つ Amazon Lex V2 ボットを作成したら、Amazon Chime SDK メッセージング APIs または CLI を使用してを作成します AppInstanceBot。 APIs [CreateAppInstanceBot](#) 「」を参照してください。

Note

InvokedBy 属性を使用して、AppInstanceBot のダイアログインタラクション動作を設定します。標準メッセージやターゲットを絞ったメッセージなど、ボットをトリガーするメッセージの種類を設定できます。

次の例は、AWS CLI を使用して MENTIONS、を含む AppInstanceBot すべての標準メッセージとターゲットメッセージが呼び出せるを作成する方法を示しています。

```
aws chime-sdk-identity create-app-instance-bot \
--app-instance-arn app-instance-arn \
--name app-instance-bot-name \
--configuration '{
  "Lex": {
    "LexBotAliasArn": "lex-bot-alias-arn",
    "LocaleId": "lex_bot_alias_locale_id",
    "InvokedBy": {
      "StandardMessages": "MENTIONS",
      "TargetedMessages": "ALL"
    }
  }
  "WelcomeIntent": "welcome-intent-name"
}
```

のチャンネルメンバーシップの作成 AppInstanceBot

を作成したら AppInstanceBot、新規または既存のチャンネルにメンバーとして追加します。詳細については、Amazon Chime SDK メッセージング API ドキュメント [CreateChannelMembership](#) の [CreateChannel](#) 「」および「」を参照してください。

次の例は、AWS CLI を使用してチャンネルを作成し、メンバーAppInstanceBotとして を追加する方法を示しています。

```
aws chime-sdk-messaging create-channel \  
--chime-bearer caller_app_instance_user_arn \  
--app-instance-arn app_instance_arn \  
--name channel_name \  
--member-arns '[  
  "app_instance_bot_arn"  
]'
```

次の例は、AWS CLI を使用して既存のチャンネルAppInstanceBotに を追加する方法を示しています。

```
aws chime-sdk-messaging create-channel-membership \  
--chime-bearer caller_app_instance_user_arn \  
--channel-arn channel_arn \  
--member-arn app_instance_bot_arn
```

へのメッセージの送信 AppInstanceBot

[SendChannelMessage](#) API を使用して にメッセージを送信します AppInstanceBot。

AppInstanceBot がメンバーであるチャンネルにメッセージを送信します。 [自然言語理解モデル](#)がメッセージの内容を認識し、Amazon Lex インテントを誘発する場合、 はチャンネルメッセージで AppInstanceBot 応答し、ダイアログを開始します。

チャンネルのメンバーにターゲットメッセージを送信することもできます。チャンネルには、AppInstanceUser または [AppInstanceBot](#) があります。ターゲットを絞ったメッセージを閲覧できるのは、ターゲットと送信者だけです。ターゲットを絞ったメッセージに対してアクションを実行できるのは、そのメッセージを表示できるユーザーのみです。ただし、管理者は表示できなくてもターゲットを絞ったメッセージを削除できます。

次の例は、AWS CLI を使用してチャンネルメッセージを送信する方法を示しています。

```
aws chime-sdk-messaging send-channel-message \  
--chime-bearer caller_app_instance_user_arn \  
--channel-arn channel_arn \  
--content content \  
--type STANDARD \  
--persistence PERSISTENT
```

Amazon Lex からのメッセージの処理

Amazon Lex にメッセージを送信すると、Amazon Chime SDK メッセージングは、チャンネルと送信者の ARN 情報をリクエスト属性として CHIME.channel.arn と CHIME.sender.arn に入力します。この属性を使用して、メッセージの送信者と、送信者が所属するチャンネルを特定できます。詳細については、「[Amazon Lex デベロッパーガイド](#)」の [AWS「Lambda 関数によるカスタムロジックの有効化」](#)を参照してください。Amazon Lex

からのレスポンスの処理 ApplInstanceBot

ユーザーがメッセージを送信すると、はチャンネルメッセージで ApplInstanceBot 応答します。チャンネルメッセージを一覧表示して、ボットのレスポンスを取得できます。

次の例は、CLI を使用してチャンネルメッセージを一覧表示する方法を示しています。

```
aws chime-sdk-messaging list-channel-messages \
--chime-bearer caller_app_instance_user_arn \
--channel-arn channel_arn
```

からの成功レスポンスは、次の形式 ApplInstanceBot になります。

```
{
  "MessageId": "messageId",
  "Content": "*{\\"Messages\\":[{\\"...\\"}]}*",
  "ContentType": "application/amz-chime-lex-msgs",
  "MessageAttributes": {
    "CHIME.LEX.sessionState.intent.name": {
      "StringValues": [
        "lex_bot_intent_name"
      ]
    },
    "CHIME.LEX.sessionState.intent.state": {
      "StringValues": [
        "lex_bot_intent_fullfilment_status"
      ]
    },
    "CHIME.LEX.sessionState.originatingRequestId": {
      "StringValues": [
        "lex_bot_originating_request_id"
      ]
    }
  },
}
```

```
    "CHIME.LEX.sessionState.sessionId": {
      "StringValues": [
        "lex_bot_session_id"
      ]
    },
    "Sender": {
      "Arn": "app_instance_bot_arn",
      "Name": "app_instance_bot_name"
    },
    "Type": "STANDARD",
  }
}
```

Content

Content フィールドには Amazon Lex V2 ボットから送信されるメッセージのリストが含まれます。これらのメッセージの詳細については、Amazon Lex V2 RecognizeText API の「[メッセージ](#)」を参照してください。

以下の例は、挨拶のメッセージで Content フィールドを使用する方法を示しています。

```
{
  "Messages":
  [
    {
      "Content": "Hello!",
      "ContentType": "PlainText"
    },
    {
      "ContentType": "ImageResponseCard",
      "ImageResponseCard":
      {
        "Title": "Hello! I'm BB, the Bank Bot.",
        "Subtitle": "I can help you with the following transactions",
        "Buttons":
        [
          {
            "Text": "Check balance",
            "Value": "Check balance"
          },
          {
            "Text": "Escalate to agent",
            "Value": "Escalate to agent"
          }
        ]
      }
    }
  ]
}
```

```
}
  ]
}
]
```

失敗レスポンスの場合、[コンテンツ] フィールドには以下の形式のエラーメッセージとコードが含まれます。

```
{
  "Code": error_code
}
```

ContentType

ContentType は Content フィールドに含まれるペイロードのタイプを指し、Content フィールドを解析するにはこれをチェックする必要があります。

Note

Lex V2 ボットは別の ContentType を使用します。

ContentType は成功レスポンスの場合は application/amz-chime-lex-msgs に、失敗レスポンスの場合は application/amz-chime-lex-error に設定されます。

MessageAttribute

MessageAttribute は、文字列キーと文字列値のマップです。からのレスポンスには、Amazon Lex ボットからのレスポンスにマッピングされた次のメッセージ属性 AppInstanceBot が含まれます。

- CHIME.LEX.sessionState.intent.name — リクエストが処理しようとした Lex ボットのインテントの名前。
- CHIME.LEX.sessionState.intent.state — インテントの現在の状態。指定できる値には、Fulfilled、InProgress、Failed などがあります。
- CHIME.LEX.sessionState .originatingRequestId – Amazon Lex ボットへの特定のリクエストの一意の識別子。これは、MessageId をトリガーした元のユーザーメッセージの に設定されず AppInstanceBot。

- CHIME.LEX.sessionState.sessionId — ユーザーとボットの間の会話を識別する一意の識別子。ユーザーがボットとのチャットを開始すると、Amazon Lex によりセッションが作成されま

す。

Amazon Lex セッションとセッション状態の詳細については、Amazon Lex API リファレンス [SessionState](#) の「」、および Amazon Lex V2 [セッションの管理](#)」を参照してください。

Amazon Lex V2 が返す属性の詳細については、「[Amazon Lex ランタイム V2](#)」API を参照してください。

ルールを使用して Amazon にイベントを送信する EventBridge

Amazon Chime SDK は、エラーにより Amazon Lex V2 ボットを呼び出せなくなったときに EventBridge イベントを配信します。これらのイベントを認識し、EventBridge ルールが一致したときに自動的にアクションを実行するルールを作成できます。詳細については、「[Amazon ユーザーガイド](#)」の「[Amazon EventBridge ルール](#)」を参照してください。EventBridge

次の例は、一般的な失敗イベントを示しています。

```
{
  version: '0',
  id: '12345678-1234-1234-1234-111122223333',
  'detail-type': 'Chime Messaging AppInstanceBot Lex Failure',
  source: 'aws.chime',
  account: 'aws-account-id',
  time: 'yyyy-mm-ddThh:mm:ssZ',
  region: "region",
  resources: [],
  detail: {
    resourceArn: 'arn:aws:chime:region:aws-account-id:app-instance/app-instance-id/bot/app-instance-bot-id',
    failureReason: "1 validation error detected: Value at 'text' failed to satisfy constraint: Member must have length less than or equal to 1024 (Service: LexRuntimeV2, Status Code: 400, Request ID: request-id)"
  }
}
```

Amazon Lex V2 ボットで AppInstanceBots 設定されたトラブルシューティング V2

以下のトピックでは、の一般的な問題のトラブルシューティング方法について説明します AppInstanceBots。

Amazon Lex V2 の障害の検出

Amazon Chime SDK メッセージングは、エラーにより [Amazon Lex V2 ボットを呼び出せない場合に Amazon EventBridge イベント](#) を配信します。Amazon Lex V2 ルールの設定と通知ターゲットの設定の詳細については、「[Amazon ユーザーガイド](#)」の「[Amazon の開始 EventBridge 方法 EventBridge](#)」を参照してください。

AWS CloudWatch Logs で EventBridge イベントを受信した場合、Logs Insights CloudWatch を使用して AWS、Amazon Chime SDK メッセージングの詳細タイプに基づいて EventBridge イベントをクエリできます。failureReason には障害の原因が一覧表示されます。

以下に、一般的なクエリの例を示します。

```
fields @timestamp, @message
| filter `detail-type` = "Chime Messaging AppInstanceBot Lex Failure"
| sort @timestamp desc
```

Amazon Chime SDK Messaging が Amazon Lex V2 ボットを呼び出すことができる場合、SDK はエラーメッセージCONTROLを含むメッセージを送信します。

Amazon Lex V2 ボットのアクセス許可エラーのトラブルシューティング

AppInstanceBot が Amazon Lex V2 ボットを呼び出すには、Amazon Chime SDK メッセージングサービスプリンシパルに Amazon Lex V2 ボットリソースを呼び出すアクセス許可が必要です。また、リソースポリシー条件AWS:SourceArnの が の ARN と一致することを確認します AppInstanceBot。

Amazon Lex V2 ボットを呼び出す AppInstanceBot ように を設定する方法の詳細については[Amazon Lex V2 ボットの作成](#)、このセクションの前半にある「」を参照してください。

Amazon Lex V2 ボットスロットリングのトラブルシューティング

Amazon Lex には、ボットエイリアスごとにテキストモードでの同時会話の最大数に関するサービスクォータがあります。クォータの増量については、Amazon Lex サービスチームにお問い合わせくだ

さい。詳細については、[Amazon Lexデベロッパーガイド](#)の「[Amazon Lex ガイドラインとクォータ](#)」を参照してください。Amazon Lex

メッセージ保持の管理

アカウントオーナーは Amazon Chime SDK API を使用してメッセージング保持を有効にできます。メッセージは、管理者が設定した期間に基づいて自動的に削除されます。保持期間は 1 日から 15 年です。API を使用して、いつでもメッセージ保持期間を更新したり、メッセージ保持をオフにしたりすることもできます。

このセクションのトピック

- [CLI 保持コマンドの例](#)
- [メッセージ保持を有効にする](#)
- [メッセージの復元と削除](#)

CLI 保持コマンドの例

以下の例は、保持の標準的な CLI コマンドを示しています。

有効化

```
aws chime-sdk-identity put-app-instance-retention-settings --app-instance-arn {appInstanceArn} --app-instance-retention-settings ChannelRetentionSettings={RetentionDays=60}
```

[更新中]

```
aws chime-sdk-identity put-app-instance-retention-settings --app-instance-arn {appInstanceArn} --app-instance-retention-settings ChannelRetentionSettings={RetentionDays=30}
```

無効化

```
aws chime-sdk-identity put-app-instance-retention-settings --app-instance-arn {appInstanceArn} --app-instance-retention-settings ChannelRetentionSettings={}
```

メッセージ保持を有効にする

Amazon Chime SDK API を使用してメッセージング保持を有効にします。API を使用して、いつでもメッセージ保持期間を更新したり、メッセージ保持をオフにしたりすることもできます。メッセージング保持の設定の詳細については、「[Amazon Chime SDK API Reference](#)」を参照してください。

メッセージの復元と削除

メッセージ保持期間を設定または更新してから 30 日以内にユーザーへのメッセージを復元できます。ただし、30 日間の猶予期間が過ぎると、保持期間に該当するすべてのメッセージは完全に削除され、新しいメッセージは保持期間を過ぎると完全に削除されます。

Note

30 日間の猶予期間中に、保持ポリシーを延長するかオフにすると、新しい保持期間を過ぎていないメッセージがアカウント内のユーザーに再び表示されるようになります。

また、AppInstanceUser がチャンネルまたはメッセージを削除すると、メッセージは完全に削除されます。

メッセージングのユーザーインターフェイスコンポーネント

コンポーネントライブラリを使用すると、チャットメッセージングのユーザーインターフェイスを構築するのに必要な労力を軽減できます。詳細については、「」の「[Amazon Chime React コンポーネントライブラリ](#) GitHub」を参照してください。

クライアントライブラリとの統合

Amazon Chime SDK のメッセージング機能を使用するには、クライアントアプリケーションを以下のクライアントライブラリと統合する必要があります。

- AWS SDK – メッセージの送信とリソースの管理のための APIs が含まれています。
- JavaScript (NPM) 用の Amazon Chime SDK クライアントライブラリ — クライアントを Amazon Chime SDK メッセージングウェブソケットと統合してメッセージを受信するのに役立つ TypeScript タイプ定義を持つ JavaScript ライブラリ。

クライアントアプリケーションを Amazon Chime SDK と統合するには、クライアントライブラリ README.md の手順を参照し、デモを使用してメッセージング機能の構築方法を学んでください。

での Amazon Chime SDK メッセージングの使用 JavaScript

JavaScript を使用して Amazon Chime SDK リソースを管理し、メッセージを送信できます。詳細については、[AWS JavaScript 「SDK」](#) を参照してください。

クライアントアプリケーションでメッセージングセッションを作成して、Amazon Chime SDK メッセージングからメッセージを受信することもできます。詳細については、「[での用の JavaScript Amazon Chime SDK クライアントライブラリの使用](#)」を参照してください GitHub。

Amazon Chime SDK PSTN Audio サービスの使用

Note

このセクションでは、Chime SDK PSTN Audio サービスについて説明します。このサービスは、以前のバージョンのドキュメントや一部のブログ記事では「SIP メディアアプリケーション (SMA)」と呼ばれていました。今後、「SIP メディアアプリケーション」と言うと、Amazon Chime SDK コンソールと PSTN Audio サービスに関連付けられている AWS SDK の設定項目を指します。

このセクションでは、Amazon Chime SDK 公衆交換電話網 (PSTN) Audio サービスの使用方法について説明します。PSTN Audio サービスを使用すると、デベロッパーはサーバーレス AWS Lambda 関数の俊敏性と運用上のシンプルさを使用してカスタムテレフォニーアプリケーションを構築できます。

AWS Lambda 関数は、Amazon Chime SDK Voice Connector を使用して、音声プロンプトの再生、デジタルの収集、通話の記録、PSTN およびセッション開始プロトコル (SIP) デバイスへの通話のルーティングなど、通話の動作を制御します。以下のトピックでは、PSTN Audio サービスの概要とアーキテクチャ情報を提供します。これには、呼び出しを制御する AWS Lambda 関数を構築する方法が含まれます。

Note

このセクションのトピックでは、サービスを理解していることを前提としています AWS Lambda。の詳細については AWS Lambda、[「 の開始方法 AWS Lambda」](#)を参照してください。また、Amazon Chime SDK のこのセクションを正しく使用するには、Amazon Chime SDK 管理者が少なくとも 1 つの SIP ルールと 1 つの SIP メディアアプリケーションを作成する必要があります。これらのタスクを完了する方法の詳細については、「Amazon Chime SDK 管理者ガイド」の [「Managing SIP media applications」](#)を参照してください。

トピック

- [Amazon Chime SDK Voice 名前空間への移行](#)
- [電話番号、SIP ルール、SIP メディアアプリケーション、および AWS Lambda 機能について](#)
- [PSTN Audio サービスのプログラミングモデルについて](#)

- [AWS Lambda 関数への呼び出しとイベントのルーティング](#)
- [PSTN Audio サービスの通話レグの使用について](#)
- [通話フローの例](#)
- [PSTN Audio サービスの AWS Lambda 関数の構築](#)

Amazon Chime SDK Voice 名前空間への移行

[Amazon Chime SDK Voice](#) 名前空間は、Amazon Chime SDK 音声リソースを作成および管理する API のための専用の場所です。名前空間を使用して、Amazon Chime SDK 音声 API エンドポイントを使用可能にする任意の AWS リージョンのエンドポイントに対処します。Amazon Chime SDK を使い始めたばかりの場合は、この名前空間を使用してください。リージョンの詳細については、このガイドの「[利用できるリージョン](#)」を参照してください。

[Amazon Chime](#) 名前空間を使用する既存のアプリケーションでは、更新された API と新機能を使用するために、専用の名前空間への移行を計画する必要があります。

トピック

- [移行すべき理由](#)
- [移行する前に](#)
- [名前空間の相違点](#)

移行すべき理由

[Amazon Chime SDK Voice](#) 名前空間への移行をお勧めするのは、以下の理由からです。

API エンドポイントの選択

Amazon Chime SDK Voice 名前空間により、[API エンドポイントを利用可能にする任意のリージョン](#)で API エンドポイントを使用できるようになります。us-east-1 以外の API エンドポイントを使用する場合は、Amazon Chime SDK Voice 名前空間を使用する必要があります。現在のエンドポイントの詳細については、このガイドの「[API マッピング](#)」を参照してください。

音声 API の更新および新規追加

Amazon Chime SDK Voice 名前空間の音声 API のみ追加または更新されます。

移行する前に

移行する前に、名前空間の相違点に注意してください。以下の表では、名前空間の一覧と説明を示しています。

	Amazon Chime SDK Voice 名前空間	Amazon Chime 名前空間
AWS 名前空間	ChimeSDKVoice	Chime
リージョン	複数	us-east-1 のみ
エンドポイント	https://voice-chime. <i>region</i> .amazonaws.com	service.chime.aws.amazon.com
サービスプリンシパル	chime.amazonaws.com	chime.amazonaws.com
API	PSTN Audio サービス用の API のみ	PSTN Audio および Amazon Chime のその他の部分用の API
音声コネクタ管理	複数のリージョン	us-east-1
音声コネクタグループ管理	複数のリージョン	us-east-1
SIP メディアアプリケーションおよび SIP ルール管理	複数のリージョン	us-east-1
電話番号管理	複数のリージョン	us-east-1
通話分析	利用可能	利用不可
音声プロファイルドメイン	利用可能	利用不可
緊急通報管理	複数のリージョン	us-east-1
プロキシ電話セッション管理	複数のリージョン	us-east-1
ストリーミング管理	複数のリージョン	us-east-1

	Amazon Chime SDK Voice 名前空間	Amazon Chime 名前空間
ログ記録およびメトリクス管理	複数のリージョン	us-east-1

利用可能なリージョンの詳細については、このガイドの前の方にある「[PSTN リージョン](#)」を参照してください。

名前空間の相違点

以下のセクションでは、Amazon Chime SDK Voice 名前空間と Amazon Chime 名前空間の相違点について説明します。

AWS 名前空間

Amazon Chime SDK 名前空間では Chime という正式名を使用します。Amazon Chime SDK Voice 名前空間では ChimeSDKVoice という正式名を使用します。名前の正確な形式はプラットフォームによって異なります。

例えば、Node.js の AWS SDK を使用して会議を作成する場合、次のコード行を使用して名前空間に対処します。

```
const chimeVoice = AWS.Chime();
```

Amazon Chime SDK Voice 名前空間に移行するには、このコード行を新しい名前空間とエンドポイントリージョンで更新します。

```
const chimeVoice = AWS.ChimeSDKVoice({ region: "eu-central-1" });
```

リージョン

[Amazon Chime](#) 名前空間は、us-east-1 リージョンの API エンドポイントのみをアドレス指定します。[Amazon Chime SDK Voice](#) 名前空間は、利用可能な任意のリージョンで Amazon Chime SDK 音声 API エンドポイントをアドレス指定できます。現在の音声リージョンのリストについては、本ガイドの「[利用できるリージョン](#)」を参照してください。

エンドポイント

[Amazon Chime SDK Voice](#) 名前空間は、[Amazon Chime](#) 名前空間とは異なる API エンドポイントを使用します。

音声アクションを変更するのに使用できるのは、その作成に使用したエンドポイントだけです。つまり、eu-central-1 のエンドポイントを介して作成された音声アクションは、eu-central-1 経由でしか変更できないということです。また、Chime 名前空間を介して作成された音声アクションは、us-east-1 の ChimeSDKVoice 名前空間ではアドレス指定できません。現在のエンドポイントの詳細については、このガイドの「[API マッピング](#)」を参照してください。

サービスプリンシパル

どちらの名前空間も chime.amazonaws.com サービスプリンシパルを使用します。サービスへのアクセスを許可するアクセスポリシーがある場合、それらのポリシーを更新する必要はありません。

API

[Amazon Chime SDK Voice](#) 名前空間には、音声アクションを作成および管理するための API のみが含まれています。[Amazon Chime](#) 名前空間には、音声や Amazon Chime サービスのその他の部分 (会議など) のための API が含まれています。

タグ付け

[Amazon Chime SDK Voice](#) 名前空間のみがタグをサポートしています。タグの詳細については、[TagResource](#) 「」および「」を参照してください [UntagResource](#)。

メディアリージョン

Chime 名前空間を使用して、音声コネクタや SIP メディアアプリケーションなどのリソースを作成する場合、これを実行できるのは us-east-1 と us-west-2 だけであり、us-east-1 エンドポイントを使用する必要があります。

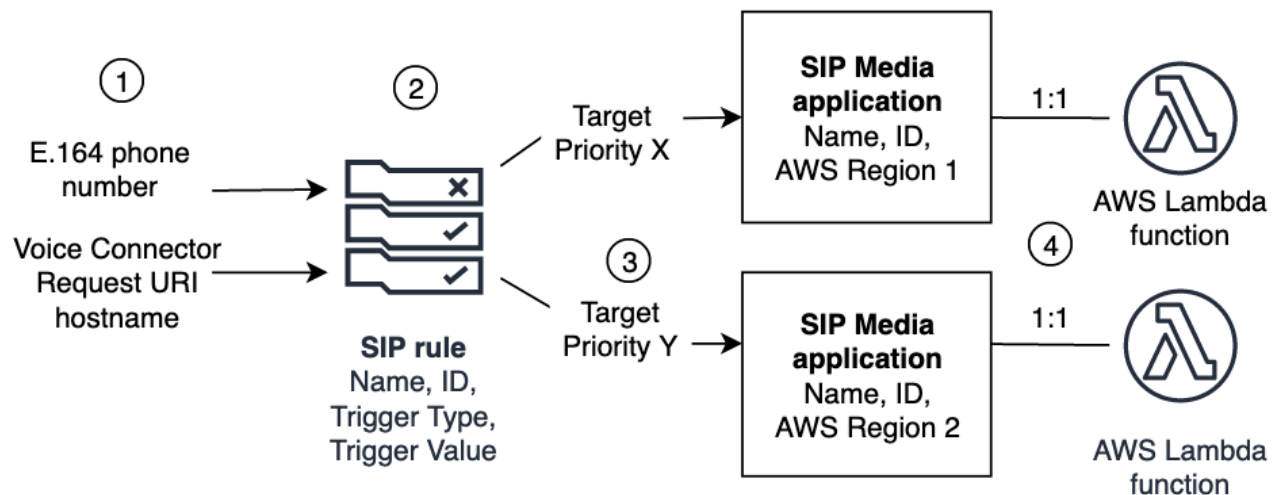
これに対して、ChimeSDKVoice 名前空間では、us-east-1 と us-west-2 だけでなく、サポートされているすべてのリージョンでリソースを作成できます。リージョンカバレッジの詳細については、「[利用できるリージョン](#)」を参照してください。

電話番号、SIP ルール、SIP メディアアプリケーション、および AWS Lambda 機能について

PSTN Audio サービスを使用する前に、Amazon Chime SDK 管理者は、電話番号をプロビジョニングし、SIP ルールおよび SIP メディアアプリケーションと呼ばれるマネージドオブジェクトを作成

する必要があります。Amazon Chime SDK コンソールまたは AWS SDK を使用して電話番号をプロビジョニングし、SIP ルールと SIP メディアアプリケーションマネージドオブジェクトをプロビジョニングできます。

以下の画像は、PSTN Audio サービスを構成するマネージドオブジェクト間の関係を示しています。画像の数字は、画像の下のテキスト内の数字に対応しています。



SIP ルール (2) には電話番号と Amazon Chime SDK Voice Connector (1) のみ割り当てることができません。また、PSTN Audio サービスで電話番号または Voice Connector をプロビジョニングする必要もあります。電話番号への受信通話、または Voice Connector からの発信通話リクエストを受け取ると、SIP ルールは、SIP メディアアプリケーションおよび関連付けられた AWS Lambda 関数を呼び出します (4)。この AWS Lambda 関数は、保留時の音楽の再生や会議への参加など、事前定義された一連のアクションを実行します。マルチリージョンの回復性を提供するために、SIP ルールでは、フェイルオーバーの優先順位に従って、異なる AWS リージョン (3) の代替ターゲット SIP メディアアプリケーションを指定できます。あるターゲットで障害が発生すると、PSTN Audio サービスは次のターゲットを試行し、またさらに次へと試行していきます。各代替ターゲットは異なる AWS リージョンに存在する必要があることに注意してください。

さらに、複数の SIP メディアアプリケーションが特定の AWS Lambda 関数を呼び出すことができます。別の言い方をすれば、AWS Lambda 関数を作成すると、SIP メディアアプリケーションでその関数を使用できます。

SIP メディアアプリケーションおよびルールのプロビジョニングの詳細については、「Amazon Chime SDK 管理者ガイド」の「[SIP メディアアプリケーションおよびルールの管理](#)」を参照してください。

PSTN Audio サービスのプログラミングモデルについて

PSTN Audio サービスは、AWS Lambda 関数を使用するリクエスト/レスポンスプログラミングモデルを使用します。AWS Lambda 関数は、着信通話と発信通話に自動的に呼び出されます。例えば、新しい着信が到着すると、PSTN Audio サービスはNEW_INCOMING_CALLイベントで AWS Lambda 関数を呼び出し、アクションと呼ばれるコマンドを待ちます。例えば、アプリケーションは、音声プロンプトの再生、ディジットの収集、音声の録音、次への通話のルーティングなどのアクションを選択できます。これらの JSON 形式のアクションは、AWS Lambda 関数からのコールバックを使用して PSTN Audio サービスに送り返されます。

以下の例は、PlayAudio アクションを示しています。

```
{
  "Type": "PlayAudio",
  "Parameters": {
    "CallId": "call-id-1",
    "ParticipantTag": "LEG-A",
    "PlaybackTerminators": ["1", "8", "#"],
    "Repeat": "5",
    "AudioSource": {
      "Type": "S3",
      "BucketName": "valid-S3-bucket-name",
      "Key": "wave-file.wav"
    }
  }
}
```

以下の例は、RecordAudio アクションを示しています。

```
{
  "Type": "RecordAudio",
  "Parameters": {
    "CallId": "call-id-1",
    "DurationInSeconds": "10",
    "SilenceDurationInSeconds": 3,
    "SilenceThreshold": 100,
    "RecordingTerminators": [
      "#"
    ],
    "RecordingDestination": {
      "Type": "S3",

```

```
        "BucketName": "valid-bucket-name",
        "Prefix": "valid-prefix-name"
    }
}
```

PSTN Audio サービスがアクションを実行すると、成功または失敗のどちらかのインジケータを使用して AWS Lambda 関数を再度呼び出します。

アプリケーションは、発信通話を行い、AWS Lambda 関数を使用して通話フロー、発信者エクスペリエンス、通話コンテキストを制御することもできます。この場合、[CreateSipMediaApplicationCall](#) API を呼び出し、AWS Lambda が NEW_OUTBOUND_CALL イベントで呼び出されます。通話が応答されたら、音声プロンプトの再生やユーザーが入力したディジットの収集などのアクションを返すことができます。[UpdateSipMediaApplicationCall](#) API を使用して AWS Lambda 関数をトリガーし、タイマー、参加者のミュート、待合室を実装することもできます。

AWS Lambda 関数への呼び出しとイベントのルーティング

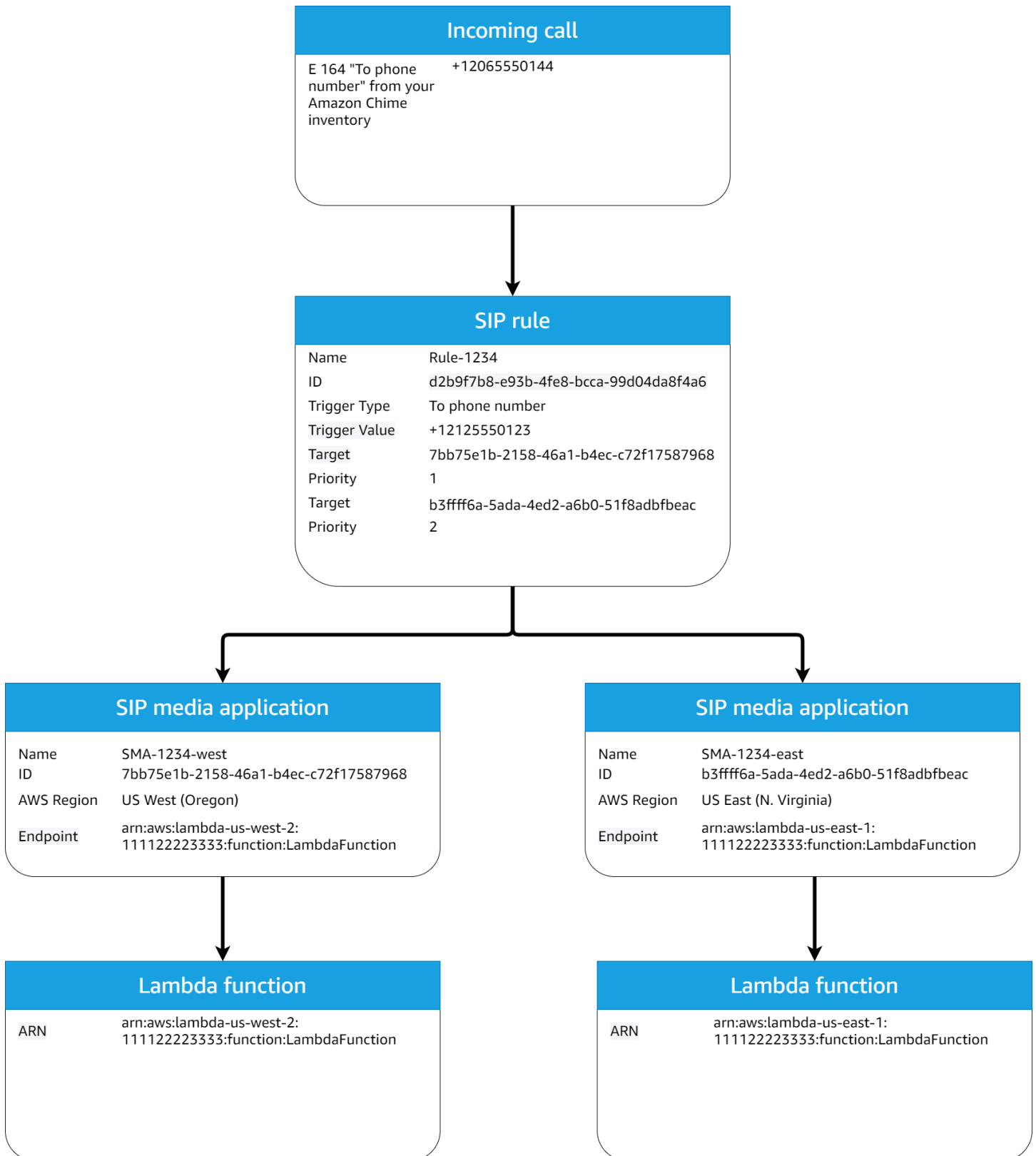
PSTN Audio サービスでは、着信通話を処理のために AWS Lambda 関数にルーティングする方法を以下に示します。

- 着信番号に基づいて通話をルーティングできます。これを行うには、Amazon Chime SDK 管理者が、[トリガータイプ] が [電話番号へ] に設定された SIP ルールを作成します。この電話番号は、SIP ルールと同じ AWS アカウントにある Amazon Chime SDK 電話番号インベントリに存在する必要があります。
- 着信 Voice Connector SIP コールのリクエスト URI に基づいて、AWS Lambda 関数にコールをルーティングできます。これを行うには、Amazon Chime SDK 管理者が、[トリガータイプ] が [リクエスト URI ホスト名] に設定された SIP ルールを作成します。このフィールドには、SIP ルールと同じ AWS アカウントでプロビジョニングされる Voice Connector の「アウトバウンドホスト名」フィールドで指定された完全修飾ドメイン名が含まれている必要があります。

次に、管理者は、ターゲット SIP メディアアプリケーションを少なくとも 1 つプロビジョニングします。オプションで、冗長性とフェイルオーバーをサポートするために、複数の SIP メディアアプリケーションを優先順位に従ってプロビジョニングすることもできます。例えば、2 つの異なる AWS リージョンに 2 つの SIP メディアアプリケーションをプロビジョニングし、優先順位を指定できます。SIP ルールに複数のターゲット SIP メディアアプリケーションがある場合、SIP メディアアプリケーションの Lambda 関数は、優先順位に従って呼び出されます。優先順位が最も高い SIP メ

メディアアプリケーションの AWS Lambda 関数 (1 など最小数) が最初に実行されます。PSTN Audio サービスがその AWS Lambda 関数を呼び出せない場合 AWS Lambda 、優先順位が次に高い SIP メディアアプリケーションの関数 (2 など、次に小さい番号) が呼び出されます。SIP ルールで指定されている SIP メディアアプリケーションを実行しようとしてすべて失敗すると、PSTN Audio サービスはハングアップします。

必要な SIP ルールと SIP メディアアプリケーションがプロビジョニングされると、PSTN Audio サービスは着信通話を AWS Lambda 関数にルーティングします。次の図は、[電話番号へ]トリガータイプを使用した一般的なシーケンスを示しています。



図中の手順を説明します。

1. PSTN Audio サービスは、同じ AWS アカウントの SIP ルールでプロビジョニングされた電話番号への着信通話を受信します。
2. PSTN Audio サービスが SIP ルールを評価し、優先順位が最も高い (この場合は優先度 1 の) SIP メディアアプリケーションをフェッチします。
3. その後、サービスは SIP メディアアプリケーションに関連付けられた AWS Lambda 関数を呼び出します。
4. オプション。サービスが優先度の高い順に関連付け AWS Lambda られている を呼び出せない場合は、優先度の高い順に (この場合は優先度 2) SIP メディアアプリケーションが存在する場合、そのアプリケーションを実行しようとします。
5. オプション。ターゲット SIP メディアアプリケーションがすべて失敗すると、PSTN Audio サービスは通話をハングアップします。

次の図は、[リクエスト URI ホスト名] トリガータイプを使用する一般的なルールを示しています。

Incoming call

Voice Connector Host name	1234567890abcdef0.voiceconnector.chime.aws
---------------------------	--



SIP rule

ID	b8be60f8-788c-4a30-b489-62531291cf
Trigger Type	Request URI hostname
Target	81bdd897-2948-474d-849e-9a754a136f28
Trigger Value	12345678cdef0.voiceconnector.chime.aws
Priority	1
Name	Rule-5678

図中の手順を説明します。

1. PSTN オーディオサービスは、同じ AWS アカウントのプロビジョニングされた SIP ルールに一致するリクエスト URI ホスト名を持つ Amazon Chime SDK Voice Connector で着信通話を受信します。
2. サービスが SIP ルールを評価し、優先順位が最も低い SIP メディアアプリケーション (この場合は優先度 1 の唯一のターゲット SIP メディアアプリケーション) をフェッチします。
3. その後、サービスは SIP メディアアプリケーションに関連付けられた AWS Lambda 関数を呼び出します。
4. オプション。優先度が最も低い AWS Lambda 関数に呼び出せない場合、サービスでは、優先度が次に低い SIP メディアアプリケーションが存在する場合は実行しようとします。ここではターゲット SIP メディアアプリケーションは 1 つだけです。
5. オプション。ターゲット SIP メディアアプリケーションがすべて失敗すると、PSTN Audio サービスは通話をハングアップします。

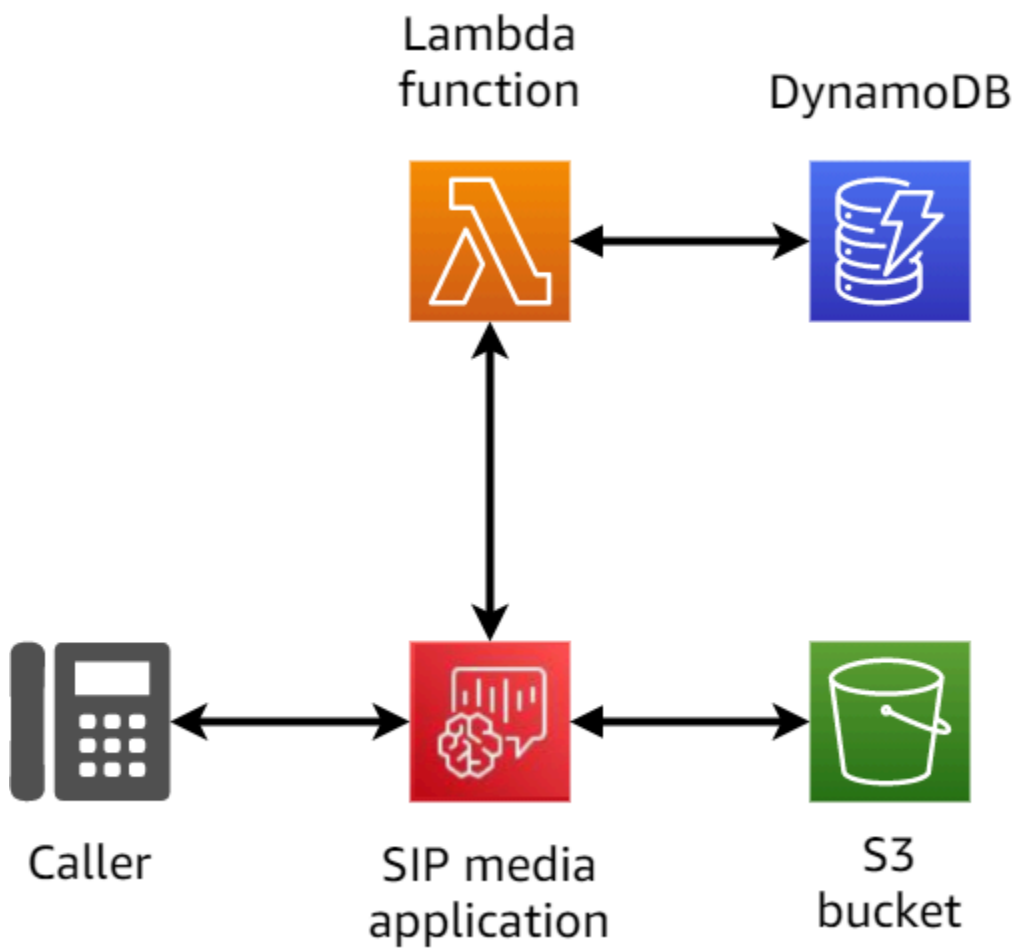
さらに、アウトバウンドコールを作成し、その後 [CreateSIPMediaApplicationCall](#) API を使用して AWS Lambda 関数を呼び出して追加の処理を行うことができます。この API を使用するには、プロビジョニングされた [SIP メディアアプリケーション ID] をパラメータとして指定します。

最後に、[UpdateSIPMediaApplicationCall](#) API を使用して呼び出しがアクティブになっている間はいつでも AWS Lambda 関数をトリガーできます。この API を使用するには、プロビジョニングされた [SIP メディアアプリケーション ID] をパラメータとして指定します。

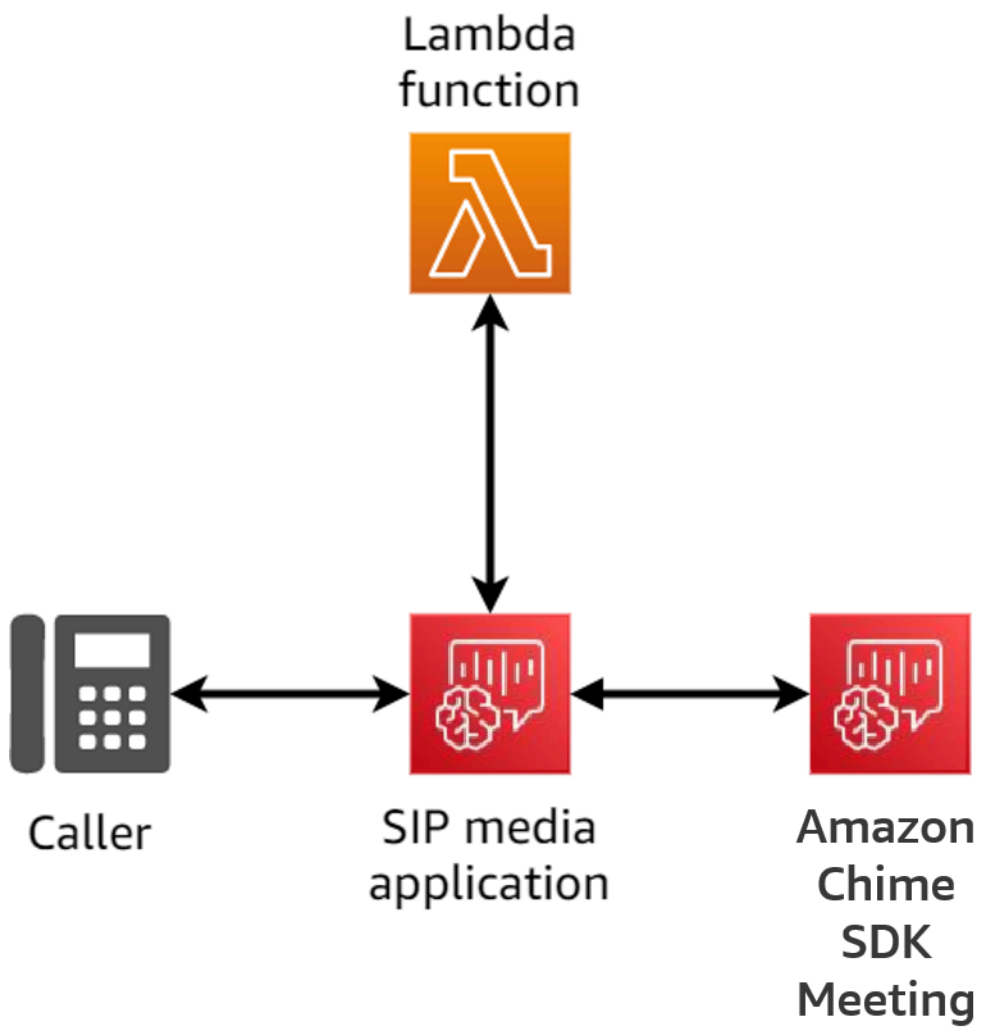
PSTN Audio サービスの通話レグの使用について

PSTN Audio サービスは 1 つ以上の通話レグで動作できます。例えば、ボイスメールを録音または配信する場合は単一の通話レグを使用し、Amazon Chime SDK ミーティングに参加する場合は複数の通話レグを使用します。

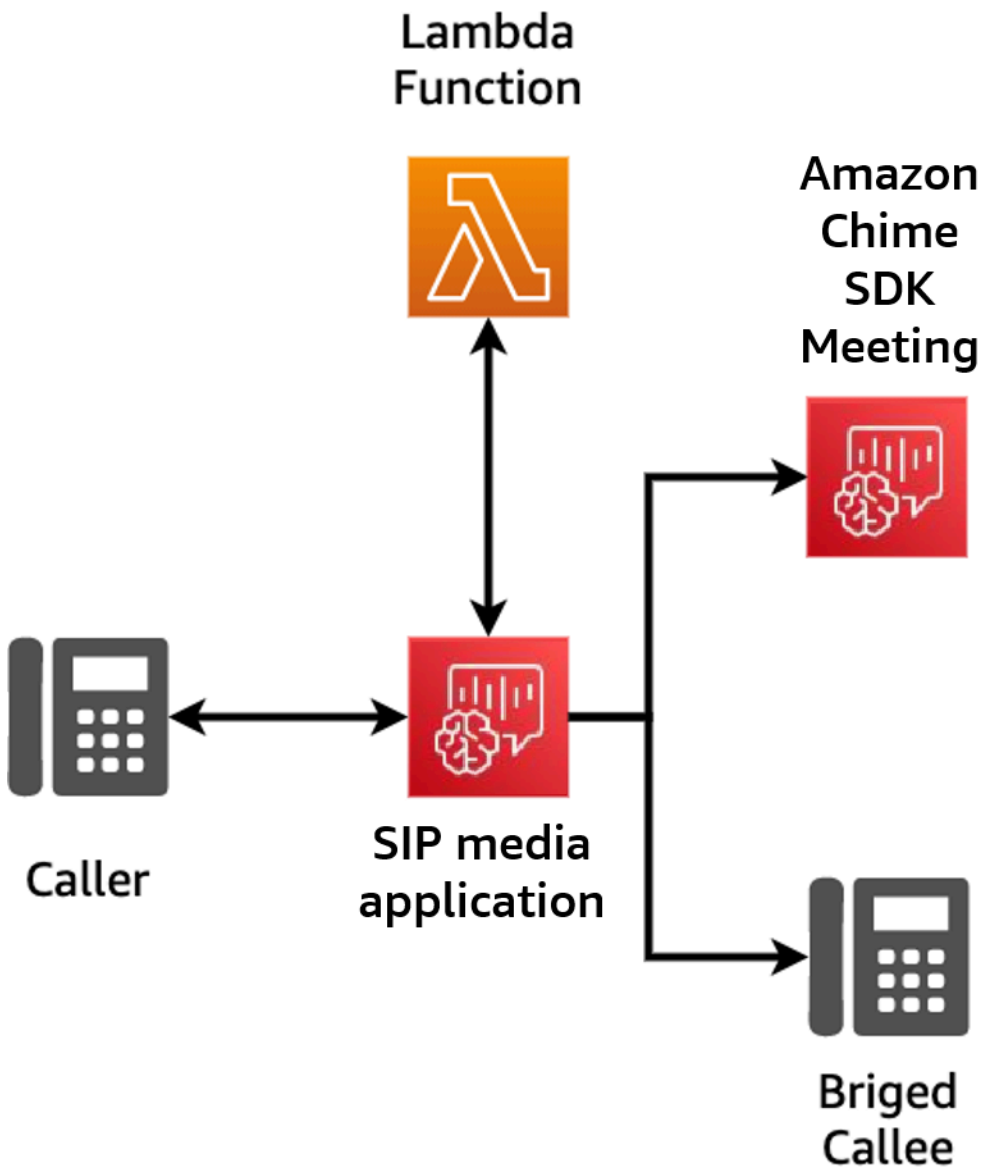
次の図は、単一レグの通話のフローを示しています。



次の図は、複数レッグの通話のアーキテクチャを示しています。



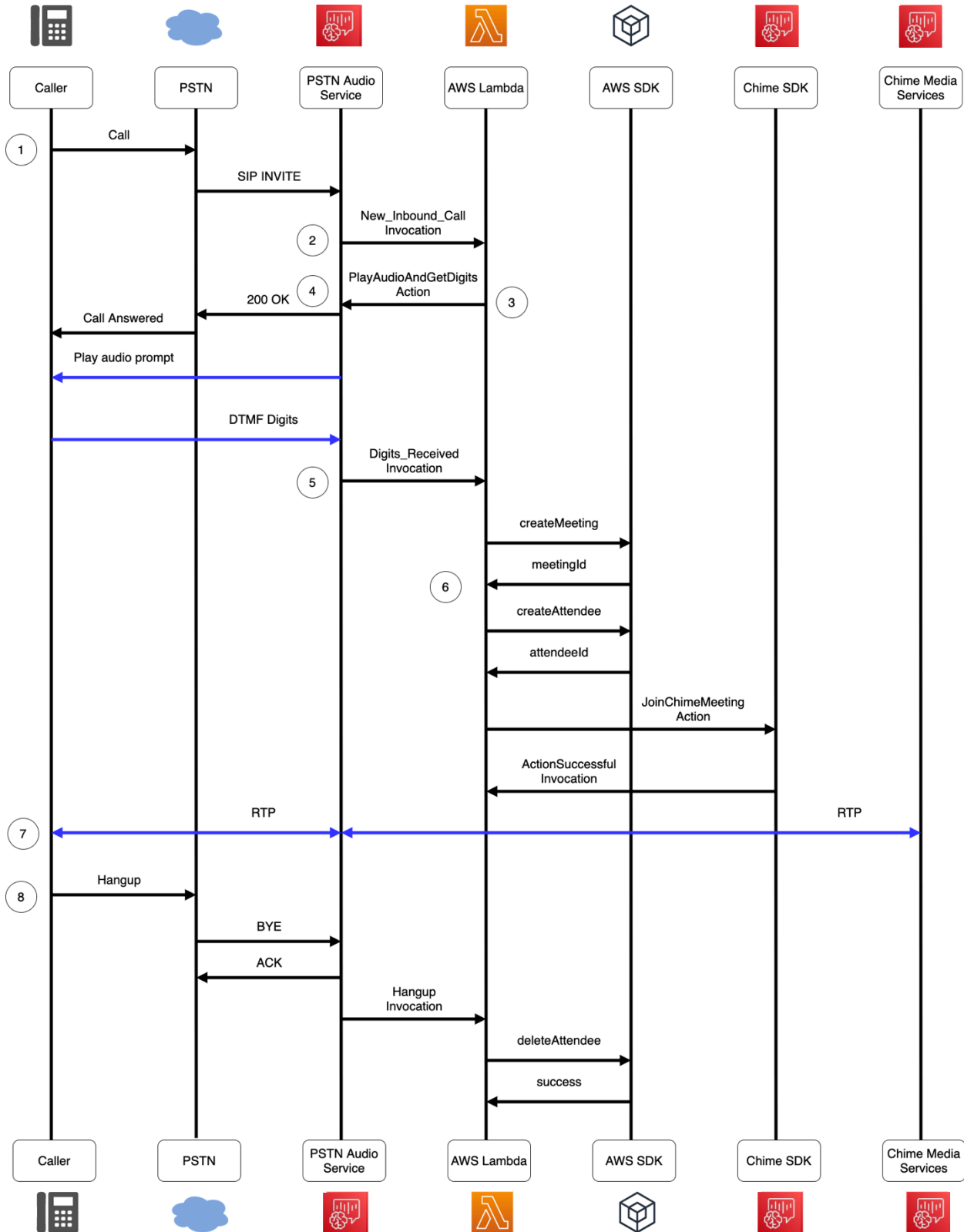
次の図は、複数レッグのブリッジ通話のフローを示しています。



通話フローの例

この図は、Amazon Chime SDK PSTN Audio サービスと顧客の AWS Lambda 関数を介した通話の流れを示しています。この例では、アプリケーションは発信者へのプロンプトを再生し、デュアルトーンマルチ周波数 (DTMF) デイジットを収集して、それを Amazon Chime SDK ミーティングに接続しています。

図中の数字は、図の下にある説明の番号に対応しています。



図中の手順を説明します。

1. Amazon Chime SDK PSTN Audio サービスは、SIP ルールでプロビジョニングされている電話番号への通話を受信します。
2. PSTN オーディオサービスは、関連付けられた SIP メディアアプリケーションを取得し、NEW_INBOUND_CALL イベント (LEG-A) で関連付けられた AWS Lambda 関数を呼び出します。
3. この AWS Lambda 関数は、を含むアクションのリストを返します。このアクションは PlayAudioAndGetDigits、PSTN Audio サービスに通話への応答、発信者への音声ファイルの再生、発信者が入力した DTMF デジットの収集を指示します。
4. PSTN Audio サービスは、通話に応答し、音声プロンプトを再生して、発信者が入力した DTMF デジットを収集します。
5. PSTN Audio サービスは、DTMF デジット入力を使用して AWS Lambda 関数を呼び出します。この AWS Lambda 関数は AWS SDK を使用して、Amazon Chime SDK 会議と会議参加者を作成します。
6. AWS SDK が MeetingId と を返すと AttendeeId、AWS Lambda 関数は Amazon Chime SDK ミーティング (LEG-B) への呼び出しに参加するアクションを返します。
7. 公衆交換電話網 (PSTN) からの発信者と Amazon Chime SDK Media サービスの間で、リアルタイム転送プロトコル (RTP) セッションが確立されます。
8. PSTN 発信者がハングアップすると、PSTN Audio サービスは HANGUP イベントで関数を呼び出し AWS Lambda、AWS Lambda 関数は参加者を削除します。

PSTN Audio サービスの AWS Lambda 関数の構築

このセクションのトピックでは、PSTN Audio サービスで使用される AWS Lambda 関数を構築する方法について説明します。

内容

- [テレフォニーイベントについて](#)
- [アクションについて](#)
- [AWS Lambda 関数を呼び出すテレフォニーイベント](#)
- [アクションリストによる呼び出しへの応答](#)
- [PSTN Audio サービスでサポートされるアクション](#)
- [SIP ヘッダーの使用](#)
- [通話詳細レコードの使用](#)

- [タイムアウトと再試行](#)
- [デバッグとトラブルシューティング](#)
- [VoiceFocus](#)
- [PSTN Audio サービス用語集](#)

テレフォニーイベントについて

Audio Service は、通話中に特定のイベントが発生したときに AWS Lambda 関数を呼び出します。次の例はイベントを示しています。それぞれのイベントについては、この例の後のテキストで説明しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 3,
  "InvocationEventType": "event-type",
  "CallDetails": {
    "TransactionId": "transaction-id-1",
    "AwsAccountId": "aws-acct-id-1",
    "AwsRegion": "us-east-1",
    "SipMediaApplicationId": "sip-media-app-id-1",
    "Participants": [
      {
        "CallId": "call-id-1",
        "ParticipantTag": "LEG-A",
        "To": "e164PhoneNumber",
        "From": "e164PhoneNumber",
        "Direction": "Inbound/Outbound",
        "StartTimeInMilliseconds": "1641998241509",
        "Status": "Connected/Disconnected"
      }
    ]
  }
}
```

SchemaVersion

このイベントオブジェクトの作成に使用されたスキーマのバージョン。

Sequence

AWS Lambda 関数を呼び出すイベントのシーケンス。通話中に関数が呼び出されるたびに、シーケンスがインクリメントされます。

InvocationEventType

AWS Lambda 呼び出しをトリガーするイベントのタイプ。詳細については、このトピックで後述する「[イベントタイプ](#)」を参照してください。

CallDetails

AWS Lambda 呼び出しに関連付けられた呼び出しに関する情報。

TransactionId

AWS Lambda 呼び出しに関連付けられた呼び出しの ID。

AwsAccountId

通話ルーティングの原因となった SIP メディアアプリケーションに関連付けられた AWS アカウント ID。

SipMediaApplicationId

通話に関連付けられた SIP メディアアプリケーションの ID。

Participants

AWS AWS Lambda 関数を呼び出す通話の参加者に関する情報。

CallId

各参加者に割り当てられた一意の ID。

ParticipantTag

各通話参加者にはタグ (LEG-A または LEG-B) が付けられます。

To

参加者の「受信者」電話番号 (E.164 形式)。

From

参加者の「発信者」電話番号 (E.164 形式)。

Direction

通話レグの発信元の方角。Inbound は Audio Service への通話を表します。Outbound は Audio Service からの通話を表します。

StartTimeInMilliseconds

参加者が通話に参加したときから始まるエポック時間 (ミリ秒単位)。

Status

参加者が `Connected` か `Disconnected` か。

イベントタイプ

Audio Service は、以下のイベントタイプを使用して Lambda 関数を呼び出します。

NEW_INBOUND_CALL

SIP メディアアプリケーションに関連付けられている電話番号から新しい通話が開始されました。

NEW_OUTBOUND_CALL

[CreateSipMediaApplicationCall](#) API を介して新しいアウトバウンドコールが行われました。

ACTION_SUCCESSFUL

AWS Lambda 関数から返されたアクションは成功しました。成功したアクションには、成功したアクションと一致する `ActionData` が含まれます。

```
"ActionData": {  
  // The previous successful action  
},
```

ACTION_FAILED

AWS Lambda 関数から返されたアクションは成功しませんでした。失敗したアクションには、失敗したアクションと一致する `ActionData`、エラータイプ、失敗について説明するエラーメッセージが含まれます。

```
"ActionData": {  
  // The previous unsuccessful action  
  "ErrorType": "error-type",  
  "ErrorMessage": "error message"
```



```
},
```

ACTION_INTERRUPTED

実行中のアクションが [UpdateSipMediaApplicationCall](#) API 呼び出しによって中断されました。ActionData には中断されたアクションが含まれます。

```
"ActionData": {  
  // The action that was interrupted  
},
```

HANGUP

ユーザーまたはアプリケーションが通話レックをハングアップしました。ActionData にはイベントに関する以下の詳細が含まれます。

```
"ActionData": {  
  "Type": "Hangup",  
  "Parameters": {  
    "SipResponseCode": 486,  
    "CallId": "c70f341a-adde-4406-9dea-1e01d34d033d",  
    "ParticipantTag": "LEG-A"  
  }  
},
```

Type

Hangup

Parameters

HANGUP イベントに関する情報:

- SipResponseCode – イベントに関連付けられたレスポンスコード。最も一般的なコードは次のとおりです。
 - 0 - 通常のクリア
 - 480 - 応答なし
 - 486 - ユーザービジー
- CallId ハングアップした参加者の ID。
- ParticipantTag ハングアップした参加者のタグ。

CALL_ANSWERED

Audio Service が応答し、着信通話が応答されました。このイベントは、通話がブリッジされない限り、ダイヤルアウト通話時に返されます。

INVALID_LAMBDA_RESPONSE

最後の AWS Lambda 呼び出しに対して提供されたレスポンスが原因で問題が発生しました。ActionData には以下の追加フィールドが含まれます。

```
"ErrorType": "error-type-1",  
"ErrorMessage": "error-msg-1"
```

DIGITS_RECEIVED

アプリケーションは、ReceiveDigits アクションの完了後に DTMF デイジットを受け取りました。ActionData には受け取ったデイジットが含まれます。

```
"ActionData": {  
  "ReceivedDigits": ###  
  // The ReceiveDigits action data  
},
```

CALL_UPDATE_REQUESTED

[UpdateSipMediaApplicationCall](#) API が呼び出されました。ActionData には更新リクエストに関する情報が含まれます。

```
"ActionData": {  
  "Type": "CallUpdateRequest",  
  "Parameters": {  
    "Arguments": {  
      "leg": "LEG-A"  
    }  
  }  
},  
}
```

RINGING

通話レグの呼び出し音が鳴っています。

アクションについて

PSTN Audio サービスでは、SIP メディアアプリケーションが AWS Lambda 関数をトリガーします。次に、AWS Lambda 関数はアクションと呼ばれる命令のリストを返すことができます。アクションとは、数字の送受信や会議への参加など、電話のレグで実行する項目のことです。アクションはデータを返すこともできるので、アクションはデータフィールドを持つオブジェクトと考えることができます。PSTN Audio サービスによって呼び出されるアクションの詳細については、「[テレフォニーイベントについて](#)」を参照してください。

AWS Lambda 関数を呼び出すテレフォニーイベント

Audio Service は、さまざまなイベントにตอบสนองして AWS Lambda 関数を呼び出します。各呼び出しは、呼び出しイベントタイプを指定し、該当する場合は、参加者を含む呼び出しの詳細を提供します。以下のトピックでは、AWS Lambda 関数を呼び出す Audio Service イベントについて説明します。

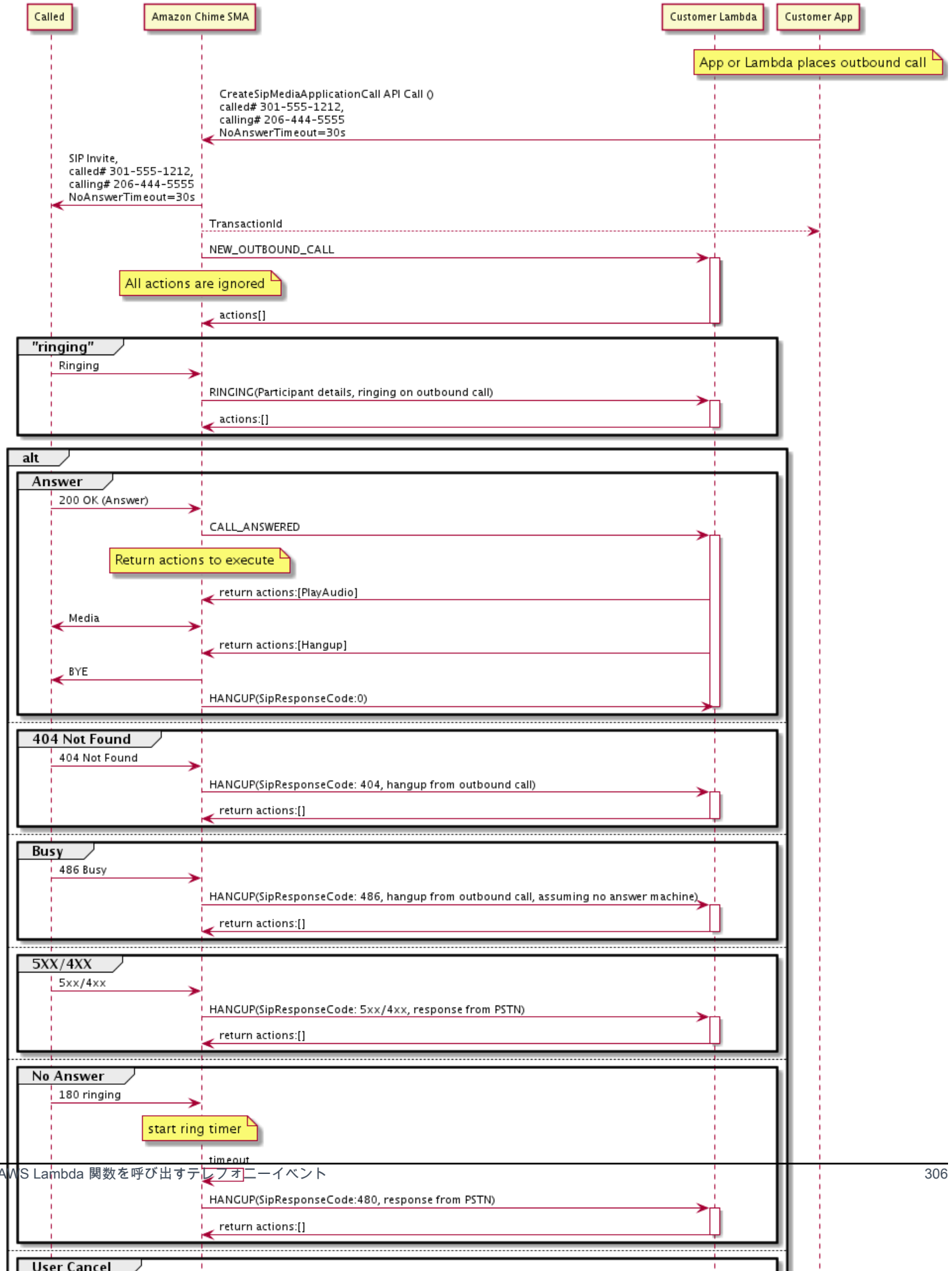
発信通話を行う

アウトバウンドコールを作成するには、[CreateSipMediaApplicationCall](#) API を使用します。API は、指定された SIP media application ID のエンドポイントを呼び出します。お客様は、エンドポイントから異なるシグナリングと[SipMediaApplication](#)アクションを提供することで、通話のフローを制御できます。

応答が成功すると、API は transactionId とともに 202 http ステータスコードを返します。このコードを [UpdateSipMediaApplicationCall](#) API とともに使用して、進行中の呼び出しを更新できます。

次の図は、アウトバウンドコールの AWS Lambda 関数エンドポイントに対して行われた呼び出しを示しています。

CreateSipMediaApplicationCall() Behavior



SIP メディアアプリケーション用に設定されたエンドポイントは、発信通話のさまざまなステータスに対して呼び出されます。顧客が通話を開始すると、Amazon Chime SDK は、NEW_OUTBOUND_CALL 呼び出しイベントタイプでエンドポイントを呼び出します。

この例は、NEW_OUTBOUND_CALL の典型的な呼び出しイベントを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 1,
  "InvocationEventType": "NEW_OUTBOUND_CALL",
  "CallDetails": {
    "TransactionId": "transaction-id",
    "AwsAccountId": "aws-account-id",
    "AwsRegion": "us-east-1",
    "SipApplicationId": "sip-application-id",
    "Participants": [
      {
        "CallId": "call-id-1",
        "ParticipantTag": "LEG-A",
        "To": "+1xxxx",
        "From": "+1xxxxxxx",
        "Direction": "Outbound",
        "StartTimeInMilliseconds": "159700958834234"
      }
    ]
  }
}
```

イベント関連の AWS Lambda 呼び出しに対する応答は無視されます。

受信者から RINGING 通知を受け取ると、Amazon Chime SDK は設定されたエンドポイントを再度呼び出します。

この例は、RINGING の典型的な呼び出しイベントを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 1,
  "InvocationEventType": "RINGING",
  "CallDetails": {
    "TransactionId": "transaction-id",
    "AwsAccountId": "aws-account-id",
    "AwsRegion": "us-east-1",
```

```
"SipApplicationId": "sip-application-id",
"Participants": [
  {
    "CallId": "call-id-1",
    "ParticipantTag": "LEG-A",
    "To": "+1xxxx",
    "From": "+1xxxxxxxx",
    "Direction": "Outbound",
    "StartTimeInMilliseconds": "159700958834234"
  }
]
}
```

イベント関連の AWS Lambda 呼び出しに対する応答は無視されます。

受信者が発信に回答しない場合、またはエラーによって発信が失敗した場合、Chime は発信を切断し、Hangup イベントタイプでエンドポイントを呼び出します。Hangup イベントタイプの詳細については、「[発信の終了](#)」を参照してください。

発信に回答した場合、Chime は CALL_ANSWERED アクションでエンドポイントを呼び出します。この例は、典型的な呼び出しイベントを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 1,
  "InvocationEventType": "CALL_ANSWERED",
  "CallDetails": {
    "TransactionId": "transaction-id",
    "AwsAccountId": "aws-account-id",
    "AwsRegion": "us-east-1",
    "SipApplicationId": "sip-application-id",
    "Participants": [
      {
        "CallId": "call-id-1",
        "ParticipantTag": "LEG-A",
        "To": "+1xxxx",
        "From": "+1xxxxxxxx",
        "Direction": "Outbound",
        "StartTimeInMilliseconds": "159700958834234",
        "Status": "Connected"
      }
    ]
  }
}
```

```
}  
}
```

この時点で、アクションリストを使用して呼び出しに 응답してアクションを返すことができます。アクションを実行しない場合は、空のリストで 응답します。AWS Lambda 呼び出しごとに最大 10 個のアクションで 응답でき、呼び出しごとに 1,000 回 Lambda 関数を呼び出すことができます。一連のアクションによる 응답の詳細については、「[アクションリストによる呼び出しへの 응답](#)」を参照してください。

受信通話の受け取り

NEW_INCOMING_CALL イベントが発生すると、Audio Service は、HANGUP イベントが発生するまで、一意の TransactionID と CallID を作成します。

NEW_INCOMING_CALL イベントにはいくつかの方法で 응답できます。例:

- PlayAudio または RecordAudio アクションを送信し、自動的に発信に 응답します。
- Pause アクションを送信します。
- Hangup アクションを送信します。この場合、発信は 응답されず、顧客への課金は発生しません。
- CallAndBridge アクションを送信し、別のユーザーを発信に追加します。
- 何もしなければ、30 秒後に発信がタイムアウトします。

新しい受信通話を受信すると、SIP メディアアプリケーションはこのペイロードで AWS Lambda 関数を呼び出します。

```
{  
  "SchemaVersion": "1.0",  
  "Sequence": 2,  
  "InvocationEventType": "NEW_INBOUND_CALL"  
  "CallDetails": {  
    "TransactionId": "transaction-id",  
    "AwsAccountId": "aws-account-id",  
    "AwsRegion": "us-east-1",  
    "SipRuleId": "sip-rule-id",  
    "SipApplicationId": "sip-application-id",  
    "Participants": [  
      {  
        "CallId": "call-id-1",  
        "ParticipantTag": "LEG-A",  
        "To": "+12065551212",
```

```
        "From": "+15105550101",
        "Direction": "Inbound",
        "StartTimeInMilliseconds": "159700958834234",
        "Status": "Connected"
    }
  ]
}
```

テレフォニーイベントに対するアクションの指定

Audio Service では、SIP メディアアプリケーションは AWS Lambda 関数を呼び出します。これにより、Lambda 関数は、アクションと呼ばれる命令のリストを返すことができます。アクションとは、数字の送受信や会議への参加など、電話のレグで実行する項目のことです。PSTN Audio サービスによって呼び出されるアクションの詳細については、「[テレフォニーイベントについて](#)」を参照してください。

SIP メディアアプリケーションがアクションのリストを正常に実行すると、アプリケーションは呼び出しイベントタイプで AWS Lambda 関数を呼び出します ACTION_SUCCESSFUL。いずれかのアクションが完了しない場合、SIP メディアアプリケーションは ACTION_FAILED イベントで AWS Lambda 関数を呼び出します。

リストのすべてのアクションが成功した場合、SIP メディアアプリケーションは ACTION_SUCCESSFUL を返すのみです。リストのいずれかのアクションが失敗した場合、SIP メディアアプリケーションは ACTION_FAILED イベントで AWS Lambda 関数を呼び出し、リスト内の失敗したアクションより後の残りのアクションをクリアします。次に、SIP メディアアプリケーションは、AWS Lambda 関数によって返される次のアクションを実行します。ActionData キーを使用して、関数を呼び出した発信を識別します。

次のイベントは、PlayAudioAndGetDigits アクション後の ACTION_SUCCESSFUL 呼び出しイベントタイプのサンプルペイロードを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 3,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "PlayAudioAndGetDigits",
    "Parameters" : {
      "CallId": "call-id-1",
      "AudioSource": {
```



```
        "Type": "S3",
        "BucketName": "bucket-name",
        "Key": "failure-audio-file.wav"
    },
    "FailureAudioSource": {
        "Type": "S3",
        "BucketName": "bucket-name",
        "Key": "failure-audio-file.wav"
    },
    "MinNumberOfDigits": 3,
    "MaxNumberOfDigits": 5,
    "TerminatorDigits": ["#"],
    "InBetweenDigitsDurationInMilliseconds": 5000,
    "Repeat": 3,
    "RepeatDurationInMilliseconds": 10000
},
"ReceivedDigits": "123"
}
"CallDetails": {
    "TransactionId": "transaction-id",
    "AwsAccountId": "aws-account-id",
    "AwsRegion": "us-east-1",
    "SipRuleId": "sip-rule-id",
    "SipApplicationId": "sip-application-id",
    "Participants": [
        {
            "CallId": "call-id-1",
            "ParticipantTag": "LEG-A",
            "To": "+12065551212",
            "From": "+15105550101",
            "Direction": "Inbound",
            "StartTimeInMilliseconds": "159700958834234",
            "Status": "Connected"
        }
    ]
}
}
}
```

リスト内のいずれかのアクションが正常に完了しなかった場合、SIP メディアアプリケーションは AWS Lambda 関数を呼び出して失敗を通知し、その呼び出しで新しい一連のアクションを実行します。次のイベントは、PlayAudio アクション実行後の ACTION_FAILED 呼び出しイベントタイプのサンプルペイロードを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 4,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {
    "Type": "PlayAudio",
    "Parameters" : {
      "CallId": "call-id-1",
      "AudioSource": {
        "Type": "S3",
        "BucketName": "bucket-name",
        "Key": "audio-file.wav"
      }
    },
    "ErrorType": "InvalidAudioSource",
    "ErrorMessage": "Audio Source parameter value is invalid."
  }
  "CallDetails": {
    "TransactionId": "transaction-id",
    "AwsAccountId": "aws-account-id",
    "AwsRegion": "us-east-1",
    "SipRuleId": "sip-rule-id",
    "SipApplicationId": "sip-application-id",
    "Participants": [
      {
        "CallId": "call-id-1",
        "ParticipantTag": "LEG-A",
        "To": "+12065551212",
        "From": "+15105550101",
        "Direction": "Inbound",
        "StartTimeInMilliseconds": "159700958834234",
        "Status": "Connected"
      }
    ]
  }
}
```

発信者の入力の受信

ReceiveDigits アクションを使用してインバウンド DTMF デジットを収集し、正規表現と照合します。SIP メディアアプリケーションは、正規表現に一致するデジットを受信する

と、ACTION_SUCCESSFUL イベントで AWS Lambda 関数を呼び出します。収集されたディジットは、ActionData オブジェクトの ReceivedDigits 値に表示されます。

例:

```
{
  "SchemaVersion": "1.0",
  "Sequence": 4,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "ReceivedDigits": "",
    "Type": "ReceiveDigits",
    "Parameters": {
      "CallId": "call-id-1",
      "InputDigitsRegex": "^\\d{2}#$",
      "InBetweenDigitsDurationInMilliseconds": 5000,
      "FlushDigitsDurationInMilliseconds": 10000
    }
  },
  "CallDetails": {
    "TransactionId": "transaction-id",
    "AwsAccountId": "aws-account-id",
    "AwsRegion": "us-east-1",
    "SipRuleId": "sip-rule-id",
    "SipApplicationId": "sip-application-id",
    "Participants": [
      {
        "CallId": "call-id-1",
        "ParticipantTag": "LEG-A",
        "To": "+12065551212",
        "From": "+15105550101",
        "Direction": "Inbound",
        "StartTimeInMilliseconds": "159700958834234",
        "Status": "Connected"
      }
    ]
  }
}
```

発信者が正規表現パターンに一致する数字を入力すると、SIP メディアアプリケーションは次のタイプのペイロードを返す AWS Lambda 関数を呼び出します。

```
{
```

```
"SchemaVersion": "1.0",
"Sequence": 5,
"InvocationEventType": "DIGITS_RECEIVED",
"ActionData": {
  "ReceivedDigits": "11#",
  "Type": "ReceiveDigits",
  "Parameters": {
    "CallId": "call-id-1",
    "InputDigitsRegex": "^\\d{2}#$",
    "InBetweenDigitsDurationInMilliseconds": 5000,
    "FlushDigitsDurationInMilliseconds": 10000
  }
},
"CallDetails": {
  "TransactionId": "transaction-id",
  "AwsAccountId": "aws-account-id",
  "AwsRegion": "us-east-1",
  "SipRuleId": "sip-rule-id",
  "SipApplicationId": "sip-application-id",
  "Participants": [
    {
      "CallId": "call-id-1",
      "ParticipantTag": "LEG-A",
      "To": "+12065551212",
      "From": "+15105550101",
      "Direction": "Inbound",
      "StartTimeInMilliseconds": "159700958834234",
      "Status": "Connected"
    }
  ]
}
}
```

GitHub 「: <https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>」の実例を参照してください。

進行中の通話の更新

PSTN Audio Service の一部として、SIP メディアアプリケーションでは、着信通話や DTMF デイジットなどの通話イベントに基づいてユーザー定義の Lambda 関数を呼び出すことで、通話で実行されるアクションを設定できます。[UpdateSipMediaApplicationCall](#) API を使用すると、呼び出しがアクティブな間はいつでも Lambda 関数をトリガーし、現在のアクションを呼び出しによって返される新しいアクションに置き換えることができます。

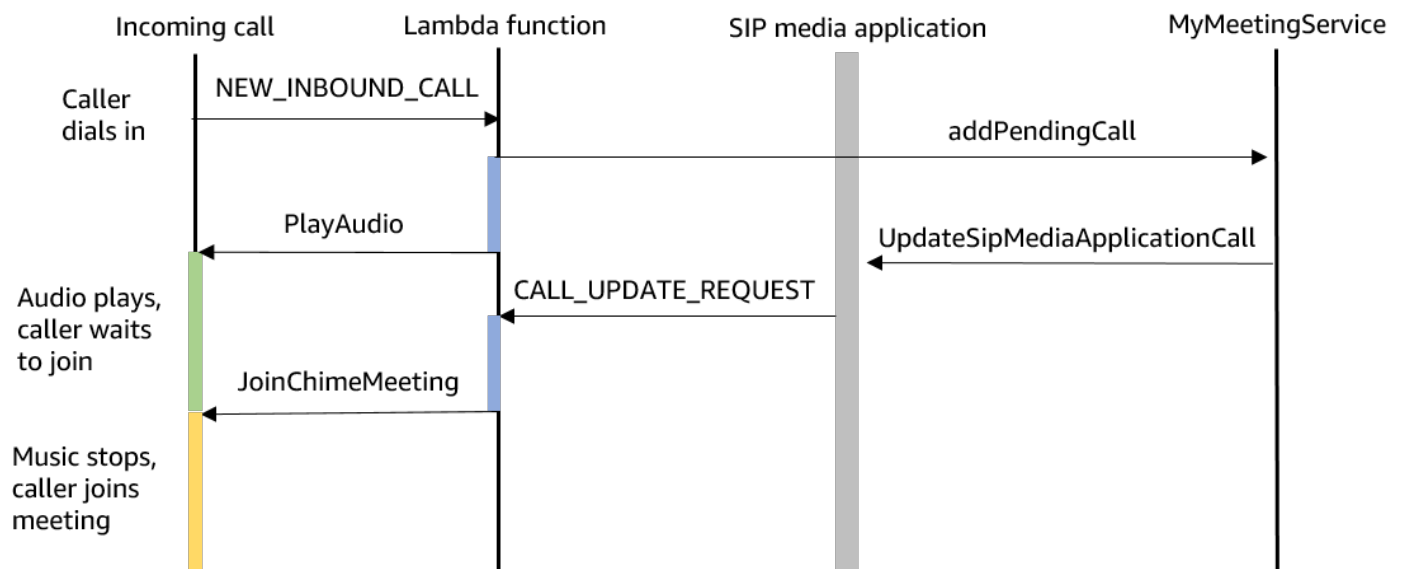
ワークフロー

[UpdateSipMediaApplicationCall](#) API は、会議への参加者の追加、ユーザーのミュートとミュート解除、ユーザーの切断など、さまざまなケースで使用します。以下のユースケースでは、一般的なワークフローについて説明しています。

Amazon Chime SDK が会議をセットアップしている間、ユーザーは電話をかけたり音楽を聴いたりしています。セットアップが完了すると、Amazon Chime SDK は音声を停止し、発信者を会議に参加させます。次に、会議を管理する別のシステム、MyMeetingService を使用すると仮定します。すべての着信通話を保留にする必要があります。Chime は着信 MyMeetingService について通知 MyMeetingService し、通話ごとに参加者を作成し、MyMeetingService が会議を開始する準備ができたなら SIP メディアアプリケーションに通知し、会議に参加するためのトークンを提供します。

このケースを処理するには、Lambda 関数は次のロジックを実装する必要があります。

- 新しい着信通話が届くと、Lambda が NEW_INBOUND_CALL イベントで呼び出されます。Lambda は MyMeetingService を呼び出し、現在の通話を識別する transactionId を渡して、PlayAudio アクションを返します。
- MyMeetingService が発信者を会議に追加する準備ができたなら、サービスは [UpdateSipMediaApplicationCall](#) API を呼び出し、引数 JoinToken の一部として呼び出しの transactionId と を渡します。この API コールにより、今度は CALL_UPDATE_REQUESTED イベントで Lambda 関数が再度トリガーされます。はイベントの一部として Lambda 関数 JoinToken に を MyMeetingService 渡し、トークンは JoinChimeMeeting アクションを SIP メディアアプリケーションに返すために使用されます。これにより、PlayAudio アクションが中断され、発信者が会議に接続されます。



Note

[UpdateSipMediaApplicationCall](#) API は HTTP 202 (Accepted) を返します。SIP メディアアプリケーションは、通話が進行中で更新可能であることを確認し、Lambda 関数を呼び出そうとします。呼び出しは非同期で実行されるため、API からの応答が成功しても、Lambda 関数が開始または完了したことは保証されません。

以下にリクエストの構文例を示します。

```

{
  "SipMediaApplicationId": "string",
  "TransactionId": "string",
  "Arguments": {
    "string": "string"
  }
}
  
```

リクエストパラメータ

- `SipMediaApplicationId` – 通話进行处理する SIP メディアアプリケーションの ID。
- `TransactionId` – 通話トランザクションの ID。受信通話の場合、`TransactionId` は、最初の呼び出しで Lambda 関数に渡された `NEW_INCOMING_CALL` イベントから取得できます。アウトバウンドコールの場合、`TransactionId` は のレスポンスで返されます [CreateSipMediaApplicationCall](#)。

- Arguments - CallUpdateRequest アクションデータの一部として Lambda 関数で使用できるカスタム引数。キーと値のペアを 0~20 個含めることができます。

一般的なリクエストの例を次に示します。

```
aws chime update-sip-media-application-call --sip-media-application-id
feb37a7e-2b66-49fb-b2dd-30f4780dc36d --transaction-id 1322a4e7-c106-4e70-aaaf-
a8fa4c77c0cb --arguments '{"JoinToken": "abc123"}'
```

レスポンスの構文

```
{
  "SipMediaApplicationCall": {
    "TransactionId": "string"
  }
}
```

レスポンス要素

- TransactionId – コールトランザクションの ID。リクエストと同じ ID。

次の例は、CALL_UPDATE_REQUESTED 呼び出しイベントを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 2,
  "InvocationEventType": "CALL_UPDATE_REQUESTED",
  "ActionData": {
    "Type": "CallUpdateRequest",
    "Parameters": {
      "Arguments": {
        "string": "string"
      }
    }
  },
  "CallDetails": {
    ...
  }
}
```

イベントの要素

- SchemaVersion – JSON スキーマのバージョン (1.0)
- Sequence - 通話でのイベントのシーケンス番号
- InvocationEventType – この場合、Lambda 呼び出しイベントのタイプ。
CALL_UPDATE_REQUESTED
- ActionData – CallUpdateRequest アクションに関連付けられたデータ。
 - Type - アクションのタイプ。ここでは CallUpdateRequest
 - Parameters - アクションのパラメータ
 - Arguments - UpdateSipMediaApplicationCall API リクエストの一部として渡される引数
- CallDetails – 現在の通話状態に関する情報

割り込み可能なアクションと割り込みできないアクションについて

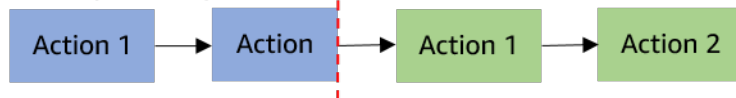
既存のアクションの実行中に Lambda 関数が新しいアクションのリストを返すと、進行中のアクションに続くすべてのアクションが新しいアクションに置き換えられます。場合によっては、新しいアクションをすぐに実行するために、Lambda 関数が進行中のアクションを中断することがあります。

次の図は、代表的な例を示しています。ロジックについては、図の下のテキストで説明しています。

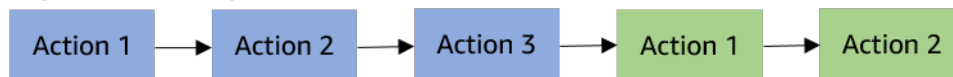
Initial actions



Interrupt and replace



Replace after completion



Action 2 が割り込み可能な場合は、これを停止し、代わりに新しい Action 1 を実行します。

Action 2 が割り込み可能でない場合は、これを完了してから新しい Action 1 を開始します。

いずれの場合も Action 3 は実行されません。

何かによってアクションが中断された場合は、ACTION_INTERRUPTED イベントによって Lambda 関数が呼び出されます。このイベントは、情報提供のみを目的として使用されます。SIP メディアアプリケーションは、この呼び出しによって返されるすべてのアクションを無視します。

割り込み可能なアクションのタイプ:

- PlayAudio
- RecordAudio
- Pause

サンプル Lambda 関数

以下の例は、音声ファイルを再生し、参加トークンを渡して、通話を更新する一般的な Lambda 関数を示しています。

```
const MMS = require('my-meeting-service');
const myMeetingServiceClient = new MMS.Client();

exports.handler = async (event) => {
  console.log('Request: ' + JSON.stringify(event));

  const playAudio = () => {
    return {
      Type: 'PlayAudio',
      Parameters: {
        ParticipantTag: 'LEG-A',
        AudioSource: {
          Type: 'S3',
          BucketName: 'chime-meetings-audio-files-bucket-name',
          Key: 'welcome.wav'
        }
      }
    }
  }

  const joinChimeMeeting = (joinToken) => {
    return {
      Type: 'JoinChimeMeeting',
      Parameters: {
        JoinToken: joinToken
      }
    }
  }
}
```

```
    }
  }

  const response = (...actions) => {
    const r = {
      SchemaVersion: '1.0',
      Actions: actions
    };
    console.log('Response: ' + JSON.stringify(r));
    return r;
  };

  switch (event.InvocationEventType) {
    case 'NEW_INBOUND_CALL':
      myMeetingServiceClient.addPendingCall(event.CallDetails.TransactionId);

      return response(playAudio());
    case 'CALL_UPDATE_REQUESTED':
      const joinToken = event.ActionData.Parameters.Arguments['JoinToken'];
      return response(joinChimeMeeting(joinToken));
    default:
      return response();
  }
}
```

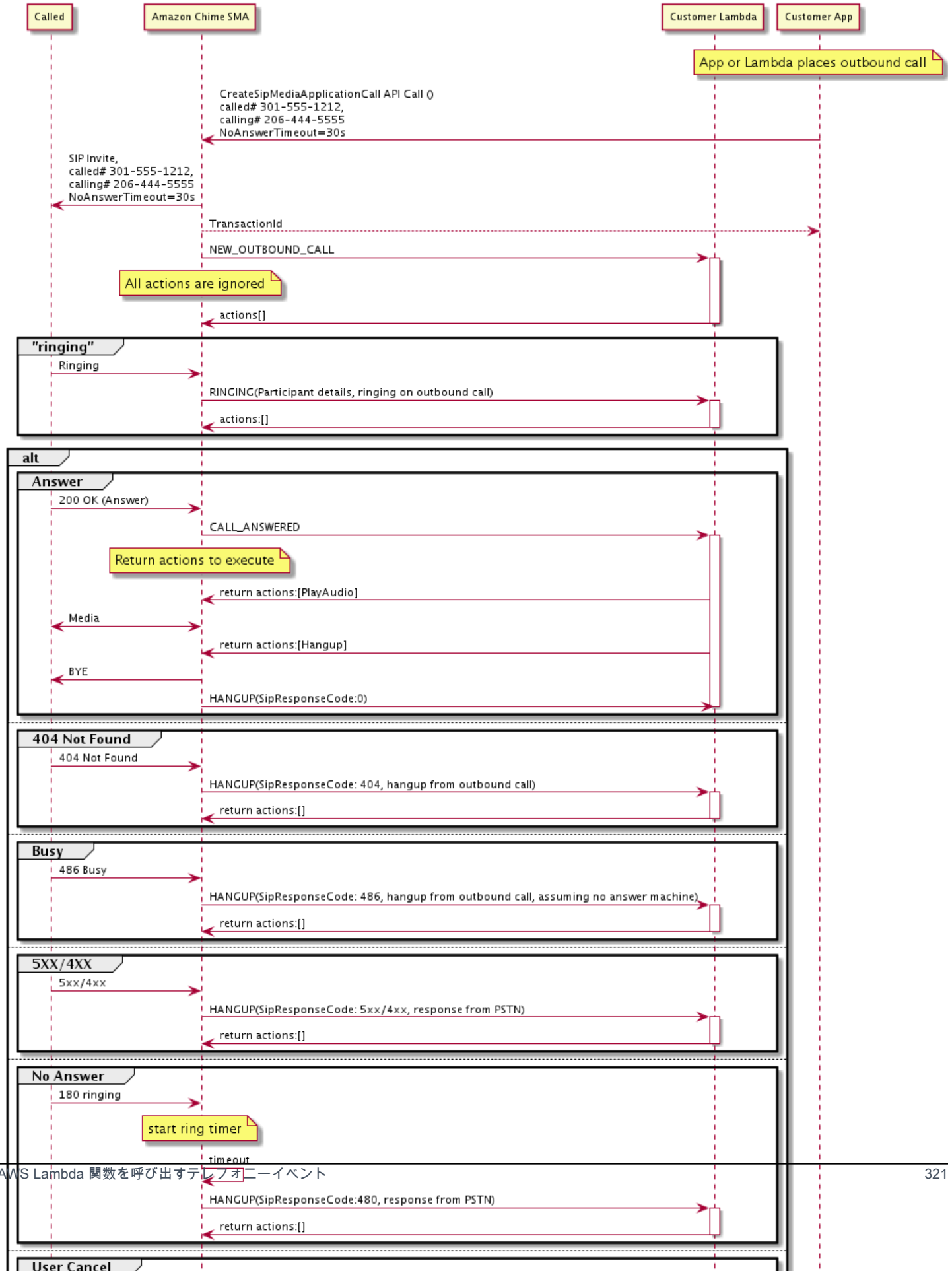
発信の終了

[CreateSipMediaApplicationCall](#) API を使用してアウトバウンドコールを終了できます。API は、指定された SIP メディアアプリケーション ID のエンドポイントを呼び出します。SIP メディアアプリケーションにアクションを返すことで、発信のフローを制御できます。

レスポンスが成功すると、API は とともに 202 http ステータスコードを返します。このコードを [UpdateSipMediaApplicationCall](#) API とともに使用して transactionId、進行中の呼び出しを更新できます。

次の図は、アウトバウンドコールの AWS Lambda 関数エンドポイントに対して行われた呼び出しを示しています。

CreateSipMediaApplicationCall() Behavior



SIP メディアアプリケーション用に設定されたエンドポイントは、発信通話のさまざまなステータスに対して呼び出されます。顧客が通話を終了すると、Amazon Chime SDK は、HANGUP 呼び出しイベントタイプでエンドポイントを呼び出します。

この例は、HANGUP の典型的な呼び出しイベントを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 6,
  "InvocationEventType": "HANGUP",
  "ActionData": {
    "Type": "Hangup",
    "Parameters": {
      "CallId": "call-id-1",
      "ParticipantTag": "LEG-A"
    }
  },
  "CallDetails": {
    "TransactionId": "transaction-id",
    "AwsAccountId": "aws-account-id",
    "AwsRegion": "us-east-1",
    "SipRuleId": "sip-rule-id",
    "SipApplicationId": "sip-application-id",
    "Participants": [
      {
        "CallId": "call-id-1",
        "ParticipantTag": "LEG-A",
        "Direction": "Inbound",
        "To": "+12065551212",
        "From": "+15105550101",
        "StartTimeInMilliseconds": "1597009588",
        "Status": "Disconnected"
      }
    ]
  }
}

// if LEG-B receives a hangup in a bridged call, such as a meeting ending
{
  "SchemaVersion": "1.0",
  "Sequence": 6,
  "InvocationEventType": "HANGUP",
  "ActionData": {
```

```
    "Type": "ReceiveDigits",
    "Parameters": {
      "CallId": "call-id-2",
      "ParticipantTag": "LEG-B"
    }
  },
  "CallDetails": {
    "TransactionId": "transaction-id",
    "AwsAccountId": "aws-account-id",
    "AwsRegion": "us-east-1",
    "SipRuleId": "sip-rule-id",
    "SipApplicationId": "sip-application-id",
    "Participants": [
      {
        "CallId": "call-id-1",
        "ParticipantTag": "Leg-A",
        "To": "+12065551212",
        "From": "+15105550101",
        "Direction": "Inbound",
        "StartTimeInMilliseconds": "1597009588",
        "Status": "Connected"
      },
      {
        "CallId": "call-id-2",
        "ParticipantTag": "Leg-B",
        "To": "+17035550122",
        "From": "SMA",
        "Direction": "Outbound",
        "StartTimeInMilliseconds": "15010595",
        "Status": "Disconnected"
      }
    ]
  }
}
```

End-to-end コールの例

このユースケースでは、PSTN 発信者からの通話の受信、音声メッセージによる発信者への挨拶、発信者からの会議 PIN の取得、音声の再生、および会議への発信者の参加の例を示します。

呼び出しイベントと アクション

Audio Service は、呼び出しイベントを JSON オブジェクトとして AWS Lambda 関数に渡します。オブジェクトには、呼び出しイベントタイプと関連するメタデータが含まれます。また、AWS Lambda 関数は、SIP メディアアプリケーションのアクションを JSON オブジェクトとして返し、それらのオブジェクトにはアクションタイプと関連するメタデータが含まれます。

次の表に、呼び出しイベントと、呼び出しイベントを受け取ったときに発生する可能性のある `ActionData.Type` の一覧を示します。

呼び出しイベント	ActionData.Type
ACTION_SUCCESSFUL	CallAndBridge ReceiveDigits PlayAudio PlayAudioAndGetDigits JoinChimeMeeting ModifyChimeMeetingAttendees RecordMeeting
ACTION_FAILED	CallAndBridge PlayAudio PlayAudioAndGetDigits ModifyChimeMeetingAttendees RecordMeeting
HANGUP	HangUp
DIGITS_RECEIVED	ReceiveDigits

Note

次のユースケースを実装するには、Amazon Chime SDK インベントリに少なくとも 1 つの電話番号、Amazon リソースネーム (ARN) を持つ AWS Lambda 関数を使用する SIP メディアアプリケーションマネージドオブジェクト、および電話番号をトリガーとして使用する SIP ルールが必要です。

Amazon Chime SDK がルールで指定された電話番号への呼び出しを受信すると、PSTN Audio サービスは呼び出しイベントタイプで AWS Lambda 関数 `NEW_INBOUND_CALL` を呼び出します。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 1,
  "InvocationEventType": "NEW_INBOUND_CALL",
  "CallDetails": {
    "TransactionId": "transaction-id",
    "AwsAccountId": "aws-account-id",
    "AwsRegion": "us-east-1",
    "SipRuleId": "sip-rule-id",
    "SipApplicationId": "sip-application-id",
    "Participants": [
      {
        "CallId": "call-id-1",
        "ParticipantTag": "LEG-A",
        "To": "+11234567890",
        "From": "+19876543210",
        "Direction": "Inbound",
        "StartTimeInMilliseconds": "159700958834234",
        "Status": "Connected"
      }
    ]
  }
}
```

AWS Lambda 関数をプログラムして、通話の詳細を検証し、将来の使用に備えて保存できます。NEW_INBOUND_CALL イベントの場合、AWS Lambda 関数はウェルカムプロンプトを再生し、会議 PIN を要求する一連のアクションで応答します。

オーディオファイルには、次の要件があります。

- Amazon Simple Storage Service (S3) バケットから音声ファイルを再生する必要があります。S3 バケットは、SIP メディアアプリケーションと同じ AWS アカウントに属している必要があります。さらに、Amazon Chime SDK Voice Connector サービスプリンシパル (`voiceconnector.chime.amazonaws.com`) に `s3:GetObject` 許可を付与する必要があります。このためには、S3 コンソールまたはコマンドラインインターフェイス (CLI) を使用できます。
- 50 MB 以下のサイズの PCM WAV ファイルを使用する必要があります。Amazon Chime SDK では、モノラル 8 kHz が推奨されます。
- 各 WAV ファイルの S3 メタデータには、`{'ContentType': 'audio/wav'}` が含まれている必要があります。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "PlayAudio",
      "Parameters": {
        "CallId": "call-id-1",

        "AudioSource": {
          "Type": "S3",
          "BucketName": "chime-meetings-audio-files-bucket-name",
          "Key": "welcome-to-meetings.wav"
        }
      }
    },
    {
      "Type": "PlayAudioAndGetDigits",
      "Parameters": {
        "ParticipantTag": "LEG-A",

        "AudioSource": {
          "Type": "S3",
          "BucketName": "chime-meetings-audio-files-bucket-name",
          "Key": "enter-meeting-pin.wav"
        },
        "FailureAudioSource": {
          "Type": "S3",
          "BucketName": "chime-meetings-audio-files-bucket-name",
          "Key": "invalid-meeting-pin.wav"
        }
      }
    }
  ]
}
```



```

    },
    "MinNumberOfDigits": 3,
    "MaxNumberOfDigits": 5,
    "TerminatorDigits": ["#"],
    "InBetweenDigitsDurationInMilliseconds": 5000,
    "Repeat": 3,
    "RepeatDurationInMilliseconds": 10000
  }
}
]
}

```

SIP メディアアプリケーションは、通話レグ A でこれらのアクションを実行します。PlayAudioAndGetDigitsアクションが数字を受信すると仮定すると、SIP メディアアプリケーションは ACTION_SUCCESSFUL イベントタイプで AWS Lambda 関数を呼び出します。

```

{
  "SchemaVersion": "1.0",
  "Sequence": 2,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "PlayAudioAndGetDigits",
    "Parameters" : {
      "ParticipantTag": "LEG-A",
      "AudioSource": {
        "Type": "S3",
        "BucketName": "chime-meetings-audio-files-bucket-name",
        "Key": "enter-meeting-pin.wav"
      },
      "FailureAudioSource": {
        "Type": "S3",
        "BucketName": "chime-meetings-audio-files-bucket-name",
        "Key": "invalid-meeting-pin.wav"
      },
      "MinNumberOfDigits": 3,
      "MaxNumberOfDigits": 5,
      "TerminatorDigits": ["#"],
      "InBetweenDigitsDurationInMilliseconds": 5000,
      "Repeat": 3,
      "RepeatDurationInMilliseconds": 10000
    },
    "ReceivedDigits": "12345" // meeting PIN
  },
}

```

```

    "CallDetails": {
      ... // same as in previous event
    }
  }
}

```

データに基づいて発信者を識別するように AWS Lambda 関数をプログラムできま
す CallDetails。以前に受信した会議 PIN を検証することもできます。正しい PIN を想定し、
[CreateMeeting](#) および [CreateAttendee](#) APIs を使用して Amazon Chime SDK 会議を作成し、会議参加
者が使用する参加トークンを生成します。この AWS Lambda 関数は、Amazon Chime SDK 会議に
参加するアクションで応答します。

```

{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "JoinChimeMeeting",
      "Parameters": {
        "JoinToken": "meeting-attendee-join-token"
      }
    }
  ]
}

```

が有効であるとする、SIP メディアアプリケーション JoinToken は Amazon Chime SDK 会議に
参加し、ACTION_SUCCESSFUL イベントで AWS Lambda 関数を呼び出します。には SIP メディア
アプリケーションと Chime メディアサービス (LEG-B) からのデータ CallDetails が含まれます。

```

{
  "SchemaVersion": "1.0",
  "Sequence": 3,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "JoinChimeMeeting",
    "Parameters": {
      "JoinToken": "meeting-attendee-join-token"
    }
  },
  "CallDetails": {
    "TransactionId": "transaction-id",
    "AwsAccountId": "aws-account-id",
    "AwsRegion": "us-east-1",

```

```

    "SipRuleId": "sip-rule-id",
    "SipApplicationId": "sip-application-id",
    "Participants": [
      {
        "CallId": "call-id-1",
        "ParticipantTag": "LEG-A",
        "To": "+11234567890",
        "From": "+19876543210",
        "Direction": "Inbound",
        "StartTimeInMilliseconds": "159700958834234",
        "Status": "Connected"
      },
      {
        "CallId": "call-id-2",
        "ParticipantTag": "LEG-B",
        "To": "SMA",
        "From": "+17035550122",
        "Direction": "Outbound",
        "StartTimeInMilliseconds": "159700958834234",
        "Status": "Connected"
      }
    ]
  }
}

```

この時点で発信または通話レグでのアクションの実行を停止する場合は、空のアクションセットで応答できます。

```

{
  "SchemaVersion": "1.0"
  "Actions": []
}

```

発信者がハングアップすると、SIP メディアアプリケーションは HANGUP イベントで AWS Lambda 関数を呼び出します。

```

{
  "SchemaVersion": "1.0",
  "Sequence": 4,
  "InvocationEventType": "HANGUP",
  "ActionData": {
    "Type": "Hangup",

```

```
    "Parameters": {
      "CallId": "call-id-1",
      "ParticipantTag": "LEG-A"
    }
  },
  "CallDetails": {
    "TransactionId": "transaction-id",
    "AwsAccountId": "aws-account-id",
    "AwsRegion": "us-east-1",
    "SipRuleId": "sip-rule-id",
    "SipApplicationId": "sip-application-id",
    "Participants": [
      {
        "CallId": "call-id-1",
        "ParticipantTag": "LEG-A",
        "To": "+11234567890",
        "From": "+19876543210",
        "Direction": "Inbound",
        "StartTimeInMilliseconds": "159700958834234",
        "Status": "Disconnected"
      },
      {
        "CallId": "call-id-2",
        "ParticipantTag": "LEG-B",
        "To": "SMA",
        "From": "+17035550122",
        "Direction": "Outbound",
        "StartTimeInMilliseconds": "159700958834234",
        "Status": "Disconnected"
      }
    ]
  }
}
```

アクションで Hangup イベントに応答すると、SIP メディアアプリケーションは、他の Participants が Status として Connected を示した場合、そのアクションを無視します。

アクションリストによる呼び出しへの応答

AWS Lambda 呼び出しイベントには、呼び出しの個々の参加者に対して実行するアクションのリストで応答できます。AWS Lambda 呼び出しごとに最大 10 個のアクションで応答でき、呼び出しごとに 1,000 回 AWS Lambda 関数を呼び出すことができます。

デフォルトでは、Lambda 関数が 20 秒経っても応答しない場合、SIP メディアアプリケーションはタイムアウトになります。

以下の例は、一般的な応答構造を示しています。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "PlayAudio",
      "Parameters": {
        "ParticipantTag": "LEG-A",
        "AudioSource": {
          "Type": "S3",
          "BucketName": "bucket-name",
          "Key": "audio-file.wav"
        }
      }
    },
    {
      "Type": "RecordAudio",
      "Parameters": {
        "DurationInSeconds": "10",
        "RecordingTerminators": ["#"],
        "RecordingDestination": {
          "Type": "S3",
          "BucketName": "bucket-name"
        }
      }
    }
  ]
}
```

AWS Lambda 関数が SIP メディアアプリケーションにアクションのリストを返すと、次のオペレーションが発生します。

1. アプリケーションは、通話に対する現在のアクションの実行を終了させます。
2. そして、アプリケーションは、最新の呼び出しイベントから受け取った新しいアクションセットで古いアクションセットを置き換えます。

SIP メディアアプリケーションが NULL アクションセットを受信すると、既存のアクションが保持されます。

PSTN Audio サービスでサポートされるアクション

AWS Lambda 関数からのレスポンスでは、さまざまなタイプのシグナリングアクションとメディアアクションを指定できます。アクションにはそれぞれ異なるプロパティがあります。以下のトピックでは、コード例を示し、アクションの使用方法について説明します。

内容

- [TransactionAttributes を使用する](#)
- [通話録音の使用](#)
- [CallAndBridge](#)
- [Hangup](#)
- [JoinChimeMeeting](#)
- [ModifyChimeMeetingAttendee \(オーディオのミュートとミュート解除\)](#)
- [Pause](#)
- [PlayAudio](#)
- [PlayAudioAndGetDigits](#)
- [ReceiveDigits](#)
- [RecordAudio](#)
- [SendDigits](#)
- [Speak](#)
- [SpeakAndGetDigits](#)
- [StartBotConversation](#)

TransactionAttributes を使用する

データ TransactionAttributes 構造を使用して、通話状態や会議 IDs などのアプリケーション固有の情報を保存し、そのデータを AWS Lambda 呼び出しに渡します。この構造により、Amazon DynamoDB などの外部データベースにデータを保存する必要がなくなります。

TransactionAttributes は、キーと値のペアが含まれる [JSON オブジェクト](#) です。オブジェクトには最大 100 個のキーと値のペアを含めることができます。オブジェクトの最大ペイロードサ

サイズは 20 KB です。TransactionAttributes 構造内のデータは、トランザクションの存続期間中、維持されます。

AWS Lambda 関数が SIP メディアアプリケーションTransactionAttributesに渡すと、アプリケーションは保存された属性を更新します。既存のキーセットが含まれるTransactionAttributes オブジェクトを渡す場合は、保存されている値を更新します。別のキーセットを渡す場合は、既存の値を、その別のキーセットの値に置き換えます。空のマップ ({}) を渡すと、保存されている値はすべて消去されます。

トピック

- [TransactionAttributes の設定](#)
- [TransactionAttributes の更新](#)
- [クリア中 TransactionAttributes](#)
- [ACTION_SUCCESSFUL イベントの処理](#)
- [無効な入力](#)

TransactionAttributes の設定

次の例は、[PlayAudio](#)アクションTransactionAttributesとともに を設定し、AWS Lambda 関数から SIP メディアアプリケーションに属性を渡す方法を示しています。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "PlayAudio",
      "Parameters": {
        "ParticipantTag": "LEG-A",
        "AudioSource": {
          "Type": "S3",
          "BucketName": "mtg1-sipmedia-app-iad",
          "Key": "Welcome3.wav"
        }
      }
    }
  ],
  "TransactionAttributes": {
    "key1": "value1",
    "key2": "value2"
  }
}
```

```
}
```

TransactionAttributes の更新

保存されている TransactionAttributes を変更するには、JSON オブジェクトの内容を新しい値で更新します。次の例では、キー NewKey1 および NewKey2 が TransactionAttributes に追加されています。これらのキーは、値 NewValue1 および NewValue2 とそれぞれペアになっています。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "PlayAudio",
      "Parameters": {
        "ParticipantTag": "LEG-A",
        "AudioSource": {
          "Type": "S3",
          "BucketName": "mtg1-sipmedia-app-iad",
          "Key": "Welcome3.wav"
        }
      }
    }
  ],
  "TransactionAttributes": {
    "NewKey1": "NewValue1",
    "NewKey2": "NewValue2"
  }
}
```

前の例で NewValue1 を key1 に渡した場合、key1 の既存の値は NewValue1 に置き換えられます。ただし、NewKey1 に値を渡すと、新しいキーと値のペアが作成されます。

クリア中 TransactionAttributes

TransactionAttributes オブジェクトの内容をクリアするには、空の JSON オブジェクトが含まれる TransactionAttributes フィールドを渡します。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
```



```
    "Type": "PlayAudio",
    "Parameters": {
      "ParticipantTag": "LEG-A",
      "AudioSource": {
        "Type": "S3",
        "BucketName": "mtg1-sipmedia-app-iad",
        "Key": "Welcome3.wav"
      }
    }
  ],
  "TransactionAttributes": {
  }
}
```

Note

値を null に設定しても、TransactionAttributes 構造からデータをクリアすることはできません。また、TransactionAttribute 構造を省略してもデータはクリアされません。オブジェクトからデータをクリアするには、必ず TransactionAttributes で空の JSON オブジェクトを渡してください。

ACTION_SUCCESSFUL イベントの処理

次の例は、成功した [PlayAudio](#) が、CallDetails の一部として、保存された TransactionAttributes をどのように送信するかを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 2,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "PlayAudio",
    "Parameters": {
      "AudioSource": {
        "Type": "S3",
        "BucketName": "mtg1-sipmedia-app-iad",
        "Key": "Welcome3.wav"
      },
      "Repeat": 1,
      "ParticipantTag": "LEG-A"
    }
  }
}
```

```
    }
  },
  "CallDetails": {
    "TransactionId": "mtg1-tx-id",
    "TransactionAttributes": {
      "key1": "value1",
      "key2": "value2"
    },
    "AwsAccountId": "166971021612",
    "AwsRegion": "us-east-1",
    "SipRuleId": "aafbd402-b7a2-4992-92f8-496b4563c492",
    "SipMediaApplicationId": "e88f4e49-dd21-4a3f-b538-bc84eae11505",
    "Participants": [
      {
        "CallId": "bbff30c5-866a-41b5-8d0a-5d23d5e19f3e",
        "ParticipantTag": "LEG-A",
        "To": "+14345550101",
        "From": "+14255550199",
        "Direction": "Inbound",
        "StartTimeInMilliseconds": "1644539405907",
        "Status": "Connected"
      }
    ]
  }
}
```

無効な入力

無効な入力の例を以下に示します。ここでは、JSON オブジェクトが SIP メディアアプリケーションに渡す項目が多すぎるという状態になっています。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "PlayAudio",
      "Parameters": {
        "ParticipantTag": "LEG-A",
        "AudioSource": {
          "Type": "S3",
          "BucketName": "mtg1-sipmedia-app-iad",
          "Key": "Welcome3.wav"
        }
      }
    }
  ]
}
```

```
    }
  }
],
"TransactionAttributes": {
  "key1": "value1",
  "key2": "value2",
  "key3": "value3",
  "key4": "value4",
  "key5": "value5",
  "key6": "value6",
  "key7": "value7",
  "key8": "value8",
  "key9": "value9",
  "key10": "value10",
  "key11": "value11"
}
}
```

次の例は、前の例の入力に対する応答を示しています。この出力は、SIP メディアアプリケーションから、アプリケーションを呼び出した AWS Lambda 関数に渡されます。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 2,
  "InvocationEventType": "INVALID_LAMBDA_RESPONSE",
  "CallDetails": {
    "TransactionId": "mtg1-tx-id",
    "AwsAccountId": "166971021612",
    "AwsRegion": "us-east-1",
    "SipRuleId": "aafbd402-b7a2-4992-92f8-496b4563c492",
    "SipMediaApplicationId": "e88f4e49-dd21-4a3f-b538-bc84eae11505",
    "Participants": [
      {
        "CallId": "72cbec69-f098-45d8-9ad6-e26cb9af663a",
        "ParticipantTag": "LEG-A",
        "To": "+14345550101",
        "From": "+14255550199",
        "Direction": "Inbound",
        "StartTimeInMilliseconds": "1644540839987"
      }
    ]
  },
  "ErrorType": "TransactionAttributesInvalidMapSize",
}
```

```
"ErrorMessage": "Transaction Attributes has too many mappings. Maximum number of mappings is 10"
}
```

通話録音の使用

SIP メディアアプリケーションの通話録音アクションを使用すると、さまざまな用途に対応する通話録音および通話後文字起こしのソリューションを構築できます。例えば、カスタマーケアの通話を録音してトレーニングに使用できます。

通話録音アクションは SIP メディアアプリケーションと連動して使用します。アクションはオンデマンドまたは SIP イベントへのレスポンスで使用することもできます。

- SIP メディアアプリケーションで通話のオンデマンド録音を開始するには、[UpdateSipMediaApplication](#) API を使用してアプリケーションを呼び出し、[StartCallRecording](#) アクションを返します。
- SIP イベントに応答して通話録音を開始するには、そのアプリケーションで [StartCallRecording](#) アクションを返します。

進行中の録音は、一時停止および再開できます。一時停止するには、[PauseCallRecording](#) アクションを使用します。再開するには、[ResumeCallRecording](#) アクションを使用します。録音を一時停止または再開するたびに、アクションは一時停止または再開を示すトーンをキャプチャします。一時停止すると、アクションは無音を録音します。Amazon Chime SDK はこれを使用して一時停止の長さを追跡し、一時停止分を請求書に含めます。録音は、必要に応じて、一時停止および再開できます。

通話録音を停止するには、[StopCallRecording](#) アクションを返します。ただし、通話が停止すると通話録音は自動的に停止するため、その場合は明示的に [StopCallRecording](#) アクションを返す必要はありません。録音を開始および停止できるのは、1 つの通話レグにつき 1 回だけです。

Amazon Chime SDK は、ユーザーが選択した Amazon S3 バケットに通話録音を配信します。バケットは AWS アカウントに属している必要があります。通話が停止すると、SIP メディアアプリケーションは [StartCallRecording](#) アクションの `Destination` パラメータで指定されたフォルダに録音を配信します。Amazon Chime SDK は、オープンな WAV フォーマットで通話を録音します。着信トラックと発信トラックを録音する通話ではステレオモードを使用します。着信トラックは左チャンネル、発信トラックは右チャンネルです。着信トラックまたは発信トラックだけを録音する場合、システムはモノモードを使用します。

Note

この機能を使用して作成された録音は、電子通信記録に関する法律または規制の対象となる場合があります。録音に関するすべての適用法を遵守することは、お客様とそのエンドユーザーの責任です。これには、録音セッションまたは通信のすべての参加者にセッションまたは通信が録音されていることを適切に通知し、同意を得ることを含みます。

通話録音の請求

Amazon Chime SDK では、通話レグの通話録音が有効になっている時間、およびすべての一時停止を含む時間について、1分単位で請求されます。通話録音が Amazon S3 バケットに配信されると、通話録音の使用料金が請求されます。

オーディオトラックの録音

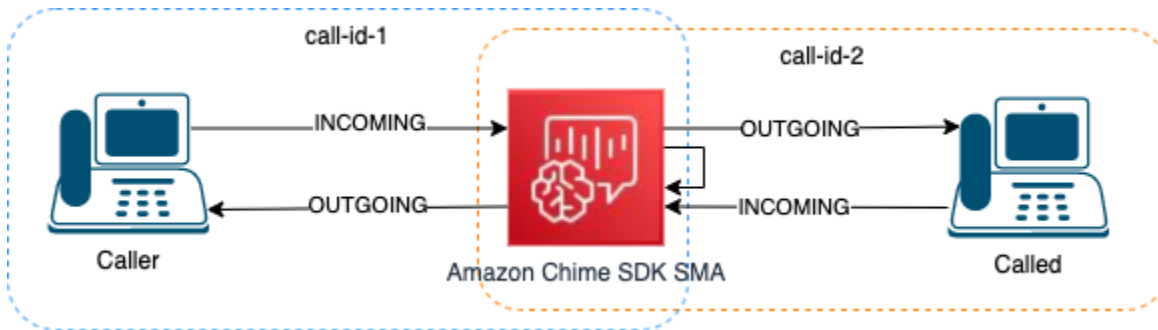
通話の着信トラックまたは発信トラックのみ、または通話の両方のトラックを録音できます。

この画像は、1レグ (ブリッジを使わない) 一般的な着信を示しています。



この通話には callID が call-id-1 である 1レグしかありません。INCOMING オーディオトラックは、発信者から SIP メディアアプリケーションへのオーディオです。OUTGOING オーディオトラックは、SIP メディアアプリケーションから発信者へのオーディオです。SIP メディアアプリケーションは、録音したい通話の CallId を指定します。通話を発信した参加者を録音するには、INCOMING を指定します。通話に応答した参加者を録音するには、OUTGOING を指定します。両方の参加者を録音するには、BOTH を指定します。

この図は、2人の参加者による一般的なブリッジ通話を示しています。



この例では、通話には2つの通話レグ `call-id-1` と `call-id-2` があり、`call-id-1` は `call-id-2` にブリッジされます。これにより、4つのオーディオトラック、つまり両方の通話 ID について着信音声ストリームと発信音声ストリームが作成されます。どの通話 ID とオーディオトラックを録音するかを指定できます。例えば、着信側参加者からのオーディオトラックを録音する場合は、`call-id-2` を `CallId` として、`INCOMING` をトラックとして指定して、`INCOMING` オーディオトラックを録音します。

発信者に聞こえるすべてを録音したい場合は、`call-id-1` を `CallId` として、`OUTGOING` をトラックとして指定して、`OUTGOING` オーディオトラックを録音します。Caller が話したり聞いたりしたオーディオをすべて録音したい場合は、`call-id-1` を `CallId` として、`BOTH` をトラックとして指定して、`BOTH` オーディオトラックを録音します。

サンプルユースケース

SIP メディアアプリケーションは、通話録音アクションを構成要素として提供します。これにより、ビジネスユースケースに合わせた通話録音ソリューションを柔軟に構築できます。以下のケースでは、一般的なユースシナリオを取り上げます。

トピック

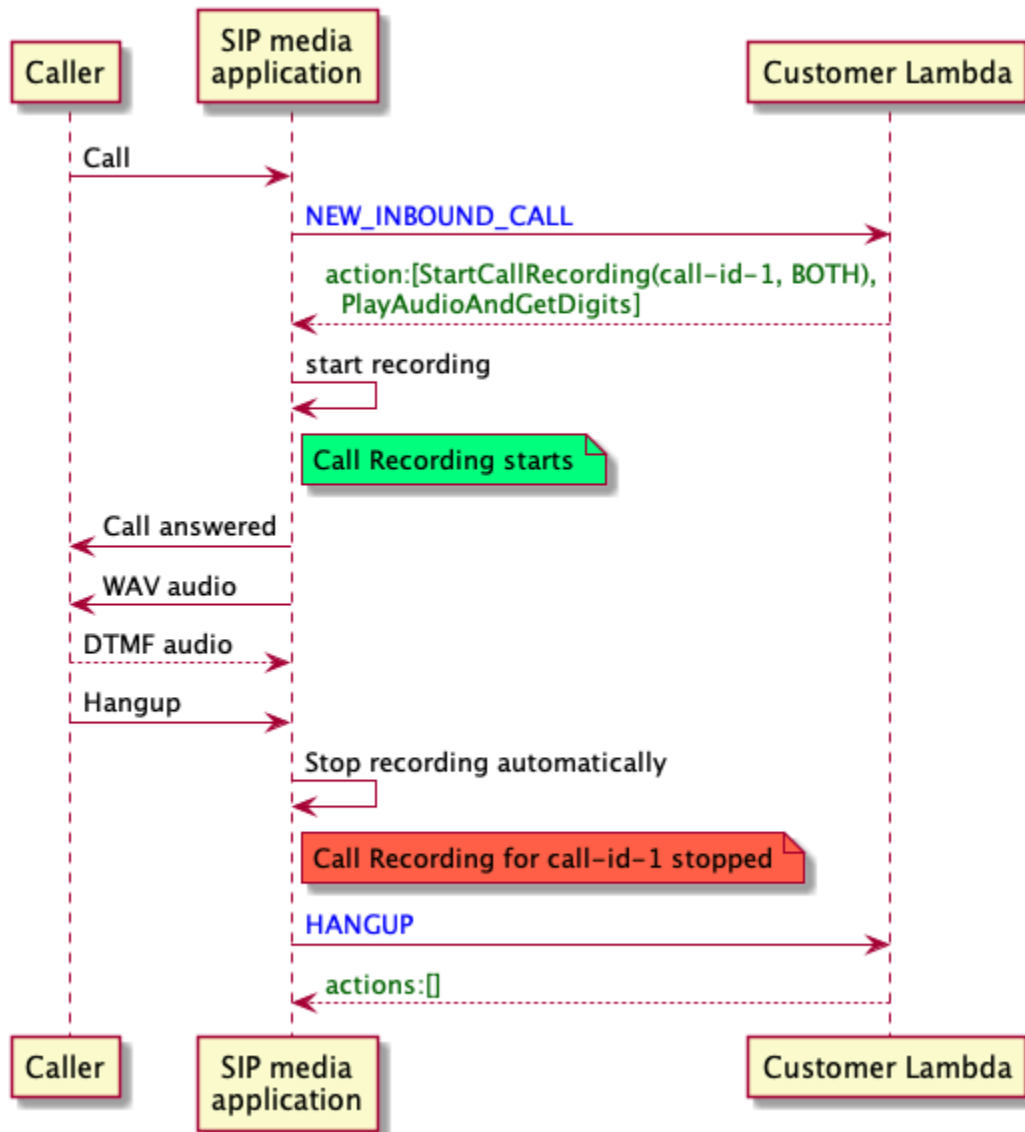
- [ケース 1: SIP アクションを伴う 1 レグ通話の録音](#)
- [ケース 2: ブリッジ通話での音声の選択的な録音](#)
- [ケース 3: 複数通話レグの録音](#)
- [ケース 4: 一時停止と再開を伴うオンデマンド録音](#)

ケース 1: SIP アクションを伴う 1 レグ通話の録音

発信者と、SIP メディアアプリケーションのアクション ([PlayAudio](#) や [PlayAudioAndGetDigits](#) アクションなど) によって生成されるすべての音声を録音できます。録音中に発信者が数字を押すと、その数字のトーンが録音されます。この例では `PlayAudioAndGetDigits` アクションを使用してい

ますが、自動音声応答 (IVR) は SIP メディアアプリケーションの複雑な一連のアクションでもかまいません。

この例では、SIP メディアアプリケーションは、発信者と SIP メディアアプリケーション自体の間の両方のオーディオトラックを録音します。通話確立されると録音が始まり、発信者が切断すると録音は停止します。通話確立されると請求が始まり、発信者が切断すると請求は停止します。

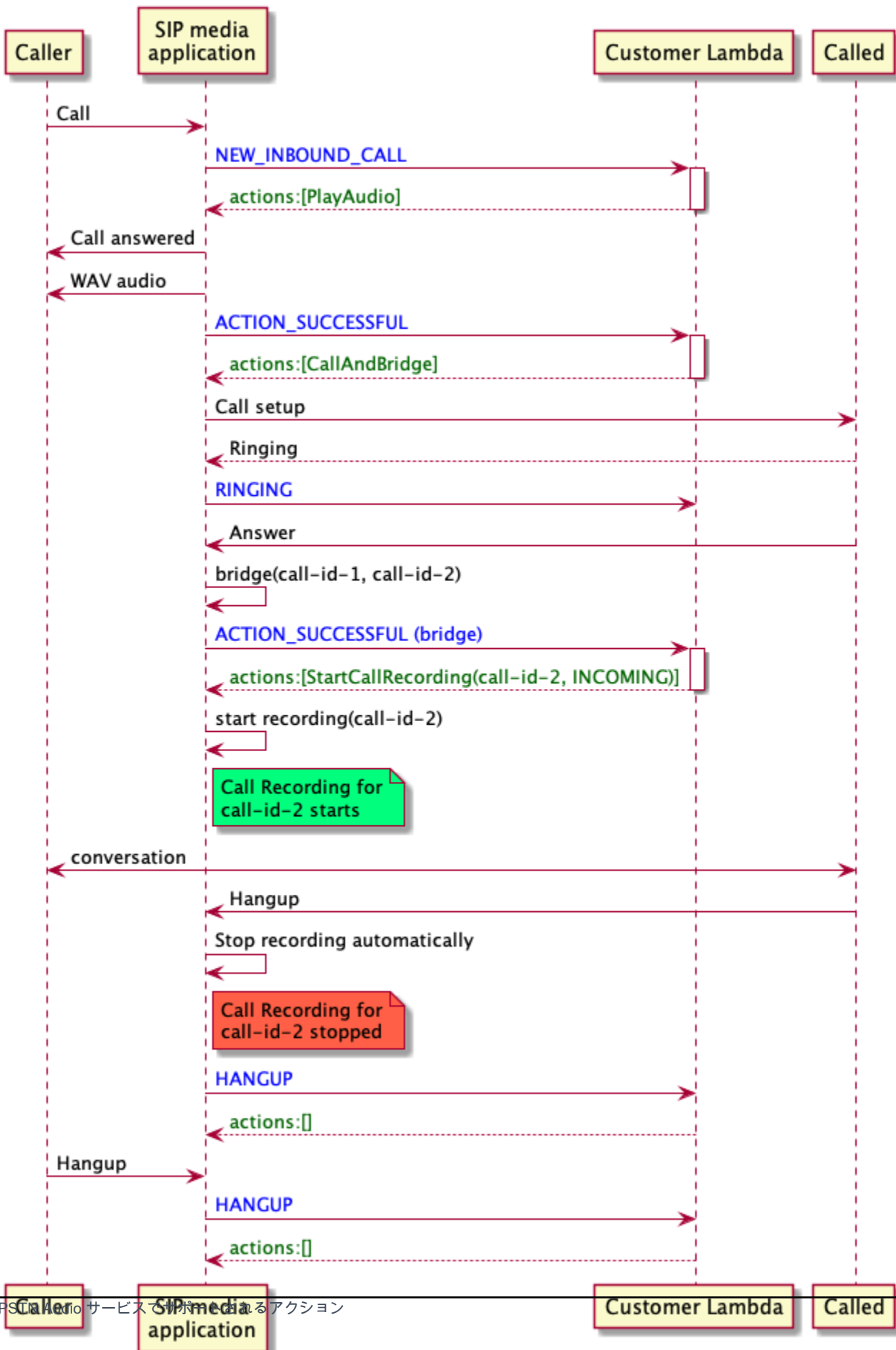


ケース 2: ブリッジ通話での音声の選択的な録音

1 人の通話参加者のオーディオトラックを選択的に録音できます。この機能を使用して、特定の参加者に対してのみ通話の録音を選択的に有効にできます。

この例では、SIP メディアアプリケーションは、call-id-2 を CallId として、INCOMING をトラックとして指定して、着信側と SIP メディアアプリケーション自体の間の着信オーディオトラックを録

音します。通話録音は、発信者が着信側にブリッジ接続された時点で開始され、請求も開始されます。着信側が切断すると録音が停止し、請求も終了します。この録音には着信側のオーディオトラックしか含まれていません。

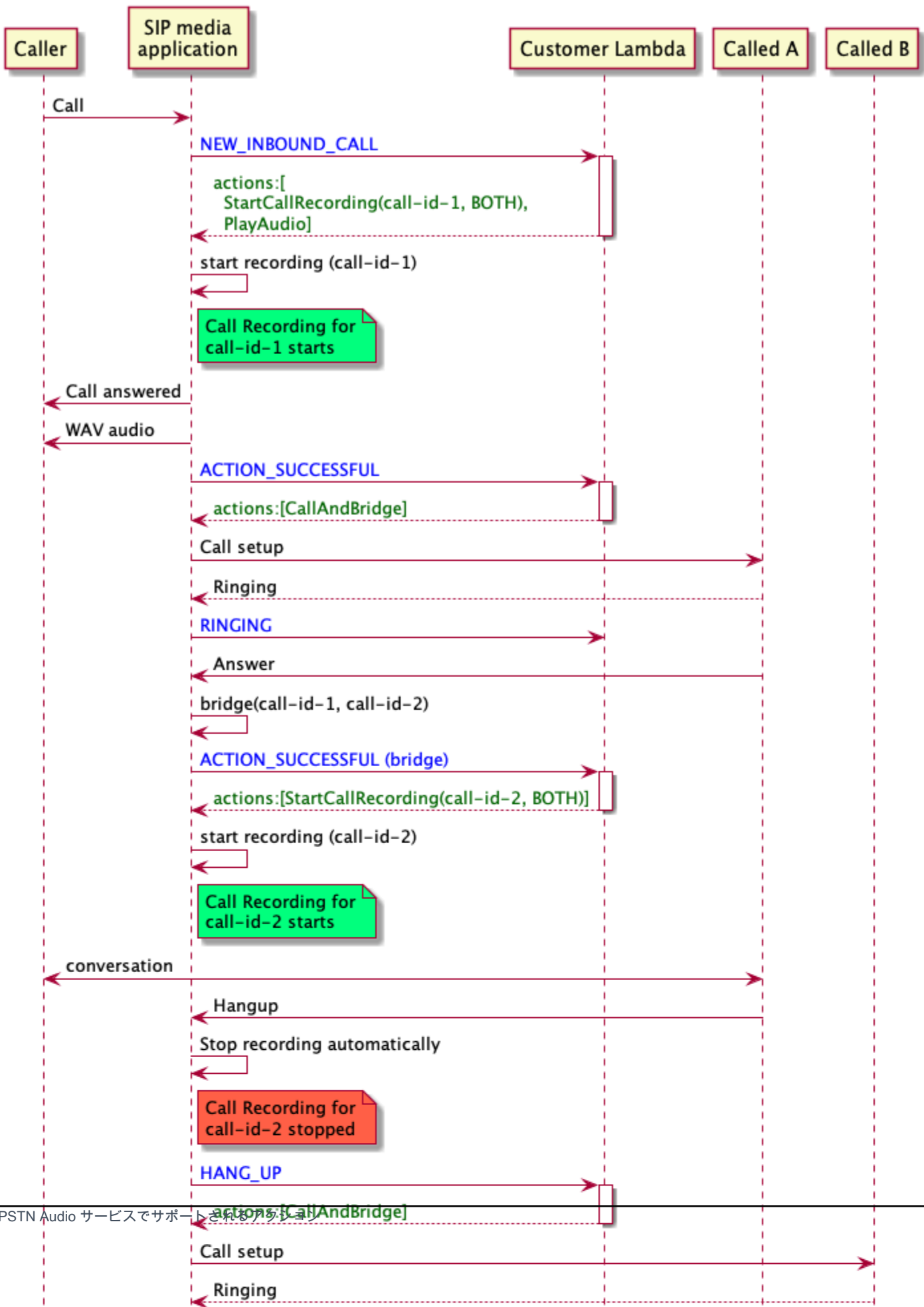


ケース 3: 複数通話レグの録音

複数の通話レグを録音できます。例えば、通話を参加者にブリッジするとします。その参加者が切断すると、通話は別の参加者にブリッジされます。3つの通話レグすべてで通話録音を有効にできます。

この例では、3つの別々の録音ファイルを示しています。最初の通話レグの録音では、発信者、アプリケーション、および通話にブリッジされた2人の参加者の間の会話がキャプチャされます。2番目の通話レグの録音では、発信者と最初の参加者との会話がキャプチャされます。3番目の通話レグの録音では、発信者と2番目の参加者との会話をキャプチャされます。

このケースでは3つの通話レグが作成され、各通話レグの開始から終了まで課金が行われます。別の言い方をすれば、システムが3つの録音をS3バケットに配信し、それぞれに課金されるということです。

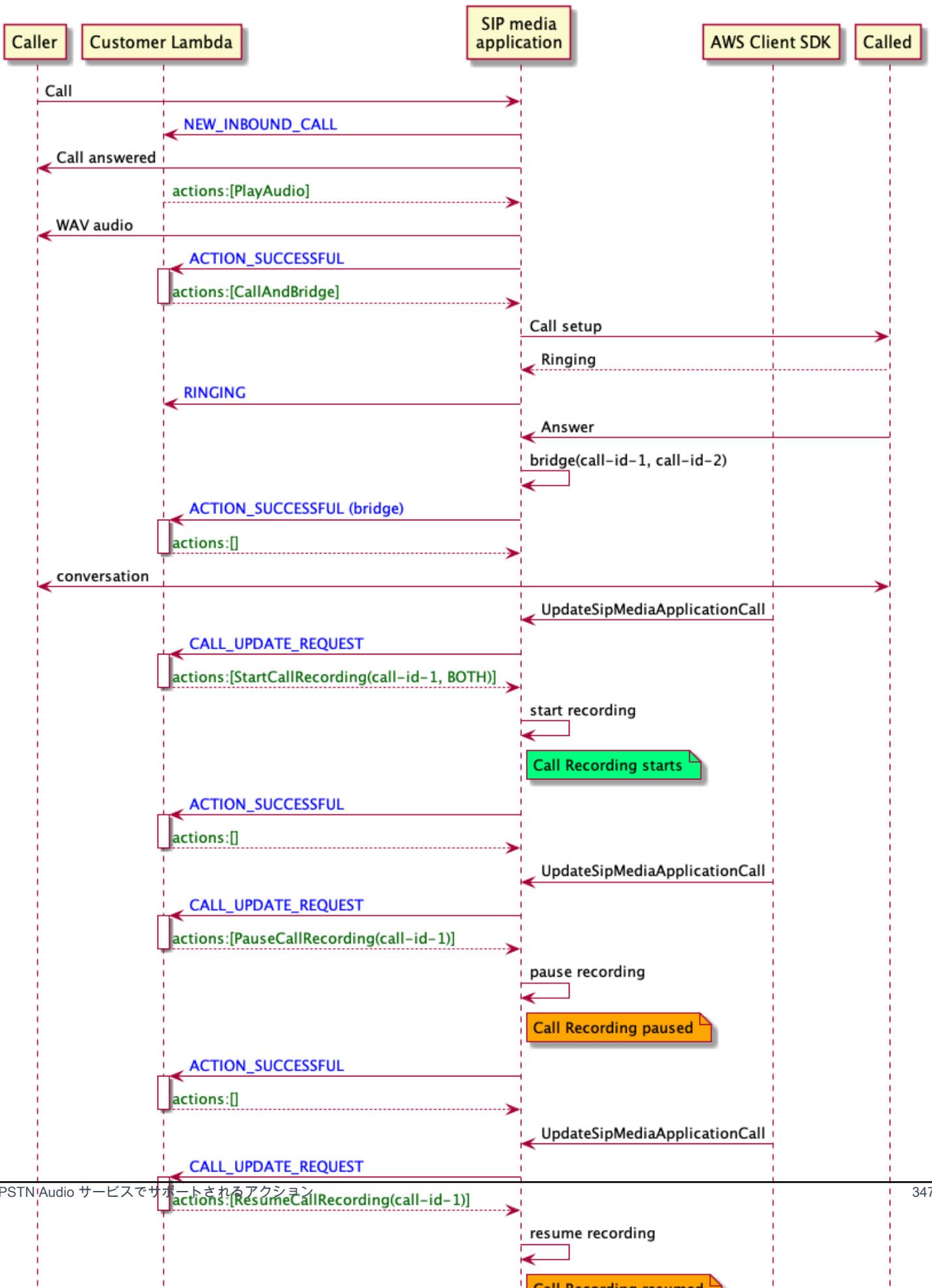


ケース 4: 一時停止と再開を伴うオンデマンド録音

[UpdateSipMediaApplicationCall](#) API を使用して、通話録音をオンデマンドで開始、停止、一時停止、再開できます。UpdateSipMediaApplicationCall API を呼び出して SIP メディアアプリケーションを呼び出して通話録音アクションを返すクライアントアプリケーションを構築できます。

エンドユーザーはクライアントアプリケーションを使用して通話録音を制御します。例えば、コールセンターのエージェントは、デスクトップクライアントアプリケーションを使用して、オンデマンドで通話録音アクションをトリガーします。コールセンターの例では、エージェントは発信者に通話の録音許可を求め、発信者が同意したらクライアントアプリケーションをクリックして録音を開始できます。別の例として、発信者が社会保障番号 (SSN) などの情報を提供する必要がある場合があります。ただし、コールセンターのポリシーでは、担当者は顧客の SSN などの情報を録音してはならないことが要求されています。エージェントは、顧客が情報を提供している間、アプリケーションをクリックして録音を一時停止し、もう一度クリックして録音を再開できます。エージェントが発信者のリクエストを処理すると、エージェントはアプリケーションをクリックして録音を停止し、通話を切断します。

このユースケースでは、SIP メディアアプリケーションは、発信者と SIP メディアアプリケーションの間のオーディオトラックを録音します。call-id-1 レッグは call-id-2 レッグにブリッジされているため、システムは発信者と着信者の両方のレッグの音声を録音します。UpdateSipMediaApplicationCall API が StartCallRecording アクションを呼び出すと、録音と請求が開始します。UpdateSipMediaApplicationCall API が StopCallRecording アクションを呼び出すと、録音と請求が停止します。注意点として、録音を一時停止しても録音時間は変わりません。一時停止していた分もすべて請求されます。



SIP メディアアプリケーションの通話録音アクション

SIP メディアアプリケーションの AWS Lambda 関数からのレスポンスで、異なる通話録音アクションを指定できます。以下のトピックでは、コード例を示し、アクションの使用方法について説明します。

トピック

- [StartCallRecording](#)
- [StopCallRecording](#)
- [PauseCallRecording](#)
- [ResumeCallRecording](#)

StartCallRecording

StartCallRecording アクションは通話レグの録音を開始します。SIP メディアアプリケーションで、オンデマンドまたは SIP イベントへのレスポンスで通話録音を開始します。

- 通話のオンデマンド録音を開始するには、UpdateSipMediaApplication API を使用してアプリケーションを呼び出し、StartCallRecording アクションを返します。
- SIP イベントに応答して通話録音を開始するには、そのアプリケーションで StartCallRecording アクションを返します。

着信レグ、発信レグ、またはその両方のオーディオトラックを録音するかどうかを指定します。このセクションでは、StartCallRecording アクションの使用方法について説明します。

Note

この機能を使用して作成された録音は、電子通信記録に関する法律または規制の対象となる場合があります。録音に関するすべての適用法を遵守することは、お客様とそのエンドユーザーの責任です。これには、録音セッションまたは通信のすべての参加者にセッションまたは通信が録音されていることを適切に通知し、同意を得ることを含みます。

トピック

- [StartCallRecording アクションのリクエスト](#)
- [録音先の指定](#)

- [Amazon S3 バケットのアクセス許可の付与](#)
- [アクション成功レスポンス](#)
- [アクションエラーレスポンス](#)

StartCallRecording アクションのリクエスト

次の例は、BOTHトラックの StartCallRecording アクションをリクエストする方法を示しています。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Type": "StartCallRecording",
      "Parameters": {
        "CallId": "call-id-1",
        "Track": "BOTH",
        "Destination": {
          "Type": "S3",
          "Location": "valid-bucket-name-and-optional-prefix"
        }
      }
    }
  ]
}
```

CallId

説明 — CallId AWS Lambda 関数呼び出しCallDetailsのにおける参加者の数

使用できる値 – 有効な通話 ID

必須 – はい

デフォルト値 – なし

Track

説明 - 通話録音の音声の Track。

使用できる値 – BOTH、INCOMING、または OUTGOING

必須 – はい

デフォルト値 – なし

Destination.Type

説明 – 送信先のタイプ。Amazon S3 のみを使用できます。

使用できる値 – Amazon S3

必須 – はい

デフォルト値 – なし

Destination.Location

説明 - 有効な Amazon S3 バケットとオプションの Amazon S3 キープレフィックス。バケットは、Amazon Chime SDK Voice Connector サービスプリンシパル (voiceconnector.chime.amazonaws.com) へのアクセス許可を持つ必要があります。

使用できる値 – Amazon Chime SDK が s3:PutObject および s3:PutObjectAcl アクションへのアクセス許可を持つ有効な Amazon S3 バケット。

必須 – はい

デフォルト値 – なし

録音先の指定

Amazon Chime SDK は、Amazon S3 バケットに通話録音を配信します。バケットは AWS アカウントに属している必要があります。StartCallRecording アクションの Destination パラメータでバケットの場所を指定します。Destination パラメータの Type フィールドは S3 でなければなりません。Location フィールドは、Amazon S3 バケットと、通話録音が配信されるオプションのオブジェクトキープレフィックスで構成されます。

SIP メディアアプリケーションは、指定された Location、通話レグの日時、トランザクション ID、通話 ID を使用して Amazon S3 オブジェクトキーをフォーマットします。StartCallRecording アクションのレスポンスでは、完全な Amazon S3 オブジェクトキーが返されます。

Location フィールドに Amazon S3 バケットのみを指定すると、SIP メディアアプリケーションは Amazon S3 パスにデフォルトプレフィックス Amazon-Chime-SMA-Call-Recordings を追加し

ます。SIP メディアアプリケーションでは、録音の整理に役立つように、通話の開始時刻の年、月、日も追加されます。次の例は、デフォルトプレフィックスが付いた Amazon S3 パスの一般的な形式を示しています。この例では、`myRecordingBucket` を Location 値として使用します。

```
myRecordingBucket/Amazon-Chime-SMA-Call-Recordings/2019/03/01/2019-03-01-17-10-00-010_c4640e3b-1478-40fb-8e38-6f6213adf70b_7ab7748e-b47d-4620-ae2c-152617d3333c.wav
```

次の例は、通話録音の Amazon S3 パスで表されるデータを示しています。

```
s3Bucket/Amazon-Chime-SMA-Call-Recordings/year/month/date/year-month-date-hour-minute-second-millisecond_transactionId_callId.wav
```

Location フィールドに Amazon S3 バケットとオブジェクトキープレフィックスを指定すると、SIP メディアアプリケーションはデフォルトプレフィックスの代わりに宛先 Amazon S3 パスのオブジェクトキープレフィックスを使用します。次の例は、プレフィックスが付いた通話録音 Amazon S3 パスの一般的な形式を示しています。例えば、`myRecordingBucket/technicalSupport/english` をとして指定できます Location。

```
myRecordingBucket/technicalSupport/english/2019/03/01/2019-03-01-17-10-00-010_c4640e3b1478-40fb-8e38-6f6213adf70b_7ab7748e-b47d-4620-ae2c-152617d3333c.wav
```

次の例は、Amazon S3 パス内のデータを示しています。

```
s3Bucket/yourObjectKeyPrefix/year/month/date/year-month-date-hour-minute-second-millisecond_transactionId_callId.wav
```

Amazon S3 バケットに送信される録音には、通話ログに関する追加の [Amazon S3 オブジェクトメタデータ](#) が含まれます。次の表に、サポートされている Amazon S3 オブジェクトメタデータを示します。

名前	説明
transaction-id	通話のトランザクション ID
call-id	CallId AWS Lambda 関数呼び出し CallDetails のの参加者の
recording-duration	通話録音時間 (秒)

名前	説明
recording-audio-file-format	インターネットメディアタイプとして表される通話録音音声オーディオファイル形式

Amazon S3 バケットのアクセス許可の付与

送信先の Amazon S3 バケットは、アプリケーションと同じ AWS アカウントに属している必要があります。さらに、Amazon Chime SDK Voice Connector サービスプリンシパル `voiceconnector.chime.amazonaws.com` に `s3:PutObject` および `s3:PutObjectAcl` 許可を付与する必要があります。次の例では、適切なアクセス許可を付与しています。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SIP media applicationRead",
      "Effect": "Allow",
      "Principal": {
        "Service": "voiceconnector.chime.amazonaws.com"
      },
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": "arn:aws:s3:::bucket-name/*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "aws-account-id"
        }
      }
    }
  ]
}
```

PSTN Audio サービスは、Sip メディアアプリケーションの代わりに S3 バケットに対して読み書きします。[「混乱した代理」の問題](#)を回避するには、S3 バケットのアクセス許可を単一の SIP メディアアプリケーションに制限できます。

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "SIP media applicationRead",
    "Effect": "Allow",
    "Principal": {
      "Service": "voiceconnector.chime.amazonaws.com"
    },
    "Action": [
      "s3:PutObject",
      "s3:PutObjectAcl"
    ],
    "Resource": "arn:aws:s3:::bucket-name/*",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "aws-account-id",
        "aws:SourceArn": "arn:aws:chime:region:aws-account-id:sma/sip-media-application-id"
      }
    }
  }
]
}

```

アクション成功レスポンス

通話レグで通話録音が正常に開始されると、SIP メディアアプリケーションは ACTION_SUCCESSFUL イベントタイプの AWS Lambda 関数を呼び出します。通話録音の場所がレスポンスで返されます。

```

{
  "SchemaVersion": "1.0",
  "Sequence": INTEGER,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "StartCallRecording",
    "Parameters": {
      "CallId": "call-id-1",
      "Track": "BOTH",
      "Destination": {
        "Type": "S3",
        "Location": "valid-bucket-name"
      }
    }
  }
}

```

```

    }
  }
  "CallRecordingDestination": {
    "Type": "S3",
    "Location": "call-recording-bucket-and-key"
  }
}
"CallDetails": {
  ...
}
}

```

アクションエラーレスポンス

検証エラーの場合、SIP メディアアプリケーションは適切なエラーメッセージで AWS Lambda 関数を呼び出します。次の表に、エラーメッセージを示します。

エラー	メッセージ	理由
InvalidActionParameter	アクションの CallId パラメータが無効です。	パラメータが無効です。
SystemException	アクションの実行中にシステムエラーが発生しました。	アクションの実行中に、別のタイプのシステムエラーが発生しました。

アクションが通話ログでメディアを録音できない場合、SIP メディアアプリケーションは、ActionFailed イベントタイプで AWS Lambda 関数を呼び出します。

次の例は一般的なエラーレスポンスを示しています。

```

{
  "SchemaVersion": "1.0",
  "Sequence": 5,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {
    "Type": "StartCallRecording",
    "Parameters": {
      "CallId": "call-id-1",

```

```
        "Track": "BOTH",
        "Destination": {
            "Type": "S3",
            "Location": "valid-bucket-name"
        }
    }
    "Error": "NoAccessToDestination: Error while accessing destination"
}
"CallDetails": {
    ...
}
}
```

GitHub「<https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>」の実例を参照してください。

StopCallRecording

StopCallRecording アクションは通話レグの録音を停止します。通話が終了すると録音が自動的に停止し、StopCallRecording アプリケーションはアクションを明示的に返す必要はありません。通話レグの録音が停止すると、再開することはできず、録音は StartCallRecording アクションで指定された宛先に配信されます。

次の例では、call-id-1 通話レグの録音を停止します。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "StopCallRecording",
      "Parameters": {
        "CallId": "call-id-1"
      }
    }
  ]
}
```

CallId

説明 — CallId AWS Lambda 関数呼び出し CallDetails のにおける参加者の

使用できる値 – 有効な通話 ID

必須 – はい

デフォルト値 – なし

GitHub 「<https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>」の実例を参照してください。

PauseCallRecording

PauseCallRecording アクションは通話レグの録音を一時停止します。録音を一時停止するたびに、一時停止を示すトーンが録音されます。一時停止しても録音は継続しますが、キャプチャされるのは無音だけです。録音を一時停止しても、録音の合計時間には影響しません。録音は、必要に応じて、一時停止および再開できます。

次の例では、録音を一時停止します。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "PauseCallRecording",
      "Parameters": {
        "CallId": "call-id-1"
      }
    }
  ]
}
```

CallId

説明 — CallId AWS Lambda 関数呼び出しCallDetailsの における参加者の数

使用できる値 – 有効な通話 ID

必須 – はい

デフォルト値 – なし

GitHub 「<https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>」の実例を参照してください。

ResumeCallRecording

ResumeCallRecording アクションは通話レグの録音を再開します。録音が再開される前に、短いトーンが再生されます。通話レグの間は、録音を複数回一時停止および再開できます。

次の例では、録音を再開します。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "ResumeCallRecording",
      "Parameters": {
        "CallId": "call-id-1"
      }
    }
  ]
}
```

CallId

説明 — CallId AWS Lambda 関数呼び出しCallDetailsのにおける参加者の数

使用できる値 – 有効な通話 ID

必須 – はい

デフォルト値 – なし

GitHub 「<https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>」の実例を参照してください。

CallAndBridge

PSTN 電話番号へ、または Amazon Chime SDK Voice Connector または Amazon Chime SDK Voice Connector グループとして設定された SIP トランクへの発信通話を作成し、それを既存の通話レグとブリッジします。電話番号に電話をかけるときは PSTN を使用し、SIP トランクに電話をかけるときは AWS を使用します。

既存の通話レグは、[CreateSIPMediaApplicationCall](#) API を使用して作成されたアウトバウンド通話レグ、またはNewInboundCallイベントで関数を呼び出す AWS Lambda SIP ルールによって

作成されたインバウンドレグです。Voice Connector または Voice Connector グループのエンドポイントに CallAndBridge アクションを実装する場合、Voice Connector または Voice Connector グループの Amazon リソース番号 (ARN) を指定する必要があります。

カスタム SIP ヘッダーを発信通話レグと AWS Lambda 関数に追加することもできます。カスタムヘッダーを使用すると、フロア番号や郵便番号などの値を渡すことができます。カスタムヘッダーの詳細については、「[SIP ヘッダーの使用](#)」を参照してください。

次のコード例は、PSTN エンドポイントにブリッジする一般的なアクションを示しています。

```
{
  "SchemaVersion": "1.0",
  "Actions": [{
    "Type": "CallAndBridge",
    "Parameters": {
      "CallTimeoutSeconds": 30,
      "CallerIdNumber": "e164PhoneNumber", // required
      "Endpoints": [{
        "BridgeEndpointType": "PSTN", // required
        "Uri": "e164PhoneNumber", // required
      }],
    }
  ]
}
```

次の例は、Voice Connector または Voice Connector グループとカスタム SIP ヘッダーを使用する一般的なアクションを示しています。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "CallAndBridge",
      "Parameters": {
        "CallTimeoutSeconds": 30,
        "CallerIdNumber": "e164PhoneNumber", // required
        "RingbackTone": { // optional
          "Type": "S3",
          "BucketName": "s3_bucket_name",
          "Key": "audio_file_name"
        }
      }
    }
  ]
}
```



```
    "Endpoints": [
      {
        "BridgeEndpointType": "AWS", // enum type, required

        "Arn": "arn:aws:chime:us-
east-1:0123456789101:vc/abcdefghijklm2nopq3rs" //VC or VCG ARN, required for AWS
endpoints
        "Uri": "ValidString", // required, see description below
      }
    ],
    "SipHeaders": {
      "x-String": "String"
    }
  }
]
```

CallTimeoutSeconds

説明 - 通話がタイムアウトするまでの間隔。タイマーは通話のセットアップから始まります。

使用できる値 - 1 ~ 120 (両端を含む)

必須 - いいえ

デフォルト値 - 30

CallerIdNumber

説明 - 顧客の番号、または A レッグの From 番号

使用できる値 - E.164 形式の有効な電話番号

必須 - はい

デフォルト値 - なし

Endpoints

説明 - 通話のエンドポイント

許可される値:

- BridgeEndpointType - Voice Connector および Voice Connector グループの場合は AWS、それ以外の場合は PSTN。

- Arn – Voice Connector または Voice Connector グループの ARN。AWS を BridgeEndpointType として使用する場合にのみ必須です。
- Uri – URI 値はエンドポイントのタイプによって異なります。

PSTN エンドポイントの場合、URI は有効な E.164 電話番号でなければなりません。

AWS エンドポイントでは、URI 値によって Request-URI の user 部分が設定されます。[拡張バックスナウア記法を使用する必要があります](#)。必要な長さ: 1~30 (両端を含む)。値は a-z, A-Z, 0-9, &, =, +, \$, /, %, -, _, !, ~, *, (,), (.) を使用します。

Request-URI のホスト値は、ターゲットの Voice Connector の着信ルートから算出されます。次の例は、AWS エンドポイントを使用する CallAndBridge アクションを示しています。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "CallAndBridge",
      "Parameters": {
        "CallTimeoutSeconds": 30,
        "CallerIdNumber": "+18005550122",
        "Endpoints": [
          {
            "BridgeEndpointType": "AWS",
            "Arn": "arn:aws:chime:us-east-1:0123456789101:vc/abcdefghijklm2nopq3rs",
            "Uri": "5550"
          }
        ],
        "SipHeaders": {
          "x-String": "String"
        }
      }
    }
  ]
}
```

着信ルートと Voice Connector の詳細については、「[Editing Amazon Chime SDK Voice Connector settings](#)」を参照してください。

必須 – はい

デフォルト値 – なし

SipHeaders

説明 - 追加の値を渡すことができます。AWS エンドポイントタイプでのみ使用してください。

使用できる値 – 有効な SIP ヘッダー

必須 – いいえ

デフォルト値 – なし

次の例は、PSTN エンドポイントを使用する成功した CallAndBridge アクションを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 3,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "CallAndBridge",
    "Parameters": {
      "CallTimeoutSeconds": 30,
      "CallerIdNumber": "e164PhoneNumber",
      "Endpoints": [
        {
          "BridgeEndpointType": "PSTN",
          "Uri": "e164PhoneNumber"
        }
      ],
      "CallId": "call-id-1"
    }
  },
  "CallDetails": {
    .....
    .....
    "Participants": [
      {
        "CallId": "call-id-1",
        "ParticipantTag": "LEG-A",
        .....
        "Status": "Connected"
      },
      {
```

```

        "CallId": "call-id-2",
        "ParticipantTag": "LEG-B",
        .....
        "Status": "Connected"
    }
  ]
}
}

```

次の例は、失敗した CallAndBridge アクションを示しています。

```

{
  "SchemaVersion": "1.0",
  "Sequence": 2,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {
    "Type": "CallAndBridge",
    "Parameters": {
      "CallTimeoutSeconds": 30,
      "CallerIdNumber": "e164PhoneNumber",
      "Endpoints": [
        {
          "BridgeEndpointType": "PSTN",
          "Uri": "e164PhoneNumber"
        }
      ],
      "CallId": "call-id-1"
    },
    "ErrorType": "CallNotAnswered",
    "ErrorMessage": "Call not answered"
  },
  "CallDetails": {
    .....
    .....
    "Participants": [
      {
        "CallId": "call-id-1",
        "ParticipantTag": "LEG-A",
        .....
      }
    ]
  }
}
}

```

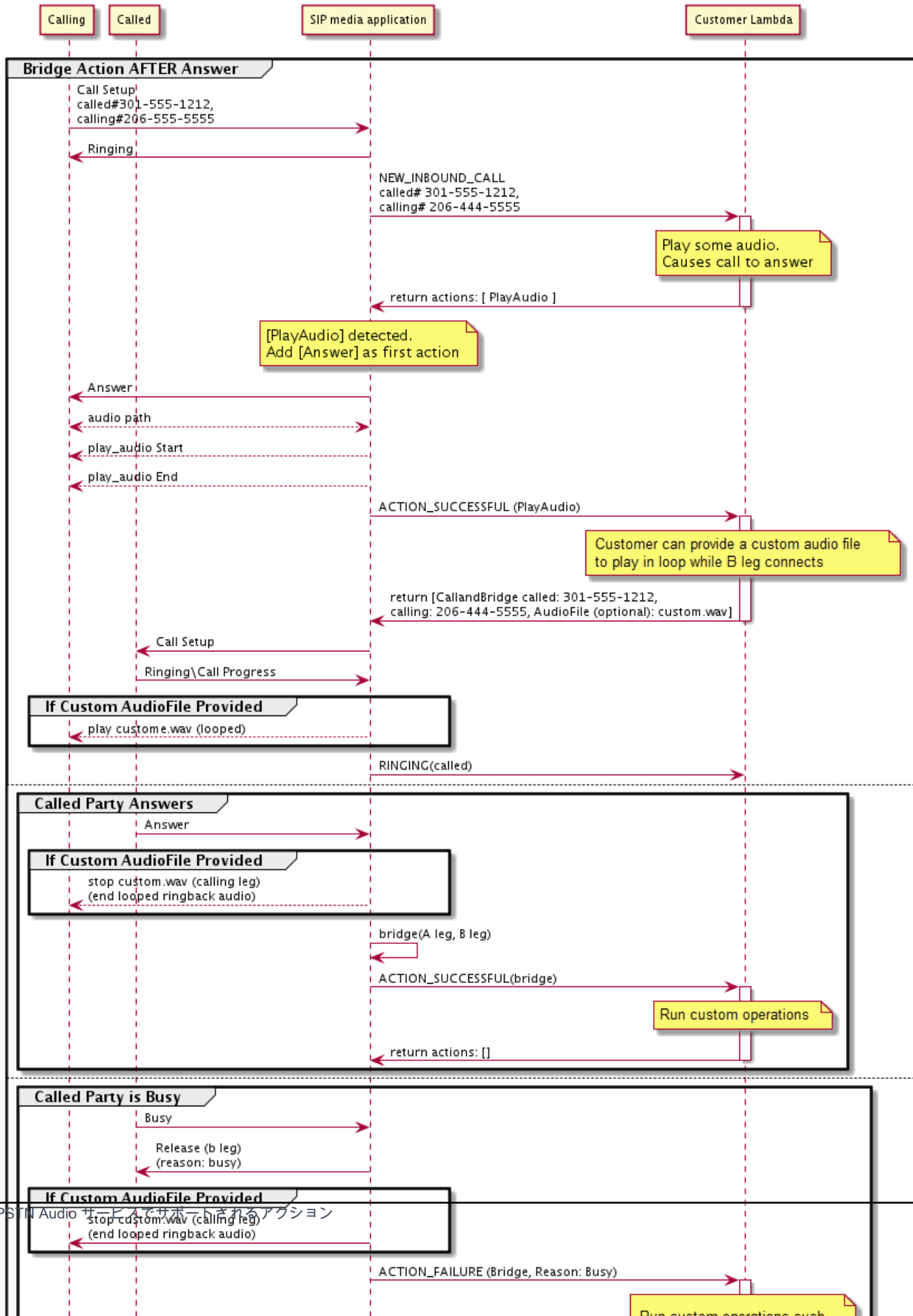
通話フロー

CallAndBridge アクションは、パラメータやレグの接続有無に応じて、既存の通話レグとは異なる通話発信とオーディオエクスペリエンスを提供します。

次の図は、着信通話レグ A が既に接続されている場合のさまざまなパラメータを使用した通話フローを示しています。

CallandBridge

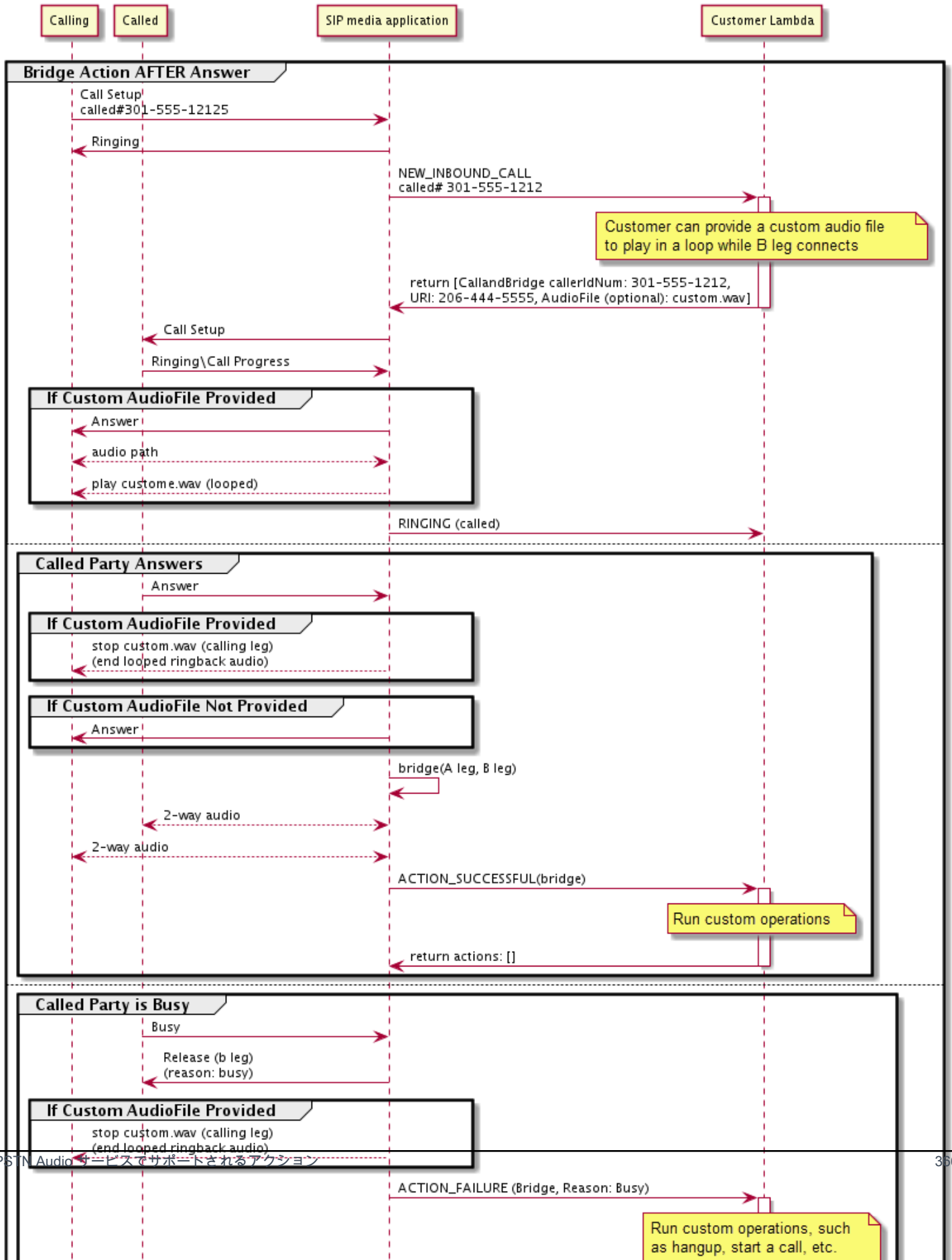
A leg has already been answered. Bridge A leg with B leg only once B leg is answered



次の図は、未応答通話の通話フローを示しています。

CallandBridge

A leg has NOT been answered. Bridge A leg with B leg only once B leg is answered



その他の詳細

CallAndBridge アクションについて、以下の点に注意してください。

- `CallTimeoutSeconds` – このタイマーは、SIP 招待が B レッグで送信されたときに開始されます。希望するターゲット値を設定できますが、上流キャリアはこの値を無視できます。
- `CallerIdNumber` – この電話番号は顧客に属するか、A レッグの送信元番号である必要があります。
- 切断動作とエッジケース – 一方の通話レッグが切断しても、もう一方の通話レッグは自動的に通話を切断しません。Hangup イベントが AWS Lambda 関数に送信されると、残りのレッグは個別に切断する必要があります。通話レッグが保留状態のままの場合、通話は切断されるまで請求されます。例えば、次のシナリオでは予期しない料金が発生する可能性があります。
- 宛先電話番号にブリッジしようとしています。宛先が通話中であり、通話をボイスメールに直接送信します。Audio サービス側から見ると、ボイスメールへの送信は応答された通話です。A レッグは切断しますが、B レッグは引き続きボイスメールメッセージを待機します。B レッグが待機している間、請求されます。
- ベストプラクティスとして、AWS Lambda 関数、または通話のもう一方の側を使用して、各通話レッグを個別に切断します。
- 請求 – CallAndBridge を使用すると、次の料金が請求されます。
 - PSTN に作成された各通話レッグ (A レッグ、B レッグなど) のアクティブな通話時間 (分)。
 - Audio サービスの使用時間 (分)。

の作業例を参照してください [GitHub](#)。

- <https://github.com/aws-samples/amazon-chime-sma-bridging>
- <https://github.com/aws-samples/amazon-chime-sma-call-forwarding>
- <https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>

Hangup

`SipStatusCode` と共に `Hangup` 値を通話の任意のレッグに送信します。

Audio Service が通話レッグで Hangup アクションを実行すると、以下の処理が行われます。

- 通話レッグが 1 つしかない通話の場合、SIP メディアアプリケーションは HANGUP イベントで AWS Lambda 関数を呼び出し、レスポンスを無視します。そして、通話が切断されます。

- 別の通話レグ (レグ B) にブリッジされている通話レグ (レグ A) については、ブリッジされた通話レグ (レグ B) に Hangup アクションが関連付けられている場合、PSTN 音声サービスはブリッジされた通話レグを切断し、レグ B の HANGUP イベントで Lambda 関数を呼び出します。PSTN 音声サービスは、その Lambda 呼び出しから返されたアクションを実行します。
- 別の通話レグ (レグ B) にブリッジされている通話レグ (レグ A) については、元の通話レグ (レグ A) に Hangup アクションが関連付けられている場合、PSTN 音声サービスは元の通話レグを切断し、レグ A の HANGUP イベントで Lambda 関数を呼び出します。PSTN 音声サービスは、その Lambda 呼び出しから返されたアクションを実行します。
- JoinMeeting アクションを使用して会議に参加した通話レグについては、会議レグ (通常はレグ B) に Hangup アクションが関連付けられている場合、発信者は会議から切断され、Hangup アクションの ACTION_SUCCESSFUL イベントを受信します。

一般的な Hangup アクションの例を次に示します。

```
{
  "Type": "Hangup",
  "Parameters": {
    "CallId": "call-id-1",
    "ParticipantTag": "LEG-A",
    "SipResponseCode": "0"
  }
}
```

CallId

説明 — CallId AWS Lambda 関数呼び出し CallDetails のにおける参加者の

使用できる値 – 有効な通話 ID

必須 – いいえ

デフォルト値 – なし

ParticipantTag

説明 – CallDetails に接続されている参加者のうちの 1 人の ParticipantTag。

使用できる値 – LEG-A または LEG-B

必須 – いいえ

デフォルト値 - 呼び出された callLeg の ParticipantTag (CallId を指定すると無視されま
す)

SipResponseCode

説明 - サポートされている任意の SIP レスポンスコード

使用できる値 - 480 - 使用不可、486 - 話中、0 - 正常終了

必須 - いいえ

デフォルト値 - 0

ユーザーが通話を終了すると、SIP メディアアプリケーションは にリストされているコードを使用
して AWS Lambda 関数を呼び出します [発信の終了](#)。

の作業例を参照してください GitHub。

- <https://github.com/aws-samples/amazon-chime-sma-bridging>
- <https://github.com/aws-samples/amazon-chime-sma-call> 転送
- <https://github.com/aws-samples/amazon-chime-sma-outbound-call-notifications>
- <https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>

JoinChimeMeeting

出席者参加トークンを指定して、Amazon Chime SDK ミーティングに参加します。これを行うに
は、 [CreateMeeting](#) および [CreateAttendee](#) APIs に AWS SDK 呼び出しを行い、トークンを取得し、
アクションで渡します。次の例を参照してください。

Note

このアクションは、ブリッジ通話では実行できません。

```
{
  "Type": "JoinChimeMeeting",
  "Parameters": {
    "JoinToken": "meeting-attendee-join-token",
    "CallId": "call-id-1",
    "ParticipantTag": "LEG-A",
```

```
    "MeetingId": "meeting-id"  
  }  
}
```

JoinToken

説明 - Amazon Chime SDK ミーティング出席者の有効な参加トークン

使用できる値 - 有効な参加トークン

必須 - はい

デフォルト値 - なし

CallId

説明 — AWS Lambda 関数呼び出しCallDetailsの におけるCallId参加者の

使用できる値 - 有効な通話 ID

必須 - いいえ

デフォルト値 - なし

ParticipantTag

説明 - CallDetails に接続されている参加者のうちの 1 人の ParticipantTag。

使用できる値 - LEG-A

必須 - いいえ

デフォルト値 - 呼び出された callLeg の ParticipantTag (CallId を指定すると無視されます)

MeetingId

説明 - JoinToken に関連付けられている有効な Amazon Chime SDK ミーティング ID。 [Amazon Chime](#) 名前空間の API を使用して会議が作成された場合、会議 ID は不要です。 [Amazon Chime SDK ミーティング](#) 名前空間の API を使用して会議が作成された場合は、会議 ID が必要です。会議に参加するには、会議の作成に使用された API エンドポイントを使用します。

使用できる値 - 有効な会議 ID

必須 - いいえ。

デフォルト値 – なし。

SIP メディアアプリケーションは、このアクションを実行した後、常に AWS Lambda 関数を呼び出します。そして、ACTION_SUCCESSFUL または ACTION_FAILED 呼び出しイベントタイプが返されます。次の例は、成功呼び出しイベントの構造を示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 4,
  "InvocationEvent": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "JoinChimeMeeting",
    "Parameters": {
      "JoinToken": "meeting-attendee-join-token",
      "CallId": "call-id-1"
      "ParticipantTag": "LEG-A"
    }
  }
  "CallDetails": {
    ...
  }
}
```

エラー処理

会議のブリッジ中に検証エラーが発生すると、SIP アプリケーションは、次の表に示すエラーメッセージのいずれかを使用して AWS Lambda 関数を呼び出します。

エラー	メッセージ	理由
InvalidActionParameter	JoinToken パラメータの値が無効です。	アクションの他のパラメータのいずれかが無効か、欠落しています。
SystemException	アクションの実行中にシステムエラーが発生しました。	アクションの実行中に、別のタイプのシステムエラーが発生しました。

次の例は、一般的な失敗イベントを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 3,
  "InvocationEvent": "ACTION_FAILED",
  "ActionData": {
    "Type": "JoinChimeMeeting",
    "Parameters": {
      "JoinToken": "meeting-attendee-join-token",
      "CallId": "call-id-1",
      "ParticipantTag": "LEG-A"
    },
    "Error": "ErrorJoiningMeeting: Error while joining meeting."
  },
  "CallDetails": {
    ...
  }
}
```

「<https://github.com/aws-samples/amazon-chime-sma-update-call>」の実例を参照してください
GitHub。

ModifyChimeMeetingAttendee (オーディオのミュートとミュート解除)

Amazon Chime SDK の会議 ID および出席者リストを提供することで、SIP メディアアプリケーションがテレフォニー出席者のステータスを変更できるようにします。

Note

このアクションは現在、テレフォニー出席者に対するミュート操作およびミュート解除操作をサポートしています。また、ユーザーは、JoinChimeMeeting アクションを使用して会議に参加する必要があります。このアクションは、participantTag="LEG-B" に対して、または対応する CallId に対して実行できます。

このアクションが適用されるのは、SIP メディアアプリケーションから "+13605550122" に参加する callLeg、LEG-B、または SIP メディアアプリケーションから会議に参加するレッグのみです。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
```

```
{
  "Type" : "ModifyChimeMeetingAttendees",
  "Parameters" : {
    "Operation": "Mute",
    "MeetingId": "meeting-id",
    "CallId": "call-id",
    "ParticipantTag": "LEG-B",
    "AttendeeList": ["attendee-id-1", "attendee-id-2"]
  }
}
```

Operation

説明 - 出席者リストに対して実行する操作

使用できる値 - Mute、Unmute

必須 - はい

デフォルト値 - なし

MeetingId

説明 - 出席者が属する会議の ID

使用できる値 - 有効な会議 ID ミュートまたはミュート解除するユーザーも会議に属している必要があります。

必須 - はい

デフォルト値 - なし

CallId

説明 - 出席者が属する会議の ID

使用できる値 - 有効な通話 ID。

必須 - いいえ

デフォルト値 - なし

ParticipantTag

説明 - 出席者に割り当てられたタグ

使用できる値 - 有効なタグ

必須 - いいえ

デフォルト値 - なし

AttendeeList

説明 - ミュートまたはミュート解除する出席者 ID のリスト

使用できる値 - 有効な出席者 ID のリスト

必須 - はい

デフォルト値 - なし、最大 100 件

このアクションを実行すると、Audio Service は常に、ACTION_SUCCESSFUL または ACTION_FAILED 呼び出しイベントタイプの AWS Lambda 関数を呼び出します。次のコード例は、一般的な ACTION_SUCCESSFUL 呼び出しイベントを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": INTEGER,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "ModifyChimeMeetingAttendees",
    "Parameters": {
      "Operation": "Mute",
      "MeetingId": "meeting-id",
      "CallId": "call-id",
      "ParticipantTag": "LEG-B",
      "AttendeeList": ["attendee-id-1", "attendee-id-2"]
    }
  }
  "CallDetails": {
    ...
  }
}
```


エラー処理

無効な命令パラメータまたは API 障害が発生した場合、SIP メディアアプリケーションは、失敗した命令または API に固有のエラーメッセージを含む AWS Lambda 関数を呼び出します。

エラー	メッセージ	理由
InvalidActionParameter	ModifyChimeMeeting Attendees Operation パラメータの値が無効です。	Operation 値は Mute または Unmute でなければなりません。
	会議 ID のパラメータ値が無効です。	会議 ID が空です。
	出席者リストのパラメータ値が無効です。	出席者 ID リストが空か、上限の 100 件を超えています。
	通話中のアクションが無効です。	通話がブリッジされていません。
	通話が Chime Meeting に接続されていません。	出席者が Chime Meeting に接続されていません。
	One or more attendees are not part of this meeting. All attendees must be part of this meeting.	この出席者には、会議の出席者を変更する権限がありません。
SystemException	アクションの実行中にシステムエラーが発生しました。	アクションの実行中に、システムエラーが発生しました。

次のコード例は、一般的な失敗イベントを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": INTEGER,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {
    "Type": "ModifyChimeMeetingAttendees",
```

```
    "Parameters" : {
      "Operation": "Mute",
      "MeetingId": "meeting-id",
      "CallId": "call-id",
      "ParticipantTag": "LEG-B",
      "AttendeeList": ["attendee-id-1", "attendee-id-2"]
    },
    "ErrorType": "",
    "ErrorMessage": "",
    "ErrorList": []
  }
  "CallDetails": {
    ...
  }
}
```

の作業例を参照してください [GitHub](#)。

- <https://github.com/aws-samples/amazon-chime-sma-bridging>。
- <https://github.com/aws-samples/amazon-chime-sma-update-call>

Pause

指定した時間だけ通話を一時停止します。

```
{
  "Type": "Pause",
  "Parameters": {
    "CallId": "call-id-1",
    "ParticipantTag": "LEG-A",
    "DurationInMilliseconds": "3000"
  }
}
```

CallId

説明 — AWS Lambda 関数呼び出しCallDetailsの におけるCallId参加者の

使用できる値 – 有効な通話 ID

必須 – いいえ

デフォルト値 – なし

ParticipantTag

説明 – CallDetails に接続されている参加者のうちの 1 人の ParticipantTag。

使用できる値 – LEG-A または LEG-B

必須 – いいえ

デフォルト値 – 呼び出された callLeg の ParticipantTag (CallId を指定すると無視されます)

DurationInMilliseconds

説明 – 一時停止の時間 (ミリ秒単位)

使用できる値 – 整数 > 0

必須 – はい

デフォルト値 – なし

の実例を参照してください [GitHub](#)。

- <https://github.com/aws-samples/amazon-chime-sma-outbound-call-notifications>
- <https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>

PlayAudio

通話の任意のレグでオーディオファイルを再生します。オーディオは何回でも繰り返すことができます。進行中のオーディオは、PlaybackTerminators で設定された DTMF デイジットを使用して終了できます。

現在、Amazon Chime SDK は Amazon Simple Storage Service (Amazon S3) バケットからのオーディオファイルの再生のみをサポートしています。S3 バケットは、SIP メディアアプリケーションと同じ AWS アカウントに属している必要があります。さらに、Amazon Chime SDK Voice Connector サービスプリンシパルに s3:GetObject 許可を付与する必要があります。この操作は、S3 コンソールまたはコマンドラインインターフェイス (CLI) を使用して実行できます。

次のコード例は、一般的なバケットポリシーを示しています。

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "SMARead",
    "Effect": "Allow",
    "Principal": {
      "Service": "voiceconnector.chime.amazonaws.com"
    },
    "Action": [
      "s3:GetObject"
    ],
    "Resource": "arn:aws:s3:::bucket-name/*",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "aws-account-id"
      }
    }
  }
]
}

```

Audio Service は、Sip メディアアプリケーションの代わりに S3 バケットに対して読み書きします。[「混乱した代理」の問題](#)を回避するには、S3 バケットアクセスを単一の SIP メディアアプリケーションに制限できます。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SMARead",
      "Effect": "Allow",
      "Principal": {
        "Service": "voiceconnector.chime.amazonaws.com"
      },
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::bucket-name/*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "aws-account-id",
          "aws:SourceArn": "arn:aws:chime:region:aws-account-id:sma/sip-media-application-id"
        }
      }
    }
  ]
}

```

```
    }  
  }  
]  
}
```

次のコード例は、一般的なアクションを示しています。

```
{  
  "Type": "PlayAudio",  
  "Parameters": {  
    "CallId": "call-id-1",  
    "ParticipantTag": "LEG-A",  
    "PlaybackTerminators": ["1", "8", "#"],  
    "Repeat": "5",  
    "AudioSource": {  
      "Type": "S3",  
      "BucketName": "valid-S3-bucket-name",  
      "Key": "wave-file.wav"  
    }  
  }  
}
```

CallId

説明 – CallDetails の参加者の CallId。

使用できる値 – 有効な通話 ID。

必須 – ParticipantTag が存在する場合は、いいえ。

デフォルト値 – なし。

ParticipantTag

説明 – CallDetails で接続されている参加者のうちの 1 人の ParticipantTag。

使用できる値 – LEG-A または LEG-B。

必須 – CallId が存在する場合は、いいえ。

デフォルト値 – 呼び出した callLeg の ParticipantTag。CallId を指定すると無視されま
す。

PlaybackTerminator

説明 – ユーザーからの DTMF 入力を使用して、進行中のオーディオを終了します。

使用できる値 – 「0」、「1」、「2」、「3」、「4」、「5」、「6」、「7」、「8」、「9」、「#」、「*」の値の配列

必須 – いいえ

デフォルト値 – なし

Repeat

説明 – 指定した回数だけオーディオを繰り返します。

使用できる値 – 0 より大きい整数

必須 – いいえ

デフォルト値 – 1

AudioSource.Type

説明 – オーディオファイルのソースタイプ。

使用できる値 – S3。

必須 – はい。

デフォルト値 – なし。

AudioSource.BucketName

説明 – S3 ソースタイプの場合、S3 バケットは SIP アプリケーションと同じ AWS アカウントに属している必要があります。バケットは、Amazon Chime SDK Voice Connector サービスプリンシパル (voiceconnector.chime.amazonaws.com) にアクセスできる必要があります。

使用できる値 – Amazon Chime SDK が `s3:GetObject` アクションにアクセスできる有効な S3 バケット。

必須 – はい。

デフォルト値 – なし。

AudioSource.key

説明 – S3 ソースタイプの場合は、AudioSource.BucketName 属性で指定された S3 バケットのファイル名。

使用できる値 – 有効なオーディオファイル。

必須 – はい。

デフォルト値 – なし。

SIP メディアアプリケーションは、ソース URL からオーディオの再生を試みます。50 MB 以下の未圧縮の PCM .wav ファイルを使用できます。Amazon Chime SDK では、モノラル 8 kHz が推奨されます。

ダイヤルプランの最後の指示が PlayAudio で、ファイルが再生を終了するか、ユーザーがキーを押すと再生を停止すると、アプリケーションは次の例に示すイベントで AWS Lambda 関数を呼び出します。

```
{
  "SchemaVersion": "1.0",
  "Sequence": INTEGER,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "PlayAudio",
    "Parameters": {
      "CallId": "call-id-1",
      "AudioSource": {
        "Type": "S3",
        "BucketName": "valid-S3-bucket-name",
        "Key": "wave-file.wav",
      }
    }
  }
}
```

有限小数によってオーディオが停止した後は、繰り返されません。

エラー処理

検証ファイルにエラーが含まれている場合、またはアクションの実行中にエラーが発生した場合、SIP メディアアプリケーションは適切なエラーコードを使用して AWS Lambda 関数を呼び出します。

エラー	メッセージ	理由
InvalidAudioSource	オーディオソースパラメータが無効です。	このエラーは、さまざまな理由で発生します。例えば、許可や URL の問題により、SIP メディアアプリケーションがファイルにアクセスできなくなります。または、形式、継続時間、サイズなどにより、オーディオファイルの検証に失敗することがあります。
SystemException	アクションの実行中にシステムエラーが発生しました。	アクションの実行中に、別のシステムエラーが発生しました。
InvalidActionParameter	CallId アクションの または ParticipantTag パラメータが無効です。	アクションに無効なパラメータが含まれています。

次のコード例は、一般的な呼び出し失敗を示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 2,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {
    "Type": "PlayAudio",
    "Parameters" : {
      "CallId": "call-id-1",
      "AudioSource": {
        "Type": "S3",
        "BucketName": "bucket-name",
        "Key": "audio-file.wav"
      }
    },
  },
  "ErrorType": "InvalidAudioSource",
  "ErrorMessage": "Audio Source parameter value is invalid."
}
```



```
"CallDetails": {  
    ...  
}  
}
```

の実例を参照してください [GitHub](#)。

- <https://github.com/aws-samples/amazon-chime-sma-bridging>。
- <https://github.com/aws-samples/amazon-chime-sma-call> 転送
- <https://github.com/aws-samples/amazon-chime-sma-outbound-call-notifications>
- <https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>
- <https://github.com/aws-samples/amazon-chime-sma-update-call>

PlayAudioAndGetDigits

音声を再生して DTMF デイジットを収集します。ユーザーが正しい DTMF デイジット数を入力しなかったなどの失敗が発生した場合、このアクションは「失敗」音声を再生し、Repeat パラメータで定義された試行回数を SIP メディアアプリケーションが使い切るまでメイン音声を再生します。

S3 バケットから音声ファイルを再生する必要があります。S3 バケットは、SIP メディアアプリケーションと同じ AWS アカウントに属している必要があります。さらに、[Amazon Chime SDK Voice Connector サービスプリンシパル](#) (`voiceconnector.chime.amazonaws.com`) に `s3:GetObject` 許可を付与する必要があります。これを行うには、S3 コンソールまたは CLI を使用できます。

次のコード例は、一般的な S3 バケットポリシーを示しています。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "SMARead",  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "voiceconnector.chime.amazonaws.com"  
      },  
      "Action": [  
        "s3:GetObject"  
      ],  
      "Resource": "arn:aws:s3:::bucket-name/*",  
      "Condition": {
```

```

        "StringEquals": {
            "aws:SourceAccount": "aws-account-id"
        }
    }
}
]
}

```

Audio Service は、Sip メディアアプリケーションの代わりに S3 バケットに対して読み書きします。[「混乱した代理」の問題](#)を回避するには、S3 バケットアクセスを単一の SIP メディアアプリケーションに制限できます。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SMARead",
      "Effect": "Allow",
      "Principal": {
        "Service": "voiceconnector.chime.amazonaws.com"
      },
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::bucket-name/*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "aws-account-id",
          "aws:SourceArn": "arn:aws:chime:region:aws-account-id:sma/sip-media-application-id"
        }
      }
    }
  ]
}

```

一般的な PlayAudioAndGetDigits アクションの例を次に示します。

```

{
  "Type" : "PlayAudioAndGetDigits",
  "Parameters" : {
    "CallId": "call-id-1",

```

```
    "ParticipantTag": "LEG-A"
    "InputDigitsRegex": "^\\d{2}#$",
    "AudioSource": {
      "Type": "S3",
      "BucketName": "bucket-name",
      "Key": "audio-file-1.wav"
    },
    "FailureAudioSource": {
      "Type": "S3",
      "BucketName": "bucket-name",
      "Key": "audio-file-failure.wav"
    },
    "MinNumberOfDigits": 3,
    "MaxNumberOfDigits": 5,
    "TerminatorDigits": ["#"],
    "InBetweenDigitsDurationInMilliseconds": 5000,
    "Repeat": 3,
    "RepeatDurationInMilliseconds": 10000
  }
}
```

CallId

説明 — AWS Lambda 関数呼び出しCallDetailsの におけるCallId参加者の数

使用できる値 – 有効な通話 ID

必須 – いいえ

デフォルト値 – なし

ParticipantTag

説明 – CallDetails に接続されている参加者のうちの 1 人の ParticipantTag。

使用できる値 – LEG-A または LEG-B

必須 – いいえ

デフォルト値 – 呼び出された callLeg の ParticipantTag (CallId を指定すると無視されます)

InputDigitsRegex

説明 - 正規表現パターン

使用できる値 - 有効な正規表現パターン

必須 - いいえ

デフォルト値 - なし

AudioSource.Type

説明 - 音声ファイルタイプのソースのタイプ

使用できる値 - S3 バケット

必須 - はい

デフォルト値 - "S3"

AudioSource.BucketName

説明 - S3 AudioSource.Type値の場合、S3 バケットは SIP メディアアプリケーションと同じ AWS アカウントに属している必要があります。S3 バケットは、[Amazon Chime SDK Voice Connector サービスプリンシパル](#)である `voiceconnector.chime.amazonaws.com` にアクセスできる必要があります。

使用できる値 - Amazon Chime SDK が `s3:GetObject` アクションにアクセスできる有効な S3 バケット。

必須 - はい

デフォルト値 - なし

AudioSource.Key

説明 - AudioSource.BucketName S3 バケット内の音声オブジェクトのキー名。

使用できる値 - 有効な音声ファイル

必須 - はい

デフォルト値 - なし

FailureAudioSource.Type

説明 - FailureAudioSource.BucketName S3 バケット内の音声オブジェクトのキー名。

使用できる値 – S3

必須 – はい

デフォルト値 – なし

FailureAudioSource.BucketName

説明 – S3 ソースタイプの場合、S3 バケツは SIP メディアアプリケーションと同じ AWS アカウントに属している必要があります。[Amazon Chime SDK Voice Connector サービスプリンシパル](#)である `voiceconnector.chime.amazonaws.com` は S3 バケツにアクセスできる必要があります。

使用できる値 – Amazon Chime SDK が `s3:GetObject` アクションにアクセスできる有効な S3 バケツ。

必須 – はい

デフォルト値 – なし

FailureAudioSource.Key

説明 - `FailureAudioSource.BucketName` S3 バケツ内の音声オブジェクトのキー名。

使用できる値 - 有効な音声ファイル

必須 – はい

デフォルト値 – なし

MinNumberOfDigits

説明 - タイムアウトになるか「通話失敗」の音声再生される前にキャプチャする最小ディジット数。

使用できる値 – ≥ 0

必須 – いいえ

デフォルト値 - 0

MaxNumberOfDigits

説明 - 終了ディジットなしで停止する前にキャプチャする最大ディジット数。

使用できる値 - >MinNumberOfDigits

必須 - いいえ

デフォルト値 - 128

TerminatorDigits

説明 - ユーザーが MaxNumberOfDigits 未満の値を入力した場合に、入力を終了するのに使用されるディジット。

使用できる値 - 次のディジットのいずれか: 0123456789#*

必須 - いいえ

デフォルト値 - #

InBetweenDigitsDurationInMilliseconds

説明 - FailureAudio を再生するまでのディジット入力間の待ち時間 (ミリ秒単位)。

使用できる値 - 0 より大きい値

必須 - いいえ

デフォルト値 - 指定しない場合、デフォルトは RepeatDurationInMilliseconds 値になります。

Repeat

説明 - ディジットの取得を試みた回数の合計。

使用できる値 - 0 より大きい値

必須 - いいえ

デフォルト値 - 1

RepeatDurationInMilliseconds

説明 - Repeat の試行間の待ち時間 (ミリ秒単位)。

使用できる値 - 0 より大きい値

必須 – はい

デフォルト値 – なし

SIP メディアアプリケーションは、ACTION_SUCCESSFULまたは呼び出しイベントタイプを使用して、PlayAudioAndGetDigitsアクションを実行した後、常にその AWS Lambda 関数ACTION_FAILEDを呼び出します。このアプリケーションは、ディジットの収集に成功すると、ActionData オブジェクトに ReceivedDigits 値を設定します。次の例は、その AWS Lambda 関数の呼び出しイベント構造を示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 3,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "PlayAudioAndGetDigits",
    "Parameters" : {
      "CallId": "call-id-1",
      "ParticipantTag": "LEG-A",
      "InputDigitsRegex": "^\\d{2}#$",
      "AudioSource": {
        "Type": "S3",
        "BucketName": "bucket-name",
        "Key": "audio-file-1.wav"
      },
      "FailureAudioSource": {
        "Type": "S3",
        "BucketName": "bucket-name",
        "Key": "audio-file-failure.wav"
      },
      "MinNumberOfDigits": 3,
      "MaxNumberOfDigits": 5,
      "TerminatorDigits": ["#"],
      "InBetweenDigitsDurationInMilliseconds": 5000,
      "Repeat": 3,
      "RepeatDurationInMilliseconds": 10000
    },
    "ErrorType": "InvalidAudioSource",
    "ErrorMessage": "Audio Source parameter value is invalid."
  },
  "ReceivedDigits": "1234"
},
```

```

    "CallDetails": {
      ...
    }
  }
}

```

エラー処理

検証エラーが発生すると、SIP メディアアプリケーションは、対応するエラーメッセージで AWS Lambda 関数を呼び出します。次の表に、発生する可能性のあるエラーメッセージを示します。

エラー	メッセージ	理由
InvalidAudioSource	Audio Source parameter value is invalid.	このエラーは、さまざまな理由で発生します。例えば、許可や S3 バケットの問題により、SIP メディアアプリケーションがファイルにアクセスできなくなります。または、継続時間、サイズ、サポートされていない形式により、音声ファイルの検証に失敗することがあります。
InvalidActionParameter	アクションの CallId または ParticipantTag パラメータが無効です。	CallId、ParticipantTag、またはそれ以外のパラメータが無効です。
SystemException	アクションの実行中にシステムエラーが発生しました。	アクションの実行中にシステムエラーが発生しました。

タイムアウトまたは再試行回数が多すぎるため、アクションが指定された桁数を収集できない場合、SIP メディアアプリケーションは ACTION_FAILED 呼び出しイベントタイプで AWS Lambda 関数を呼び出します。

```

{
  "SchemaVersion": "1.0",
  "Sequence": 4,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {

```



```
"Type": "PlayAudioAndGetDigits",
"Parameters" : {
  "CallId": "call-id-1",
  "ParticipantTag": "LEG-A",
  "InputDigitsRegex": "^\\d{2}#$",
  "AudioSource": {
    "Type": "S3",
    "BucketName": "bucket-name",
    "Key": "audio-file-1.wav"
  },
  "FailureAudioSource": {
    "Type": "S3",
    "BucketName": "bucket-name",
    "Key": "audio-file-failure.wav"
  },
  "MinNumberOfDigits": 3,
  "MaxNumberOfDigits": 5,
  "TerminatorDigits": ["#"],
  "InBetweenDigitsDurationInMilliseconds": 5000,
  "Repeat": 3,
  "RepeatDurationInMilliseconds": 10000
},
"ErrorType": "InvalidAudioSource",
"ErrorMessage": "Audio Source parameter value is invalid."
}
"CallDetails": {
  ...
}
}
```

の作業例を参照してください GitHub。

- <https://github.com/aws-samples/amazon-chime-sma-bridging>。
- <https://github.com/aws-samples/amazon-chime-sma-update-call>

ReceiveDigits

ユーザーがこのアクションで指定された正規表現パターンに一致する数字を入力すると、SIP メディアアプリケーションは AWS Lambda 関数を呼び出します。

```
{
  "Type": "ReceiveDigits",
```

```
"Parameters": {
  "CallId": "call-id-1",
  "ParticipantTag": "LEG-A",
  "InputDigitsRegex": "^d{2}#$",
  "InBetweenDigitsDurationInMilliseconds": 1000,
  "FlushDigitsDurationInMilliseconds": 10000
}
```

CallId

説明 — AWS Lambda 関数呼び出しCallDetailsの におけるCallId参加者の

使用できる値 – 有効な通話 ID

必須 – いいえ

デフォルト値 – なし

ParticipantTag

説明 – CallDetails に接続されている参加者のうちの 1 人の ParticipantTag。

使用できる値 – LEG-A または LEG-B

必須 – いいえ

デフォルト値 – 呼び出された callLeg の ParticipantTag (CallId を指定すると無視されます)

InputDigitsRegex

説明 - 正規表現パターン

使用できる値 - 有効な正規表現パターン

必須 – はい

デフォルト値 – なし

InBetweenDigitsDurationInMilliseconds

説明 - 入力が正規表現パターンと一致するかどうかを確認する前のデジタルの間隔

使用できる値 - 所要時間 (ミリ秒単位)

必須 – はい

デフォルト値 – なし

FlushDigitsDurationInMilliseconds

説明 – 受信した DTMF デイジットがフラッシュされて AWS Lambda 関数に送信される間隔。間隔が終了した後、SIP メディアアプリケーションが新しいデイジットを受信すると、タイマーが再び開始されます。

使用できる値 – InBetweenDigitsDurationInMilliseconds

必須 – はい

デフォルト値 – なし

SIP メディアアプリケーションは、新しい ReceiveDigits アクションを受信するまで、通話中の DTMF デイジットを破棄します。FlushDigitsDurationInMilliseconds 間隔は、SIP メディアアプリケーションが最初の DTMF デイジットを受信したときから始まります。間隔が期限切れになる前にユーザーが正しい桁を入力すると、SIP メディアアプリケーションは [で説明されている AWS Lambda 関数を呼び出します](#) [発信者の入力の受信](#)。

ユーザー入力が正規表現パターンと一致しない場合、SIP メディアアプリケーションは、アプリケーションが繰り返し回数を使い切るか、ユーザーが有効なデイジットを入力するまで、「失敗」音声ファイルメッセージを繰り返します。

の実例を参照してください [GitHub](#)。

- <https://github.com/aws-samples/amazon-chime-sma-outbound-call-notifications>
- <https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>
- <https://github.com/aws-samples/amazon-chime-sma-update-call>

RecordAudio

SIP メディアアプリケーションが、特定の通話 ID からメディアを録音できるよう許可します。例えば、音声メールアプリケーションや、会議参加者のお知らせなどです。アプリケーションは、設定した時間に達するまで、またはユーザーがいずれかの RecordingTerminators を押すか、アプリケーションが無音を検出するまで録音を実行します。このような場合、アクションは、結果のメディアファイルを指定された S3 バケットに配置するようアプリケーションに指示します。S3 バケットは、SIP メディアアプリケーションと同じ AWS アカウントに属している必要があります。さらに、アクションは、Amazon Chime SDK Voice Connector サービスプリンシパル、[Amazon Chime SDK](#)

[Voice Connector サービスプリンシパル](#)、に `s3:PutObject` および `アクセスs3:PutObjectAcl` 許可を付与する必要があります `voiceconnector.chime.amazonaws.com`。

Note

この機能を使用して作成された録音は、電子通信記録に関する法律または規制の対象となる場合があります。録音に関するすべての適用法を遵守することは、お客様とそのエンドユーザーの責任です。これには、録音セッションまたは通信のすべての参加者にセッションまたは通信が録音されていることを適切に通知し、同意を得ることを含みます。

次の例では、Amazon Chime SDK Voice Connector サービスプリンシパルに `s3:PutObject` および `s3:PutObjectAcl` 許可を付与します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SMARead",
      "Effect": "Allow",
      "Principal": {
        "Service": "voiceconnector.chime.amazonaws.com"
      },
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": "arn:aws:s3:::bucket-name/*"
    }
  ]
}
```

次の例では、発信者がシャープキー (#) を押したとき、アクティビティなしで 10 秒経過したとき、または発信者が 3 秒間無音状態になったときに録音を停止し、RecordingDestination パラメータで定義された場所に、結果のメディアファイルを書き込みます。

Note

この例では CallId パラメータを使用します。代わりに ParticipantTag パラメータを使用できますが、両方を使用することはできません。

```
{
  "Type": "RecordAudio",
  "Parameters": {
    "CallId": "call-id-1",
    "DurationInSeconds": "10",
    "SilenceDurationInSeconds": 3,
    "SilenceThreshold": 100,
    "RecordingTerminators": [
      "#"
    ],
    "RecordingDestination": {
      "Type": "S3",
      "BucketName": "valid-bucket-name",
      "Prefix": "valid-prefix-name"
    }
  }
}
```

CallId

説明 — CallId AWS Lambda 関数呼び出しCallDetailsの における参加者の

使用できる値 – 有効な通話 ID

必須 – いいえ

デフォルト値 – なし

ParticipantTag

説明 – CallDetails に接続されている参加者のうちの 1 人の ParticipantTag。

使用できる値 – LEG-A または LEG-B

必須 – いいえ

デフォルト値 – 呼び出された callLeg の ParticipantTag (CallId を指定すると無視されます)

RecordingDestination.Type

説明 – 送信先のタイプ。S3 のみ。

使用できる値 – S3

必須 – はい

デフォルト値 – なし

RecordingDestination.BucketName

説明 – 有効な S3 バケット名。バケットは、[Amazon Chime SDK Voice Connector サービスプリンシパル](#)である `voiceconnector.chime.amazonaws.com` にアクセスできる必要があります。

使用できる値 – Amazon Chime SDK が `s3:PutObject` および `s3:PutObjectAcl` アクションにアクセスできる有効な S3 バケット。

必須 – はい

デフォルト値 – なし

RecordingDestination.Prefix

説明 – 録音ファイルの S3 プレフィックス

使用できる値 – 979 文字までのセーフ文字を含む有効なプレフィックス名。セーフ文字の詳細については、Amazon Simple Storage Service ユーザーガイドの「[セーフ文字](#)」を参照してください。

必須 – いいえ

デフォルト – なし。指定しない場合、録音は S3 バケットのルートに保存されます。

DurationInSeconds

説明 – 録音の長さ (秒単位)。

使用できる値 – 0 より大きい値

必須 – いいえ

デフォルト値 – なし

SilenceDurationInSeconds

説明 – 録音が停止するまでの無音の継続時間 (秒単位)。指定しない場合、無音検出は無効になります。

使用できる値 – [1;1000]

必須 – いいえ

デフォルト値 – 200

SilenceThreshold

説明 – 「無音」と見なされるノイズのレベル。SilenceDurationInSeconds を指定しない場合、このパラメータは無視されます。

基準値 (ノイズを無音として扱うためのノイズレベルとしきい値):

- 1 – 30 dB 以下 (静かな部屋など)
- 100 – 40 ~ 50 dB (ささやきや静かなオフィスなど)
- 200 – 60 dB (混雑したオフィスなど)
- 1000 – 75 dB (大声や音楽など)

使用できる値 – [1;1000]

必須 – いいえ

デフォルト値 – 200

RecordingTerminators

説明 – 使用可能なすべての録音ターミネータを一覧表示します。

使用できる値 – [123456789*0#] の 1 桁の数字と記号の配列

必須 – はい

デフォルト値 – なし

ACTION_SUCCESS イベントの処理

録音が終了すると、Amazon Chime SDK SIP メディアアプリケーションは AWS Lambda 関数を呼び出し、呼び出し結果とともに ACTION_SUCCESSFUL イベントを渡します。

```
{
  "SchemaVersion": "1.0",
  "Sequence": INTEGER,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type" : "RecordAudio",
    "Parameters": {
      ...
    }
  },
}
```

```
    "RecordingDestination": {
      "Type": "S3",
      "BucketName": "valid-bucket-name",
      "Key": "valid-S3-key"
    },
    "RecordingTerminatorUsed": "#"
  },
  "CallDetails": {
    ...
  }
}
```

ACTION_SUCCESSFUL イベントには、ActionData が含まれます (以下のフィールドがあります)。

Type

説明 – アクションのタイプ (RecordAudio)。

Parameters

説明 – アクションのパラメータ。

RecordingDestination.Type

説明 – 送信先のタイプ。S3 のみ。

RecordingDestination.BucketName

説明 – 録音ファイルが含まれる S3 バケット。

RecordingDestination.Key

説明 – 録音ファイルの S3 キー。

RecordingTerminatorUsed

説明 – 録音を停止するために使用されるターミネーター (RecordingTerminators パラメータに渡されるいずれかのターミネーター)。最大持続時間 (DurationInSeconds) に達したか、無音 (SilenceDurationInSeconds) により録音が停止した場合、このキーと値のペアは出力に含まれません。

エラー処理

検証エラーの場合、SIP メディアアプリケーションは適切なエラーメッセージで AWS Lambda 関数を呼び出します。次の表に、発生する可能性のあるエラーメッセージを示します。

エラー	メッセージ	理由
InvalidActionParameter	<p>アクションの CallId または ParticipantTag パラメータが無効です。</p> <p>DurationInSeconds パラメータの値が無効です。</p> <p>SilenceDurationInSeconds パラメータの値が無効です。</p> <p>SilenceThreshold パラメータの値が無効です。</p> <p>RecordingDestination パラメータの値が無効です。</p> <p>S3 バケットへの録音のアップロード中にエラーが発生しました。</p>	パラメータが無効です。
SystemException	アクションの実行中にシステムエラーが発生しました。	アクションの実行中に、別のタイプのシステムエラーが発生しました。

ACTION_FAILED イベントの処理

アクションが通話レグにメディアを記録できない場合、SIP メディアアプリケーションは ACTION_FAILED イベントタイプで AWS Lambda 関数を呼び出します。次の例を参照してください。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 5,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {
    "Type": "RecordAudio",
```

```
    "Parameters": {
      ...
    },
    "ErrorType": "InvalidActionParameter",
    "ErrorMessage": "RecordingDestination parameter value is invalid."
  },
  "CallDetails": {
    ...
  }
}
```

「」の実例<https://github.com/aws-samples/amazon-chime-sma-bridging> GitHub。

SendDigits

任意の通話レグで、最大 50 のデュアルトーンマルチ周波数 (DTMF) デイジットを送信できます。シグナルには以下を含めることができます。

- 0 から 9 までの数字
- 特殊文字: 星 (*) とポンド (#)
- ネットワーク制御シグナル A、B、C、D
- カンマ文字 (,)。このシグナルにより、前のシグナルと次のシグナルの間に 0.5 秒の遅延が追加されます。

トピック

- [SendDigits アクションの使用](#)
- [ACTION_SUCCESSFUL イベントの処理](#)
- [ACTION_FAILED イベントの処理](#)
- [通話フロー](#)

SendDigits アクションの使用

次の例は、一般的な SendDigits アクションを示しています。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
```

```
    "Type": "SendDigits",
    "Parameters": {
      "CallId": "call-id-1", // required
      "Digits": ",,*1234,56,7890ABCD#", // required
      "ToneDurationInMilliseconds": 100 // optional
    }
  ]
}
```

CallId

説明 – AWS Lambda 関数呼び出しの CallDetails における、参加者の CallId

使用できる値 – 有効な通話 ID

必須 – はい

デフォルト値 – なし

Digits

説明 - CallId に対応する通話レックで送信されるディジット

使用できる値 – 0~9、*、#、A、B、C、D、カンマ (,)

必須 – はい

デフォルト値 – なし

ToneDurationInMilliseconds

説明 - 各ディジットの送信で許容される時間 (ミリ秒単位)

使用できる値 - 50 ~ 24000 の任意の整数

必須 – いいえ

デフォルト値 – 250

ACTION_SUCCESSFUL イベントの処理

SendDigits アクションの一般的な ACTION_SUCCESSFUL イベントの例を次に示します。

```
{
```

```
"SchemaVersion": "1.0",
"Sequence": 3,
"InvocationEventType": "ACTION_SUCCESSFUL",
"ActionData": {
  "Type": "SendDigits",
  "Parameters": {
    "Digits": "1,2A#",
    "ToneDurationInMilliseconds": 100,
    "CallId": "call-id-1"
  },
  "CallDetails": {
    ...
  }
}
```

ACTION_FAILED イベントの処理

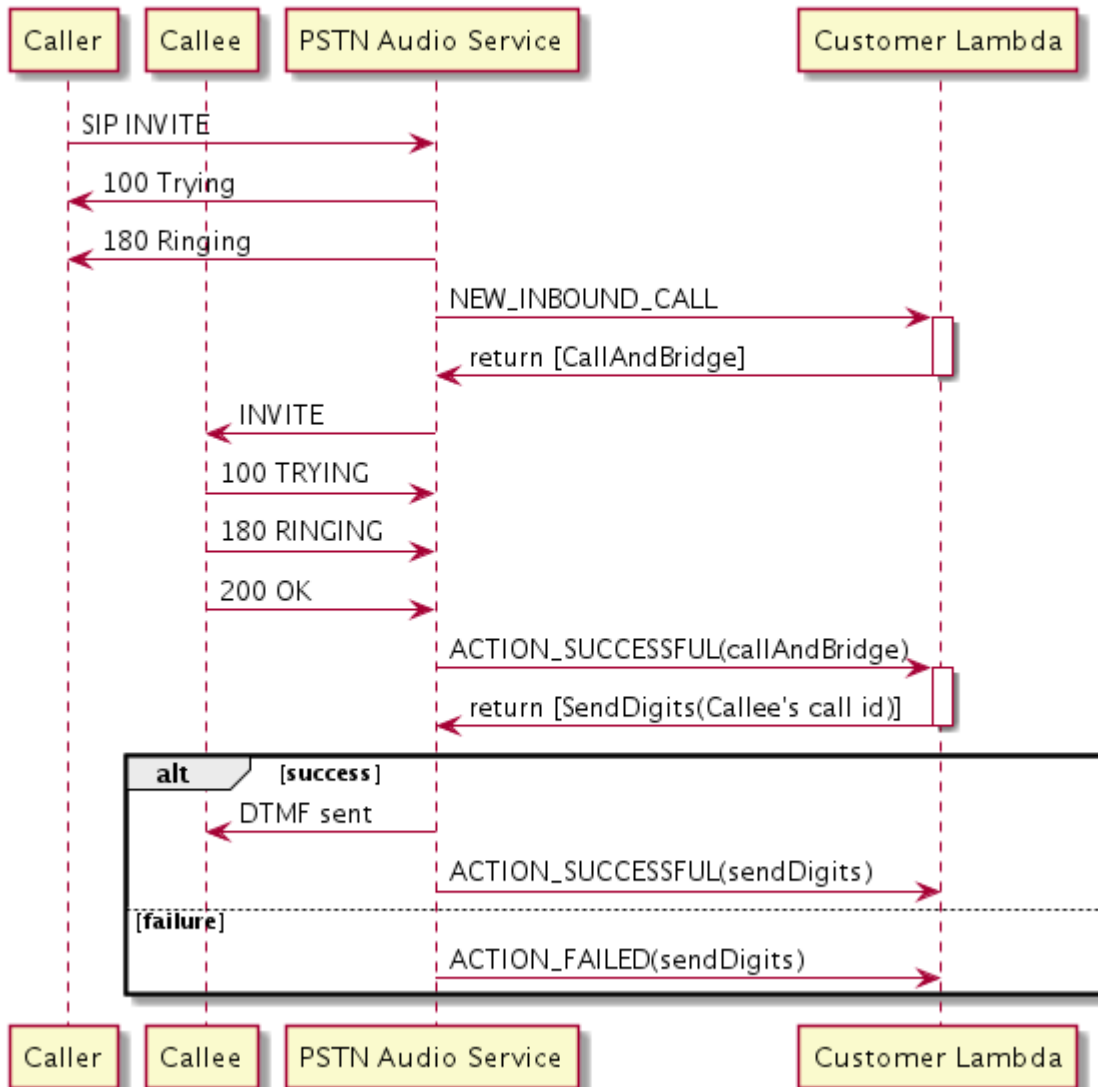
SendDigits アクションの一般的な ACTION_FAILED イベントの例を次に示します。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 3,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {
    "Type": "SendDigits",
    "Parameters": {
      "Digits": "1,2A#",
      "ToneDurationInMilliseconds": 20000000,
      "CallId": "call-id-1"
    },
    "ErrorType": "InvalidActionParameter",
    "ErrorMessage": "ToneDuration parameter value is invalid."
  },
  "CallDetails": {
    ...
  }
}
```

通話フロー

次の図は、発信者から受信者にディジットを送信するプログラムフローを示しています。

Send Digits from Caller to Callee



Speak

テキストを入力すれば、どの通話レグでも音声再生できます。プレーンテキストまたは音声合成マークアップ言語 (SSML) を使用できます。SSML では、一時停止を追加したり、特定の単語を強調したり、話し方を変更したりすることで、Amazon Chime SDK が音声を生成する方法をより細かく制御できます。

Amazon Chime SDK は Amazon Polly サービスを使用して を変換します text-to-speech。Amazon Polly では、音声品質を向上させるために、標準エンジンとニューラルエンジンのどちらかを選択できます。Amazon Polly は、20 以上の言語と 60 の音声をサポートしており、アプリケーションのユーザーエクスペリエンスをカスタマイズすることができます。Amazon Chime SDK では音

声機能は無料で提供されますが、Amazon Polly の使用には料金がかかります。価格情報については、Amazon Polly の [料金ページ](#) または請求ダッシュボードを参照してください。

⚠ Important

Amazon Polly の使用には、AWS Machine Learning および人工知能サービスに固有の条件を含む、[AWS のサービス条件](#) が適用されます。

トピック

- [Speak アクションの使用](#)
- [ACTION_SUCCESSFUL イベントの処理](#)
- [ACTION_FAILED イベントの処理](#)
- [プログラムフロー](#)

Speak アクションの使用

次の例は、Speak アクションの一般的な使用方法を示しています。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "Speak",
      "Parameters": {
        "Text": "Hello, World!",           // required
        "CallId": "call-id-1",          // required
        "Engine": "neural",              // optional. Defaults to standard
        "LanguageCode": "en-US",        // optional
        "TextType": "text",              // optional
        "VoiceId": "Joanna"              // optional. Defaults to Joanna
      }
    }
  ]
}
```

CallId

説明 – Lambda 関数呼び出しの CallDetails における、参加者の CallId

使用できる値 – 有効な通話 ID

必須 – はい

デフォルト値 – なし

Text

説明 - 音声に合成する入力テキストを指定します。ssml を TextType として指定した場合は、入力テキストの SSML 形式に従います。

使用できる値 - 文字列

必須 – はい

デフォルト値 – なし

Engine

説明 - 音声合成用のテキストを処理するときに使用するエンジン (標準またはニューラル) を指定します。

使用できる値 – standard | neural

必須 – いいえ

デフォルト値 - standard

LanguageCode

説明 - 言語コードを指定します。バイリンガル音声を使用する場合にのみ必要となります。言語コードなしでバイリンガル音声を使用した場合は、バイリンガル音声のデフォルト言語が使用されます。

使用できる値 - [Amazon Polly の言語コード](#)

必須 – いいえ

デフォルト値 – なし

TextType

説明 - 入力テキストのタイプ、つまりプレーンテキストか SSML かを指定します。入力タイプが指定されていない場合は、デフォルトとしてプレーンテキストが使用されます。SSML の詳細に

については、「Amazon Polly デベロッパーガイド」の「[SSML ドキュメントから音声を生成する](#)」を参照してください。

使用できる値 – `ssml | text`

必須 – いいえ

デフォルト値 – なし

Voiceld

説明 - 使用する音声の ID を指定します。

使用できる値 - [Amazon Polly の音声 ID](#)

必須 – いいえ

デフォルト値 - `Joanna`

ACTION_SUCCESSFUL イベントの処理

次の例は、Amazon Polly の Joanna の音声を使用して「Hello World」というテキストを英語の音声に合成するアクションの一般的な ACTION_SUCCESSFUL イベントを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 3,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "Speak",
    "Parameters": {
      "CallId": "call-id-1",
      "Engine": "neural",
      "LanguageCode": "en-US",
      "Text": "Hello World",
      "TextType": "text",
      "VoiceId": "Joanna"
    }
  },
  "CallDetails": {
    ...
  }
}
```



```
}

```

ACTION_FAILED イベントの処理

次の例は、前の例で使用された同じイベントの一般的な ACTION_FAILED イベントを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 2,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {
    "Type": "Speak",
    "Parameters": {
      "CallId": "call-id-1",
      "Engine": "neural",
      "LanguageCode": "en-US",
      "Text": "Hello World",
      "TextType": "text",
      "VoiceId": "Joanna"
    }
  },
  "ErrorType": "SystemException",
  "ErrorMessage": "System error while running action"
},
"CallDetails": {
  ...
}
}
```

エラー処理

以下の表では、Speak アクションによってスローされるエラーメッセージを列挙し、説明しています。

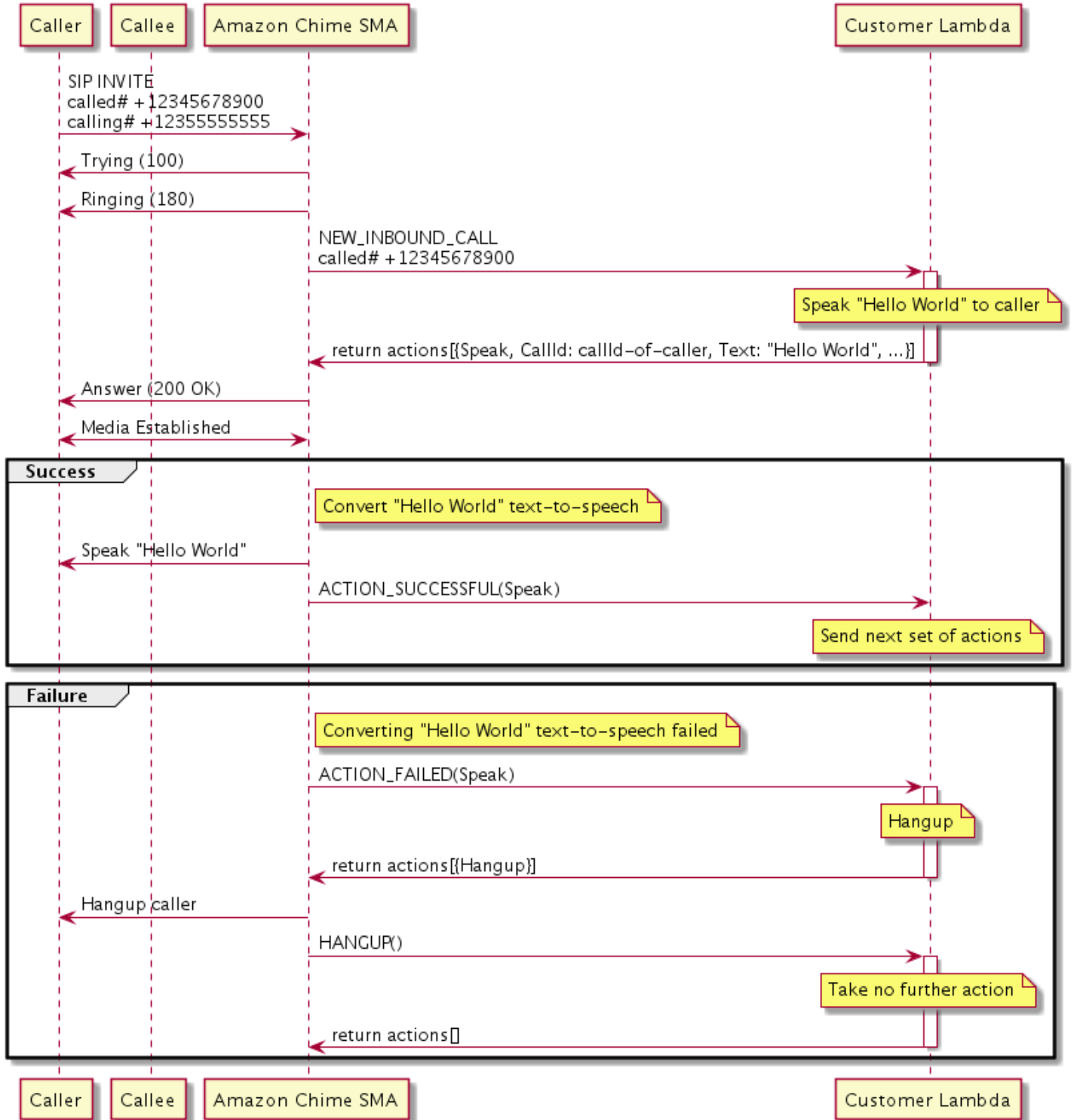
エラー	メッセージ	理由
AccessDenied	AWSServiceRoleForAmazonChimeVoiceConnector サービスにリンクさ	Amazon Polly へのリクエストに使用された、サービスにリンクされたロールが、存在しないかアクセス許可がありません。これを解決するには、

エラー	メッセージ	理由
	れたロールが正しく設定されていません。	「 Amazon Chime SDK 音声コネクタサービスにリンクされたロールの使用 」セクションの手順を参照してください。
InvalidActionParameter		アクションのパラメータの検証中にエラーが発生しました。パラメータの詳細については、Amazon Pollyデベロッパーガイド」の SynthesizeSpeech 「API」 を参照してください。
ActionExecutionThrottled	Amazon Polly は、音声合成のリクエストをスロットリングしています。	Amazon Polly へのリクエストがスロットリング例外を返しています。Amazon Polly のスロットリング制限の詳細については、「 https://docs.aws.amazon.com/polly/latest/dg/limits.html#limits-throttle 」を参照してください。
MissingRequiredActionParameter	Text は必須パラメータです。	このアクションのパラメータには Text 値が必要です。
MissingRequiredActionParameter	Text は最大 1,000 文字に制限されています。	テキストが文字数制限を超えています。
SystemException	アクションの実行中にシステムエラーが発生しました。	アクションの実行中にシステムエラーが発生しました。

プログラムフロー

次の図は、発信者の Speak アクションを可能にするプログラムフローを示しています。この例では、発信者には次のようなテキストが聞こえます。

Enable Speak action for Caller in SMA



図の説明

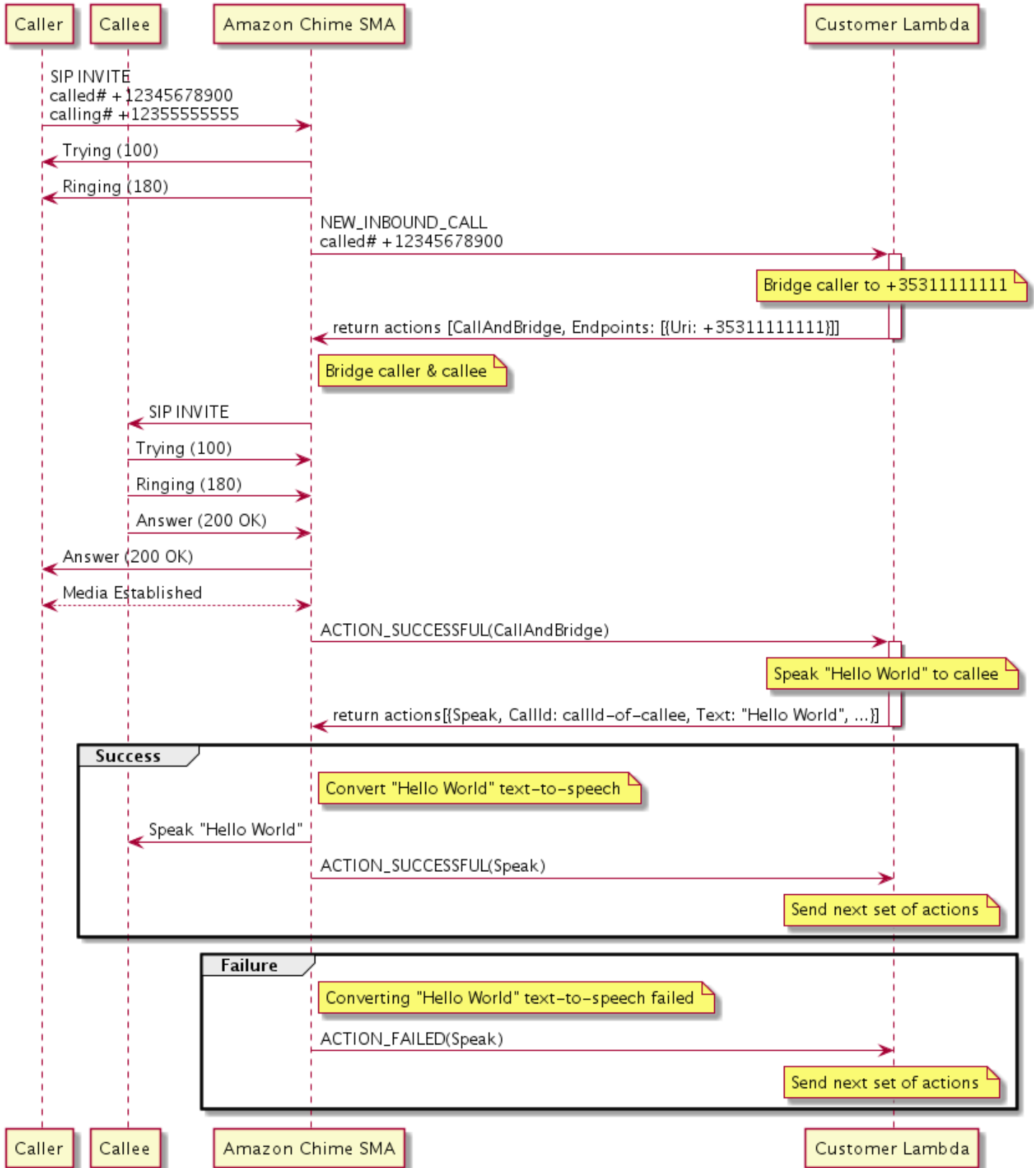
発信者は、ソフトフォンを使用して、SIP メディアアプリケーションに登録されている番号を入力します。アプリケーションは、SIP INVITE メソッドを使用して、発信者に Trying (100) 応答を送信します。これは、ネクストホップサーバーが呼び出しリクエストを受信したことを示しています。SIP アプリケーションは、INVITE を使用してエンドポイントに連絡します。接続が確立されると、アプリケーションは発信者に Ringing (180) 応答を送信し、アラートが開始されます。

次に、SIP メディアアプリケーションが Lambda 関数に NEW_INBOUND_CALL イベントを送信します。Lambda 関数は、発信者の ID と、音声に変換するテキストが含まれた Speak アクションで応答します。そして、SIP アプリケーションは、呼び出しが応答されたことを示す 200 (OK) 応答を送信します。このプロトコルにより、メディアも有効になります。

Speak アクションが成功してテキストが音声に変換されると、SIP メディアアプリケーションに ACTION_SUCCESSFUL イベントが返されます。そして、SIP メディアアプリケーションは次のアクションセットを返します。アクションが失敗した場合、SIP メディアアプリケーションは Lambda 関数に ACTION_FAILED イベントを送信します。Lambda 関数は Hangup アクションセットで応答します。アプリケーションは発信者をハングアップし、Lambda 関数に HANGUP イベントを返します。Lambda 関数は、それ以上アクションを実行しません。

次の図は、受信者の Speak アクションを可能にするプログラムフローを示しています。

Enable Speak action for Callee in SMA



図の説明

発信者が、SIP メディアアプリケーションに登録されている番号を入力すると、アプリケーションは、前の図で説明したように応答します。Lambda 関数は、NEW_INBOUND_CALL イベントを受信すると、SIP アプリケーションに [the section called “CallAndBridge”](#) アクションを返します。アプリケーションは、SIP INVITE メソッドを使用して、受信者に Trying (100) 応答と Ringing (180) 応答を送信します。

受信者が応答すると、SIP メディアアプリケーションは 200 (OK) 応答を受信し、その同じ応答を発信者に送信します。これによりメディアが確立され、SIP アプリケーションは [the section called “CallAndBridge”](#) アクションの ACTION_SUCCESSFUL イベントを Lambda 関数に送信します。この関数は、Speak アクションとデータを SIP アプリケーションに返します。SIP アプリケーションはこれを変換します。

SpeakAndGetDigits

テキストを提供して音声を再生し、ユーザーからデュアルトーンマルチ周波数 (DTMF) デイジットを収集します。テキストには、プレーンテキスト、または音声合成マークアップ言語 (SSML) で強化されたテキストを使用できます。サポートされている SSML 機能の中でも特に一時停止の追加、特定の単語の強調、話し方の変更によって、Amazon Chime SDK が音声を生成する方法をより細かく制御できます。ユーザーが正しい DTMF デイジット数を入力しなかったなどの失敗が発生した場合、このアクションは「失敗」音声を再生し、Repeat パラメータで定義された試行回数を SIP メディアアプリケーションが使い切るまでメイン音声を再生します。

Amazon Chime SDK は、テキストを肉声に近い音声に変換するクラウドサービスである Amazon Polly を使用しています。Amazon Polly は、標準エンジンと、音声品質が改善されているニューラルエンジンを提供しています。サポートする言語は 20 を超え、60 の音声に対応しています。Amazon Polly では音声機能は無料で提供されますが、Amazon Polly の使用には料金がかかります。価格情報については、Amazon Polly の [料金ページ](#) または請求ダッシュボードを参照してください。

Important

Amazon Polly の使用には、AWS Machine Learning および人工知能 [AWS サービス](#) に固有の条件を含むサービス条件が適用されます。

トピック

- [SpeakAndGetDigits アクションの使用](#)

- [ACTION_SUCCESSFUL イベントの処理](#)
- [ACTION_FAILED イベントの処理](#)
- [Amazon Chime SDK 音声コネクタサービスにリンクされたロールの使用](#)

SpeakAndGetDigits アクションの使用

次の例は、SpeakAndGetDigits アクションの一般的な使用方法を示しています。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "SpeakAndGetDigits",
      "Parameters": {
        "CallId": "call-id-1", // required
        "InputDigitsRegex": "^\\d{2}#$", // optional
        "SpeechParameters": {
          "Text": "Hello World", // required
          "Engine": "neural", // optional. Defaults to standard
          "LanguageCode": "en-US", // optional
          "TextType": "text", // optional
          "VoiceId": "Joanna" // optional. Defaults to Joanna
        },
        "FailureSpeechParameters": {
          "Text": "Hello World", // required
          "Engine": "neural", // optional. Defaults to the Engine
          "LanguageCode": "en-US", // optional. Defaults to the
          "LanguageCode value in SpeechParameters
          "TextType": "text", // optional. Defaults to the TextType
          "TextType value in SpeechParameters
          "VoiceId": "Joanna" // optional. Defaults to the VoiceId
          "VoiceId value in SpeechParameters
        },
        "MinNumberOfDigits": 3, // optional
        "MaxNumberOfDigits": 5, // optional
        "TerminatorDigits": ["#"], // optional
        "InBetweenDigitsDurationInMilliseconds": 5000, // optional
        "Repeat": 3, // optional
        "RepeatDurationInMilliseconds": 10000 // required
      }
    }
  ]
}
```

```
    ]  
}
```

CallId

説明 - Lambda 関数呼び出し CallDetails の におけるCallId参加者の。

使用できる値 - 有効な callID

必須 - はい

デフォルト値 - なし

InputDigitsRegex

説明 - ユーザーが正しいディジットと文字を入力できるようにするための正規表現パターン。

使用できる値 - 有効な正規表現パターン

必須 - いいえ

デフォルト値 - なし

SpeechParameters.Engine

説明 - 音声合成用のテキストを処理するときに使用するエンジン (標準またはニューラル) を指定します。

使用できる値 - standard | neural

必須 - いいえ

デフォルト値 - standard

SpeechParameters.LanguageCode

説明 - 言語コードを指定します。これはバイリンガル音声を使用する場合にのみ必要となります。バイリンガル音声の使用時に言語コードが指定されていない場合は、バイリンガル音声のデフォルト言語が使用されます。

使用できる値 - [Amazon Polly の言語コード](#)

必須 - いいえ

デフォルト値 - なし

SpeechParameters.Text

説明 - 入力テキストを指定します。ssml を `SpeechParameters.TextType` として指定した場合は、入力テキストの SSML 形式に従う必要があります。SSML の詳細については、「Amazon Polly デベロッパーガイド」の「[SSML ドキュメントから音声を作成する](#)」を参照してください。

使用できる値 - 文字列

必須 - はい

デフォルト値 - なし

SpeechParameters.TextType

説明 - `SpeechParameters.Text` のテキスト形式を指定します。指定されない場合、デフォルトで `text` が使用されます。SSML の詳細については、「Amazon Polly デベロッパーガイド」の「[SSML ドキュメントから音声を作成する](#)」を参照してください。

使用できる値 - `ssml` | `text`

必須 - いいえ

デフォルト値 - `text`

SpeechParameters.VoiceId

説明 - `SpeechParameters.Text` のテキストを読み上げるために使用される Amazon Polly の音声の ID。

使用できる値 - [Amazon Polly の音声 ID](#)

必須 - いいえ

デフォルト値 - `Joanna`

FailureSpeechParameters.Engine

説明 - 顧客が音声合成に関する無効な応答を入力したときに使用される失敗メッセージの処理に使用するエンジン (標準またはニューラル) を指定します。

使用できる値 - `standard` | `neural`

必須 - いいえ

デフォルト値 - `SpeechParameters.Engine` 値

`FailureSpeechParameters.LanguageCode`

説明 - 顧客が無効な応答を入力したときに使用する言語コードを指定します。バイリンガル音声を使用する場合にのみ必要となります。言語コードを指定せずにバイリンガル音声を使用した場合は、バイリンガル音声のデフォルト言語が使用されます。

使用できる値 - [Amazon Polly の言語コード](#)

必須 - いいえ

デフォルト値 - `SpeechParameters.LanguageCode` 値

`FailureSpeechParameters.Text`

説明 - 顧客が無効な応答を入力したときに読み上げられる入力テキストを指定します。ssml を `FailureSpeechParameters.TextType` として指定した場合は、入力テキストの SSML 形式に従う必要があります。

使用できる値 - 文字列

必須 - はい

デフォルト値 - なし

`FailureSpeechParameters.TextType`

説明 - `FailureSpeechParameters.Text` で指定される入力テキストがプレーンテキストか SSML かを指定します。デフォルト値はプレーンテキストです。詳細については、「Amazon Polly デベロッパーガイド」の「[SSML ドキュメントから音声を生成する](#)」を参照してください。

使用できる値 - `ssml` | `text`

必須 - いいえ

デフォルト値 - `SpeechParameters.Text` 値

`FailureSpeechParameters.VoiceId`

説明 - `FailureSpeechParameters.Text` の文字列を読み上げるために使用される音声の ID。

使用できる値 - [Amazon Polly の音声 ID](#)

必須 - はい

デフォルト値 - `SpeechParameters.VoiceId` 値

MinNumberOfDigits

説明 - タイムアウトになるか「通話失敗」メッセージが再生される前にキャプチャする最小ディジット数。

使用できる値 - 0 以上

必須 - いいえ

デフォルト値 - 0

MaxNumberOfDigits

説明 - 終了ディジットなしで停止する前にキャプチャする最大ディジット数。

使用できる値 - MinNumberOfDigits より大きい値

必須 - いいえ

デフォルト値 - 128

TerminatorDigits

説明 - ユーザーが未満を入力した場合の入力終了に使用される桁数 MaxNumberOfDigits

使用できる値 - 0 1 2 3 4 5 6 7 8 9 # * のいずれか

必須 - いいえ

デフォルト値 - #

InBetweenDigitsDurationInMilliseconds

説明 - 失敗音声を再生するまでのディジット入力間の待ち時間 (ミリ秒単位)。

使用できる値 - 0 より大きい値

必須 - いいえ

デフォルト値 - 指定しない場合、デフォルトは RepeatDurationInMilliseconds 値になります。

Repeat

説明 - デジットの取得を試みた回数の合計。このパラメータを省略すると、デフォルトでデジットの収集が 1 回試行されます。

使用できる値 - 0 より大きい値

必須 - いいえ

デフォルト値 - 1

RepeatDurationInMilliseconds

説明 - デジットを取得する各試行のタイムアウト (ミリ秒単位)。

使用できる値 - 0 より大きい値

必須 - はい

デフォルト値 - なし

ACTION_SUCCESSFUL イベントの処理

次の例は、一般的な ACTION_SUCCESSFUL イベントを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 3,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "SpeakAndGetDigits",
    "Parameters": {
      "CallId": "call-id-1",
      "InputDigitsRegex": "\\d{2}#",
      "SpeechParameters": {
        "Engine": "neural",
        "LanguageCode": "en-US",
        "Text": "Hello World",
        "TextType": "text",
        "VoiceId": "Joanna"
      },
      "FailureSpeechParameters": {
        "Engine": "neural",
```

```

        "LanguageCode": "en-US",
        "Text": "Hello World",
        "TextType": "text",
        "VoiceId": "Joanna"
    },
    "MinNumberOfDigits": 3,
    "MaxNumberOfDigits": 5,
    "TerminatorDigits": ["#"],
    "InBetweenDigitsDurationInMilliseconds": 5000,
    "Repeat": 3,
    "RepeatDurationInMilliseconds": 10000
},
"ReceivedDigits": "1234"
},
"CallDetails":{
    ...
}
}

```

ACTION_FAILED イベントの処理

次の例は、一般的な ACTION_FAILED イベントを示しています。

```

{
  "SchemaVersion": "1.0",
  "Sequence": 2,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {
    "Type": "SpeakAndGetDigits",
    "Parameters": {
      "CallId": "call-id-1",
      "InputDigitsRegex": "^\\d{2}#$",
      "SpeechParameters": {
        "Engine": "neural",
        "LanguageCode": "en-US",
        "Text": "Hello World",
        "TextType": "text",
        "VoiceId": "Joanna"
      },
    },
    "FailureSpeechParameters": {
      "Engine": "neural",
      "LanguageCode": "en-US",
      "Text": "Hello World",
      "TextType": "text",
    }
  }
}

```

```

        "VoiceId": "Joanna"
    },
    "MinNumberOfDigits": 3,
    "MaxNumberOfDigits": 5,
    "TerminatorDigits": ["#"],
    "InBetweenDigitsDurationInMilliseconds": 5000,
    "Repeat": 3,
    "RepeatDurationInMilliseconds": 10000
},
"ErrorType": "SystemException",
"ErrorMessage": "System error while running action"
},
"CallDetails":{
    ...
}
}

```

エラー処理

以下の表では、Speak アクションによってスローされるエラーメッセージを列挙し、説明しています。

エラー	メッセージ	理由
AccessDenied	AWSServiceRoleForAmazonChimeVoiceConnector ロールが正しく設定されていません。	Amazon Polly へのリクエストに使用されたロールが存在しないか、アクセス許可がありません。これを解決するには、「 Amazon Chime SDK 音声コネクタサービスにリンクされたロールの使用 」セクションの手順を参照してください。
InvalidActionParameter		アクションのパラメータの検証中にエラーが発生しました。このアクションで使用可能なパラメータとそのオプションを確認するには、Amazon Polly デベロッパーガイ

エラー	メッセージ	理由
		ド SynthesizeSpeech 」の「」を参照してください。
MissingRequiredActionParameter	Text は必須パラメータです。	このアクションのパラメータには Text 値が必要です。
MissingRequiredActionParameter	Text は最大 1,000 文字に制限されています。	テキストが文字数制限を超えています。
SystemException	アクションの実行中にシステムエラーが発生しました。	アクションの実行中にシステムエラーが発生しました。

Amazon Chime SDK 音声コネクタサービスにリンクされたロールの使用

Speak アクションまたは SpeakAndGetDigits アクションのサービスにリンクされたロールを手動で作成する必要はありません。Amazon Chime SDK コンソール、または AWS API で SIP メディアアプリケーションを作成 AWS Command Line Interface または更新すると、Amazon Chime SDK によってサービスにリンクされたロールが作成されます。

詳細については、「Amazon Chime SDK 管理者ガイド」の「[Using the Amazon Chime service-linked role](#)」を参照してください。

StartBotConversation

StartBotConversation アクションにより、エンドユーザーと Amazon Lex v2 ボット間の音声会話が確立されます。ユーザーは必要な情報をボットに提供します。次に、ボットは情報を公衆交換電話網 (PSTN) Audio Lambda 関数に返し、関数は要求されたタスクを実行します。

例えば、ボットは会話の開始時にオプションのウェルカムメッセージを再生して、PSTN Audio Lambda 関数が実行できるタスクを簡単に説明できます。ボットが必要な情報を収集するまで、ユーザーとボットの間で会話が繰り返されます。会話が終了すると、Amazon Chime SDK はボットによって収集された情報を含むアクション成功イベントで PSTN Audio Lambda 関数を呼び出します。PSTN Audio Lambda 関数は情報を処理し、要求されたタスクを実行します。

Audio サービスは、ユーザーと本物そっくりの会話形式を提供します。例えば、音声プロンプトが終了する前に、ユーザーがボットに割り込み、質問に答えることができます。さらに、ユーザーは音声と DTMF の数字を自由に組み合わせて情報を提供できます。ボットは、ユーザーによる情報入力を

待機してから応答します。ユーザーが話し終わってから音声入力を解釈するまでボットが待機する時間を設定できます。ユーザーは、通話中にクレジットカード番号などの追加情報を取得する時間が必要な場合、ボットに待機するように指示できます。

StartBotConversation アクションでは、ボットとの会話の間、Amazon Lex と Amazon Polly が使用されます。Amazon Lex と Amazon Polly の標準料金が適用されます。料金の詳細については、[Amazon Lex ストリーミング会話の料金表](#)と [Amazon Polly 料金](#)のページを参照してください。

Note

このアクションは、ブリッジ通話や Amazon Chime SDK ミーティングに参加した通話では実行できません。

Important

Amazon Lex と Amazon Polly の使用には[AWS](#)、AWS Machine Learningおよび人工知能サービスに固有の条件を含む サービス条件が適用されます。

トピック

- [StartBotConversation の構文](#)
- [StartBotConversation アクションの使用](#)
- [ACTION_SUCCESSFUL イベントの処理](#)
- [ACTION_FAILED イベントの処理](#)
- [ボットを使用するアクセス許可の付与](#)
- [音声と DTMF のタイムアウトの設定](#)
- [会話中に DTMF 入力の使用](#)
- [請求とサービスクォータ](#)

StartBotConversation の構文

一般的な StartBotConversation 構文の例を次に示します。

```
{
  "SchemaVersion": "1.0",
```



```
"Actions":[
  {
    "Type": "StartBotConversation",
    "Parameters": {
      "CallId": "string",
      "ParticipantTag": "string",
      "BotAliasArn": "string",
      "LocaleId": "string",
      "Configuration": {
        "SessionState": {
          "SessionAttributes": {
            "string": "string"
          },
          "DialogAction" : {
            "Type": "string"
          }
        },
        "WelcomeMessages": [
          {
            "Content": "string",
            "ContentType": "string"
          }
        ]
      }
    }
  }
]
```

CallId

説明 – AWS Lambda 関数呼び出し CallID の CallDetails における参加者の。StartBotConversation アクションはこの ID をボットの SessionId として使用します。通話中に行われるすべてのボットの会話は、同じ会話セッションを共有します。[Amazon Lex PutSession](#) API を使用して、ユーザーとボット間のセッション状態を変更できます。詳細については、「Amazon Lex Developer Guide」の「[Managing sessions with the Amazon Lex v2 API](#)」を参照してください。

使用できる値 – 有効な通話 ID。

必須 – ParticipantTag が存在する場合は、いいえ。

デフォルト値 – なし。

ParticipantTag

説明 - CallDetails で接続されている参加者のうちの 1 人の ParticipantTag。

使用できる値 - LEG-A。

必須 - CallId が存在する場合は、いいえ。

デフォルト値 - 呼び出した callLeg の ParticipantTag。CallDetails を指定すると無視されます。

BotAliasArn

説明 - Lex ボットのボットエイリアス ARN。ボットは PSTN Audio アプリケーションと同じ AWS リージョンに作成する必要があります。有効な Amazon Lex ボットエイリアスの形式は `arn:aws:lex:region:awsAccountId:bot-alias/botId/botAliasId` です。ここで `region` はボットが存在する AWS リージョンです。`awsAccountId` は Amazon Lex ボットが作成された AWS アカウント ID です。botId 値は、その作成時にボットに割り当てた識別子です。ボット ID は、Amazon Lex コンソールの [ボットの詳細] ページにあります。botAliasId は、その作成時にボットエイリアスに割り当てた識別子です。ボットエイリアス ID は、Amazon Lex コンソールの [エイリアス] ページにあります。

使用できる値 - 有効なボット ARN。

必須 - はい。

デフォルト値 - なし。

LocaleId

説明 - ボットに使用したロケールの識別子。ロケールおよび言語コードのリストについては、「[Languages and locales supported by Amazon Lex](#)」を参照してください。

使用できる値 - [Amazon Lex でサポートされている言語とロケール](#)。

必須 - いいえ。

デフォルト値 - en_US。

Configuration

説明 - セッション状態とウェルカムメッセージを含む会話設定。Configuration オブジェクトの JSON 文字列表現の合計サイズは 10 KB に制限されています。

使用できる値 - Configuration オブジェクト。

必須 - いいえ。

デフォルト値 - なし。

Configuration.SessionState

説明 - Amazon Lex v2 とのユーザーのセッションの状態。

使用できる値 - SessionState オブジェクト。

必須 - いいえ。

デフォルト値 - なし。

Configuration.SessionState.SessionAttributes

説明 - セッション固有のコンテキスト情報を表すキー/値のペアのマップ。このマップには、Amazon Lex v2 とクライアントアプリケーションの間に渡されるアプリケーション情報が含まれます。

使用できる値 - 文字列間のマッピング。

必須 - いいえ。

デフォルト値 - なし。

Configuration.SessionState.DialogAction.Type

説明 - ユーザーとのインタラクションにおいて、ボットが取る次のアクション。使用できる値:

- Delegate Amazon Lex v2 が次のアクションを決定します。
- ElicitIntent 次のアクションは、ユーザーからインテントを引き出します。

使用できる値 - Delegate | ElicitIntent。

必須 - いいえ。

デフォルト値 - なし。

Configuration.WelcomeMessages

説明 - 会話の開始時にユーザーに送信するメッセージのリスト。welcomeMessage フィールドを設定する場合、DialogAction.Type 値を ElicitIntent に設定する必要があります。

使用できる値 - メッセージオブジェクト

必須 - いいえ。

デフォルト値 - なし。

Configuration.WelcomeMessages.Content

説明 - ウェルカムメッセージのテキスト。

使用できる値 - 文字列。

必須 - いいえ。

デフォルト値 - なし。

Configuration.WelcomeMessages.ContentType

説明 - ウェルカムメッセージのタイプを示します。

使用できる値 - PlainText | SSML

- PlainText - メッセージにはプレーン UTF-8 テキストが含まれています。
- SSML - メッセージには音声出力のテキスト形式が含まれています。

必須 - はい。

デフォルト値 - なし。

StartBotConversation アクションの使用

一般的な StartBotConversation アクションの例を次に示します。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "StartBotConversation",
      "Parameters": {
        "CallId": "call-id-1",
        "BotAliasArn": "arn:aws:lex:us-east-1:123456789012:bot-alias/ABCDEFGH/IH/MNOPQRSTU/V",
        "LocaleId": "en_US",
        "Configuration": {
          "SessionState": {
            "SessionAttributes": {
              "mykey1": "myvalue1"
            }
          },
          "DialogAction" : {
```



```
        "string": {
          "Value": {
            "OriginalValue": "string",
            "InterpretedValue": "string",
            "ResolvedValues": ["string"]
          },
          "Values": []
        }
      },
      "State": "string",
      "ConfirmationState": "string"
    }
  },
  "Interpretations": [
    {
      "NluConfidence": {
        "Score": number
      },
      "Intent": {
        "Name": "string",
        "Slots": {
          "string": {
            "Value": {
              "OriginalValue": "string",
              "InterpretedValue": "string",
              "ResolvedValues": ["string"]
            },
            "Values": []
          }
        }
      },
      "State": "string",
      "ConfirmationState": "string"
    }
  ]
}
}
```

IntentResult

ユーザーとボットの間のお話の結果。

SessionId

ボット会話セッションの識別子。ユーザーがボットとの会話を開始すると、Amazon Lex によりセッションが作成されます。セッションは、ユーザーとボットの間で交換される情報をカプセル化します。StartBotConversation アクションは通話 ID をボットの SessionId として使用します。Lex [PutSession](#) API を使用して、ユーザーとボット間のセッション状態を変更できます。詳細については、「Amazon Lex Developer Guide」の「[Managing sessions with the Amazon Lex V2 API](#)」を参照してください。

SessionState

ユーザーの Amazon Lex v2 セッションの状態。

SessionState.SessionAttributes

セッション固有のコンテキスト情報を表すキー/値のペアのマップ。マップには、ボットにアタッチされた Lambda 関数と PSTN Audio Lambda 関数の間で渡されるボット会話情報が含まれています。

Interpretations

Amazon Lex によって導き出された、顧客の発話を満たす可能性のあるインテントのリスト。NluConfidence スコアの最も高いインテントが SessionState のインテントになります。

解釈NluConfidenceスコア

インテントがユーザーのインテントを満たしているという Amazon Lex v2 の信頼度を示すスコア。0.00 ~ 1.00 までの範囲になります。スコアが高いほど、信頼度が高くなります。

Intent

ユーザーが実行したいアクション。

Intent.Name

インテントの名前。

Intent.Slots

インテントのすべてのスロットのマップ。スロットの名前は、スロットの値にマップされます。スロットがいっぱいになっていない場合は、値は null です。

Intent.Slots.Value

スロットの値。

Intent.Slots.Values

スロットにユーザーが指定した 1 つ以上の値のリスト。

Intent.Slots.Value.OriginalValue

スロットに対して入力された、ユーザーの返信のテキスト。

Intent.Slots.Value.InterpretedValue

説明 - Amazon Lex v2 がスロットについて決定する値。実際の値は、ボットの値選択戦略の設定によって異なります。ユーザーが入力した値を使用できるか、Amazon Lex v2 に resolvedValues のリスト中の値を選ばせるかが行えます。

Intent.Slots.Value.ResolvedValues

Amazon Lex v2 がスロットで認識した追加の値のリスト。

Intent.State

説明 - インテントの履行情報。使用できる値:

- Failed - Lambda 関数がインテントを達成できませんでした。
- Fulfilled - Lambda 関数がインテントを満たしました。
- ReadyForFulfillment - インテントの情報が存在し、Lambdafunction がインテントを満たすことができます。

Intent.ConfirmationState

説明 - インテントの確認を示します。使用できる値:

- Confirmed - インテントが履行されます。
- Denied - ユーザーは確認プロンプトで「いいえ」と答えました。
- None - ユーザーは確認を求められなかったか、または確認を求められたが肯定も否定もしませんでした。

ACTION_FAILED イベントの処理

StartBotConversation アクションの一般的な ACTION_FAILED イベントの例を次に示します。

```
{
  "SchemaVersion": "1.0",
  "Sequence": number,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {
```



```

    "CallId": "string",
    "Type": "StartBotConversation",
    "Parameters": {
        // parameters provided in the StartBotConversation action
    },
    "ErrorType": "string",
    "ErrorMessage": "string"
},
"CallDetails":{
}
}

```

ErrorType

エラー状態を個別に識別する文字列。

ErrorMessage

エラー状態の一般的な説明。

エラーコード

次の表は、Lambda 関数が ACTION_FAILED イベントで返す可能性のあるエラーメッセージを示しています。

エラー	説明
InvalidActionParameter	1つ以上のアクションパラメータが無効です。エラーメッセージには、無効なパラメータが示されます。
SystemException	アクションの実行中に、システムエラーが発生しました。
ResourceNotFound	指定したポットは見つかりませんでした。
ResourceAccessDenied	ポットへのアクセスは拒否されます。
ActionExecutionThrottled	ポット会話サービスの制限を超えています。エラーメッセージには、特定のサービスの制限を超えていることが示されます。

ボットを使用するアクセス許可の付与

次の例では、Amazon Lex API を呼び出すアクセス許可を Amazon Chime SDK APIs。

[StartConversation](#) ユーザーにボットを使用する Audio サービスのアクセス許可を明示的に付与する必要があります。条件ブロックはサービスプリンシパルに必要です。条件ブロックはグローバルコンテキストキー `AWS:SourceAccount` と `AWS:SourceArn` を使用する必要があります。`AWS:SourceAccount` は AWS アカウント ID です。`AWS:SourceArn` は、Lex ボットを呼び出す PSTN Audio アプリケーションのリソース ARN です。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowChimePstnAudioUseBot",
      "Effect": "Allow",
      "Principal": {
        "Service": "voiceconnector.chime.amazonaws.com"
      },
      "Action": "lex:StartConversation",
      "Resource": "arn:aws:lex:region:awsAccountId:bot-alias/botId/aliasId",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "awsAccountId"
        },
        "ArnEquals": {
          "AWS:SourceArn": "arn:aws:voiceconnector:region:awsAccountId:sma/smaId"
        }
      }
    }
  ]
}
```

音声と DTMF のタイムアウトの設定

ユーザー入力をキャプチャするときの音声と DTMF のタイムアウトを設定できます。ボットとの会話を開始するときにセッション属性を使用してタイムアウトを設定し、必要に応じて Lex ボットの Lambda 関数で上書きできます。Amazon Lex では、インテントまたはボットに複数のスロットを設定できます。セッション属性をインテントレベルとスロットレベルに適用するように指定できるため、特定のタイプの入力を収集するときのみ属性を設定するように指定できます。例えば、アカウント番号を収集する場合は、日付を収集する場合よりも長いタイムアウトを指定できます。セッション属性キーにはワイルドカードを使用できます。

例えば、すべてのインテントのすべてのスロットの音声タイムアウトを 4000 ミリ秒に設定するには、セッション属性名として `x-amz-lex:start-timeout-ms:*:*`、セッション属性値として `4000` を使用して、セッション属性を指定できます。詳細については、「Amazon Lex Developer Guide」の「[Configuring timeouts for capturing user input](#)」を参照してください。

会話中に DTMF 入力の使用

Amazon Lex ボットは、会話中の音声入力とキーパッド入力をサポートします。ボットはキーパッド入力を DTMF の数字として解釈します。入力をシャープ記号 (#) で終了し、星の記号 (*) を使用して会話をキャンセルするよう問い合わせに求めることができます。顧客にシャープ記号で入力を終了するように促さない場合、Lex は 5 秒後に追加のキー押下を待たなくなります。

請求とサービスクォータ

AWS は、次のコストを請求します。

- 通話での Amazon Chime SDK の使用。詳細については、「[Amazon Chime SDK の料金](#)」を参照してください。
- ユーザーの音声を解釈するための Amazon Lex の使用。詳細については、「[Amazon Lex 料金](#)」を参照してください。
- ボットからのテキスト応答を合成するための Amazon Polly の使用。詳細については、「[Amazon Polly 料金](#)」を参照してください。

また、次のサービスクォータに注意する必要があります。

- Amazon Chime SDK には、PSTN Audio [StartBotConversation](#) アクションで使用できる Amazon Lex ボットの最大数に対するサービスクォータがあります。詳細については、「AWS 全般のリファレンス」の「[SIP トランキングと音声クォータ](#)」を参照してください。
- Amazon Lex には、Lex ボットあたりの同時音声会話の最大数に関するサービスクォータがあります。クォータの増量については、Amazon Lex サービスチームにお問い合わせください。詳細については、「Amazon Lex Developer Guide」の「[Guidelines and quotas](#)」を参照してください。
- Amazon Polly には、テキスト応答の合成に関するサービスクォータがあります。クォータの増量については、Amazon Polly サービスチームにお問い合わせください。Amazon Polly サービスのクォータの詳細については、「Amazon Polly Developer Guide」の「[Quotas in Amazon Polly](#)」を参照してください。

SIP ヘッダーの使用

通話コンテキスト情報を SIP インフラストラクチャと交換する場合に、AWS Lambda 関数で User-to-User ヘッダー、Diversion ヘッダー、およびカスタム SIP ヘッダーを送受信できるようになりました。

- User-to-User (UUI) ヘッダーを使用して、通話制御データを送信できます。このデータは、セッションを開始するアプリケーションによって挿入され、セッションを受け付けるアプリケーションによって使用されます。基本的な SIP 機能には使用されません。例えば、コールセンターで UUI ヘッダーを使用して、通話に関する情報をエージェント間でやり取りできます。
- Diversion ヘッダーは、通話がどこから転送されたのか、またその理由を示すために使用されます。このヘッダーを使用して、他の SIP エージェントからの転送情報を確認する、または伝えることができます。
- カスタム SIP ヘッダーを使用すると、必要なその他の情報を伝えることができます。例えば、アカウント ID を伝える場合は、「X-Account-Id」という X ヘッダーを作成し、この情報を追加できます。

カスタム SIP ヘッダーには x- というプレフィックスを付ける必要があります。ヘッダーは AWS Lambda 関数で公開され、インバウンド呼び出し中に NEW_INBOUND_CALL イベントの一部として受信されます。[CallAndBridge](#) アクションまたは [CreateSipMediaApplicationCall](#) API をトリガーするときに、これらのヘッダーをアウトバウンドコールログに含めることもできます。

Lambda 関数の Participants セクションには SipHeaders フィールドが含まれています。このフィールドは、カスタムヘッダーを受け取るとき、あるいは、User-to-User ヘッダーまたは Diversion ヘッダーにデータを入力するときに使用できます。

この例では、Lambda 呼び出しに SIP AWS ヘッダーが含まれている場合に予想されるレスポンスを示します。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 3,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "actionType",
    "Parameters": {
      // Parameters vary by actionType
    }
  },
}
```

```

"CallDetails":{
  .....
  .....
  "Participants":[
    {
      "CallId": "call-id-1",
      "ParticipantTag": "LEG-A",
      .....
      "Status": "Connected"
      "SipHeaders": {
        "X-Test-Value": "String",
        "User-to-User":
"616d617a6f6e5f6368696d655f636f6e6e6563745f696e746567726174696f6e";encoding=hex",
        "Diversion": "sip:
+11234567891@public.test.com;reason=unconditional"
      }
    },
    {
      "CallId": "call-id-2",
      "ParticipantTag": "LEG-B",
      .....
      "Status": "Connected"
    }
  ]
}
}

```

次の例は、SipHeadersパラメータのエントリが無効であるために成功した[CallAndBridge](#)アクションを示しています。

```

{
  "SchemaVersion": "1.0",
  "Actions":[
    {
      "Type": "CallAndBridge",
      "Parameters":{
        "CallTimeoutSeconds": 30,
        "CallerIdNumber": "e164PhoneNumber", // required
        "RingbackTone": { // optional
          "Type": "S3",
          "BucketName": "s3_bucket_name",
          "Key": "audio_file_name"
        }
      },
    },
  ],
}

```

```

    "Endpoints": [
      {
        "Uri": "e164PhoneNumber", // required
        "BridgeEndpointType": "PSTN" // required
      }
    ],
    "SipHeaders": {
      "X-Test-Value": "String",
      "User-to-User":
"616d617a6f6e5f6368696d655f636f6e6e6563745f696e746567726174696f6e;encoding=hex",
      "Diversion": "sip:+11234567891@public.test.com;reason=unconditional"
    }
  }
}

```

次の例は、無効なSipHeadersパラメータが原因で失敗した[CallAndBridge](#)アクションを示しています。

```

{
  "SchemaVersion": "1.0",
  "Sequence": 3,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {
    "Type": "actionType",
    "Parameters": {
      // Parameters vary by Action Type
      "SipHeaders": {
        "X-AMZN": "String",
        "User-to-User":
"616d617a6f6e5f6368696d655f636f6e6e6563745f696e746567726174696f6e;encoding=hex",
        "Diversion": "sip:+11234567891@public.test.com;reason=unconditional"
      },
    },
    "ErrorType": "InvalidActionParameter",
    "ErrorMessage": "Invalid SIP header(s) provided: X-AMZN"
  },
  "CallDetails": {
    .....
    "Participants": [
      {
        "CallId": "call-id-1",

```

```
        "ParticipantTag": "LEG-A",
        .....
        "Status": "Connected"
    },
    {
        "CallId": "call-id-2",
        "ParticipantTag": "LEG-B",
        .....
        "Status": "Connected"
    }
]
}
```

SipHeaders フィールドの使用

[CreateSipMediaApplicationCall](#) API をトリガーするとき、オプションの SipHeaders フィールドを使用すると、カスタム SIP ヘッダーをアウトバウンド通話ログに渡すことができます。有効なヘッダーキーには、次のいずれかが含まれている必要があります。

- x- プレフィックス
- User-to-User ヘッダー
- Diversion ヘッダー

X-AMZN は予約ヘッダーです。このヘッダーを API コールで使用すると、失敗します。ヘッダーの最大長は 2048 文字です。

次の例は、コマンドラインインターフェイスの一般的な [CreateSipMediaApplicationCall](#) API とオプションの SipHeaders パラメータを示しています。

```
create-sip-media-application-call
  --from-phone-number value // (string)
  --to-phone-number value // (string)
  --sip-media-application-id value // (string)
  --sip-headers // (map)
```

詳細については、「[A Mechanism for Transporting User-to-User Call Control Information in SIP](#)」と「[Diversion Indication in SIP](#)」を参照してください。

通話詳細レコードの使用

Amazon Chime SDK 管理者は、通話詳細レコード (CDR) を保存するように Amazon Chime SDK Voice Connector を設定できます。CDR を保存するように Amazon Chime SDK Voice Connector を設定する方法の詳細については、「Amazon Chime SDK 管理ガイド」の「[Managing global settings in Amazon Chime SDK](#)」を参照してください。

CDR を有効にすると、各通話の後で、SIP メディアアプリケーションが S3 バケット内の Amazon-Chime-SMADR という名前のフォルダにレコードを送信します。

以下の表は、CDR の属性とその適切なフォーマットを示しています。レコードには、すべての通話について、ここにリストされているすべてのフィールドが含まれています。

値	説明
"AwsAccountId": " <i>AWS-account-ID</i> "	PSTN の使用を開始した SIP メディアアプリケーションに関連付けられた AWS アカウント ID
"TransactionId": " <i>transaction-ID</i> "	通話のトランザクション ID
"CallId": " <i>SIP-media-application-call-ID</i> "	参加者の通話 ID (関連する使用向け)
"VoiceConnectorId": " <i>voice-connector-ID</i> "	Amazon Chime SDK Voice Connector ID UUID
"Status": " <i>status</i> "	通話のステータス (完了、失敗)
"BillableDurationSeconds": " <i>billable-duration-in-seconds</i> "	請求可能な通話時間 (秒単位)
"SchemaVersion": " <i>schema-version</i> "	CDR スキーマバージョン
"SourcePhoneNumber": " <i>12075550155</i> "	発信元の電話番号 (E.164 形式)
"SourcePhoneNumberName": " <i>North Campus Reception</i> "	発信元の電話番号に割り当てられた名前

値	説明
"DestinationPhoneNumber": " <i>13605551214</i> "	送信先の電話番号 (E.164 形式)。
"DestinationPhoneNumberName": " <i>South Campus Reception</i> "	送信先の電話番号に割り当てられた名前
"UsageType": " <i>usage-type</i> "	Price List API の明細項目の使用状況の詳細
"ServiceCode": " <i>service-code</i> "	Price List API のサービスのコード
"Direction": " <i>direction</i> "	通話の方向 (Outbound または Inbound)
"TimeStampEpochSeconds": " <i>start-time-epochseconds</i> "	エポック/Unix タイムスタンプ形式のレコードのタイムスタンプ
"Region": " <i>AWS-region</i> "	AWS Amazon Chime SDK Voice Connector のリージョン
"SipRuleId": " <i>sip-rule-id</i> "	通話が PSTN Audio サービスに届いたときにトリガーされる SIP ルールの ID
"SipApplicationId": " <i>sip-application-id</i> "	通話进行处理する SIP アプリケーションの ID
"CallLegTriggerType": " <i>trigger-type</i> "	通話をトリガーしたイベントのタイプ
"BillableVoiceFocusSeconds": " <i>billable-voice-focus-in-seconds</i> "	請求可能な Voice Focus の使用量 (秒単位)

タイムアウトと再試行

PSTN Audio サービスは、AWS Lambda 関数と同期的にやり取りします。アプリケーションは、呼び出しを再試行する前に AWS Lambda 関数が応答するまで 5 秒待ちます。関数が 4XX ステータスコードのいずれかでエラーを返した場合、デフォルトでは、SIP メディアアプリケーションは呼び出しを 1 回だけ再試行します。再試行回数を使い切ると、通話は 480 Unavailable エラーコード

で終了します。AWS Lambda エラーの詳細については、[「」の「呼び出しの問題のトラブルシューティング AWS Lambda」](#)を参照してください。

デバッグとトラブルシューティング

以下の情報を使用して、Amazon Chime SDK PSTN Audio デバイスの使用時に発生する可能性がある一般的な問題の診断と修正に役立てます。

トピック

- [ログの確認](#)
- [予期しない切断のデバッグ](#)
- [予想しない ACTION_FAILED イベントのデバッグ](#)

ログの確認

SIP メディアアプリケーションをデバッグする場合は、アプリケーションに関連付けられた AWS Lambda 関数の Cloudwatch ログを確認します。

次に、SIP メディアアプリケーションに関連するログを確認します。必要に応じて、SIP メディアアプリケーションをログ記録用に設定できます。詳細については、「Amazon Chime SDK Administrator Guide」の「[Using SIP media applications](#)」を参照してください。ログ記録を有効にすると、Cloudwatch の `/aws/ChimeSipMediaApplicationSipMessages/` SIP メディアアプリケーション ID ロググループでログを検索できます。

予期しない切断のデバッグ

- AWS Lambda ポリシーが voiceconnector.chime.amazonaws.com サービスプリンシパルに `aws:lambda:InvokeFunction` 許可を付与していることを確認します。
- AWS Lambda 関数のログをチェックして、正常に呼び出されていることを確認します。
- ログに受信イベントと返されたアクションが表示されている場合は、AWS Lambda 関数が呼び出されたときに `hangup` アクションを返さないことを確認します。
- SIP メディアアプリケーションの Cloudwatch ログを確認します。返されるメッセージの一部を次の表に示します。

メッセージ	解決方法
AWS Lambda クライアントオペレーションがタイムアウトしました。	関数が完了するまでに 20 秒以上かかりました。応答時間を 20 秒未満に短縮してください。
AWS Lambda 関数の呼び出し中にアクセスが拒否されました。	この AWS Lambda 関数は、サービスが Amazon Chime SDK Voice Connector サービスプリンシパルにアクセスすることを許可するポリシーを提供しません。AWS Lambda サービスvoiceconnector.chime.amazonaws.com プリンシパルにポリシーの lambda:InvokeFunction アクセス許可を付与します。
AWS Lambda 関数がスロットリングされました。	AWS Lambda 関数がスロットリングされたため、Audio Service は関数を呼び出すことができませんでした。詳細については、 https://aws.amazon.com/premiumsupport/knowledge-center/lambda-troubleshoot-throttling/ を参照してください。
Error while reading actions list.	PSTN Audio サービスは、AWS Lambda 関数によって返されたアクションを解析できませんでした。ログに ACTION_FAILED イベントがないか確認し、失敗したアクションのドキュメントを参照して、正しくコーディングされていることを確認してください。
The schema version in the invocation request does not match the schema version in the response.	ログをチェックして、リクエストとレスポンスが同じスキーマバージョンを使用していることを確認してください。
Unsupported action name specified	AWS Lambda 関数は、PSTN Audio サービスが認識しなかったアクションを返しました。アクションのスペルが正しいことを確認し、アクションのマニュアルを参照してください。

メッセージ	解決方法
Actions list is empty.	NEW_INCOMING_CALL イベントへのレスポンスからアクションが何も返されませんでした。そのイベントに対するアクションを返してください。
Too many actions specified in a response.	呼び出しに応答して 10 を超えるアクションを返 AWS Lambda しました。10 個以下のアクションを返してください。
Response is blank or empty	NULL または空の文字列が返されました。レスポンスオブジェクトに少なくとも SchemaVersion フィールドが含まれていることを確認してください。

予想しない ACTION_FAILED イベントのデバッグ

予想しない ACTION_FAILED イベントが発生した場合は、次の点を確認します。

アクション	エラータイプ	エラーメッセージ	解決方法
CallAndBridge 、 PlayAudio 、および PlayAudioAndGetDigits	InvalidAudioSource	Cannot access S3 bucket or audio file.	<ul style="list-style-type: none"> S3 バケットが SIP メディアアプリケーションと同じ AWS アカウントにあることを確認します。 S3 バケットが voiceconnector.chime.amazonaws.com サービスプリンシパルに s3:GetObject アクセス許可を付与し

アクション	エラータイプ	エラーメッセージ	解決方法
			ていることを確認してください。
PlayAudio 、および PlayAudioAndGetDigits	InvalidAudioSource	Audio Source parameter value is invalid.	<ul style="list-style-type: none">• S3 などの有効な Type を使用していることを確認します。• S3 バケットが <code>voiceconnector.chime.amazonaws.com</code> サービスプリンシパルに <code>s3:GetObject</code> アクセス許可を付与していることを確認してください。• <code>BucketName</code> フィールドが <code>null</code> でも空でもないことを確認します。• <code>Key</code> フィールドが <code>null</code> でも空でもないことを確認します。

アクション	エラータイプ	エラーメッセージ	解決方法
CallAndBridge	InvalidAudioSource	Ringtone parameter value is invalid.	<ul style="list-style-type: none">• S3 などの有効な Type を使用していることを確認します。• BucketName フィールドが null でも空でもないことを確認します。• Key フィールドが null でも空でもないことを確認します。
	InvalidActionParameter	Invalid number of endpoints provided.	エンドポイントが null でも 0 でもなく、1 以下であることを確認してください。

アクション	エラータイプ	エラーメッセージ	解決方法
	InvalidActionParameter	Endpoint parameter is invalid.	<ul style="list-style-type: none"> エンドポイント URI 値が指定されていることを確認してください。 エンドポイントタイプが PSTN の場合は、URI フィールドに入力された電話番号が有効な E.164 電話番号であることを確認してください。 エンドポイントタイプが PSTN の場合は、ARN フィールドが設定されていないこと、または null に設定されていることを確認します。
	InvalidActionParameter	Invalid caller ID.	有効な E.164 形式の電話番号を CallerId フィールドに入力してください。
	InvalidActionParameter	Caller ID not defined.	有効な E.164 形式の電話番号を CallerId フィールドに入力してください。
	InvalidActionParameter	The MaxCallTimeout parameter is invalid. Timeout must be between 0 and 120 seconds.	MaxCallTimeout 間隔には 0~120 秒の値を設定してください。

アクション	エラータイプ	エラーメッセージ	解決方法
	InvalidActionParameter	Provided caller ID number is invalid. 番号は、この AWS アカウントが所有するか、LEG-A Fromの数である必要があります。	<ul style="list-style-type: none"> • CallerId 番号がプロビジョニングされ、SIP メディアアプリケーションと同じ AWS アカウントに関連付けられていることを確認します。 • 番号がアカウントに関連付けられていない場合は、LEG-A の From フィールドの番号と一致する必要があります。
	InvalidActionParameter	Invalid SIP header(s) provided: {Header}.	<ul style="list-style-type: none"> • 内部カスタムヘッダー x-vine、x-amzn、x-vc、x-canary、x-voice をすべて削除してください。 • カスタムヘッダーが x- で始まっていることを確認します。user-to-user または diversion に設定することもできます。

アクション	エラータイプ	エラーメッセージ	解決方法
JoinChimeMeeting	InvalidActionParameter	JoinToken パラメータ値が無効です。	<ul style="list-style-type: none"> 会議参加トークンが正しいことを確認してください。 トークンが関連付けられている参加者がまだ会議の有効な出席者であることを確認してください。 会議がまだ存在していることを確認してください。
ModifyChimeMeetingAttendee (オーディオのミュートとミュート解除)	InvalidActionParameter	The value in the Operation field of the ModifyChimeMeetingAttendees action is invalid.	サービスがミュート操作とミュート解除操作をサポートしていることを確認してください。
	InvalidActionParameter	The meeting ID parameter is invalid.	会議 ID が正しいことを確認してください。
	InvalidActionParameter	Attendee List parameter is invalid.	参加者が指定されていないか、100 人を超える出席者が指定されました。参加者は 1 ~ 100 人にしてください。

アクション	エラータイプ	エラーメッセージ	解決方法
	InvalidActionParameter	One or more attendees are not part of this meeting. All attendees must be part of this meeting.	アクションで指定された参加者のうちの 1 人が指定された会議の有効な参加者ではありません。会議に出席していない参加者をすべて削除してください。
Pause	InvalidActionParameter	The Duration parameter is invalid.	一時停止時間を 100 ~ 30000 の間に設定してください。
PlayAudioAndGetDigits	InvalidActionParameter	The MaxNumberOfDigits parameter is invalid.	MaxNumberOfDigits が 0 ~ 128 の間で、MinNumberOfDigits / より大きいことを確認してください
	InvalidActionParameter	The RepeatDurationInMilliseconds parameter is invalid.	RepeatDurationInMilliseconds 値が正であることを確認してください。
	InvalidActionParameter	The InputDigitsRegex parameter is invalid.	InputDigitsRegex が有効な正規表現パターンであることを確認してください。
ReceiveDigits	InvalidActionParameter	The InBetweenDigitsDurationInMilliseconds parameter is invalid.	値が 0 より大きいことを確認してください。

アクション	エラータイプ	エラーメッセージ	解決方法
	InvalidActionParameter	The FlushDigitsDurationInMilliseconds parameter is invalid.	FlushDigitsDurationInMilliseconds 間隔が InBetweenDigitsDurationInMilliseconds 間隔以下になっています。InBetweenDigitsDurationInMilliseconds 間隔を FlushDigitsDurationInMilliseconds 間隔より大きくしてください。
	InvalidActionParameter	The InputDigitsRegex parameter is invalid.	値が空でも null でもないことを確認してください。

アクション	エラータイプ	エラーメッセージ	解決方法
RecordAudio	InvalidActionParameter	The Recording Destination parameter is invalid.	<ul style="list-style-type: none">• Type フィールドが S3 など有効であることを確認してください。• BucketName フィールドが空でも NULL でもないことを確認してください。• プレフィックスが有効な文字で構成されていることを確認してください。• プレフィックスが 979 バイト以下であることを確認してください。
	InvalidActionParameter	The DurationInSeconds parameter is invalid.	DurationInSeconds は null にすることはできず、0 より大きい必要があります。
	InvalidActionParameter	The SilenceThreshold parameter is invalid.	SilenceThreshold は null にすることはできず、1 ~ 1000 の範囲の値にする必要があります。

アクション	エラータイプ	エラーメッセージ	解決方法
	InvalidActionParameter	The SilenceDurationInSeconds parameter is invalid.	SilenceDurationInSeconds は null にすることはできず、0 より大きい必要があります。
	InvalidActionParameter	An error occurred while uploading the recording to the S3 bucket.	<ul style="list-style-type: none"> S3 バケットが SIP メディアアプリケーションと同じ AWS アカウントにあることを確認します。 S3 バケットが voiceconnector.chime.amazonaws.com サービスプリンシパルに s:PutObject および s:PutObjectAccess 許可の使用を付与していることを確認してください。
VoiceFocus	MissingRequiredActionParameter	Missing a required action parameter.	Enable パラメータに有効なブール値を指定してください。

VoiceFocus

Amazon Voice Focus のノイズ抑制を、公衆交換電話網 (PSTN) 通話の受信および発信通話レグに適用できます。Amazon Voice Focus を適用すると、人間の音声に影響を与えることなくバックグラウンドノイズが低減します。これにより、現在の話者が聞き取りやすくなります。

受信通話レグを作成するには、NewInboundCall イベントで AWS Lambda 関数を呼び出す [SIP ルール](#)を使用します。アウトバウンドコールレグを作成するには、[CallAndBridge](#)アクションを使用するか、[CreateSIPMediaApplicationCall](#) API オペレーションを使用します。Amazon Voice Focus の詳細については、「[How the Amazon Chime SDK's noise cancellation works](#)」を参照してください。

Amazon Voice Focus は、次のような音声以外の不要なノイズを低減させます。

- 環境ノイズ — 風、換気扇、流水
- バックグラウンドノイズ - 芝刈り機、犬の鳴き声
- フォアグラウンドノイズ - タイピング、書類をめくる音

Note

Amazon Voice Focus を使用すると、は通話レグごとのアクティブな通話分と SIP メディアアプリケーションの使用分ごとに AWS 課金します。

以下の例は、一般的な VoiceFocus アクションを示しています。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "VoiceFocus",
      "Parameters": {
        "Enable": True|False,           // required
        "CallId": "call-id-1",         // required
      }
    }
  ]
}
```

Enable

説明 - Amazon Voice Focus を有効または無効にします。

使用できる値 - True | False

必須 - はい

デフォルト値 – なし

CallId

説明 — AWS Lambda 関数呼び出しCallDetailsの における CallId 参加者の

使用できる値 – 有効な通話 ID

必須 – はい

デフォルト値 – なし

以下の例は、VoiceFocus アクションの成功した ACTION_SUCCESSFUL イベントを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 3,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "VoiceFocus",
    "Parameters": {
      "Enable": True,
      "CallId": "call-id-1"
    }
  },
  "CallDetails": {
    .....
    .....
    "Participants": [
      {
        "CallId": "call-id-of-caller",
        .....
        "Status": "Connected"
      },
      {
        "CallId": "call-id-of-callee",
        .....
        "Status": "Connected"
      }
    ]
  }
}
```

以下の例は、VoiceFocus アクションの一般的な ACTION_FAILED イベントを示しています。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 2,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {
    "Type": "VoiceFocus",
    "Parameters": {
      "Enable": True,
      "CallId": "call-id-1"
    }
  },
  "ErrorType": "SystemException",
  "ErrorMessage": "System error while running action"
},
"CallDetails": {
  .....
  .....
  "Participants": [
    {
      "CallId": "call-id-of-caller",
      .....
    }
  ]
}
}
```

エラー処理

セキュリティ上の理由から、PSTN Audio アクションでは、顧客アカウントごとに 1 秒あたり 5 件の通話リクエスト (CPS) に制限されています。通話リクエストが 5 CPS の制限を超えると、アクションはエラーメッセージを返します。VoiceFocus アクションから返されるエラーメッセージを以下の表に示します。

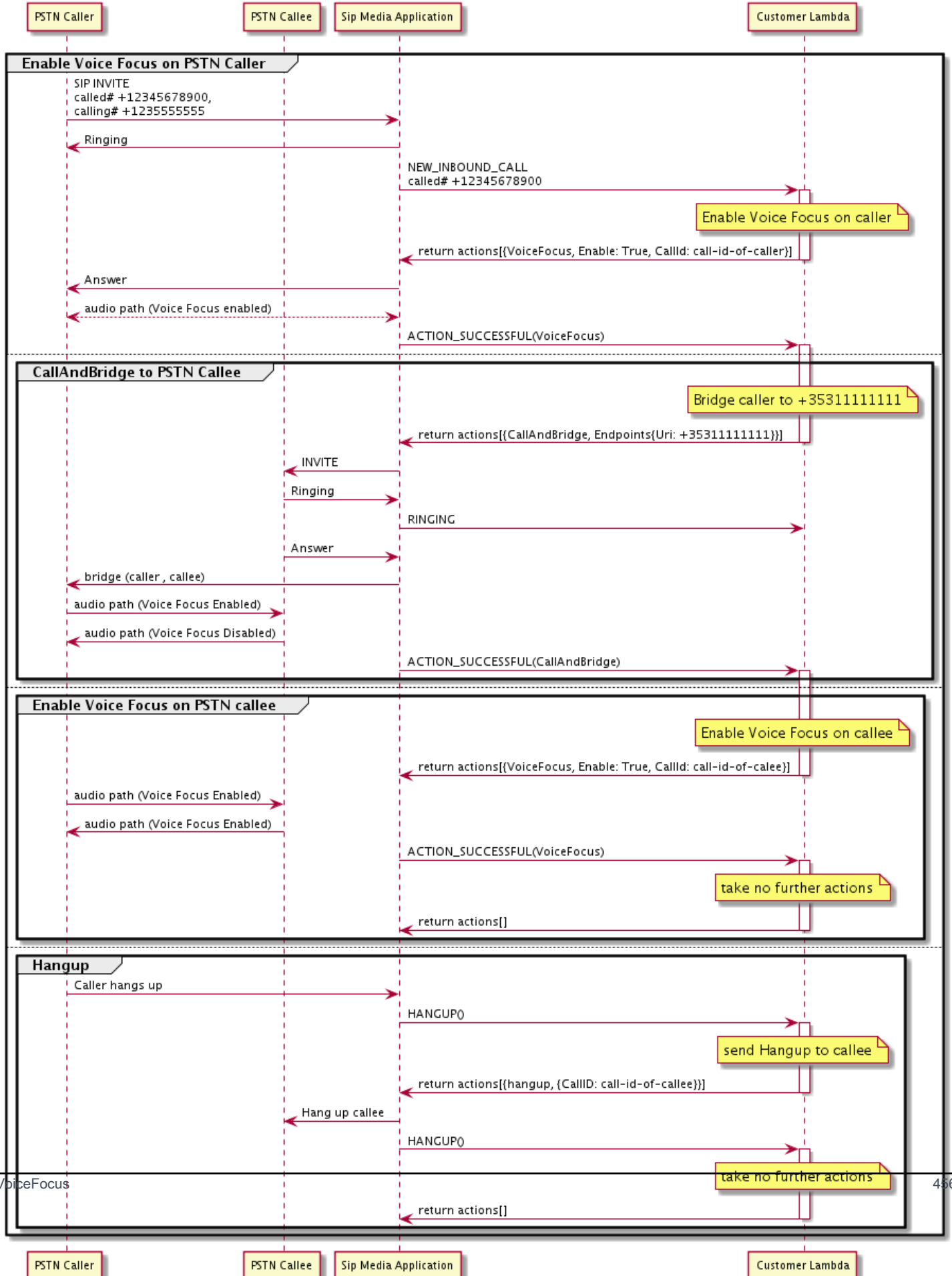
エラー	メッセージ	理由
ActionExecutionThrottled	アクションを実行できませんでした。1 秒あたりのアクションの最大数に達しました。	1 秒あたりの Voice Focus アクションリクエスト数がシステム制限を超えました。

エラー	メッセージ	理由
MissingRequiredActionParameter	アクションの必須パラメータがありません。	アクションの実行中に必須パラメータが 1 つ以上不足しています。
SystemException	アクションの実行中にシステムエラーが発生しました。	アクションの実行中にシステムエラーが発生しました。

通話フロー

以下の図は、2 つの PSTN 通話間の CallAndBridge アクションに対して Amazon Voice Focus を有効または無効にする通話フローを示しています。

Voice Focus between 2 PSTN parties



発信通話レグの場合、AWS Lambda 関数は発信者の Amazon Voice フォーカスを有効にし、を含む一連のアクションを返します CallAndBridge。通話がブリッジされると、VoiceFocus アクションは ACTION_SUCCESSFUL イベントを返します。そして、Lambda 関数は、通話されている人に対して Amazon Voice Focus を有効にする別のイベントセットを返します。この一連のアクションには、VoiceFocus、Enable、True、および発信者の ID が含まれます。発信者がハングアップするまで、それ以上のアクションは実行されません。次に、Lambda 関数が SIP メディアアプリケーションに Hangup アクションを送信します。アプリケーションは、通話されている人をハングアップし、Hangup 関数を Lambda 関数に返します。Lambda 関数は、それ以上アクションを実行しません。

PSTN Audio サービス用語集

| [A](#) | [C](#) | [E](#) | [I](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [S](#) | [T](#) | [V](#) |

A

アクション

AWS Lambda 関数では、アクションは、数字の送受信、会議への参加など、通話のレグで実行する項目です。PSTN Audio サービスが対応しているアクションの詳細については、「[PSTN Audio サービスでサポートされるアクション](#)」を参照してください。

AWS Lambda

サーバーのプロビジョニングや管理を行わなくても、ほぼすべてのタイプのアプリケーションやバックエンドサービスのコードを実行できるようにするコンピューティングサービスです。

AWS Lambda 関数

PSTN Audio サービスの文脈では、発信などによって SIP メディアアプリケーションから渡されたデータに応じて実行可能な関数です。

C

通話詳細レコード

Amazon Chime SDK Voice Connector の通話から取得されたデータ (アカウント ID、発信元の電話番号、発信先の国など)。このレコードは、アカウントの Amazon Simple Storage Service (S3) バケットにオブジェクトとして格納されます。詳細については、「Amazon Chime SDK Administrator Guide」の「[Managing global settings in Amazon Chime SDK](#)」を参照してください

い。レコードスキーマの詳細については、このガイドの「[通話詳細レコードの使用](#)」を参照してください。

通話 ID

すべての着信のログに割り当てられる ID。

通話ログ

通話の一部を意味します。Amazon Chime SDK アプリケーションでは、有効な電話番号、PSTN、Amazon Chime SDK Voice Connector から通話を受けることができます。詳細については、このガイドの「[PSTN Audio サービスの通話ログの使用について](#)」を参照してください。

Carrier

モバイルサービスを提供する会社を意味し、ワイヤレスキャリアの略称でもあります。

Amazon Chime

が提供する統合コミュニケーションおよびコラボレーションサービス AWS。

Amazon Chime SDK

ソフトウェア開発キット。開発者がカスタムコミュニケーションアプリケーションにリアルタイムメディアやコミュニケーション機能を追加するために使用します。

E

E.164

PSTN Audio サービスで唯一認められている電話番号形式です。ITU-T 推奨の番号では、1~3桁の国コードが使用され、その後に最大 12 桁のサブスクライバー番号が続きます。例: 米国: +14155552671、英国: +442071838750 44、オーストラリア: +61285993444

エンドポイント

電話機やユニファイドコミュニケーションアプリケーションなどのハードウェアデバイスまたはソフトウェアサービス。

EventBridge

さまざまなソースからのデータにアプリケーションを接続できるサーバーレスイベントバスサービス。

Note

SIP メディアアプリケーションは にデータを送信しません EventBridge。詳細については、[「Amazon Chime SDK 管理者ガイド」の「による EventBridge Amazon Chime SDK の自動化」](#)を参照してください。

I

IVR

Interactive Voice Response (自動音声応答システム) の略語。音声認識またはタッチトーンキーパッドを使用して、コンピュータによって操作される電話システムと対話できるシステム。

L

レグ

[Call leg](#) を参照してください。

M

メディア

Amazon Chime SDK ミーティングで使用できる音声、動画、チャットメッセージ。カスタムコミュニケーションアプリケーションでは、各メディアタイプを1つ以上使用できます。

メディアパイプライン

Amazon Chime SDK ミーティングで音声、動画、メッセージ、イベントをストリーミングしてキャプチャするためのメカニズム。詳細については、このガイドの「[Amazon Chime SDK メディアパイプラインを作成する](#)」を参照してください。

N

電話番号ポータビリティ

電話会社間やユニファイドコミュニケーションシステム間で電話番号を移転する機能。

O

発信

PSTN から通話を受信し、その通話を VoIP エンドポイントに渡すプロセス。

P

参加者タグ

各通話参加者に割り当てられる識別子 (LEG-A または LEG-B)。

ポリシー

Amazon Chime SDK には、次のタイプのポリシーが必要です。

- IAM ユーザーポリシー — Identity and Access Management ユーザーの権限を定義するポリシー。
- 会議ポリシー — 会議中に画面を共有しているときに、ユーザーが別のユーザーのコンピュータを制御できるポリシー。このポリシーにより、Amazon Chime SDK から通話を受けたユーザーが会議に参加するオプションも有効にできます。

PSTN

公衆交換電話網。通話機能を提供するインフラストラクチャとサービス。

PSTN Audio サービス

コミュニケーションソリューションソリューションに音声機能を追加できるようにする Amazon Chime SDK サービス。

R

ルーティング

Amazon Chime SDK で作成したアプリケーションでは、1 つ以上のルーティングタイプが使用されます。

- ネットワークルーティング — ネットワーク内、複数のネットワーク間、複数のネットワークをまたぐトラフィックのパスを選択するプロセス。
- インタラクショナルルーティング — 通話を正しい受信者またはエンドポイントに正確に転送するためのプロセス。

- 通話ルーティング — 着信をキューに入れ、あらかじめ定義された受信者またはエンドポイントに配信する通話管理機能。

S

SBC

セッションボーダーコントローラー。SIP ベースの Voice over Internet Protocol (VoIP) ネットワークを保護するために導入するネットワーク機器。

[Sequence] (シーケンス)

AWS Lambda 関数を呼び出すイベントのシーケンス。シーケンスは、通話中に関数が呼び出されるたびに増加します。

サービス制限/サービスクォータ

Amazon Chime SDK で許可される、会議、音声ストリーム、コンテンツ共有といったリソースの最大数。詳細については、このガイドの「[Audio](#)」を参照してください。

SIP

シグナリングプロトコルである Session Initiation Protocol の略語。リアルタイムセッションの開始、維持、終了に使用されます。音声、動画、メッセージングのアプリケーションを組み合わせる場合にも使用できます。詳細については、「[SIP: Session Initiation Protocol](#)」を参照してください。

SIP ヘッダー

通話制御データ、およびユーザーアカウント IDs などのその他のデータを含む AWS Lambda 関数のパラメータ。

SIP メディアアプリケーション

SIP ルールからターゲット AWS Lambda 関数に値を渡すマネージドオブジェクト。デベロッパーは [CreateSipMediaApplication](#) API を呼び出して SIP メディアアプリケーションを作成できますが、そのためには管理者権限が必要です。

SIP ルール

Amazon Chime SDK Voice Connector URI の電話番号を対象の SIP メディアアプリケーションに渡す管理対象オブジェクト。

SIP トランク

[Amazon Chime SDK Voice Connector](#) を参照してください。

SMA

SIP メディアアプリケーションを参照してください。

SMA ID

SIP メディアアプリケーションを参照してください。

T

Telco

通信サービスプロバイダー。

終了

通話を終了するプロセス。

トランザクション

1 つ以上の通話レグで構成される通話。詳細については、このガイドの「[PSTN Audio サービスの通話レグの使用について](#)」を参照してください。

トランザクション ID

複数の通話レグで構成されるトランザクションの ID。詳細については、このガイドの「[PSTN Audio サービスの通話レグの使用について](#)」を参照してください。

V

Amazon Chime SDK Voice Connector

電話システム用の Session Initiation Protocol (SIP) トランキングサービスを提供するオブジェクト。Voice Connector を作成し管理するには、Amazon Chime SDK 管理コンソールを使用します。詳細については、「Amazon Chime SDK Administrator Guide」の「[Managing Amazon Chime SDK Voice Connectors](#)」を参照してください。

Amazon Chime SDK Voice Connector グループ

異なる AWS リージョンの複数の Voice Connector を含むラッパー。このグループによって、リージョン全体で着信をフェイルオーバーすることで、耐障害性のメカニズムを実現できます。詳細については、「Amazon Chime SDK Administrator Guide」の「[Managing Amazon Chime SDK Voice Connector groups](#)」を参照してください。

通話分析を使用した通話からのインサイトの生成

このセクションのトピックでは、Amazon Chime SDK 通話分析を使用して通話データからインサイトを生成する方法について説明します。

Amazon Chime SDK 通話分析の導入により、ローコードソリューションを実現して、リアルタイム音声から費用対効果の高いインサイトを生成でき、音声データの取り込み、分析、アラート、データレイク統合なども行えます。通話分析を使用すると、Amazon Transcribe と Transcribe Call Analytics (TCA) の統合に加え、Amazon Chime SDK 音声分析をネイティブに行うことで、インサイトを生成できます。また、Amazon S3 バケットに通話を録音することも可能です。

通話分析を設定および実行するには、次の方法を使用します。

- Amazon Chime SDK コンソールを使用して通話分析を設定し、それを Amazon Chime SDK Voice Connector に関連付けます。その過程で、通話の録音と分析を有効にできます。この手順を完了するためにコードを記述する必要はありません。
- [Amazon Chime SDK API](#) のセットを使用して、プログラムで設定を行い、実行します。

詳細については、このセクションで後述する「[通話分析を設定する](#)」と「[通話分析の設定を使用する](#)」を参照してください。

トピック

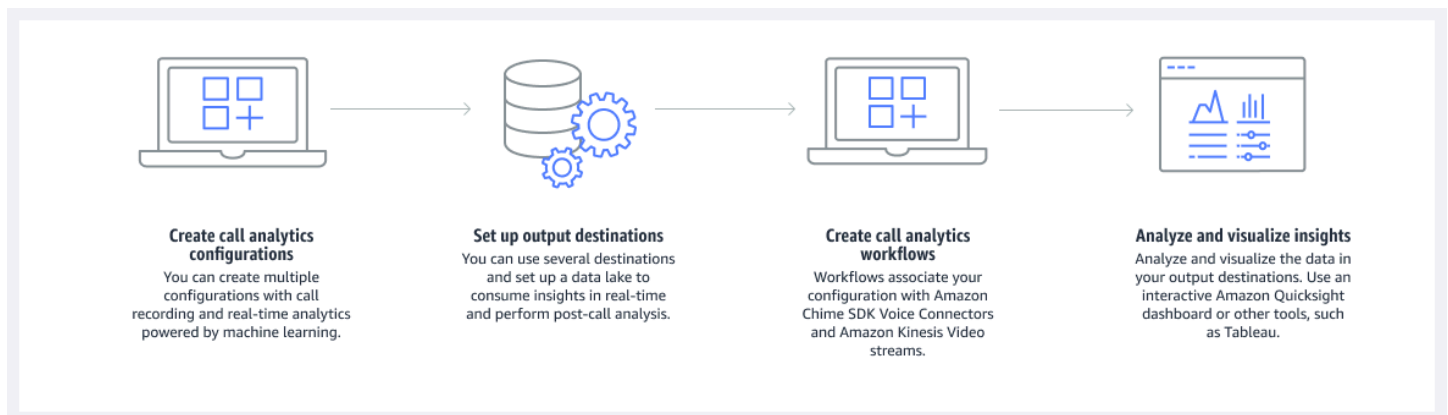
- [Amazon Chime SDK 通話分析とは](#)
- [通話分析の用語](#)
- [通話分析を設定する](#)
- [通話分析の設定を使用する](#)
- [通話分析パイプラインを管理する](#)
- [通話分析パイプラインの一時停止と再開を行う](#)
- [通話分析のリソースアクセスロールを使用する](#)
- [通話分析のステータスを理解する](#)
- [Amazon での通話分析パイプラインのモニタリング CloudWatch](#)
- [通話分析プロセッサと出力先](#)
- [通話分析のデータモデル](#)

- [Amazon Chime SDK 音声分析を使用する](#)
- [通話分析のサービスクォータ](#)

Amazon Chime SDK 通話分析とは

Amazon Chime SDK の導入により、ローコードソリューションを実現して、リアルタイム音声から費用対効果の高いインサイトを生成でき、音声データの取り込み、録音、音声分析、アラート、データレイクの利用なども行えます。また、機械学習を活用したインサイト生成も可能になります。これを実現するために、再利用可能な通話分析の設定を行い、ワークフローで有効にする AWS 機械学習統合と音声処理機能を決定します。その通話分析設定は、Voice Connector や Amazon Kinesis Video Streams などのさまざまなメディアソースにも使用します。通話分析では、Amazon Transcribe と Transcribe Call Analytics (TCA) の統合に加え、[Amazon Chime SDK 音声分析](#) (通話分析で実行されるサービス) をネイティブに行うことでインサイトを生成します。

通話分析を使用するには、次の手順に従います。



図中の手順を説明します。

1. 最初に通話分析を設定します。
2. 出力先とデータレイク (オプション) を設定します。
3. その設定を Voice Connector と Amazon Kinesis Video Streams に関連付けるワークフローを作成します。
4. インサイトを分析し、必要に応じて視覚化します。

Amazon Chime SDK コンソールを使用して、通話分析を設定し、通話分析の自動開始を有効にすることができます。特定のタイプの呼び出しに適用する設定を制御する必要がある場合は、API を使用して設定を行います。いずれの場合も、設定には、通話音声に対して呼び出す AWS 機械学習サービ

スの詳細、通話録音の有効化、インサイト、メタデータ、録音の送信先が含まれます。通話分析では次の送信先を使用できます。

- 1 つの Amazon Kinesis Data Streams (KDS)。KDS を使用して通話のインサイトをライブで受信し、それをアプリケーションに統合できます。例えば、ライブインサイトを統合して顧客との通話中に販売エージェントやカスタマーサポートエージェントを支援したり、インサイトを使用して生成 AI のプロンプトや概要を改善したりできます。
- データウェアハウスとして設定された Amazon S3 バケット。バケットには Parquet 形式でデータを格納します。Parquet は、大量のデータを圧縮し、保存することを目的としたオープンソースのファイル形式です。データを格納したら、Amazon Athena を利用して、単純なクエリ言語 (SQL) でそれらのデータを参照したり、データを既存のデータウェアハウスに移動してビジネスデータと統合したりできます。例えば、通話後の集計を分析して、顧客との通話の有効性や製品の問題点を把握することも、従業員のトレーニング機会を把握して顧客満足度を向上させることも可能です。

こうした送信先に加え、通話分析は、インサイトに基づいて事前構成できるリアルタイムアラートにも対応しています。アラートは Amazon に送信されます EventBridge。

Note

通話分析の設定では、特定の音声ソースを選択しません。これにより、複数の音声ソース間で設定を再利用できます。例えば、通話録音の有効化と、通話の文字起こしを 1 つの設定で行えます。その設定は、Chime SDK Voice Connector や、Kinesis Video Streams 経由のオーディオストリームでも使用できます。複数の Voice Connector 間で共有することも可能です。通話分析の各設定は一意であり、ARN によって識別されます。

通話分析の用語

Amazon Chime SDK 通話分析を理解するために重要な用語と概念を次に示します。

Amazon Athena

標準 SQL を使用して Amazon S3 のデータを分析できるインタラクティブなクエリサービス。Athena は、サーバーレスであるため、インフラストラクチャを管理する必要がなく、料金は実行したクエリに対してのみ発生します。Athena を利用するには、Amazon S3 内のデータを指定してスキーマを定義し、標準 SQL クエリを使用します。また、ワークグループでユーザーをグループ化し、クエリの実行時にユーザーがアクセスできるリソースを制御することもできます。ワークグループを使用すると、クエリの同時実行を管理して、さまざまなユーザーグ

ループやワークロードによるクエリ実行に優先順位を付けることが可能です。詳細については、「[Amazon Athena とは](#)」を参照してください。

Amazon Kinesis Data Firehose

抽出、変換、ロード (ETL) サービス。これにより、ストリーミングデータをデータレイク、データストア、分析サービスに、確実にキャプチャ、変換、配信することが可能です。詳細については、「[Amazon Kinesis Data Firehose](#)」を参照してください。

通話分析データウェアハウス

通話分析データ用のオプションストレージ。このウェアハウスでは、Parquet ベースのデータファイル形式で Amazon S3 バケットにデータを格納します。データのクエリには、標準 SQL を使用できます。ウェアハウスは通話分析設定で有効にします。

Glue Data Catalog

さまざまなデータソースのデータ資産を一元管理できるメタデータリポジトリであり、データベースとテーブルで構成されています。通話分析を行う際、Athena では、このテーブル内のメタデータによって Amazon S3 バケットの場所が判断されます。また、列名、データ型、テーブル名などのデータ構造も指定されます。データベースで保持されるのは、データセットのメタデータとスキーマ情報のみです。詳細については、このセクションで後述する「[Glue データカタログのテーブル構造](#)」を参照してください。

メディアインサイトパイプライン

一意の MediaPipelineId によって識別される一時的なリソース。これを作成するには、通話分析パイプラインの設定と、ランタイムパラメータを使用します。このランタイムパラメータによって、パイプラインのデータソースを指定します。

メディアインサイトのパイプライン設定

メディアインサイトパイプラインの作成に使用する静的な設定。1 つ以上のパイプラインをインスタンス化するには、この設定を使用します。

メディアインサイトパイプライン設定の構成要素

メディアインサイトパイプライン設定は、プロセッサ要素によってメディアを処理するための命令や、シンク要素によって生成したインサイトを配信するための命令などで構成されます。

メディアインサイトパイプラインタスク

メディアインサイトパイプラインの一時的なサブリソース。このタスクにより、特定のストリーム ARN およびチャンネル ID のプロセスステータスに関するメタデータが保持されます。タスク

は、固有の ID で識別され、メディアインサイトパイプラインで音声分析を開始する際に作成されます。

発話者検索

通話参加者の認識に役立つ音声分析機能。

音声分析

発話者検索やボイストーン分析などを行える Amazon Chime SDK の機能。

音声埋め込み

発信者の音声を、一意の ID を付け、ベクター形式で表現した情報。

音声エンハンスメント

通話の音質を向上させるシステム。

音声プロファイル

音声埋め込み、ID、有効期限を組み合わせた情報。

音声プロファイルドメイン

音声プロファイルの集合体。

音声トーン分析

音声分析機能。これにより、positive、negative または neutral のセンチメントを分析できます。

通話インサイトの設定、パイプラインの開始、音声分析の実行に使用する API の詳細については、「Amazon Chime SDK API Reference」の「[Amazon Chime SDK Media Pipelines](#)」を参照してください。

Note

通話分析の実行には、メディアインサイトパイプライン API の使用を強くお勧めします。新しい機能が利用できるのは、メディアインサイトパイプライン API のみだからです。メディアパイプラインと音声名前空間の違いについては、このセクションで後述する「[音声 API を使用して音声分析を実行する](#)」で詳しく確認できます。

通話分析を設定する

通話分析を使用するには、最初に設定を行います。設定とは、通話分析パイプラインの作成に必要な情報を保持する静的構造を意味します。Amazon Chime SDK コンソールを使用して設定を作成するか、[CreateMediaInsightsPipelineConfiguration](#) API を呼び出すことができます。

通話分析の設定では、録音、音声分析、Amazon Transcribe などの音声プロセッサに関する詳細設定に加え、インサイト送信先とアラートイベントも設定します。必要に応じて、通話データを Amazon S3 バケットに保存し、さらに分析することもできます。

ただし、設定では、特定の音声ソースを指定しません。これにより、その設定を複数の通話分析ワークフローで再利用できます。例えば、同じ通話分析設定を別の Voice Connector で使用したり、別の Amazon Kinesis Video Streams (KVS) ソースで使用したりできます。

この設定を使用すると、Voice Connector を介して SIP 通話が発生したとき、または新しいメディアを Amazon Kinesis Video Streams (KVS) に送信するときに、パイプラインを作成できます。これにより、設定内の指定に従って、パイプラインでメディアが処理されます。

パイプラインは、いつでもプログラムで停止できます。Voice Connector の通話が終了すると、パイプラインのメディアの処理は停止します。パイプラインは、一時停止することもできます。これにより、基盤となる Amazon 機械学習サービスへの呼び出しを無効にし、必要に応じて再開することが可能です。ただし、パイプラインを一時停止している間も、通話の録音は続きます。

次のセクションでは、通話分析の設定に必要な前提条件と、設定方法について説明します。

トピック

- [前提条件](#)
- [Amazon Chime SDK コンソールを使用して、通話分析を設定する](#)
- [API を使用して通話分析の設定を行う](#)
- [設定と Voice Connector を関連付ける](#)

前提条件

通話分析を設定する前に必要な事項を次に示します。AWS コンソールを使用して作成できます。

- Amazon Chime SDK Voice Connector。これを用意できない場合は、「[Creating Amazon Chime SDK Voice Connectors](#)」を参照してください。次の操作も必要です。

- Voice Connector のストリーミングを有効にします。詳細については、[「Amazon Chime SDK 管理者ガイド」の「による EventBridge Amazon Chime SDK の自動化」](#)を参照してください。
- 通話分析を使用するよう、Voice Connector を設定します。詳細については、「Amazon Chime SDK Administrator Guide」の[「Configuring Voice Connectors to use call analytics」](#)を参照してください。
- Amazon EventBridge ターゲット。そうでない場合は、「[による Amazon Chime SDK のモニタリング EventBridge](#)」、「Amazon Chime SDK 管理者ガイド」を参照してください。
- Voice Connector が EventBridge ターゲット上のアクションにアクセスできるようにするサービスにリンクされたロール。詳細については、「Amazon Chime SDK Administrator Guide」の[「Using the Amazon Chime SDK Voice Connector service linked role policy」](#)を参照してください。
- 1 つの Amazon Kinesis Data Streams。これを用意できない場合は、「Amazon Kinesis Streams Developer Guide」の[「Creating and Managing Streams」](#)を参照してください。音声分析と文字起こしには Kinesis Data Stream が必要です。
- 通話をオフラインで分析するには、Amazon Chime SDK データレイクを作成する必要があります。これを行うには、本ガイドで後述する[「Amazon Chime SDK データレイクを作成する」](#)を参照してください。

Amazon Chime SDK コンソールを使用して、通話分析を設定する

前のセクションに記載されている前提条件を満たすことができれば、Amazon Chime SDK コンソールを使用して 1 つ以上の通話分析を設定できます。コンソールからは、1 つまたは複数の Voice Connector を関連付けることもできます。このプロセスを完了すると、設定時に有効にした機能を使用して通話分析が実行されます。

通話分析を設定するには、次の手順に従います。

1. 名前やオプションのタグなど、設定の詳細を指定します。
2. 録音の設定を行います。録音に加え、機械学習を活用したインサイトといった、通話分析を設定します。
3. 分析サービスを設定します。
4. リアルタイムでのインサイト取得に必要な出力先を選択します。データレイクを作成し、通話後分析を行います (オプション)。
5. 新しいサービスロールを作成するか、既存のロールを使用します。
6. 特定の条件が満たされ EventBridge たときに Amazon 経由で通知を送信するリアルタイムアラートを設定します。

7. 指定した内容を確認し、設定を完了します。

設定を完了したら、Voice Connector をその設定に関連付けて通話分析を有効にします。これにより、その Voice Connector で通話の着信が発生したタイミングで通話分析が自動的に開始されます。詳細については、このセクションで後述する「[設定と Voice Connector を関連付ける](#)」を参照してください。

次のセクションでは、プロセスの各手順を完了する方法について説明します。リストされている項目をこの順序で展開してください。

設定の詳細を指定する

設定の詳細を指定するには、

1. <https://console.aws.amazon.com/chime-sdk/home> から Amazon Chime コンソールを開きます。
2. ナビゲーションペインの [通話分析] で [設定] を選択し、[設定の作成] を選択します。
3. [基本的な情報] で、以下を実行します。
 - a. 設定の名前を入力します。ユースケースとタグがわかる名前にする必要があります。
 - b. (オプション) [タグ] で [新しいタグの追加] を選択し、タグキーとオプションの値を入力します。キーと値を定義します。タグは、設定のクエリに役立ちます。
 - c. [次へ] をクリックします。

録音を設定する

録音を設定するには

- [記録を設定] ページで、次の手順を実行します。
 - a. [通話録音を有効にする] チェックボックスを選択します。これにより、Voice Connector の通話や KVS のストリームを録音し、そのデータを Amazon S3 バケットに送信できます。
 - b. [ファイル形式] で [WAV と PCM] を選択します。これにより、最良の音質が得られます。

または

[OGG と OPUS] を選択すると、音声を圧縮し、ストレージを最適化できます。

- c. (オプション) 必要に応じて、[Amazon S3 バケットの作成] リンクを選択し、次の手順に従って Amazon S3 バケットを作成します。

- d. Amazon S3 バケットの URI を入力するか、[参照] を選択してバケットを検索します。
- e. (オプション) [音声エンハンスメントを有効にする] を選択すると、録音の音質が向上します。
- f. [次へ] をクリックします。

音声エンハンスメントを理解する

通話分析の設定が完了したら、通話録音を有効にして、録音した通話を Amazon S3 バケットに保存できます。その手順中に音声エンハンスメントを有効にし、保存した通話の音質を向上させることも可能です。音声エンハンスメントは、それを有効にした後に生成された録音にのみ適用されます。音声エンハンスメントが有効な場合、元の録音に加え、拡張録音が行われ、それらは Amazon S3 バケットに同じ形式で保存されます。音声エンハンスメントでは、通話の拡張録音を最大 30 分行えます。30 分を超える通話では、拡張録音は行えません。

通話は、ナローバンドフィルタ処理され、8 kHz でサンプリングされます。音声エンハンスメントでは、サンプリングレートを 8 kHz から 16 kHz に引き上げ、機械学習モデルを使用して周波数成分を狭帯域から広帯域に拡張することで、より自然な音声を実現しています。また、Amazon Voice Focus と呼ばれるノイズリダクションモデルを使用し、拡張機能を使用した音声でバックグラウンドノイズも低減します。

音声エンハンスメントでは、Voice Focus と呼ばれるノイズリダクションモデルも使用します。このモデルにより、拡張機能を使用した音声で、バックグラウンドノイズを低減できます。このモデルは、アップグレードされた 16 kHz 音声に適用されます。

Note

音声エンハンスメントは、米国東部 (バージニア北部) リージョンと米国西部 (オレゴン) リージョンでのみサポートされています。

音声エンハンスメント録音メタデータは、設定された KDS を介して既存の AWS Glue データカタログテーブル `call_analytics_recording_metadata` に発行されます。音声拡張通話録音から元の通話録音レコードを識別するために、値を持つ `detail-subtype` という新しいフィールド `VoiceEnhancement` が KDS 通知とグルーテーブル `call_analytics_recording_metadata` に追加されます。データウェアハウススキーマの詳細については、「[通話分析のデータモデル](#)」を参照してください。

音声エンハンスメントで 사용되는ファイル形式

音声の拡張機能のファイルについては、次の点に注意します。

- 拡張機能による録音は、通常の録音と同じ Amazon S3 バケットに書き込まれます。送信先を設定するには、[S3RecordingSinkConfiguration](#) または [S3RecordingSinkRuntimeConfiguration](#) APIs を呼び出すか、Amazon Chime SDK コンソールを使用します。
- 音声の拡張機能では、ベースファイル名に `_enhanced` が追加されます。
- 音声の拡張機能では、元の録音と同じファイル形式が維持されます。ファイル形式を設定するには、[S3RecordingSinkConfiguration](#) または [S3RecordingSinkRuntimeConfiguration](#) APIs コンソールを使用します。

次の例は、一般的なファイル名の形式を示しています。

```
s3://original_file_name_enhanced.wav
```

または

```
s3://original_file_name_enhanced.ogg
```

分析サービスを設定する

Amazon Transcribe を使用すると、通話の文字起こしを行えます。その後、文字起こししたテキストを使用して、Amazon Comprehend といった他の機械学習サービスや独自の機械学習モデルを強化できます。

Note

Amazon Transcribe は、自動言語認識機能も備えていますが、この機能は、カスタム言語モデルやコンテンツ編集では使用できません。また、他の機能で言語識別を使用する場合は、その機能が対応している言語しか使用できません。詳細については、「Amazon Transcribe Developer Guide」の「[Language identification with streaming transcriptions](#)」を参照してください。

Amazon Transcribe Call Analytics は、機械学習を活用した API であり、これによって、通話の文字起こし、センチメント分析、会話に関するインサイトのリアルタイム取得などが可能になります。このサービスを利用すると、メモを取る必要がなくなり、見つかった問題にすぐに対処できます。ま

た、発信者の感情、通話の要因、会話のない時間、会話のさえぎり、会話の速度、会話の特徴などに通話後分析も行えます。

 Note

デフォルトの場合、通話後分析は、Amazon S3 バケットにストリーミングされます。録音の重複を防ぐには、通話録音と通話後分析を同時に有効化しないようにします。

さらに、Transcribe Call Analytics では、特定のフレーズに基づいて会話に自動的にタグを付けることで、音声やテキストから機密情報を削除できます。通話分析のメディアプロセッサ、これらのプロセッサで生成するインサイト、出力先については、このセクションで後述する「[通話分析プロセッサと出力先](#)」で詳しく確認できます。

分析サービスを設定するには

1. [分析サービスを設定] ページで、[音声分析] または [文字起こしサービス] の横にあるチェックボックスを選択します。両方の選択も可能です。

[音声分析] チェックボックスを選択すると、[発話者検索] と [ボイストーン分析] を任意に組み合わせることができます。

[文字起こしサービス] チェックボックスを選択して、Amazon Transcribe または Transcribe Call Analytics を有効にします。

a. 発話者検索を有効にするには

- [はい、Amazon Chime SDK 音声分析の同意確認を認めます] チェックボックスを選択し、[承諾] を選択します。

b. ボイストーン分析を有効にするには

- [ボイストーン分析] チェックボックスを選択します。

c. Amazon Transcribe を有効にするには

- i. [Amazon Transcribe] ボタンを選択します。
- ii. [言語設定] で、次のいずれかを実行します。

- A. 発信者が 1 つの言語を話す場合は、[特定の言語] を選択して [言語] リストを開き、対象の言語を指定します。

- B. 発信者が複数の言語を話す場合は、自動的に識別されます。[言語の自動検出] を選択します。
 - C. [自動言語識別の言語オプション] リストを開き、少なくとも 2 つの言語を選択します。
 - D. (オプション) [優先言語] リストを開き、優先言語を指定します。前の手順で選択した言語の信頼スコアが一致すると、優先言語での文字起こしが実行されます。
 - E. (オプション) [コンテンツ削除設定] を展開して 1 つまたは複数のオプションを選択し、表示される追加オプションから 1 つ以上を選択します。各オプションの説明は、ヘルパーテキストで確認できます。
 - F. (オプション) [その他の設定] を展開して 1 つまたは複数のオプションを選択し、表示される追加オプションから 1 つ以上を選択します。各オプションの説明は、ヘルパーテキストで確認できます。
- d. Amazon Transcribe Call Analytics を有効にするには
- i. [Amazon Transcribe Call Analytics] ボタンを選択します。
 - ii. [言語] リストを開き、言語を選択します。
 - iii. (オプション) [コンテンツ削除設定] を展開して 1 つまたは複数のオプションを選択し、表示される追加オプションから 1 つ以上を選択します。各オプションの説明は、ヘルパーテキストで確認できます。
 - iv. (オプション) [その他の設定] を展開して 1 つまたは複数のオプションを選択し、表示される追加オプションから 1 つ以上を選択します。各オプションの説明は、ヘルパーテキストで確認できます。
 - v. (オプション) [通話後分析] 設定を展開し、次の操作を行います。
 - A. [通話後分析] チェックボックスを選択します。
 - B. Amazon S3 バケットの URI を入力します。
 - C. コンテンツ編集タイプを選択します。
2. 選択したら、[次へ] を選択します。

出力の詳細設定を行う

メディア処理の手順を完了したら、分析結果の出力先を選択します。通話分析では、Amazon Kinesis Data Streams を介して、また、オプションにより、選択した Amazon S3 バケット内のデータウェアハウスを介して、インサイトをライブで取得できます。データウェアハウスを作成するには、CloudFormation テンプレートを使用します。このテンプレートにより、通話のメタデータとイ

ンサイトを Amazon S3 バケットに配信するインフラストラクチャを構築できます。データウェアハウス作成の詳細については、このセクションで後述する「[Amazon Chime SDK データレイクを作成する](#)」を参照してください。データウェアハウススキーマの詳細については、同じくこのセクションで後述する「[通話分析のデータモデル](#)」を参照してください。

前のセクションで音声分析を有効にした場合は、AWS Lambda、Amazon Simple Queue Service、Amazon Simple Notification Service などの音声分析通知の送信先を追加することもできます。以下では、その手順を説明します。

出力の詳細を設定するには

1. [Kinesis Data Stream] リストを開き、データストリームを選択します。

Note

データを視覚化するには、Amazon S3 バケットと Amazon Kinesis Data Firehose で使用する Kinesis Data Stream を選択する必要があります。

2. (オプション) [その他の音声分析通知先] を展開し、送信先として [AWS Lambda]、[Amazon SNS]、[Amazon SQS] を任意に組み合わせ、選択します。
3. (オプション) [インサイトの分析と視覚化] で、[データレイクを使用した過去データ分析を実行] チェックボックスを選択します。データレイクの詳細については、このセクションで後述する「[Amazon Chime SDK データレイクを作成する](#)」を参照してください。
4. 完了したら、[次へ] を選択します。

アクセス許可を設定する

通話分析を有効にするには、機械学習サービスなどのリソースに、データメディアにアクセスしてインサイトを提供するための権限が必要です。既存のサービスロールを使用することも、コンソールから新しいロールを作成することもできます。ロールの詳細については、このセクションで後述する「[通話分析のリソースアクセスロールを使用する](#)」を参照してください。

アクセス許可を設定するには

1. [アクセス許可を設定] ページで、次の操作のいずれかを行います。
 1. [新しいサービスロールを作成し使用する] を選択します。
 2. [サービスロール名のサフィックス] ボックスに、どのようなロールかがわかるようなサフィックスを入力します。

または

1. [既存のサービスロールを使用する] を選択します。
 2. [サービスロール] リストを開き、ロールを選択します。
2. [次へ] をクリックします。

(オプション) リアルタイムアラートを設定する

⚠ Important

リアルタイムアラートを使用するには、最初に Amazon Transcribe または Amazon Transcribe Analytics を有効にする必要があります。

Amazon にリアルタイムアラートを送信する一連のルールを作成できます EventBridge。Amazon Transcribe または Amazon Transcribe Call Analytics によって生成されたインサイトが、分析セッション中に指定されたルールと一致すると、アラートが送信されます。アラートの詳細タイプは `Media Insights Rules Matched`。は、Amazon Lambda、Amazon SQS、Amazon SNS などのダウンストリームサービスとの統合 EventBridge をサポートし、エンドユーザーへの通知をトリガーしたり、その他のカスタムビジネスロジックを開始したりできます。詳細については、このセクションで後述する「[Amazon EventBridge 通知の使用](#)」を参照してください。

アラートを設定するには

1. [リアルタイムアラート] で [リアルタイムアラートを有効にする] を選択します。
2. [ルール] で [ルールを作成] を選択します。
3. [ルール名] ボックスにルールの名前を入力します。
4. [ルールタイプ] リストを開き、使用するルールタイプを選択します。
5. 表示されるコントロールを使用して、ルールにキーワードを追加したり、[言及あり] や [言及なし] などのロジックを適用したりできます。
6. 次へ を選択します。

確認と作成

設定を行うには

1. 各セクションの設定内容を確認します。必要に応じて [編集] を選択し、設定を変更します。
2. [Create configuration] (設定を作成) をクリックします。

設定内容は、Amazon Chime SDK コンソールの [設定] ページに表示されます。

API を使用して通話分析の設定を行う

Voice Connector の作成と通話分析の設定をプログラムで行い、それらを関連付けることで、通話分析ワークフローを開始できます。このガイドは、読者の皆様にプログラミングのご経験があることを前提としています。

使用する API は、ワークフローのタイプによって異なります。例えば、音声を録音するには、まず [CreateMediaInsightsPipelineConfiguration](#) API を呼び出して通話分析設定を作成します。次に、[CreateVoiceコネクタ](#) を呼び出して Voice Connector を作成します。最後に、[PutVoiceConnectorStreamingConfiguration](#) API を使用して設定を Voice Connector に関連付けます。

対照的に、Kinesis ビデオストリームプロデューサーで音声を録音するには、[PutVoiceConnectorStreamingConfiguration](#) を呼び出し [CreateMediaInsightsPipelineConfiguration](#)、次に [CreateMediaInsightsPipeline](#) API を呼び出します。

通話分析の設定を使用してさまざまなワークフローを有効にする方法については、このセクションで後述する「[通話分析の設定を使用する](#)」のワークフローで詳しく確認できます。

設定と Voice Connector を関連付ける

コンソールから通話分析の設定を行ったら、その設定に Voice Connector を関連付けます。これにより、設定で指定した通話分析サービスが Voice Connector によって自動的に起動します。通話分析の起動は、通話ごとに行われます。

Voice Connector を関連付けるには

1. <https://console.aws.amazon.com/chime-sdk/home> から Amazon Chime コンソールを開きます。
2. ナビゲーションペインの [SIP トランキング] で、[Voice Connector] を選択します。
3. 設定に関連付ける Voice Connector の名前を選択し、[ストリーミング] タブを選択します。

4. [開始] を選択していない場合は、それを選択して、Kinesis Video Streams へのストリーミングを開始します。
5. [通話分析] で [有効化] を選択し、表示されるメニューで、対象の通話分析設定 ARN を選択します。
6. [保存] を選択します。

Note

Voice Connector に関連付けた設定を有効化、無効化、または変更した後、新しい設定がサービスに反映され有効になるまで 5 分かかります。

通話分析設定の詳細については、「Amazon Chime SDK Administrator Guide」の「[Managing call analytics](#)」を参照してください。

通話分析の設定を使用してさまざまなワークフローを有効にする方法については、このセクションで後述する「[通話分析の設定を使用する](#)」で詳しく確認できます。

通話分析の設定を使用する

通話分析設定を使用して音声进行处理するには、メディアインサイトパイプラインとも呼ばれる通話分析パイプラインを作成する必要があります。パイプラインを通話中に作成して音声进行处理させ、通話の終了時に停止します。通話パイプラインには、通話分析設定にある ARN と、音声ソースに関する情報が必要です。通話分析の設定では、音声プロセッサ、インサイトの送信先、アラートイベントなどを指定しますが、音声ソースは指定しません。これにより、さまざまな Voice Connector や KVS ソースといった、各種通話分析ワークフロー間で設定を再利用できます。通話分析パイプラインによって、設定で指定した機械学習サービスを起動し、音声を録音します。パイプラインは、通話の終了時に手動または自動で停止できます。

通話分析パイプラインは、さまざまなユースケースで使用できます。次のワークフローは、想定される通話分析設定およびパイプラインの使用方法を示しています。

トピック

- [通話録音を目的としたワークフロー](#)
- [機械学習ベースの分析を目的としたワークフロー](#)

通話録音を目的としたワークフロー

本セクションのこのトピックでは、通話と Kinesis Video Streams の録音および録画に関する各種ワークフローについて説明します。

Voice Connector の通話を録音する

このワークフローの用途:

- Voice Connector を既に使用し SIP メディアを通話分析に取り込んでいる、または使用する予定がある。

Note

Voice Connector は SIP と SIPREC に対応しています。詳細については、「Amazon Chime SDK Administrator Guide」の「[Managing Amazon Chime SDK Voice Connectors](#)」を参照してください。

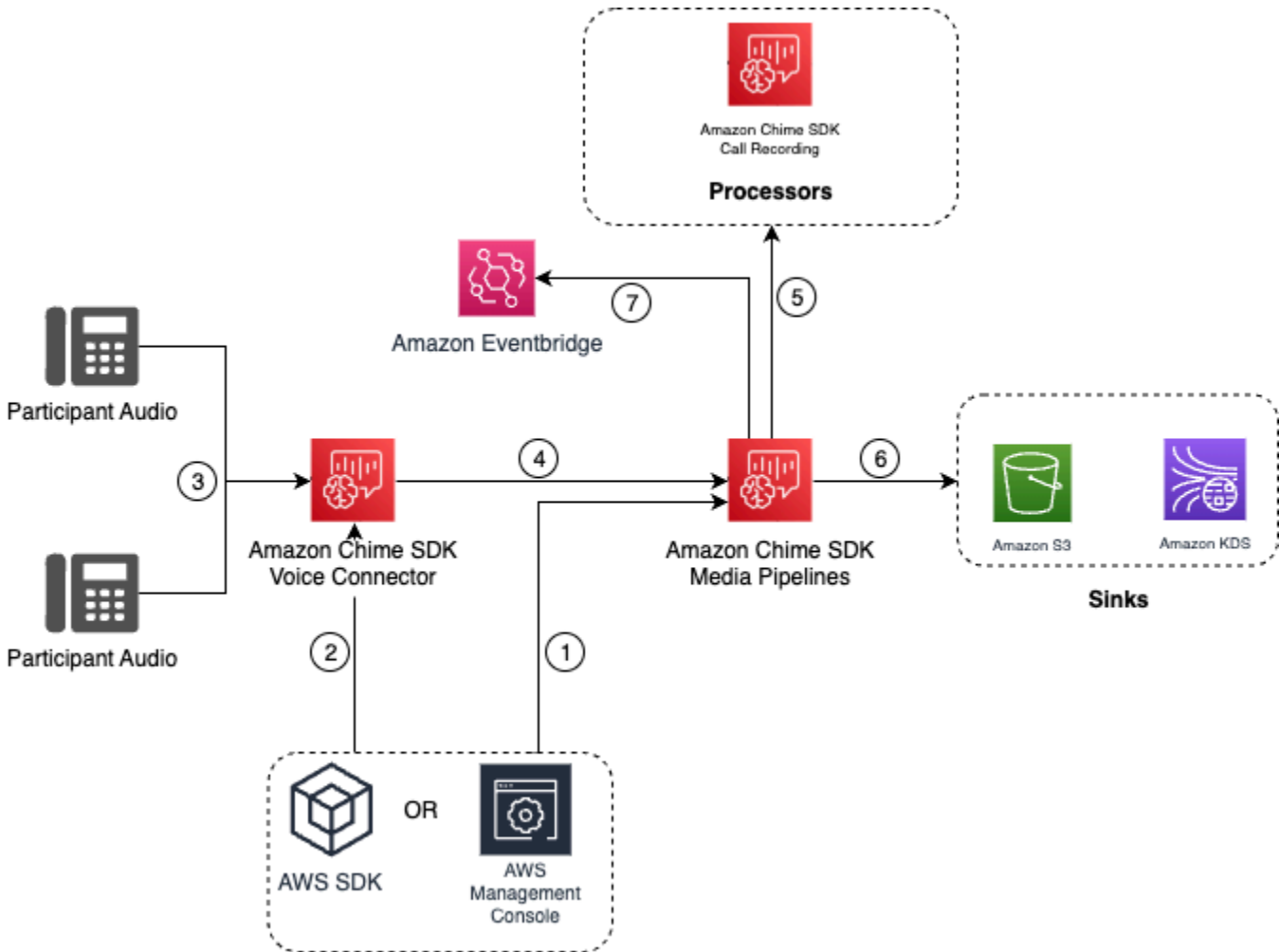
- 選択した Amazon Simple Storage Service の送信先に、SIP または SIPREC の通話を低レイテンシーかつ自動的に録音する。
- Amazon Chime SDK コンソールから設定を行い、それを Voice Connector に関連付ける。
- すべての Voice Connector の通話に同じ録音設定を適用する。1 つまたは複数の Voice Connector に複数の設定を適用する場合は、次のセクションを参照してください。

プログラムによる呼び出しを有効にするには、次の Amazon Chime SDK API を使用してください。

- [CreateMediaInsightsPipelineConfiguration](#) API を使用して通話分析設定を作成する
- を使用して Voice Connector [CreateVoiceConnector](#) を作成します。
- [PutVoiceConnectorStreamingConfiguration](#) API を使用して、設定を Voice Connector に関連付けます。

詳細については、「Amazon Chime SDK Administrator Guide」の「[Configuring Voice Connectors to use call analytics](#)」を参照してください。

次の図は、Voice Connector によって通話録音セッションを開始するときのデータフローを示しています。図の数字は、次の各説明の番号に対応しています。



図中の手順を説明します。

1. Amazon Chime SDK コンソールまたは [CreateMediaInsightsPipelineConfiguration](#) API を使用して、通話分析設定を作成します。設定のプロセスでは、通話録音の有効化、対象の録音ファイル形式の選択、録音ファイルの送信先となる Amazon S3 の指定のみ行います。詳細については、「Amazon Chime SDK Administrator Guide」の「[Creating call analytics configurations](#)」を参照してください。
2. Amazon Chime SDK コンソールまたは [PutVoiceConnectorStreamingConfiguration](#) API を使用して、設定を Voice Connector に関連付けます。コンソールを使用するには、「[Configuring Voice Connectors to use call analytics](#)」を参照してください。
3. 発信中には、Voice Connector によって、各通話参加者の音声を受信します。
4. 通話分析の録音設定を Voice Connector にアタッチしている場合は、Voice Connector サービスによって、メディアパイプラインサービスを使用した通話分析録音セッションが開始されます。

5. メディアパイプラインサービスによって、進行中の通話をモニタリングする通話録音プロセッサが起動します。
6. 通話が終了すると、メディアパイプラインサービスによって通話録音ファイルが指定の Amazon S3 バケットに配信され、Amazon Kinesis Data Streams を介して録音メタデータが提供されます。データウェアハウスを有効にしている場合、通話メタデータも Amazon Simple Storage Service データウェアハウスに送信されます。SIPREC を使用して通話分析に SIP 音声を取り入れる場合は、通話メタデータにテーブル形式の SIPREC メタデータが含まれることになります。録音テーブルの詳細については、このセクションで後述する「[Glue データカタログのテーブル](#)」を参照してください。
7. メディアパイプラインサービスは、パイプラインステータスイベントをデフォルトの Amazon に送信します EventBridge。詳細については、このガイドの [EventBridge 「通知の使用」](#) を参照してください。

Note

Voice Connector による録音を有効にするには、Voice Connector のストリーミングを有効にする必要があります。この機能によって、アカウント内の Voice Connector で管理する Kinesis Video Streams に通話データをストリーミングできるようになるからです。詳細については、「Amazon Chime SDK Administrator Guide」の「[Streaming Amazon Chime SDK Voice Connector media to Kinesis Video Streams](#)」を参照してください。

また、Voice Connector で作成した通話データは、数時間、数日、数年まで、さまざまな期間、Kinesis Video Streams に保存できます。データ保持を行わない場合、通話データをすぐに使用しなければならなくなります。Kinesis Video Streams のコストは、使用する帯域幅と合計ストレージに基づいて判断します。データ保持期間は、Voice Connector のストリーミング設定内でいつでも調整できます。通話分析の記録を有効にするには、通話分析の実行に十分な期間のデータを Kinesis Video Streams 内に保持する必要があります。これを行うには、適切なデータ保持期間を指定します。

通話インサイトパイプライン設定は、必要な数の Voice Connector に関連付けることができます。Voice Connector ごとに異なる設定を行うことも可能です。Voice Connector は、を使用して `AWSServiceRoleForAmazonChimeVoiceConnector`、トランザクション ID ごとに 1 回、ユーザーに代わって [CreateMediaInsightsPipeline](#) API を呼び出します。詳細については、「Amazon Chime SDK Administrator Guide」の「[Using the Amazon Chime SDK service-linked role for Amazon Chime SDK Voice Connectors](#)」を参照してください。

Amazon Kinesis Video Streams プロデューサーを使用して録画する

Amazon Kinesis Video Streams の録画が必要なケースを次に示します。

- Voice Connector のすべての通話に同じ設定を使用せず、各通話に異なる設定を適用する。
- Voice Connector で処理しない SIP 音声または SIP 以外の音声を録音する。

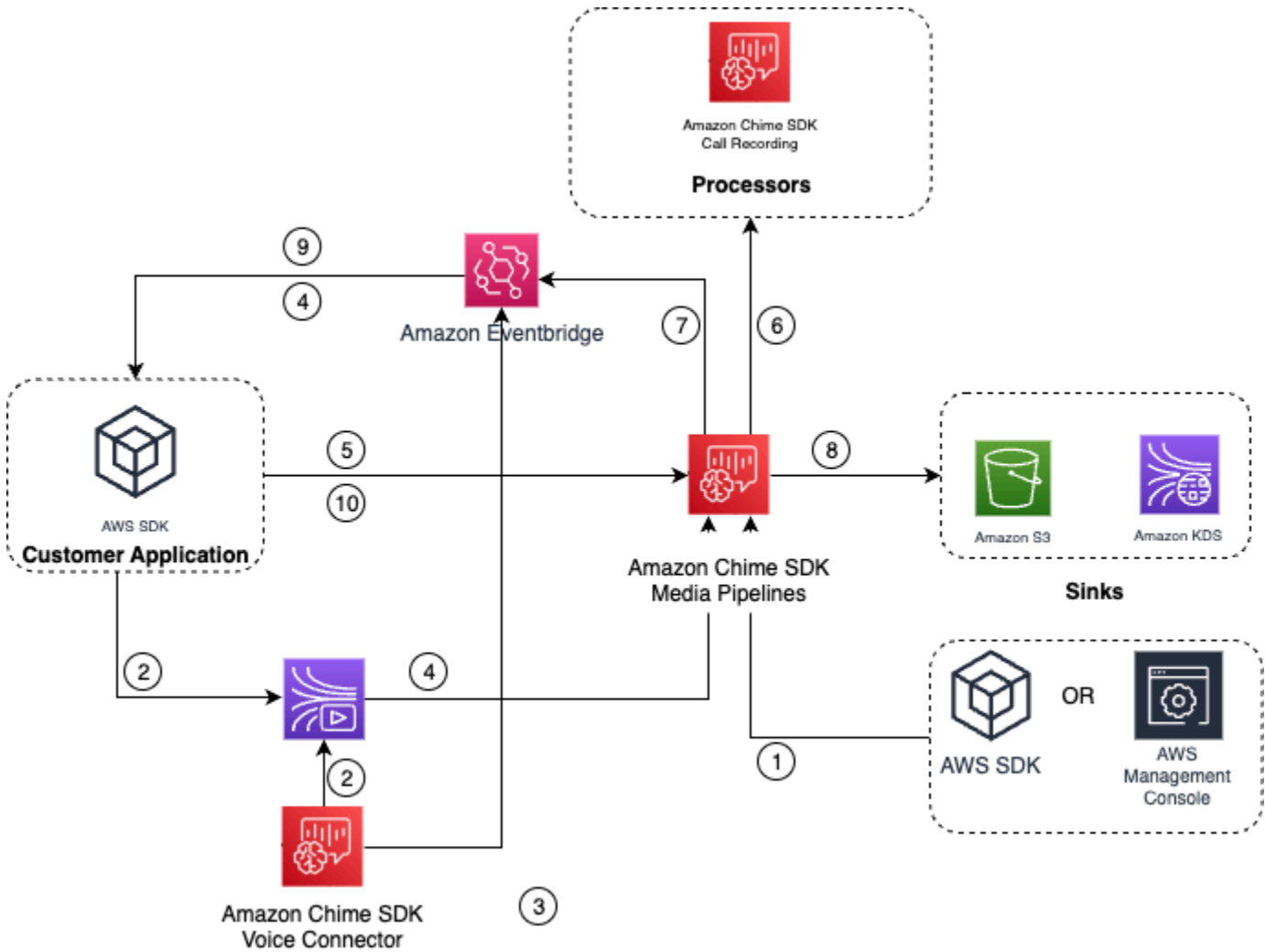
この通話録音オプションを使用するには、音声を Kinesis Video Streams (KVS) に公開し、KVS ストリームチャンネル情報と通話分析設定 ARN を使用して [CreateMediaInsightsPipeline](#) API を呼び出す必要があります。

Note

通話分析 API は最大 2 つの音声チャンネルに対応しています。Voice Connector ストリーミングを有効にし、Voice Connector EventBridge の通知で発行された KVS 情報を使用して通話録音を開始することもできます。

[CreateMediaInsightsPipeline](#) API を呼び出すときに、KVS ストリームチャンネル定義ごとにフラグメント番号を指定するかどうかを選択できます。フラグメント番号を指定すると、そのフラグメントからストリームの処理が開始されます。フラグメント ID を指定しない場合、利用可能な最新のフラグメントからストリームの処理が開始されます。

次の図は、Voice Connector によって通話録音セッションを開始するときのデータフローを示しています。図の数字は、次の各説明の番号に対応しています。



図中の手順を説明します。

1. Amazon Chime SDK コンソールまたは [CreateMediaInsightsPipelineConfiguration](#) API を使用して、通話録音設定を作成できます。
2. AWS SDK を使用して、外部オーディオを KVS にプッシュするアプリケーションを作成するか、Voice Connector ストリーミングを有効にして通話オーディオを自動的に KVS に発行します。詳細については、「Amazon Chime SDK Administrator Guide」の「[Streaming Amazon Chime SDK Voice Connector media to Kinesis Video Streams](#)」を参照してください。
3. Voice Connector ストリーミングが有効になっている場合、Voice Connector サービスはデフォルトの **EventBridge** に通知を送信します。
4. Voice Connector ストリーミングの場合、アプリケーションは **Amazon Chime Voice Connector** ストリーミング **STARTED** イベントを使用して **EventBridge**、通話のログに関する **KVS** ストリーム情報を収集できます。

5. アプリケーションが Voice Connector ストリーミングイベントまたは外部ソースからの音声情報を取得すると、アプリケーションは Amazon Chime SDK [CreateMediaInsightsPipeline](#) API を呼び出します。
6. メディアパイプラインサービスによって、進行中の通話をモニタリングする通話録音プロセッサが起動します。
7. メディアパイプラインサービスは、パイプラインステータスイベントをデフォルトの Amazon に送信します EventBridge。詳細については、「[通知の使用 EventBridge](#)」を参照してください。
8. 通話が終了すると、メディアパイプラインサービスによって通話録音ファイルが指定の Amazon S3 バケットに配信され、Amazon Kinesis Data Streams を介して録音メタデータが提供されます。データウェアハウスが有効にしている場合、通話メタデータも Amazon S3 データウェアハウスに送信されます。SIPREC を使用して通話分析に SIP 音声を取り入れる場合は、便利なテーブル形式の SIPREC メタデータが通話メタデータに含まれることになります。録音テーブルの詳細については、このセクションで後述する「[Glue データカタログのテーブル](#)」を参照してください。
9. アプリケーションはパイプラインをモニタリングでき、Voice Connector の場合は、Amazon に公開されたイベントを使用して通話ステータスをモニタリングできます EventBridge。詳細については、このガイドの [EventBridge](#) 「[通知の使用](#)」を参照してください。
- 10 録音を終了するには、[DeleteMediaPipeline](#) API を呼び出して通話録音を終了します。

API ベースの録音と例については、このガイドの「[Amazon S3 recording sink](#)」を参照してください。

CLI を使用して録音を開始する

このセクションの例では、次の操作について説明します。

- CLI を使用して通話分析設定を実行し、[CreateMediaInsightsPipeline](#) を呼び出します。
- CLI を使用して、録音先、音声ファイル形式、音声ファイル名を指定します。

トピック

- [設定を実行してパイプラインを開始する](#)
- [録音先、ファイル名、ファイル形式を設定する](#)

設定を実行してパイプラインを開始する

次のコマンドを使用して設定を実行し、メディアインサイトパイプラインを開始します。pipeline.json ファイルには設定情報が記述されています。

```
aws chime-sdk-media-pipeline create-media-insights-pipeline --cli-input-json file://  
pipeline.json
```

一般的な pipeline.json ファイルの例を次に示します。

```
{  
  "MediaInsightsPipelineConfigurationArn": arn:aws:chime:region;account_id:media-  
insights-pipeline-configuration/MyConfiguration,  
  "KinesisVideoStreamRecordingSourceRuntimeConfiguration": {  
    "Streams": [  
      {  
        "StreamArn": kinesis_video_stream_arn_1  
      },  
      {  
        "StreamArn": kinesis_video_stream_arn_2  
      }  
    ],  
    "FragmentSelector": {  
      "FragmentSelectorType": "selector_type", // Specify "server_timestamp" or  
"producer_timestamp" as the fragment selector type  
      "TimestampRange": {  
        "StartTimestamp": epoch_time_seconds,  
        "EndTimestamp": epoch_time_seconds  
      }  
    }  
  },  
  "S3RecordingSinkRuntimeConfiguration": {  
    "Destination": arn:aws:s3:::bucket_name/prefix/optional_file_name,  
    "RecordingFileFormat": file_format // Specify "Opus" or "WAV" as the recording  
file format, if you want to override the configuration  
  }  
}
```

MediaInsightsPipelineConfigurationArn は、通話分析を設定した後に受け取る設定 ARN を表しています。

録音先、ファイル名、ファイル形式を設定する

次の例では、MyRecordingBucket という名前のフォルダを S3SinkConfiguration.Destination 値として使用し、Opus を RecordingFileFormat 値として使用しています。

```
arn:aws:s3:::MyRecordingBucket/voice-connector-id/transaction-id_year-month-date-hour-minute-second-millisecond.ogg
```

次の例では、MyRecordingBucket を S3SinkConfiguration.Destination 値として使用し、Wav を RecordingFileFormat 値として使用しています。

```
arn:aws:s3:::MyRecordingBucket/voice-connector-id/transaction-id_year-month-date-hour-minute-second-millisecond.wav
```

機械学習ベースの分析を目的としたワークフロー

このセクションでは、Amazon Chime SDK 通話分析が備える機械学習分析機能の使用方法について説明します。

Note

同じ Kinesis Video Streams で複数の機械学習分析を実行する際には、ビデオストリームの GetMedia および GetMediaForFragmentList の接続レベル制限を引き上げなければならない場合があります。詳細については、「[Kinesis Video Streams Developer Guide](#)」の「[Kinesis Video Streams limits](#)」を参照してください。

Voice Connector を使用して通話分析を自動的に開始する

このワークフローの用途:

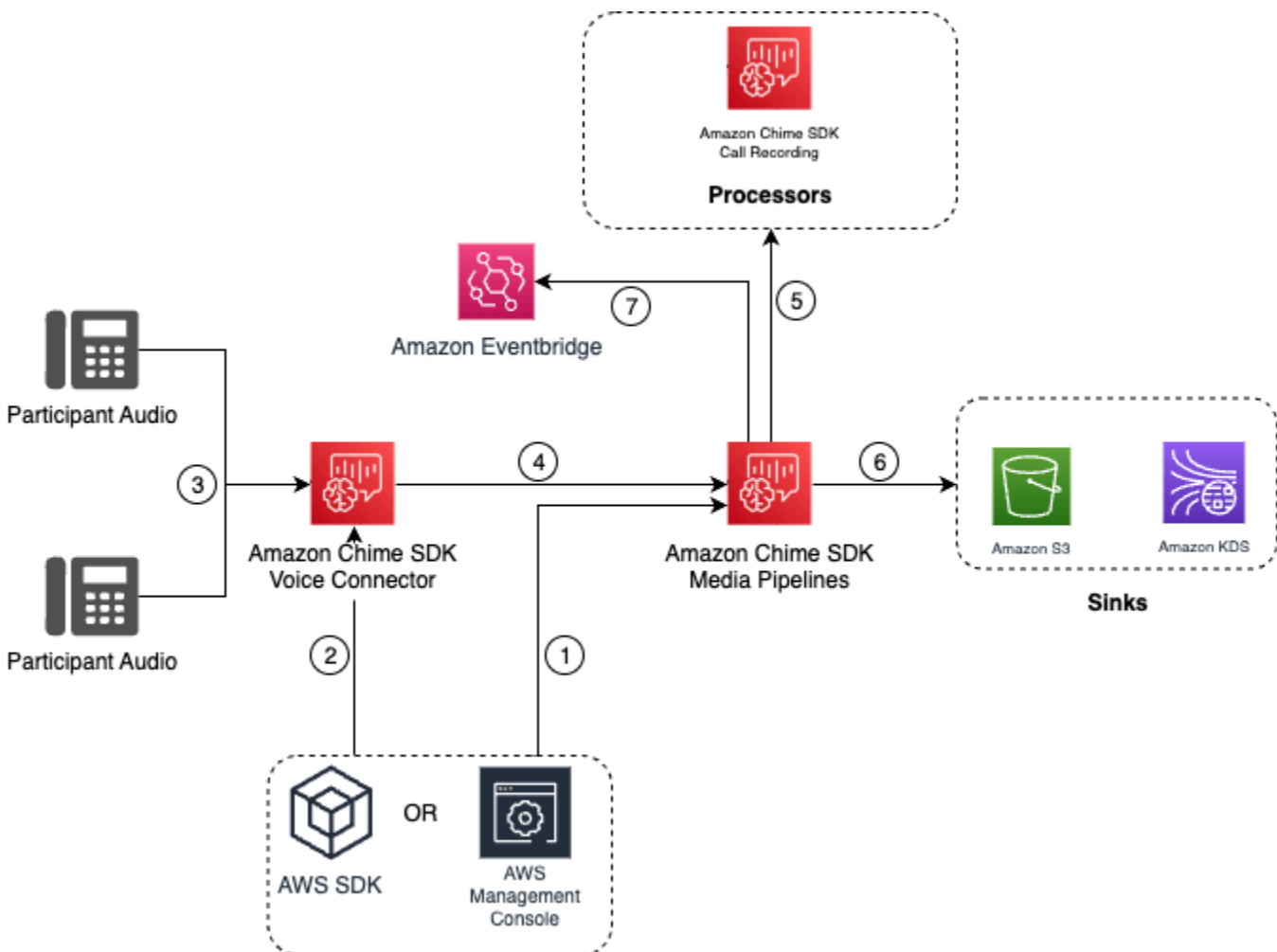
- コンソールを主に使用してセットアップを行う。
- Voice Connector を既に使用し SIP メディアを通話分析に取り込んでいる、または使用する予定がある。Voice Connector は SIP と SIPREC の両方に対応しています。Voice Connector の設定については、「[Managing Amazon Chime SDK Voice Connector](#)」で詳しく確認できます。
- すべての Voice Connector の通話に同じメディアインサイト設定を適用する。

- Amazon Chime SDK 音声分析を使用する必要があるため、Voice Connector またはメディアインサイトパイプラインが、その要件に設定されている。

Amazon Chime SDK コンソールでこのワークフローを有効にするには、「[Configuring Voice Connectors to use call analytics](#)」に記載の録音設定手順に従ってください。

このワークフローをプログラムで有効にするには、次の APIs を使用してコール分析設定を作成し、[PutVoiceConnectorStreamingConfigurationAPI](#) を使用してその設定を Voice Connector に関連付けます。[CreateMediaInsightsPipelineConfiguration](#) 詳細については、「Amazon Chime SDK Administrator Guide」の「[Configuring Voice Connectors to use voice analytics](#)」を参照してください。

次の図は、Voice Connector によって通話分析セッションを開始するときのデータフローを示しています。図の数字は、次の各説明の番号に対応しています。



図中の手順を説明します。

1. Amazon Chime SDK コンソールまたは [CreateMediaInsightsPipelineConfiguration](#) API を使用して、メディアインサイトパイプライン設定を作成します。
2. Amazon Chime SDK コンソールまたは [PutVoiceConnectorStreamingConfiguration](#) API を使用して、設定を Voice Connector に関連付けます。詳細については、「Amazon Chime SDK Administrator Guide」の「[Configuring Voice Connectors to use call analytics](#)」を参照してください。
3. 発信中には、Voice Connector によって、各通話参加者の音声を受信します。
4. Voice Connector サービスは通話分析と既に統合されているため、通話分析設定をこのサービスにアタッチすると、メディアパイプラインサービスを使用した通話分析セッションが開始されます。
5. メディアパイプラインサービスによって、1 つ以上のメディアプロセッサが設定の指定どおりに呼び出されます。
6. メディアパイプラインサービスによって、出力データが 1 つ以上の宛先に設定どおりに送信されます。例えば、Amazon Kinesis Data Streams を介してリアルタイム分析を送信できます。設定によっては、通話のメタデータおよび分析を Amazon S3 データウェアハウスに送信することも可能です。
7. メディアパイプラインサービスは、パイプラインステータスイベントをデフォルトの Amazon に送信します EventBridge。ルールを設定している場合、それらの通知 EventBridge も Amazon に送信されます。詳細については、「[通知の使用 EventBridge](#)」を参照してください。

Note

- 音声分析プロセッサは、[StartSpeakerSearchTask](#)または [StartVoiceToneAnalysisTask](#) APIs。
- Voice Connector で通話分析を使用するには、Voice Connector ストリーミングを有効にする必要があります。この機能によって、アカウント内の Voice Connector で管理する Kinesis Video Streams に通話データをストリーミングできるようになるからです。詳細については、「Amazon Chime SDK Administrator Guide」の「[Streaming Amazon Chime SDK Voice Connector media to Kinesis Video Streams](#)」を参照してください。

また、Voice Connector の通話データは、数時間から数年まで、さまざまな期間、Kinesis Video Streams に保存できます。データ保持を行わない場合、通話データをすぐに使用しなければならなくなります。Kinesis Video Streams のコストは、使用する帯域幅と合計ストレージに基づいて判断し

ます。データ保持期間は、Voice Connector のストリーミング設定を編集することで、いつでも調整できます。通話分析の記録を有効にするには、通話分析の終了時まで Kinesis Video Streams にデータが保持されるようにする必要があります。これを行うには、適切なデータ保持期間を指定します。

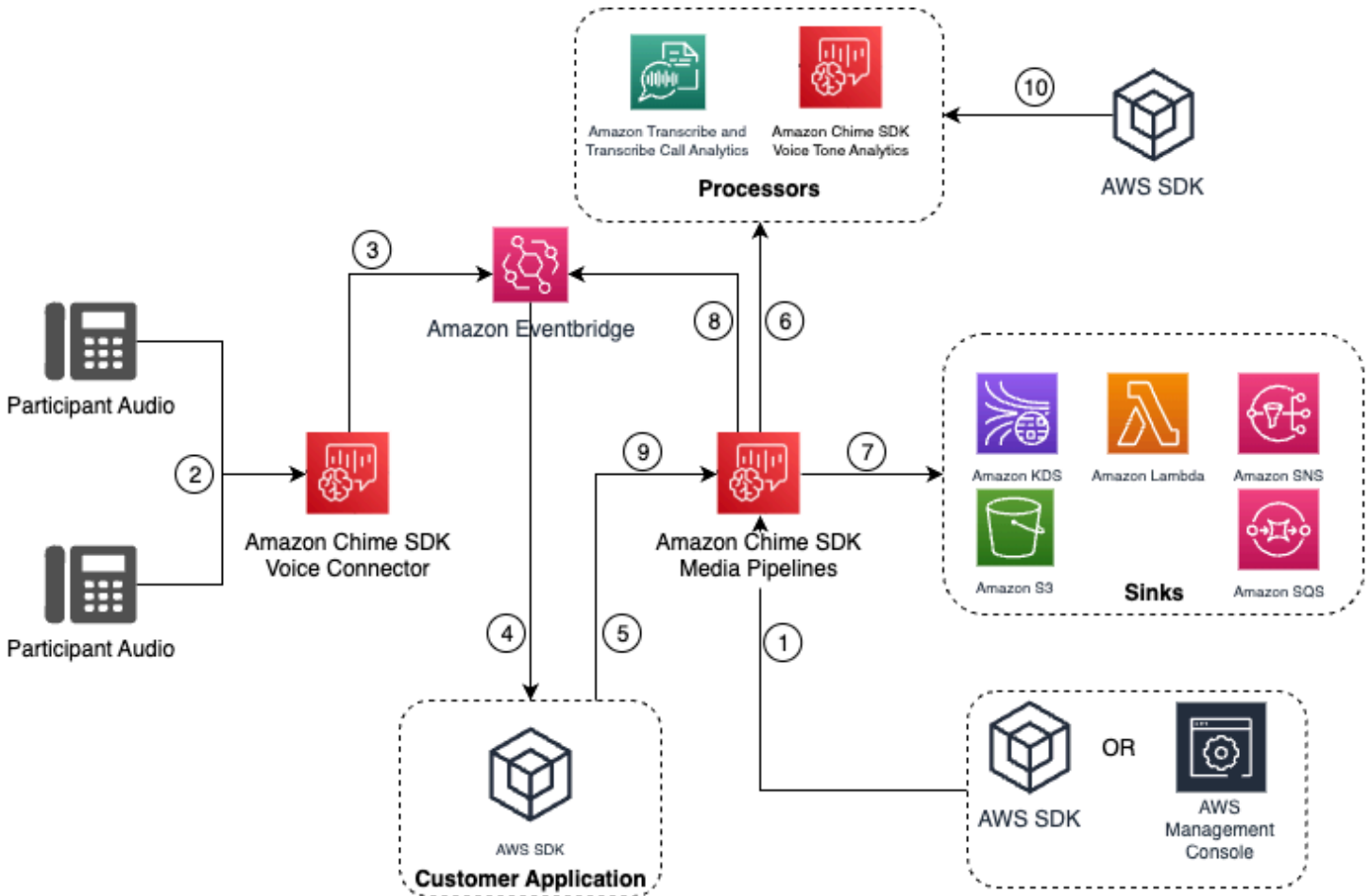
メディアインサイトパイプライン設定は、必要な数の Voice Connector に関連付けることができます。Voice Connector ごとに異なる設定を行うことも可能です。Voice Connector は、を使用して `AWSServiceRoleForAmazonChimeVoiceConnector`、トランザクション ID ごとに 1 回、ユーザーに代わって [CreateMediaInsightsPipeline](#) API を呼び出します。詳細については、「Amazon Chime SDK Administrator Guide」の「[Using the Amazon Chime SDK service-linked role for Amazon Chime SDK Voice Connectors](#)」を参照してください。

Voice Connector で通話分析 API を使用する

Voice Connector を導入済みで、通話分析設定を適用するタイミングや、適用先の通話を制御する必要がある場合は、このワークフローを使用すると良いでしょう。

この方法を使用するには、Voice Connector が発行するイベントの EventBridge ターゲットを作成し、そのイベントを使用してコール分析パイプライン APIs をトリガーする必要があります。詳細については、「[Amazon Chime SDK 管理者ガイド](#)」の「[による EventBridge Amazon Chime SDK の自動化](#)」を参照してください。

次の図は、Voice Connector で通話分析を使用する際に、より詳細に制御する方法を示しています。図の数字は、次の各説明の番号に対応しています。



図中の手順を説明します。

1. Amazon Chime SDK コンソールまたは [CreateMediaInsightsPipelineConfiguration](#) API を使用して、メディアインサイトパイプライン設定を作成します。
2. 発信中には、Voice Connector によって、参加者の音声を受信します。
3. Voice Connector は通話音声を Kinesis Video Stream に送信し、対応するイベントを EventBridge に送信します。これらのイベントには、ストリームおよび通話のメタデータが含まれています。
4. アプリケーションは EventBridge ターゲット EventBridge 経由でサブスクライブされます。
5. アプリケーションは Amazon Chime SDK [CreateMediaInsightsPipeline](#) API を呼び出します。
6. メディアパイプラインサービスによって、1 つ以上のメディアプロセッサが呼び出されます。呼び出しは、メディアインサイトパイプライン設定内のプロセッサ要素に基づいて行われます。
7. メディアパイプラインサービスによって、出力データが 1 つ以上の宛先に設定どおりに送信されます。Amazon Chime SDK 通話分析によって、Amazon Kinesis Data Streams 経由でのリアルタイム分析が可能になり、設定済みの場合、通話のメタデータ分析を Amazon S3 データウェアハウスに送信できます。

- メディアパイプラインサービスは、イベントを Amazon に送信します EventBridge。ルールを設定している場合、それらの通知 EventBridge も Amazon に送信されます。
- [UpdateMediaInsightsPipelineStatus](#) API を呼び出すことで、通話分析セッションを一時停止または再開できます。

Note

通話録音は、通話の一時停止と再開に対応していません。また、通話中に開始した音声分析タスクも、セッションを一時停止すると停止します。再起動するには、[StartSpeakerSearchTask](#)または [StartVoiceToneAnalysisTask](#) APIsを呼び出す必要があります。

- 設定中に音声トーン分析を選択した場合は、[StartSpeakerSearchTask](#)または [StartVoiceToneAnalysisTask](#) APIsを呼び出して音声分析を開始します。

Kinesis Video Streams プロデューサーで通話分析を使用する

このオプションを使用するには、音声データを Kinesis Video Streams (KVS) に公開し、KVS ストリームチャンネル情報を使用して [CreateMediaInsightsPipeline](#) API を呼び出す必要があります。

Note

通話分析 API は最大 2 つの音声チャンネルに対応しています。

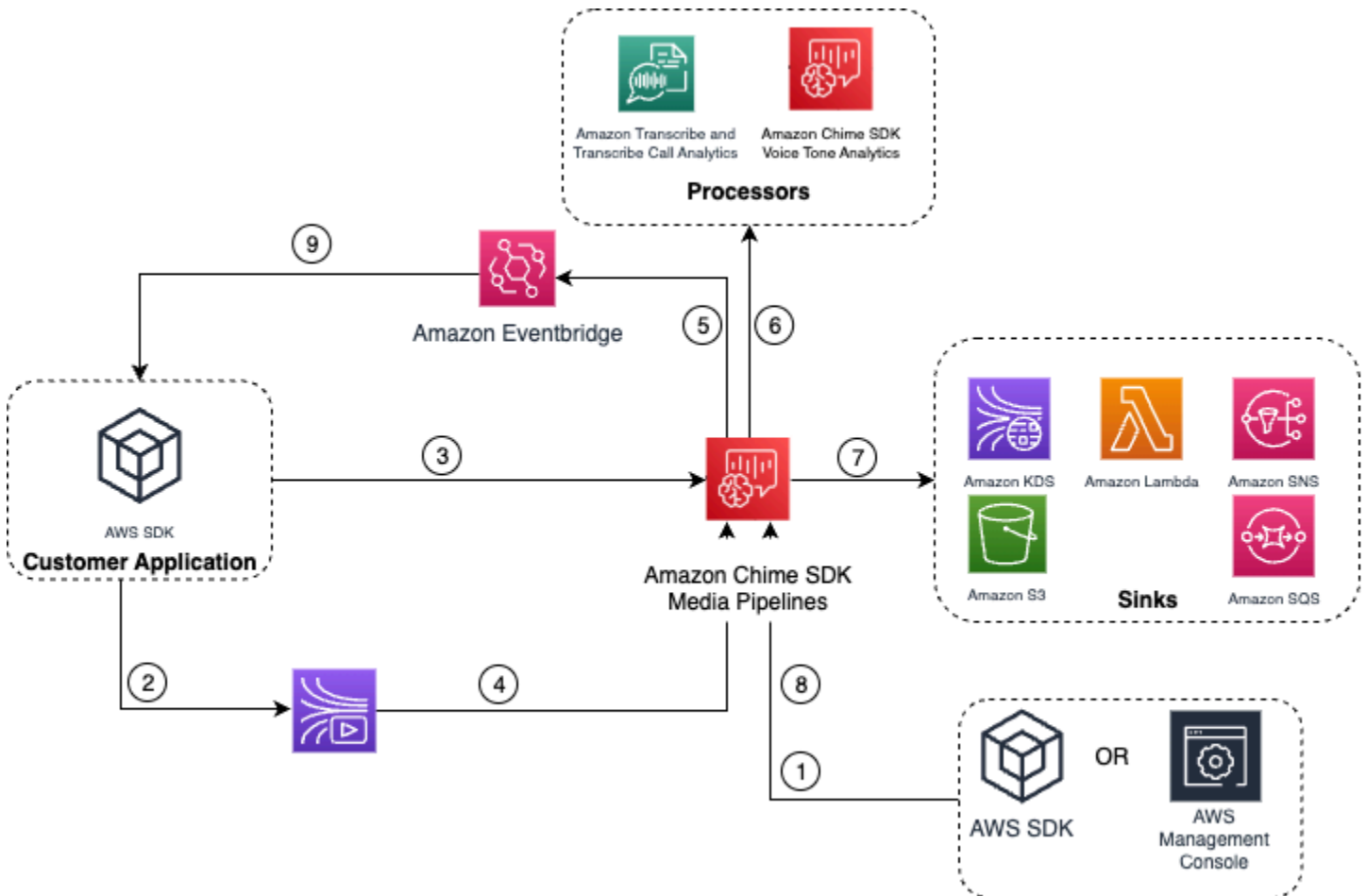
[CreateMediaInsightsPipeline](#) API を呼び出すときに、KVS ストリームチャンネル定義ごとにフラグメント番号を指定できます。フラグメント番号を指定すると、そのフラグメントからストリームの処理が開始されます。フラグメント ID を指定しない場合、利用可能な最新のフラグメントからストリームの処理が開始されます。

通話分析は、音声サンプルレートが 8 kHz~48 kHz の PCM 音声 (符号付き 16 ビットリトルエンディアン音声形式のみ。WAV には未対応) に対応しています。電話音声などの低品質音声では一般的に約 8000 Hz のレートが使用され、高品質の音声では通常 16,000 Hz~48,000 Hz の範囲のレートが使用されます。指定するサンプルレートは、音声のサンプルレートと一致させる必要があります。詳細については、「Amazon Chime SDK API リファレンス[KinesisVideoStreamSourceRuntimeConfiguration](#)」の「」を参照してください。

Kinesis Video Streams プロデューサー SDK には、音声データを Kinesis Video Streams にストリーミングするために使用できるライブラリセットが用意されています。詳細については、「Amazon

Kinesis Video Streams Developer Guide」の「[Kinesis Video Streams Producer Libraries](#)」を参照してください。

次の図は、カスタム Kinesis Video Streams プロデューサーで通話分析を使用する場合のデータフローを示しています。図の数字は、次の各説明の番号に対応しています。



1. AWS コンソールまたは [CreateMediaInsightsPipelineConfiguration](#) API を使用して、メディアインサイトパイプライン設定を作成します。
2. Kinesis Video Streams プロデューサーを使用して Kinesis Video Streams に音声を書き込みます。
3. アプリケーションは [CreateMediaInsightsPipeline](#) API を呼び出します。
4. メディアパイプラインサービスによって、顧客の Kinesis Video Streams から音声を読み取られます。
5. メディアパイプラインサービスは、イベントを Amazon に送信します EventBridge。ルールを設定している場合、それらの通知 EventBridge も Amazon に送信されます。
6. メディアパイプラインサービスによって、1 つ以上のプロセッサ要素が呼び出されます。
7. メディアパイプラインサービスによって、出力データが 1 つ以上のシンク要素に送信されます。

8. [UpdateMediaInsightsPipelineStatus](#) API を呼び出すことで、通話分析セッションを一時停止または再開できます。

Note

通話録音は、一時停止と再開に対応していません。

9. アプリケーションは Amazon EventBridge イベントを処理して、カスタムビジネスワークフローをトリガーできます。
- 10 設定の作成時に音声分析を選択した場合、アプリケーションは [StartSpeakerSearchTask](#) または [StartVoiceToneAnalysisTask](#) APIs を呼び出して音声分析を開始できます。

通話分析パイプラインを管理する

、および [DeleteMediaPipeline](#) APIs を呼び出すことで、メディアインサイトパイプラインの読み取り、一覧表示 [GetMediaPipelineListMediaPipelines](#)、削除を行うことができます。

次のいずれかの条件が満たされると、メディアインサイトパイプラインは停止します。

- いずれの Kinesis Video Streams から 15 秒間、InProgress パイプラインに新しいフラグメントが送信されていない。
- [DeleteMediaPipeline](#) API が呼び出されます。
- メディアインサイトパイプラインが 8 時間以上前に作成された。パイプラインは自動的に停止します。
- メディアインサイトパイプラインが 2 時間以上、一時停止している。パイプラインは自動的に停止します。

通話分析パイプラインの一時停止と再開を行う

メディアインサイトパイプラインを一時停止して再開するには、Pause または Resume アクションを使用して [UpdateMediaInsightsPipelineStatus](#) API を呼び出します。これを行うには、パイプラインの ID または ARN のいずれかを Identifier フィールドに渡します。

Warning

警告: [UpdateMediaInsightsPipelineStatus](#) API に Pause ステータスを指定すると、メディアインサイトパイプラインで開始したすべての音声分析タスクが停止します。Resume ステータスを指定しても、タスクは再開されないため、それらを再度開始する

必要があります。タスクの再開前には、必要なすべての通知を行い、スピーカーから必要なすべての同意を得る必要があります。詳細については、「Amazon Chime SDK API リファレンス [StartVoiceToneAnalysisTask](#)」の [StartSpeakerSearchTask](#) 「」または「」を参照してください。

一時停止中、パイプラインでは、プロセッサへのメディアの送信と、Kinesis Data Streams およびデータウェアハウスへのデータの書き込みが停止します。パイプラインを Resume (再開) すると、ストリームで利用可能な最新のチャンクが送信されます。メディアインサイトパイプラインは、2 時間以上一時停止すると、自動的に停止します。通話録音は一時停止と再開に対応していない点に留意してください。

詳細については、次のトピックを参照してください。

- [EventBridge 通知の使用](#)
- [StartSelectorType.NOW](#) Amazon Kinesis Video Streams 「」。
- 「[Amazon Transcribe call analytics processor](#)」

Note

パイプラインを一時停止している間の通話分析の使用量には課金が発生します。ただし、Amazon Transcribe や Amazon Kinesis などのリソースアクセスロールを介してアクセスされた AWS サービスに対しては請求されません。

識別子フィールドに設定名または ARN を渡すことで [GetMediaInsightsPipelineConfiguration](#)、[DeleteMediaInsightsPipelineConfiguration](#) APIs を使用して、既存のコール分析設定の読み取り [UpdateMediaInsightsPipelineConfiguration](#)、更新、削除を行うことができます。

[ListMediaInsightsPipelineConfigurations](#) API を呼び出して設定を一覧表示できます。

通話分析のリソースアクセスロールを使用する

通話アカウントでは、メディアインサイトパイプライン設定で使用するリソースアクセスロールを作成する必要があります。クロスアカウントロールは、使用できません。

通話分析の設定で有効にする機能によっては、追加のリソースポリシーを使用する必要があります。詳細については、次のセクションを展開してください。

最低限必要なポリシー

このロールには少なくとも次のポリシーが必要です。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "transcribe:StartCallAnalyticsStreamTranscription",
      "transcribe:StartStreamTranscription"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesisvideo:GetDataEndpoint",
      "kinesisvideo:GetMedia"
    ],
    "Resource": "arn:aws:kinesisvideo:us-east-1:111122223333:stream/Chime*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesisvideo:GetDataEndpoint",
      "kinesisvideo:GetMedia"
    ],
    "Resource": "arn:aws:kinesisvideo:us-east-1:111122223333:stream/*",
    "Condition": {
      "StringLike": {
        "aws:ResourceTag/AWSServiceName": "ChimeSDK"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": ["kms:Decrypt"],
    "Resource": "arn:aws:kms:us-east-1:111122223333:key/*",
    "Condition": {
      "StringLike": {
        "aws:ResourceTag/AWSServiceName": "ChimeSDK"
      }
    }
  }
}
```

```
    }  
  ]  
}
```

次の信頼ポリシーも使用する必要があります。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "mediapipelines.chime.amazonaws.com"  
      },  
      "Action": "sts:AssumeRole",  
      "Condition": {  
        "StringEquals": {  
          "aws:SourceAccount": "111122223333"  
        },  
        "ArnLike": {  
          "aws:SourceARN": "arn:aws:chime:*:111122223333:*"  
        }  
      }  
    }  
  ]  
}
```

KinesisDataStreamSink ポリシー

KinesisDataStreamSink を使用する場合は、次のポリシーを追加します。

```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Effect": "Allow",  
    "Action": [  
      "kinesis:PutRecord"  
    ],  
    "Resource": [  
      "arn:aws:kinesis:us-east-1:111122223333:stream/output_stream_name"  
    ]  
  },  
  {
```

```

    "Effect": "Allow",
    "Action": [
      "kms:GenerateDataKey"
    ],
    "Resource": [
      "arn:aws:kms:us-east-1:111122223333:key/*"
    ],
    "Condition": {
      "StringLike": {
        "aws:ResourceTag/AWSServiceName": "ChimeSDK"
      }
    }
  }
]
}

```

S3RecordingSink ポリシー

S3RecordingSink を使用する場合は、次のポリシーを追加します。

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "s3:PutObject",
      "s3:PutObjectAcl",
      "s3:PutObjectTagging",
    ],
    "Resource": [
      "arn:aws:s3:::input_bucket_path/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesisvideo:GetDataEndpoint",
      "kinesisvideo:ListFragments",
      "kinesisvideo:GetMediaForFragmentList"
    ],
    "Resource": [
      "arn:aws:kinesisvideo:us-east-1:111122223333:stream/*"
    ]
  }
]
}

```

```

    "Condition": {
      "StringLike": {
        "aws:ResourceTag/AWSServiceName": "ChimeSDK"
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:ListFragments",
        "kinesisvideo:GetMediaForFragmentList"
      ],
      "Resource": [
        "arn:aws:kinesisvideo:us-east-1:111122223333:stream/Chime*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/*"
      ],
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/AWSServiceName": "ChimeSDK"
        }
      }
    }
  ]
}

```

通話後分析ポリシー

AmazonTranscribeCallAnalyticsProcessor の通話後分析機能を使用する場合は、次のポリシーを追加します。

```

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

```

```
    "Action": [
      "iam:PassRole"
    ],
    "Resource": [
      "arn:aws:iam::111122223333:role/transcribe_role_name"
    ],
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "transcribe.streaming.amazonaws.com"
      }
    }
  }
]
```

VoiceEnhancementSinkConfiguration ポリシー

VoiceEnhancementSinkConfiguration の要素を使用する場合は、次のポリシーを追加します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:PutObjectAcl",
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3::input_bucket_path/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:GetDataEndpoint",
        "kinesisvideo:ListFragments",
        "kinesisvideo:GetMediaForFragmentList"
      ],
      "Resource": [
```

```

        "arn:aws:kinesisvideo:us-east-1:111122223333:stream/*"
    ],
    "Condition":{
        "StringLike":{
            "aws:ResourceTag/AWSServiceName":"ChimeSDK"
        }
    }
},
{
    "Effect":"Allow",
    "Action":[
        "kinesisvideo:ListFragments",
        "kinesisvideo:GetMediaForFragmentList"
    ],
    "Resource":[
        "arn:aws:kinesisvideo:us-east-1:111122223333:stream/Chime*"
    ]
},
{
    "Effect":"Allow",
    "Action":[
        "kms:GenerateDataKey"
    ],
    "Resource":[
        "arn:aws:kms:us-east-1:111122223333:key/*"
    ],
    "Condition":{
        "StringLike":{
            "aws:ResourceTag/AWSServiceName":"ChimeSDK"
        }
    }
}
]
}

```

VoiceAnalyticsProcessor ポリシー

VoiceAnalyticsProcessor を使用する場合は、定義したシンクに応じて、LambdaFunctionSink、SqsQueueSink、SnsTopicSink のポリシーを追加してください。

LambdaFunctionSink ポリシー:

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Action": [
      "lambda:InvokeFunction",
      "lambda:GetPolicy"
    ],
    "Resource": [
      "arn:aws:lambda:us-east-1:111122223333:function:function_name"
    ],
    "Effect": "Allow"
  }
]
```

SqsQueueSink ポリシー

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sqs:SendMessage",
        "sqs:GetQueueAttributes"
      ],
      "Resource": [
        "arn:aws:sqs:us-east-1:111122223333:queue_name"
      ],
      "Effect": "Allow"
    },
    {
      "Effect": "Allow",
      "Action": ["kms:GenerateDataKey", "kms:Decrypt"],
      "Resource": "arn:aws:kms:us-east-1:111122223333:key/*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/AWSServiceName": "ChimeSDK"
        }
      }
    }
  ]
}
```

SnsTopicSink ポリシー:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sns:Publish",
        "sns:GetTopicAttributes"
      ],
      "Resource": [
        "arn:aws:sns:us-east-1:111122223333:topic_name"
      ],
      "Effect": "Allow"
    },
    {
      "Effect": "Allow",
      "Action": ["kms:GenerateDataKey", "kms:Decrypt"],
      "Resource": "arn:aws:kms:us-east-1:111122223333:key/*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/AWSServiceName": "ChimeSDK"
        }
      }
    }
  ]
}
```

通話分析のステータスを理解する

メディアインサイトパイプラインでは、次のいずれかまたは両方を行うと、一連のステータスが追跡されます。

- Amazon Transcribe や音声分析など、機械学習処理の要素を複数使用する。
- 機械学習処理の有無にかかわらず、通話録音を有効にする。

パイプラインと要素のステータスを取得するには、[GetMediaPipeline](#) API と [EventBridge 通知](#) を使用します。

音声分析タスクのステータスを取得するには、[GetSpeakerSearchTask](#)と[GetVoiceToneAnalysisTask](#) APIs、および[音声分析通知ターゲット](#)を使用します。

メディアインサイトパイプラインでは、次のステータスが追跡されます。

- **パイプラインステータス** — 通話分析パイプライン (メディアインサイトパイプラインとも呼ばれる) の全体的なステータス。これは、その要素のステータスによって決まります。
- **要素ステータス** — 各メディアインサイトパイプライン設定要素の処理ステータス。
- **タスクステータス** — 音声分析用に開始したメディアインサイトパイプラインタスクの処理ステータス。VoiceAnalyticsProcessor 要素のステータスは、タスクステータスによって決まります。通話分析パイプラインの他の要素には、タスクステータスはありません。

メディアインサイトパイプラインタスクの詳細については、このガイドで前述した「[通話分析の用語](#)」を参照してください。

必ずしもすべてのメディアインサイト設定要素タイプに要素ステータスがあるわけではありません。一般的に、「プロセッサ」タイプのメディアインサイト設定要素にのみ要素ステータスがあります。また、Amazon S3 録音シンクと音声エンハンスメントシンクにはプロセッサステータスがあります。具体的には、次のメディアインサイト設定要素タイプに要素ステータスがあります。

- AmazonTranscribeProcessor
- AmazonTranscribeCallAnalyticsProcessor
- S3RecordingSink
- VoiceAnalyticsProcessor
- VoiceEnhancementSink

パイプラインのステータスは、次に示すように、要素ステータスによって決まります。

パイプラインのステータス	条件
NotStarted	すべての要素ステータスが開始になっていません。
Initializing	少なくとも 1 つの要素が初期化中で、それ以外は開始されていません。

パイプラインのステータス	条件
InProgress	少なくとも 1 つの要素が稼働中です。
[失敗]	少なくとも 1 つの要素に障害が発生し、それ以外の要素は停止しています。
停止中	停止条件をすべて確認するには、 通話分析パイプラインを管理する を参照してください。
停止	すべての要素が停止しています。
一時停止中	すべての要素が一時停止しています。

他の要素のステータスとは異なり、VoiceAnalyticsProcessor 要素には、やや特別な性質があります。前述のように、Amazon Chime SDK 音声分析機能に対応する VoiceAnalyticsProcessor 要素のステータスは、[StartSpeakerSearchTask](#) および から作成されたタスクステータスによって決まります [StartVoiceToneAnalysisTask](#)。

- VoiceAnalyticsProcessor の要素のステータスは NotStarted の状態で始まります。これは、StartSpeakerSearchTask と StartVoiceToneAnalysisTask を呼び出さないと、要素のステータスが Initializing から InProgress に変わらないからです。
- VoiceAnalyticsProcessor では、1 つのタスクを開始し、タスクの実行中に [停止条件](#) を満たさない限り、InProgress が維持されます。
- VoiceAnalyticsProcessor が InProgress の場合にも、タスクの処理時間に対してのみ課金が発生します。
- 少なくとも 1 つの音声分析タスクを開始し、それ以外のタスクを実行していないメディアインサイトパイプラインをクリーンアップするには、DeleteMediaPipeline を呼び出す必要があります。
- タスクが実行中、または正常に完了している限り、VoiceAnalyticsProcessor 要素のステータスは InProgress が維持されます。

Amazon での通話分析パイプラインのモニタリング CloudWatch

Amazon を使用して CloudWatch、Amazon Chime SDK コール分析パイプラインをモニタリングできます。また、特定のしきい値をモニタリングするアラームを設定し、しきい値に達したときに通

知を送信したりアクションを実行したりできます。の詳細については CloudWatch、[「Amazon ユーザーガイド CloudWatch」](#) を参照してください。

トピック

- [前提条件](#)
- [コール分析メトリクス](#)
- [CloudWatch パイプラインメトリクスの デイメンション](#)

前提条件

CloudWatch メトリクスを使用するには、まず、Amazon にサービスメトリクスを発行するアクセス許可を付与するメディアパイプラインサービスにリンクされたロールを作成する必要があります CloudWatch。サービスリンクロールの詳細については、このガイドの「[メディアパイプライン用のサービスにリンクされたロールを作成する](#)」を参照してください。

コール分析メトリクス

Amazon Chime SDK コール分析は、メディアインサイト設定を使用して作成したメディアインサイトパイプラインの AWS/ChimeSDK 名前空間に以下のメトリクスを公開します。

メトリクス	説明
MediaInsightsPipelineCreated	メディアインサイトパイプラインは正常に作成されました。 単位: 個
MediaInsightsPipelineStopped	メディアインサイトパイプラインは正常に停止されました。 単位: 個
MediaInsightsPipelineFailed	メディアインサイトパイプラインは失敗しました。 単位: 個
MediaInsightsPipelineDuration	パイプラインの作成から停止/失敗までの時間。

メトリクス	説明
	単位: 秒
MediaInsightsPipelineBillingDuration	メディアインサイトパイプラインの課金期間。 単位: 個
RecordingFileSize	記録ファイルのサイズ。 単位: バイト
RecordingDuration	記録の期間。 単位: 秒

CloudWatch パイプラインメトリクスの デイメンション

次の表に、通話分析パイプラインのモニタリングに使用できる CloudWatch デイメンションを示します。

デイメンション	説明
MediaInsightsPipelineConfigurationId	メディアインサイトパイプライン設定の ID。
MediaInsightsPipelineConfigurationName	メディアインサイトパイプライン設定の名前。

通話分析プロセッサと出力先

各メディアインサイトパイプライン設定では、一意の要素を 1 回のみ指定できます。すべてのプロセッサとシンクは同じ AWS アカウントに存在し、呼び出すエンドポイントと同じ AWS リージョンに作成する必要があります。例えば、Amazon Chime SDK メディアパイプライン向けに us-east-1 のエンドポイントを使用する場合、us-west-2 リージョンから Kinesis Data Stream を渡すことはできません。

各送信先に関する情報については、各セクションを展開してください。

Amazon Transcribe Call Analytics プロセッサの送信先

対応シンク: KinesisDataStreamSink

このプロセッサを Amazon Transcribe プロセッサと組み合わせることはできません。Amazon Transcribe Call Analytics の詳細については、「Amazon Transcribe Developer Guide」の「[Real-time call analytics](#)」を参照してください。AmazonTranscribeCallAnalyticsProcessorConfiguration API コールで PostCallAnalyticsSettings を指定して [通話後分析](#) を有効にすると、メディアインサイトパイプラインが停止し、処理が終了したタイミングで、送信先に指定した Amazon S3 にアーティファクトが保存されます。

Note

パイプラインを 35 秒以上一時停止した後に再開すると、通話後分析のアーティファクトは、異なるセッション ID を持つ個別のファイルとして Amazon S3 バケット内に保存されます。

通話後分析のアーティファクトは、分析 JSON ファイルと音声録音 (WAV または Opus ファイル) で構成されています。編集済み (コンテンツの編集を有効にした場合) および編集済みでない録音ファイルの Amazon S3 バケット URL は、メタデータセクションの onetimeMetadata の一部として、Amazon Transcribe Call Analytics の通話後分析セッションごとに 1 回 Kinesis Data Stream に送信されます。

Amazon Transcribe Call Analytics による通話分析では、Kinesis Video Streams から音声データの入力を取得します。

- 対応メディアエンコーディング: PCM 符号付き 16 ビットリトルエンディアン音声。
- 対応メディアサンプリングレート: 8,000 Hz ~ 48,000 Hz

Amazon Transcribe Analytics プロセスの StreamConfiguration 入力:

- ストリームごとに KinesisVideoStreamArn を指定する必要があります。
- (オプション) KVS FragmentNumber により、指定したフラグメントの後のチャンクを使用して通話分析ジョブが開始されます。指定していない場合は、Kinesis Video Streams の最新のチャンクが使用されます。
- StreamChannelDefinition により、スピーカーを定義します。Amazon Transcribe Call Analytics には 2 チャンネルの音声が必要です。 [CreateMediaInsightsPipeline](#) API を呼び出すとき

は、どのスピーカーがどのチャンネルにあるかを指定する必要があります。例えば、エージェントが最初に話す場合、ChannelId を 0 に設定して最初のチャンネルを示し、ParticipantRole を AGENT に設定してエージェントがスピーカーであることを示します。

Note

Voice Connector を使用して、Amazon Transcribe Call Analytics プロセッサで MediaInsightsPipeline を作成する場合、Voice Connector アカウントレグの音声には AGENT を、PSTN レグの音声には ParticipantRole の CUSTOMER を設定します。Voice Connector で SIPREC を使用する場合は、SIPREC メタデータによってそれらが決定されます。ほとんどの場合、辞書順に基づく値が最も小さいストリームラベルのスピーカーが AGENT と見なされます。

次の例は、1 つのデュアルチャンネルオーディオストリームを使用する Kinesis Video Streams の入力を示しています。

```
"StreamChannelDefinition" : {
  "NumberOfChannels" : 2
  "ChannelDefinitions": [
    {
      "ChannelId": 0,
      "ParticipantRole": "AGENT"
    },
    {
      "ChannelId": 1,
      "ParticipantRole": "CUSTOMER"
    }
  ]
}
```

対照的に、次の例は、2 つの異なる Kinesis Video Streams から取得した 2 つのモノラル入力を示しています。

```
KVS-1:
  "StreamChannelDefinition" : {
    "NumberOfChannels" : 1
    "ChannelDefinitions": [
      {
        "ChannelId": 0,
```

```

        "ParticipantRole": "AGENT"
    }
]
}
KVS-2:
  "StreamChannelDefinition" : {
    "NumberOfChannels" : 1
    "ChannelDefinitions": [
      {
        "ChannelId": 1,
        "ParticipantRole": "CUSTOMER"
      }
    ]
  }
}

```

Amazon Transcribe Call Analytics の出力

各 Amazon Transcribe レコードには、UtteranceEvent または CategoryEvent が含まれますが、両方は含まれません。CategoryEvents には、detail-type の TranscribeCallAnalyticsCategoryEvent があります。

次の例は、Amazon Transcribe に使用する 1 回限りのメタデータ出力形式を示しています。

```

{
  "time": "string", // ISO8601 format
  "service-type": "CallAnalytics",
  "detail-type": "CallAnalyticsMetadata",
  "mediaInsightsPipelineId": "string",
  "metadata": "string" // JSON encoded string of the metadata object
}

// metadata object
{
  "voiceConnectorId": "string",
  "callId": "string",
  "transactionId": "string",
  "fromNumber": "string",
  "toNumber": "string",
  "direction": "string",
  "oneTimeMetadata": "string" // JSON encoded string of oneTimeMetadata object
}

// onetimeMetadata object

```

```
{
  "inviteHeaders": "string", // JSON encoded string of SIP Invite headers key-value
  pair
  "siprecMetadata": "string", // siprec metadata in XML
  "siprecMetadataJson": "string", // siprec metadata in JSON (converted from above
  XML)

  // If PostcallSettings are enabled for Amazon Transcribe Call Analytics
  "s3RecordingUrl": "string",
  "s3RecordingUrlRedacted": "string"
}

// inviteHeaders object
{
  "string": "string"
}
```

次の例は、Amazon Transcribe Call Analytics の出力形式を示しています。

```
{
  "time": "string", // ISO8601 format
  "service-type": "CallAnalytics",
  "detail-type": "TranscribeCallAnalytics",
  "mediaInsightsPipelineId": "string",
  "metadata": {
    "voiceConnectorId": "string",
    "callId": "string",
    "transactionId": "string",
    "fromNumber": "string",
    "toNumber": "string",
    "direction": "string"
  },
  "UtteranceEvent": {
    "UtteranceId": "string",
    "ParticipantRole": "string",
    "IsPartial": boolean,
    "BeginOffsetMillis": number,
    "EndOffsetMillis": number,
    "Transcript": "string",
    "Sentiment": "string",
    "Items": [{
      "Content": "string",
      "Confidence": number,
```



```

{
  "time": "string", // ISO8601 format
  "service-type": "CallAnalytics",
  "detail-type": "CallAnalyticsMetadata",
  "callevent-type": "Update",
  "metadata": "string" // JSON encoded string of the metadata object
}

// metadata object
{
  "voiceConnectorId": "string",
  "callId": "string",
  "transactionId": "string",
  "fromNumber": "string",
  "toNumber": "string",
  "direction": "string",
  "oneTimeMetadata": "string" // JSON encoded string of oneTimeMetadata object
}

// onetimeMetadata object
{
  "sipHeaders": "string", // JSON encoded string of SIP Invite headers key-value pair
  "siprecMetadata": "string", // siprec metadata in XML
  "siprecMetadataJson": "string" // siprec metadata in JSON (converted from above
XML)
}

// sipHeaders object
{
  "string": "string"
}

```

次の例は、Call Analytics Amazon S3 Recording のメタデータ更新形式を示しています。

```

{
  "time": "string", // ISO8601 format
  "service-type": "CallAnalytics",
  "detail-type": "Recording",
  "callevent-type": "Update",
  "metadata": "string" // JSON encoded string of the metadata object
}

// metadata object

```

```

{
  "voiceConnectorId": "string",
  "callId": "string",
  "transactionId": "string",
  "fromNumber": "string",
  "toNumber": "string",
  "direction": "string",
  "oneTimeMetadata": "string" // JSON encoded in string of oneTimeMetadata object
}

// onetimeMetadata object
{
  "sipHeaders": "string", // JSON encoded string of SIP Invite headers key-value pair
  "siprecMetadata": "string", // siprec metadata in XML
  "siprecMetadataJson": "string" // siprec metadata in JSON (converted from above
XML)
}

// sipHeaders object
{
  "string": "string"
}

```

SIP コール記録メタデータ

次の例は、Alice と Bob の 2 人の間の SIP 通話を記録するためのメタデータを示しています。両方の参加者が音声と動画を送受信します。わかりやすくするために、この例では SIP と SDP のスニペットのみを使用しており、SRC は各参加者のストリームをミックスせずに SRS に記録しています。

```

INVITE sip:recorder@example.com SIP/2.0
Via: SIP/2.0/TCP src.example.com;branch=z9hG4bKdf6b622b648d9
From: <sip:2000@example.com>;tag=35e195d2-947d-4585-946f-09839247
To: <sip:recorder@example.com>
Call-ID: d253c800-b0d1ea39-4a7dd-3f0e20a
Session-ID: ab30317f1a784dc48ff824d0d3715d86
;remote=00000000000000000000000000000000
CSeq: 101 INVITE
Max-Forwards: 70
Require: siprec
Accept: application/sdp, application/rs-metadata,
application/rs-metadata-request
Contact: <sip:2000@src.example.com>;+sip.src

```

```
Content-Type: multipart/mixed;boundary=boundary
```

```
Content-Length: [length]
```

```
Content-Type: application/SDP
```

```
...
```

```
m=audio 49170 RTP/AVP 0
```

```
a=rtpmap:0 PCMU/8000
```

```
a=label:96
```

```
a=sendonly
```

```
...
```

```
m=video 49174 RTP/AVPF 96
```

```
a=rtpmap:96 H.264/90000
```

```
a=label:97
```

```
a=sendonly
```

```
...
```

```
m=audio 51372 RTP/AVP 0
```

```
a=rtpmap:0 PCMU/8000
```

```
a=label:98
```

```
a=sendonly
```

```
...
```

```
m=video 49176 RTP/AVPF 96
```

```
a=rtpmap:96 H.264/90000
```

```
a=label:99
```

```
a=sendonly
```

```
....
```

```
Content-Type: application/rs-metadata
```

```
Content-Disposition: recording-session
```

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<recording xmlns='urn:ietf:params:xml:ns:recording:1'>
```

```
  <datamode>complete</datamode>
```

```
    <group group_id="7+0TCyoxTmqmqyA/1weDAg==">
```

```
      <associate-time>2010-12-16T23:41:07Z</associate-time>
```

```
      <!-- Standardized extension -->
```

```
      <call-center xmlns='urn:ietf:params:xml:ns:callcenter'>
```

```
        <supervisor>sip:alice@atlanta.com</supervisor>
```

```
      </call-center>
```

```
      <mydata xmlns='http://example.com/my'>
```

```
        <structure>structure!</structure>
```

```
        <whatever>structure</whatever>
```

```
      </mydata>
```

```
    </group>
```

```
  <session session_id="hVpd7YQgRW2nD22h7q60JQ==">
```

```

        <sipSessionID>ab30317f1a784dc48ff824d0d3715d86;
            remote=47755a9de7794ba387653f2099600ef2</
sipSessionID>
        <group-ref>7+0TCyoxTmqmqyA/1weDAg==
</group-ref>
        <!-- Standardized extension -->
        <mydata xmlns='http://example.com/my'>
            <structure>F00!</structure>
            <whatever>bar</whatever>
        </mydata>
    </session>
    <participant
        participant_id="srfBEImCRp2QB23b7Mpk0w==">
        <nameID aor="sip:alice@atlanta.com">
            <naSRCme xml:lang="it">Alice</name>
        </nameID>
        <!-- Standardized extension -->
        <mydata xmlns='http://example.com/my'>
            <structure>F00!</structure>
            <whatever>bar</whatever>
        </mydata>
    </participant>
    <participant
        participant_id="zSfPoSvdSDCmU3A3TRDxAw==">
        <nameID aor="sip:bob@biloxi.com">
            <name xml:lang="it">Bob</name>
        </nameID>
        <!-- Standardized extension -->
        <mydata xmlns='http://example.com/my'>
            <structure>F00!</structure>
            <whatever>bar</whatever>
        </mydata>
    </participant>
    <stream stream_id="UAAMm5GRQKSCMVvLy14rFw=="
        session_id="hVpd7YQgRW2nD22h7q60JQ==">
        <label>96</label>
    </stream>
    <stream stream_id="i1Pz3to5hGk8fuX1+PbwCw=="
        session_id="hVpd7YQgRW2nD22h7q60JQ==">
        <label>97</label>
    </stream>
    <stream stream_id="8zc6e01YTLWIINA6GR+3ag=="
        session_id="hVpd7YQgRW2nD22h7q60JQ==">
        <label>98</label>

```

```

</stream>
<stream stream_id="EiXGlc+4TruqqoDaNE76ag=="
  session_id="hVpd7YQgRW2nD22h7q60JQ==">
  <label>99</label>
</stream>
<sessionrecordingassoc session_id="hVpd7YQgRW2nD22h7q60JQ==">
  <associate-time>2010-12-16T23:41:07Z</associate-time>
</sessionrecordingassoc>
<participantsessionassoc
  participant_id="srfBEImCRp2QB23b7Mpk0w=="
  session_id="hVpd7YQgRW2nD22h7q60JQ==">
  <associate-time>2010-12-16T23:41:07Z</associate-time>
</participantsessionassoc>
<participantsessionassoc
  participant_id="zSfPoSvdSDCmU3A3TRDxAw=="
  session_id="hVpd7YQgRW2nD22h7q60JQ==">
  <associate-time>2010-12-16T23:41:07Z</associate-time>
</participantsessionassoc>
<participantstreamassoc
  participant_id="srfBEImCRp2QB23b7Mpk0w==">
  <send>i1Pz3to5hGk8fuX1+PbwCw==</send>
  <send>UAAMm5GRQKSCMVvLy14rFw==</send>
  <recv>8zc6e01YTLWIINA6GR+3ag==</recv>
  <recv>EiXGlc+4TruqqoDaNE76ag==</recv>
</participantstreamassoc>
<participantstreamassoc
  participant_id="zSfPoSvdSDCmU3A3TRDxAw==">
  <send>8zc6e01YTLWIINA6GR+3ag==</send>
  <send>EiXGlc+4TruqqoDaNE76ag==</send>
  <recv>UAAMm5GRQKSCMVvLy14rFw==</recv>
  <recv>i1Pz3to5hGk8fuX1+PbwCw==</recv>
</participantstreamassoc>
</recording>

```

次の例は、ある通話参加者がもう片方を保留にしたときの更新されたメタデータを示しています。この場合、participant_id srfBEImCRp2QB23b7Mpk0w== は、メディアストリームのみを受信し、メディアは送信しないため、send XML 要素は省略されます。これとは対照的に、participant_id zSfPoSvdSDCmU3A3TRDxAw== は、他の参加者にはメディアを送信しますが、相手からはメディアを受信しないため、recv XML 要素は省略されます。

```

INVITE sip:recorder@example.com SIP/2.0
  Via: SIP/2.0/TCP src.example.com;branch=z9hG4bKdf6b622b648d9

```

```
From: <sip:2000@example.com>;tag=35e195d2-947d-4585-946f-09839247
To: <sip:recorder@example.com>
Call-ID: d253c800-b0d1ea39-4a7dd-3f0e20a
Session-ID: ab30317f1a784dc48ff824d0d3715d86
;remote=f81d4fae7dec11d0a76500a0c91e6bf6
CSeq: 101 INVITE
Max-Forwards: 70
Require: siprec
Accept: application/sdp, application/rs-metadata,
application/rs-metadata-request
Contact: <sip:2000@src.example.com>;+sip.src
Content-Type: multipart/mixed;boundary=foobar
Content-Length: [length]
```

```
Content-Type: application/SDP
```

```
...
```

```
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:96
a=sendonly
```

```
...
```

```
m=video 49174 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:97
a=sendonly
```

```
...
```

```
m=audio 51372 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:98
a=sendonly
```

```
...
```

```
m=video 49176 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:99
a=sendonly
```

```
....
```

```
Content-Type: application/rs-metadata
Content-Disposition: recording-session
```

```
<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording:1'>
    <datamode>partial</datamode>
    <participantstreamassoc
```

```

    participant_id="srfBEImCRp2QB23b7Mpk0w==">
    <recv>8zc6e0lYtLWIINA6GR+3ag==</recv>
    <recv>EiXGlc+4TruqqoDaNE76ag==</recv>
  </participantstreamassoc>
  <participantstreamassoc
    participant_id="zSfPoSvdSDCmU3A3TRDxAw==">
    <send>8zc6e0lYtLWIINA6GR+3ag==</send>
    <send>EiXGlc+4TruqqoDaNE76ag==</send>
  </participantstreamassoc>
</recording>

```

次の例は、通話が再開されたときのメタデータの更新を示しています。これで、ペイロードには send と recv XML 要素が含まれるようになりました。

```

INVITE sip:recorder@example.com SIP/2.0
Via: SIP/2.0/TCP src.example.com;branch=z9hG4bKdf6b622b648d9
From: <sip:2000@example.com>;tag=35e195d2-947d-4585-946f-09839247
To: <sip:recorder@example.com>
Call-ID: d253c800-b0d1ea39-4a7dd-3f0e20a
Session-ID: ab30317f1a784dc48ff824d0d3715d86
;remote=f81d4fae7dec11d0a76500a0c91e6bf6
CSeq: 101 INVITE
Max-Forwards: 70
Require: siprec
Accept: application/sdp, application/rs-metadata,
application/rs-metadata-request
Contact: <sip:2000@src.example.com>;+sip.src
Content-Type: multipart/mixed;boundary=foobar
Content-Length: [length]

Content-Type: application/SDP
...
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:96
a=sendonly
...
m=video 49174 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:97
a=sendonly
...
m=audio 51372 RTP/AVP 0

```



```
a=rtpmap:0 PCMU/8000
a=label:98
a=sendonly
...
m=video 49176 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:99
a=sendonly
....
```

```
Content-Type: application/rs-metadata
Content-Disposition: recording-session
```

```
<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording:1'>
    <datamode>partial</datamode>
    <participantstreamassoc
      participant_id="srfBEImCRp2QB23b7Mpk0w==">
      <send>i1Pz3to5hGk8fuXl+PbwCw==</send>
      <send>UAAMm5GRQKSCMVvLy14rFw==</send>
      <recv>8zc6e01YTLWIINA6GR+3ag==</recv>
      <recv>EiXGlc+4TruqqoDaNE76ag==</recv>
    </participantstreamassoc>
    <participantstreamassoc
      participant_id="zSfPoSvdSDCmU3A3TRDxAw==">
      <send>8zc6e01YTLWIINA6GR+3ag==</send>
      <send>EiXGlc+4TruqqoDaNE76ag==</send>
      <recv>i1Pz3to5hGk8fuXl+PbwCw==</recv>
      <recv>UAAMm5GRQKSCMVvLy14rFw==</recv>
    </participantstreamassoc>
  </recording>
```

Amazon Transcribe プロセッサの送信先

対応シンク: KinesisDataStreamSink

このプロセッサを Amazon Transcribe Call Analytics と組み合わせることはできません。Amazon Transcribe への入力と出力の詳細については、「Amazon Transcribe Developer Guide」の「[Transcribe streaming audio](#)」を参照してください。

Amazon Transcribe による通話分析セッションでは、Kinesis Video Streams から音声データの入力を取得します。

- サポートされている MediaEncoding: PCM 署名付き 16 ビットリトルエンディアンオーディオ。
- サポートされている MediaSampleRate サンプルレート: 8,000 Hz ~ 48,000 Hz。

Amazon Transcribe プロセッサの StreamConfiguration 入力:

- ストリームごとに KinesisVideoStreamArn を指定する必要があります。
- (オプション) KVS FragmentNumber により、指定したフラグメントの後のチャンクを使用して通話分析ジョブが開始されます。指定していない場合は、Kinesis Video Streams の利用可能な最新のチャンクが使用されます。
- StreamChannelDefinition Amazon Transcribe は現在、2 つのチャンネルの音声に対応しています。NumberOfChannels は、ランタイム StreamChannelDefinition で指定する必要があります。さらに、モノラル音声を 2 つの個別チャンネルで送信する場合は、ChannelId を渡さなければなりません。トランスクリプトでは、チャンネルに ch_0 および ch_1 というラベルが割り当てられます。次の例は、1 つのモノラル音声チャンネルストリームに使用する KVS 入力を示しています。

```
"StreamChannelDefinition" : {  
  "NumberOfChannels" : 1  
}
```

次の例は、2 つの異なるストリームの 2 つのモノラル音声入力に使用する KVS 入力を示しています。

```
KVS-1:  
  "StreamChannelDefinition" : {  
    "NumberOfChannels" : 1  
    "ChannelDefinitions": [  
      {  
        "ChannelId": 0  
      }  
    ]  
  }  
KVS-2:  
  "StreamChannelDefinition" : {  
    "NumberOfChannels" : 1  
    "ChannelDefinitions": [  
      {  
        "ChannelId": 1  
      }  
    ]  
  }
```

```
]
}
```

Note

Amazon Transcribe プロセッサを使用して MediaInsightsPipeline を作成する Voice Connector の場合、Voice Connector アカウントレグの音声は channel-0 に、PSTN レグの音声は channel-1 に割り当てられます。

Voice Connector で SIPREC を使用する場合は、SIPREC メタデータによってそれらが決定されます。ほとんどの場合、辞書順に基づく値が最も小さいストリームラベルの音声は channel-0 に割り当てられます。

Amazon Transcribe および Amazon Transcribe Call Analytics プロセッサでは、2つの Kinesis Video Streams を渡し、各ストリームにモノラル音声チャンネルが含まれている場合、両方のチャンネルが1つの音声ストリームにインターリーブされた後に、Transcribe または Transcribe Call Analytics のデータが処理されます。

Amazon Transcribe の出力

次の例は、Amazon Transcribe に使用する 1 回限りのメタデータ出力形式を示しています。

```
{
  "time": "string", // ISO8601 format
  "service-type": "CallAnalytics",
  "detail-type": "CallAnalyticsMetadata",
  "mediaInsightsPipelineId": "string",
  "metadata": "string" // JSON encoded string of the metadata object
}

// metadata object
{
  "voiceConnectorId": "string",
  "callId": "string",
  "transactionId": "string",
  "fromNumber": "string",
  "toNumber": "string",
  "direction": "string",
  "oneTimeMetadata": "string" // JSON encoded string of oneTimeMetadata object
}

// onetimeMetadata object
```

```
{
  "inviteHeaders": "string", // JSON encoded string of SIP Invite headers key-value
  pair
  "siprecMetadata": "string", // siprec metadata in XML
  "siprecMetadataJson": "string" // siprec metadata in JSON (converted from above
  XML)
}

// inviteHeaders object
{
  "string": "string"
}
```

次の例は、Amazon Transcribe の出力形式を示しています。

```
{
  "time": "string", // ISO8601 format
  "service-type": "CallAnalytics",
  "detail-type": "Transcribe",
  "mediaInsightsPipelineId": "string",
  "metadata": {
    "voiceconnectorId": "string",
    "callId": "string",
    "transactionId": "string",
    "fromNumber": "string",
    "toNumber": "string",
    "direction": "string"
  }
  "TranscriptEvent": {
    "Transcript": {
      "Results": [{
        "Alternatives": [{
          "Entities": [{
            "Category": "string",
            "Confidence": number,
            "Content": "string",
            "EndTime": number,
            "StartTime": number,
            "Type": "string"
          }],
        }],
      "Items": [{
        "Confidence": number,
        "Content": "string",
```

```

        "EndTime": number,
        "Speaker": "string",
        "Stable": boolean,
        "StartTime": number,
        "Type": "string",
        "VocabularyFilterMatch": boolean
    }],
    "Transcript": "string"
}],
"ChannelId": "string",
"EndTime": number,
"IsPartial": boolean,
"LanguageCode": "string",
"LanguageIdentification": [{
    "LanguageCode": "string",
    "Score": number
}],
"ResultId": "string",
"StartTime": number
}]
}
}
}
}

```

音声分析プロセッサの送信先

対応シンク

ク:KinesisDataStreamSink、SqsQueueSink、SnsTopicSink、LambdaFunctionSink。

このプロセッサは、Amazon Transcribe Call Analytics プロセッサ、Amazon Transcribe プロセッサ、または通話録音と組み合わせることができます。音声分析プロセッサを呼び出すには、[StartSpeakerSearchTask](#)または [StartVoiceToneAnalysisTask](#) APIs を使用する必要があります。音声分析の使用の詳細については、「[Using Amazon Chime SDK voice analytics](#)」を参照してください。

Kinesis Data Stream をシンクとして使用する

通話分析によって生成される Kinesis Data Stream (KDS) レコードには、メディアパイプライン ID、詳細タイプ、メタデータ、プロセッサ固有のセクションが含まれます。Kinesis Data Stream からのデータを利用する方法については、「Amazon Kinesis Streams Developer guide」の「[Reading Data from Amazon Kinesis Data Streams](#)」を参照してください。このシンクを使用して設定を行うには、指定されたストリームの `kinesis:DescribeStream` 権限が必要です。

Metadata

生成された KDS レコードの metadata セクションには、[CreateMediaInsightsPipelineAPI](#) コール `CallAnalyticsRuntimeMetadata` 中に指定されたキーと値のペアが含まれます。通話分析セッションを Voice Connector によって開始した場合、メタデータセクションには次のパラメータが自動的に入力されます。

- `transactionId`
- `fromNumber`
- `toNumber`
- `callId`
- `voiceConnectorId`
- `direction`

上記のパラメータに加えて、Voice Connector が開始した通話分析セッションのメタデータセクションには、以下を含む `oneTimeMetadata` フィールドが入力されます。

- `inviteHeaders`
- `siprecMetadata`

これはセッションの開始時に 1 回だけ Kinesis Data Streams に発行され、`detail-type` の `CallAnalyticsMetadata` があります。

Kinesis Data Streams に配信される各レコードのソースを一意に識別できるよう

に、[CreateMediaInsightsPipelineAPI](#) コール `MediaInsightsRuntimeMetadata` ごとに一意の識別子を渡すことができます。

Amazon S3 通話録音

通話分析の録音では、KVS ストリームから音声を読み取り、録音を音声ファイルにし、そのファイルを指定した Amazon S3 バケットにアップロードします。録音後、ファイルの場所と共に通話メタデータも KDS に送信します。データウェアハウスを有効にすると、通話メタデータ (SIPREC を使用している場合は、SIPREC メタデータも含まれる) は、クエリ可能な Parquet テーブルのセットとしてデータウェアハウスに配信されます。

他の通話分析プロセッサと同様に、最初にパイプラインの設定を行う必要があります。この設定を行うには、Amazon Chime SDK コンソールまたは CLI を使用します。次に、CLI を使用してパイプラインを作成します。コンソールを使用して録音設定を行う方法については、このセクションで前述し

た「[通話分析を設定する](#)」で詳しく確認できます。録音ワークフローの使用方法については、このセクションで前述した「[通話録音を目的としたワークフロー](#)」で詳しく確認できます。

CLI を使用して設定を行うには

次のコマンドを実行します。

```
aws chime-sdk-media-pipeline create-media-insights-pipeline-configuration --cli-input-json file://configuration.json
```

次の例は、記録のみ有効にする設定用 JSON ファイルを示しています。

```
{
  "MediaInsightsPipelineConfigurationName": configuration_name,
  "ResourceAccessRoleArn": role_arn,
  "Elements": [
    {
      "KinesisDataStreamSinkConfiguration": {
        "InsightsTarget": KDS_arn //Where recording live metadata will be
delivered.
      },
      "Type": "KinesisDataStreamSink"
    },
    {
      "S3RecordingSinkConfiguration": {
        "Destination": "arn:aws:s3:::kvs-recording-testing",
        "RecordingFileFormat": file_format // Specify "Opus" or "WAV" as the
recording file format.
      },
      "Type": "S3RecordingSink"
    }
  ]
}
```

次の点に注意してください。

- Kinesis Video Streams を介して通話録音を有効にするには、オーディオは PCM 署名付き 16 ビットリトルエンディアンである必要があります。サンプルレートは 8 kHz でなければなりません。
- ビルダーは、フラグメントがコール分析によって保持され、使用できるようにするために、Kinesis Video Streams に十分なデータ保持期間を設定する必要があります。

- 通話録音を有効にする場合は、単独で使用する場合も、他のプロセッサと組み合わせて使用する場合も、録音用の Kinesis Video Streams ARN を 2 つ指定する必要があります。通話録音は 1 つのステレオ音声入力には対応していません。

Amazon S3 に通話録音する場合のメタデータ出力

次の例は、通話分析 Amazon S3 録音のメタデータ出力形式を示しています。

```
{
  "time": "string", // ISO8601 format
  "service-type": "CallAnalytics",
  "detail-type": "Recording",
  "mediaInsightsPipelineId": "string",
  "s3MediaObjectConsoleUrl": "string",
  "recordingDurationSeconds": "number",
  "metadata": "string" // JSON encoded string of the metadata object
}

// metadata object
{
  "voiceConnectorId": "string",
  "callId": "string",
  "transactionId": "string",
  "fromNumber": "string",
  "toNumber": "string",
  "direction": "string",
  "startTime": "string", // ISO8601 format
  "endTime": "string", // ISO8601 format
  "oneTimeMetadata": "string" // JSON encoded in string of oneTimeMetadata object
}

// onetimeMetadata object
{
  "sipHeaders": "string", // JSON encoded string of SIP Invite headers key-value pair
  "siprecMetadata": "string", // siprec metadata in XML
  "siprecMetadataJson": "string" // siprec metadata in JSON (converted from above
XML)
}

// sipHeaders object
{
  "string": "string"
}
```



```
}
```

音声エンハンスメントを有効にする

音声エンハンスメントを有効にするには、[CreateMediaInsightsPipelineConfiguration](#) API コールに `VoiceEnhancementSinkConfiguration` 要素を含めます。

一般的な要素の例を次に示します。

```
{
  "Type": "VoiceEnhancementSink",
  "VoiceEnhancementSinkConfiguration": {
    "Disabled": Boolean (string) // FALSE ==> Voice Enhancement will be performed
  }
}
```

設定を更新するには、[UpdateMediaInsightsPipelineConfiguration](#) API コールに `VoiceEnhancementSinkConfiguration` 要素を追加します。これを行うと、[GetMediaInsightsPipelineConfiguration](#) API は結果に `VoiceEnhancementSinkConfiguration` 要素を含めます。

このリクエスト例は、音声エンハンスメントと Amazon S3 への録音を有効にする方法を示しています。

```
POST /media-insights-pipeline-configurations HTTP/1.1
Content-type: application/json

{
  "MediaInsightsPipelineConfigurationName": "media_insights_configuration_name",
  "ResourceAccessRoleArn": "arn:aws:iam::account_id:role/resource_access_role",
  "Elements": [
    {
      "Type": "S3RecordingSink",
      "S3RecordingSinkConfiguration": {
        "Destination": "arn:aws:s3::input_bucket_path",
        "RecordingFileFormat": "Wav"
      }
    },
    {
      "Type": "VoiceEnhancementSink",
      "VoiceEnhancementSinkConfiguration": {
        "disabled": "false"
      }
    }
  ]
}
```

```
    }  
  }  
],  
"ClientRequestToken": "client_request_token"  
}
```

Note

VoiceEnhancementSink 要素を使用するには、通話分析の設定で S3RecordingSink 要素を常に設定する必要があります。

文字起こしの機能と録音シンクを連携させる

Amazon Transcribe と Amazon Transcribe Call Analytics のプロセッサを Amazon S3 録音シンクと連携させることができます。ビルダーは、[CreateMediaInsightsPipelineConfiguration](#) API コールで、またはコンソールを使用して、Amazon Transcribe プロセッサに加えて S3RecordingSinkConfiguration を渡すことができます。

Amazon S3 録音シンクと連携させた場合、Amazon Transcribe または Amazon Transcribe Call Analytics のプロセッサを使用できますが、両方を使用することはできません。これと同じ設定には、文字起こしプロセッサの有無にかかわらず、録音シンクの他に音声分析を追加できます。

Note

前述のプロセッサのいずれでも、録音を有効にできます。ただし、Amazon Transcribe Call Analytics と Amazon S3 通話録音の両方を有効にすると、録音ファイルが重複して配信されます。1 つは Amazon Transcribe Call Analytics から、もう 1 つは Amazon S3 通話録音からのファイルです。

次の点に注意してください。

- 一意の MediaInsightsPipelineConfigurationName を使用する必要があります。
- ResourceAccessRoleArn 詳細については、このガイドの「[通話分析のリソースアクセスロールを使用する](#)」を参照してください。
- Destination 値には S3 バスの ARN を指定する必要があります。同じアカウントで所有している Amazon S3 バケットを指定する必要があります。

- Transcribe と録音の指定がある設定を使用してパイプラインを作成する場合、一時停止と再開は Kinesis Data Stream で生成したインサイトにものみ表示されます。KVS ストリーム内のすべてのデータが記録され、Amazon S3 にアップロードされます。
- 設定で、録音に加え、Amazon Transcribe または Transcribe Call Analytics (TCA) を指定すると、メディアインサイトパイプラインによって、文字起こしまたは Transcribe Call Analytics のインサイトがリアルタイムで生成されます。その後、通話が終了すると、Amazon S3 への録音が行われます。通話分析中に文字起こしサービスで障害が発生しても、S3 録音ジョブの実行は試行されます。Amazon S3 への録音は、文字起こし完了後に実行されるため、録音が失敗しても、文字起こしのインサイトは影響を受けません。

この例は、Amazon Transcribe プロセッサと Amazon S3 録音シンクを指定した設定を示しています。この例を使用すると、結果を部分的に安定化でき、出力のレイテンシーも抑えられますが、精度上の影響が出る可能性があります。詳細については、「Amazon Transcribe Developer Guide」の「[Partial-result stabilization](#)」を参照してください。

```
{
  "MediaInsightsPipelineConfigurationName": unique_configuration_name,
  "ResourceAccessRoleArn": role_arn,
  "Elements": [{
    "AmazonTranscribeProcessorConfiguration": {
      "ContentIdentificationType": "string",
      "ContentRedactionType": "string",
      "EnablePartialResultsStabilization": boolean, //Enables partial result
stabilization. Can reduce latency. May impact accuracy.
      "FilterPartialResults": boolean, //To control partial utterance events
      "LanguageCode": "string",
      "LanguageModelName": "string",
      "PartialResultsStability": "string",
      "PiiEntityTypeTypes": "string",
      "ShowSpeakerLabel": boolean,
      "VocabularyFilterMethod": "string",
      "VocabularyFilterName": "string",
      "VocabularyName": "string"
    },
    "Type": "AmazonTranscribeProcessor"
  },
  {
    "KinesisDataStreamSinkConfiguration": {
      "InsightsTarget": KDS_arn //Where recording and insights live metadata
will be delivered.
    },
  }
```

```
        "Type": "KinesisDataStreamSink"
    },
    {
        "S3RecordingSinkConfiguration": {
            "Destination": S3_Arn,
            "RecordingFileFormat": file_format // Specify "Opus" or "WAV" as the
recording file format.
        },
        "Type": "S3RecordingSink"
    }
]
}
```

Amazon EventBridge 通知の使用

Amazon Chime SDK コール分析では、メディアインサイトパイプラインの状態が変化したとき、またはコール分析のリアルタイムアラート条件が満たされたときに、デフォルトの EventBridge バスにイベントを送信できます。メディアインサイトパイプラインのエラーステータスの更新では、リソースが非同期的に失敗した場合に通知するように EventBridge ターゲットを設定することをお勧めします。通話分析の通知には、aws.chime ソースに加え、さまざまな詳細タイプがあります。この点については、次のセクションで説明します。詳細については、[「Amazon ユーザーガイド EventBridge」](#)を参照してください。

トピック

- [ステータスの更新](#)
- [リアルタイムアラート](#)

ステータスの更新

メディアインサイトパイプラインは、通話分析セッションが進行し、正常に終了するか、エラーが発生すると EventBridge、通知を送信します。次の場合、「Media Insights の状態変更」の詳細タイプの EventBridge 通知を受け取ります。

- メディアインサイトパイプラインのステータスが変化した。
- メディアインサイトパイプライン要素のステータスが変化した。
- いずれかのパイプライン要素が停止した。
- いずれかのパイプライン要素で障害が発生した。

詳細セクションには、次のフィールドが常に含まれています。

- version
- mediaInsightsPipelineArn
- eventType

メディアインサイトパイプラインで分析プロセッサやデータシンクなどの要素を複数使用している場合、詳細セクションには `mediaInsightsPipelineElementStatuses` フィールドもあります。このフィールドは、パイプライン内の各要素のステータスを示しています。想定される各パイプライン要素のステータスを次に示します。

- NotStarted
- InProgress
- Stopped
- Failed

詳細セクションには、[CreateMediaInsightsPipeline](#) API コー

ル `MediaInsightsRuntimeMetadata` 中に で指定されたキーと値のペアも含まれます。通話分析セッションを Voice Connector によって開始した場合、メタデータセクションには次のパラメータが自動的に入力されます。

- transactionId
- fromNumber
- toNumber
- callId
- voiceConnectorId
- direction

メディアインサイトパイプライン内に要素が 1 つしかない場合は、次のイベントタイプが示されることがあります。詳細については、各セクションを展開してください。

Amazon Chime SDK メディアインサイトは稼働している

次の例は、一般的なイベント構造を示しています。

```
{
  "version": "0",
  "id": "string",
```

```
"detail-type": "Media Insights State Change",
"source": "aws.chime",
"account": number,
"region": "string",
"time": "yyyy-mm-ddThh:mm:ssZ",
"resources": []
"detail": {
  "version": "0",
  "mediaInsightsPipelineArn": "string",
  "eventType": "chime:MediaInsightsInProgress",
  "version": "0",
  "callId": "string",
  "transactionId": "string",
  "fromNumber": "string",
  "toNumber": "string",
  "voiceConnectorId": "string",
  "direction": "string"
}
}
```

Amazon Chime SDK メディアインサイトは一時停止している

次の例は、一般的なイベント構造を示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "region": "string",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsPaused",
    "callId": "string",
    "transactionId": "string",
    "fromNumber": "string",
    "toNumber": "string",
    "voiceConnectorId": "string",
    "direction": "string"
  }
}
```

```
}
```

Amazon Chime SDK メディアインサイトは停止している

次の例は、一般的なイベント構造を示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "region": "string",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsStopped",
    "callId": "string",
    "transactionId": "string",
    "fromNumber": "string",
    "toNumber": "string",
    "voiceConnectorId": "string",
    "direction": "string"
  }
}
```

Amazon Chime SDK メディアインサイトに一時的な障害が発生している

サービスに一時的な障害が発生し、再試行されることを示しています。ユーザーのアクションは必要ありません。

次の例は、一般的なイベント構造を示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "region": "string",
```

```

"time": "yyyy-mm-ddThh:mm:ssZ",
"resources": []
"detail": {
  "version": "0",
  "mediaInsightsPipelineArn": "string",
  "eventType": "chime:MediaInsightsTemporaryFailure",
  "callId": "string",
  "transactionId": "string",
  "fromNumber": "string",
  "toNumber": "string",
  "voiceConnectorId": "string",
  "direction": "string"
}
}

```

Amazon Chime SDK メディアインサイトに永続的な障害が発生している

ユーザーのアクションが必要な障害を示しています。failureReason に基づいて、トラブルシューティングを行います。一般的な障害として、次の状況が挙げられます。

- リソースアクセスロールの権限が不足している
- リソースが見つからないか、削除されている
- Amazon Transcribe や Amazon Kinesis など、分析を呼び出す AWS サービスからのスロットリング。
- KVS ストリームのメディア形式に互換性がない

次の例は、一般的なイベント構造を示しています。

```

{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "region": "string",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail": {
    "version": "0",
    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsPermanentFailure",

```



```
"callId": "string",
"transactionId": "string",
"fromNumber": "string",
"toNumber": "string",
"voiceConnectorId": "string",
"direction": "string",
"failureReason": "string"
}
}
```

Note

failureReason フィールドはオプションです。一般的な理由として Access denied when assuming resource access role が示されることがあります。

Amazon Chime SDK Voice Connector で開始した通話分析セッションでは、メディアインサイトパイプラインが作成されるか、作成の試行が失敗すると、次のイベントタイプが示される場合があります。詳細については、各セクションを展開してください。

Amazon Chime SDK メディアインサイトが作成された

この例は、一般的な正常終了イベントを示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "region": "string",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail":
  {
    "version": "0",
    "mediaInsightsPipelineConfigurationArn": "string",
    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsCreated",
    "callId": "string",
    "transactionId": "string",
    "fromNumber": "string",
```

```
    "toNumber": "string",
    "voiceConnectorId": "string",
    "direction": "string",
  }
}
```

Amazon Chime SDK メディアインサイトの作成に失敗した

この例は、一般的な障害イベントを示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "region": "string",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": []
  "detail":
  {
    "version": "0",
    "mediaInsightsPipelineConfigurationArn": "string",
    "eventType": "chime:MediaInsightsCreateFailed",
    "callId": "string",
    "transactionId": "string",
    "fromNumber": "string",
    "toNumber": "string",
    "voiceConnectorId": "string",
    "direction": "string",
    "failureOrigin": "Voice Connector",
    "httpStatusCode": "string",
    "failureReason": "string"
  }
}
```

メディアインサイトパイプライン内に複数の要素がある場合は、次のイベントタイプが示されることがあります。この例は、AmazonTranscribeProcessor と S3RecordingSink を連携させた場合の通知を示しています。詳細については、各セクションを展開してください。

AmazonTranscribeProcessor は進行中で、S3RecordingSink は起動していません

次の例は、一般的なイベント構造を示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "region": "string",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": [],
  "detail": {
    "version": "0",
    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsInProgress",
    "mediaInsightsPipelineElementStatuses": [
      {
        "type": "AmazonTranscribeProcessor",
        "status": "InProgress",
        "updatedOn": 1686184070655
      },
      {
        "type": "S3RecordingSink",
        "status": "NotStarted",
        "updatedOn": 1686184070655
      }
    ]
  },
  "callId": "string",
  "transactionId": "string",
  "fromNumber": "string",
  "toNumber": "string",
  "voiceConnectorId": "string",
  "direction": "string"
}
```

AmazonTranscribeProcessor は成功し、S3RecordingSink は進行中です

次の例は、一般的なイベント構造を示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
```

```
"account": number,
"region": "string",
"time": "yyyy-mm-ddThh:mm:ssZ",
"resources": [],
"detail": {
  "version": "0",
  "mediaInsightsPipelineArn": "string",
  "eventType": "chime:MediaInsightsInProgress",
  "mediaInsightsPipelineElementStatuses": [
    {
      "type": "AmazonTranscribeProcessor",
      "status": "Stopped",
      "updatedOn": 1686184070655
    },
    {
      "type": "S3RecordingSink",
      "status": "InProgress",
      "updatedOn": 1686184070655
    }
  ]
}
"callId": "string",
"transactionId": "string",
"fromNumber": "string",
"toNumber": "string",
"voiceConnectorId": "string",
"direction": "string"
}
```

AmazonTranscribeProcessor が失敗し、S3RecordingSink が進行中である

次の例は、一般的なイベント構造を示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "region": "string",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": [],
  "detail": {
    "version": "0",
```

```

    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsInProgress",
    "mediaInsightsPipelineElementStatuses": [
      {
        "type": "AmazonTranscribeProcessor",
        "status": "Failed",
        "updatedOn": 1686184070655
      },
      {
        "type": "S3RecordingSink",
        "status": "InProgress",
        "updatedOn": 1686184070655
      }
    ]
    "callId": "string",
    "transactionId": "string",
    "fromNumber": "string",
    "toNumber": "string",
    "voiceConnectorId": "string",
    "direction": "string"
  }
}

```

AmazonTranscribeProcessor が失敗し、S3RecordingSink が成功しました

次の例は、一般的なイベント構造を示しています。

```

{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "region": "string",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": [],
  "detail": {
    "version": "0",
    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsPermanentFailure",
    "mediaInsightsPipelineElementStatuses": [
      {
        "type": "AmazonTranscribeProcessor",
        "status": "Failed",

```

```
        "updatedOn": 1686184070655
      },
      {
        "type": "S3RecordingSink",
        "status": "Stopped",
        "updatedOn": 1686184070655
      }
    ]
    "callId": "string",
    "transactionId": "string",
    "fromNumber": "string",
    "toNumber": "string",
    "voiceConnectorId": "string",
    "direction": "string",
    "failureReason": "string"
  }
}
```

AmazonTranscribeProcessor は成功し、S3RecordingSink は失敗しました

次の例は、一般的なイベント構造を示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "region": "string",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": [],
  "detail": {
    "version": "0",
    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsPermanentFailure",
    "mediaInsightsPipelineElementStatuses": [
      {
        "type": "AmazonTranscribeProcessor",
        "status": "Stopped",
        "updatedOn": 1686184070655
      },
      {
        "type": "S3RecordingSink",
        "status": "Failed",

```

```
        "updatedOn": 1686184070655
      }
    ]
    "callId": "string",
    "transactionId": "string",
    "fromNumber": "string",
    "toNumber": "string",
    "voiceConnectorId": "string",
    "direction": "string",
    "failureReason": "string"
  }
}
```

AmazonTranscribeProcessor は一時停止し、S3RecordingSink は起動しません

次の例は、一般的なイベント構造を示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "region": "string",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": [],
  "detail": {
    "version": "0",
    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsPaused",
    "mediaInsightsPipelineElementStatuses": [
      {
        "type": "AmazonTranscribeProcessor",
        "status": "Paused",
        "updatedOn": 1686184070655
      },
      {
        "type": "S3RecordingSink",
        "status": "NotStarted",
        "updatedOn": 1686184070655
      }
    ]
  }
  "callId": "string",
  "transactionId": "string",
}
```

```
    "fromNumber": "string",
    "toNumber": "string",
    "voiceConnectorId": "string",
    "direction": "string"
  }
}
```

AmazonTranscribeProcessor が一時的に失敗し、S3RecordingSink が起動していない

次の例は、一般的なイベント構造を示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "region": "string",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": [],
  "detail": {
    "version": "0",
    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsTemporaryFailure",
    "mediaInsightsPipelineElementStatuses": [
      {
        "type": "AmazonTranscribeProcessor",
        "status": "TemporarilyFailed",
        "updatedOn": 1686184070655
      },
      {
        "type": "S3RecordingSink",
        "status": "NotStarted",
        "updatedOn": 1686184070655
      }
    ]
  },
  "callId": "string",
  "transactionId": "string",
  "fromNumber": "string",
  "toNumber": "string",
  "voiceConnectorId": "string",
  "direction": "string"
}
```



```
}
```

AmazonTranscribeProcessor および S3RecordingSink 成功

次の例は、一般的なイベント構造を示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "region": "string",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": [],
  "detail": {
    "version": "0",
    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsStopped",
    "mediaInsightsPipelineElementStatuses": [
      {
        "type": "AmazonTranscribeProcessor",
        "status": "Stopped",
        "updatedOn": 1686184070655
      },
      {
        "type": "S3RecordingSink",
        "status": "Stopped",
        "updatedOn": 1686184070655
      }
    ]
  },
  "callId": "string",
  "transactionId": "string",
  "fromNumber": "string",
  "toNumber": "string",
  "voiceConnectorId": "string",
  "direction": "string"
}
```

S3RecordingSink が成功し VoiceEnhancement、進行中

次の例は、一般的なイベント構造を示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "string",
  "detail": {
    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsInProgress",
    "version": "0",
    "mediaInsightsPipelineElementStatuses": [
      {
        "type": "VoiceEnhancementSink",
        "status": "InProgress",
        "updatedOn": 1686184070655
      },
      {
        "type": "S3RecordingSink",
        "status": "Stopped",
        "updatedOn": 1686184070655
      }
    ]
  }
}
```

30 分を超える呼び出しにより S3RecordingSink3 が成功し VoiceEnhancement、失敗しました
次の例は、一般的なイベント構造を示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "string",
  "detail": {
    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsStopped",
    "version": "0",
```

```

"mediaInsightsPipelineElementStatuses": [
  {
    "type": "VoiceEnhancement",
    "status": "NotSupported",
    "updatedOn": 1686184070655,
    "statusDetail": "Unsupported recording length"
  },
  {
    "type": "S3RecordingSink",
    "status": "Stopped",
    "updatedOn": 1686184070655
  }
]
}
}

```

呼び出しが 30 分未満であるため、S3RecordingSink3 は成功し、VoiceEnhancement 失敗しました。次の例は、一般的なイベント構造を示しています。

```

{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights State Change",
  "source": "aws.chime",
  "account": number,
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "string",
  "detail": {
    "mediaInsightsPipelineArn": "string",
    "eventType": "chime:MediaInsightsPermanentFailure",
    "version": "0",
    "mediaInsightsPipelineElementStatuses": [
      {
        "type": "VoiceEnhancement",
        "status": "Failed",
        "updatedOn": 1686184070655
      },
      {
        "type": "S3RecordingSink",
        "status": "Stopped",
        "updatedOn": 1686184070655
      }
    ]
  }
}

```

```
}  
}
```

リアルタイムアラート

Note

Amazon Transcribe と Amazon Transcribe Call Analytics のプロセッサのみがリアルタイムアラートに対応しています。

Amazon Chime SDK 通話分析では、分析セッション中にプロセッサを介して、リアルタイムアラートの送信ルールを設定できます。アラートは、詳細タイプ EventBridge で Amazon に送信されます。Media Insights Rules Matched。は、Lambda、Amazon SQS、Amazon SNS などのダウンストリームサービスとの統合 EventBridge をサポートし、エンドユーザーへの通知をトリガーしたり、その他のカスタムビジネスロジックを開始したりできます。

リアルタイムアラートは、MediaInsightsPipelineConfiguration の RealTimeAlertConfiguration フィールドの一部として設定します。Amazon Chime SDK コンソールを使用してフィールドを設定するか、[CreateMediaInsightsPipelineConfiguration](#) または [UpdateMediaInsightsPipelineConfiguration](#) APIs を呼び出すことができます。

次の例は、API を使用して、リアルタイムアラートを設定または更新する方法を示しています。

```
{  
  "MediaInsightsPipelineConfigurationName": "config_name",  
  "ResourceAccessRoleArn": "arn:aws:iam::account_id:role/role_name",  
  "RealTimeAlertConfiguration": {  
    "Disabled": false,  
    "Rules": [{  
      "Type": "KeywordMatch",  
      "KeywordMatchConfiguration": {  
        "RuleName": "rule_name_1",  
        "Keywords": [  
          "hello",  
          "thank you"  
        ],  
        "Negate": false  
      }  
    }  
  ],  
}
```

```
{
  {
    "Type": "Sentiment",
    "RuleName": "rule_name_2",
    "SentimentType": "NEGATIVE",
    "TimePeriod": 60
  },
  {
    "Type": "IssueDetection",
    "RuleName": "rule_name_3"
  }
]
},
"Elements": [{
  "Type": "AmazonTranscribeCallAnalyticsProcessor",
  "AmazonTranscribeCallAnalyticsProcessorConfiguration": {
    "LanguageCode": "en-US"
  }
},
{
  "Type": "KinesisDataStreamSink",
  "KinesisDataStreamSinkConfiguration": {
    "InsightsTarget": "arn:aws:kinesis:us-
east-1:account_id:stream/stream_name"
  }
}
]
```

リアルタイムアラート設定の各ルールは、個別にトリガーされます。複数のルール条件が同時に満たされた場合、複数の EventBridge 通知を受け取ることがあります。アラートルールリストの作成では、次のルールタイプを選択できます。

キーワードの一致

指定したキーワードまたはフレーズのセットが発話イベントまたは文字起こしイベントで一致したときにアラートが送信されます。次の場合にイベントが発行されるよう、アラートを設定できます。

- Negate を false に設定しており、指定したキーワードが発話された。
- Negate を true に設定しており、指定したいずれのキーワードも通話中に発話されなかった。

Amazon Transcribe と Amazon Transcribe Analytics は、このルールタイプに対応しています。

センチメント分析

特定のセンチメントタイプが、ローリングウィンドウの期間継続した場合にアラートが発行されます。Transcribe Call Analytics のみ、このルールに対応しています。

問題検出

発話イベントで、特定の問題が検出されるとアラートが発行されます。Transcribe Call Analytics のみ、このルールタイプに対応しています。

次の例は、KeywordMatch ルールのリアルタイムアラートイベントを示しています。

```
{
  "version": "0",
  "id": "string",
  "detail-type": "Media Insights Rules Matched",
  "source": "aws.chime",
  "account": number,
  "region": "us-east-1",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "resources": [],
  "detail": {
    "version": "0",
    "sourceMetadata": {}
    "ruleName": "string"
    "utteranceId": "string",
    "beginTimestamp": "yyyy-mm-ddThh:mm:ssZ",
  }
}
```

一部の EventBridge フィールドは、一致するルールタイプに固有です。

キーワード一致のフィールド

utteranceId: 一致するキーワードが含まれる文字起こし文の ID (Amazon Transcribe Call Analytics を使用する場合)。発話されたキーワードの一致のみが対象。

resultId: 一致するキーワードが含まれる文字起こし文の ID (Amazon Transcribe を使用する場合)。発話されたキーワードの一致のみが対象。

beginTimestamp: 一致したキーワードが含まれる文字起こし文の開始時間。発話されたキーワードの一致のみが対象。

センチメント分析のフィールド

`beginTimestamp`: 一致したセンチメントのローリングウィンドウが開始した時間。

`endTimestamp`: 一致したセンチメントのローリングウィンドウが終了した時間。

Amazon Chime SDK データレイクを作成する

Amazon Chime SDK 通話分析データレイクを使用すると、機械学習を活用したインサイトやメタデータを Amazon Kinesis Data Streams から Amazon S3 バケットにストリーミングできます。例えば、データレイクを使用して、記録用の URL にアクセスすることが可能です。データレイクを作成するには、Amazon Chime SDK コンソールから、または `awscli` を使用してプログラムで一連の AWS CloudFormation テンプレートをデプロイします AWS CLI。データレイクを使用すると、Amazon Athena の AWS Glue データテーブルを参照して、通話メタデータと音声分析データをクエリできます。

トピック

- [前提条件](#)
- [データレイクの用語と概念](#)
- [複数のデータレイクを作成する](#)
- [リージョン別に見たデータレイクの可用性](#)
- [データレイクのアーキテクチャ](#)
- [データレイクをセットアップする](#)

前提条件

Amazon Chime SDK データレイクを作成するには、次の機能が必要です。

- 1 つの Amazon Kinesis Data Streams。詳細については、「Amazon Kinesis Streams Developer Guide」の「[Creating a Stream via the AWS Management Console](#)」を参照してください。
- S3 バケット 詳細については、「Amazon S3 ユーザーガイド」の「[最初の S3 バケットを作成する](#)」を参照してください。

データレイクの用語と概念

データレイクの仕組みを理解するには、次の用語と概念を参考にしてください。

Amazon Kinesis Data Firehose

抽出、変換、ロード (ETL) サービス。これにより、ストリーミングデータをデータレイク、データストア、分析サービスに、確実にキャプチャ、変換、配信することが可能です。詳細については、「Amazon Kinesis Data Firehose とは何ですか?」を参照してください。

Amazon Athena

Amazon Athena は、標準 SQL を使用して Amazon S3 のデータを分析できるインタラクティブなクエリサービスです。Athena は、サーバーレスであるため、インフラストラクチャを管理する必要がなく、料金は実行したクエリに対してのみ発生します。Athena を利用するには、Amazon S3 内のデータを指定してスキーマを定義し、標準 SQL クエリを使用します。また、ワークグループでユーザーをグループ化し、クエリの実行時にユーザーがアクセスできるリソースを制御することもできます。ワークグループを使用すると、クエリの同時実行を管理して、さまざまなユーザーグループやワークロードによるクエリ実行に優先順位を付けることが可能です。

Glue Data Catalog

Amazon Athena のテーブルとデータベースには、基礎となるソースデータのスキーマを詳細に示すメタデータが格納されています。また、Athena では、データセットごとにテーブルが存在する必要があります。このテーブル内のメタデータによって Amazon S3 バケットの場所が判断されます。また、列名、データ型、テーブル名などのデータ構造も指定されます。データベースで保持されるのは、データセットのメタデータとスキーマ情報のみです。

複数のデータレイクを作成する

一意の Glue データベース名を指定して通話のインサイトを保存する場所を指定することで、複数のデータレイクを作成できます。特定のアカウントには AWS、対応するデータレイクを持つ複数の通話分析設定があります。つまり、特定のユースケースにデータ分離を適用することが可能です。例えば、保持ポリシーや、データの保存方法に関するアクセスポリシーなどをカスタマイズできます。インサイト、録音データ、メタデータへのアクセスに、さまざまなセキュリティポリシーを適用することも可能です。

リージョン別に見たデータレイクの可用性

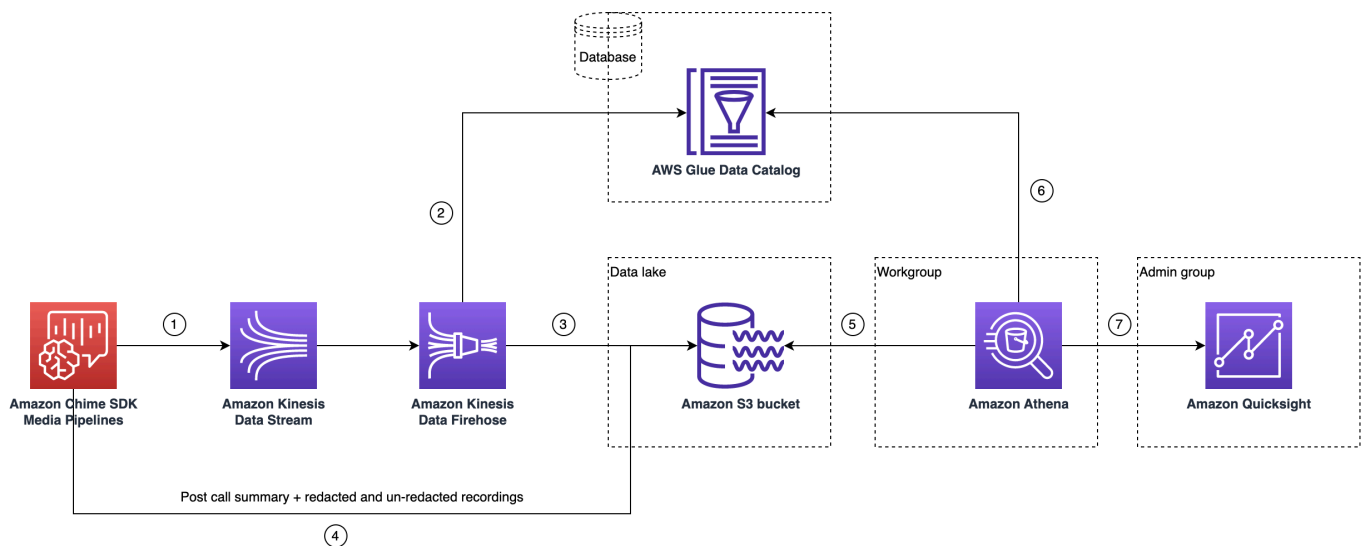
Amazon Chime SDK データレイクは、次のリージョンで利用できます。

リージョン	Glue テーブル	Amazon QuickSight
us-east-1	利用可能	利用可能

リージョン	Glue テーブル	Amazon QuickSight
us-west-2	利用可能	利用可能
eu-central-1	利用可能	利用可能

データレイクのアーキテクチャ

データレイクのアーキテクチャを次の図に示します。図の数字は、次の各説明の番号に対応しています。



この図では、AWS コンソールを使用してメディアインサイトパイプライン設定ワークフローから CloudFormation テンプレートをデプロイすると、次のデータが Amazon S3 バケットに流れます。

1. Amazon Chime SDK 通話分析によって、顧客の Kinesis Data Stream に対しリアルタイムデータのストリーミングが開始されます。
2. Amazon Kinesis Firehose では、128 MB まで蓄積される、または 60 秒が経過するまで (どちらか早い方が適用される)、このリアルタイムデータをバッファリングします。次の Firehose では、Glue データカタログの `amazon_chime_sdk_call_analytics_firehose_schema` を使用して、データを圧縮し、JSON レコードを Parquet 形式のファイルに変換します。
3. Parquet ファイルは、パーティション形式で Amazon S3 バケットに格納されます。
4. リアルタイムデータの他に、Amazon Transcribe Call Analytics の通話後概要データ .wav ファイル (設定で指定している場合、編集済みと未編集のファイル) と、通話録音の .wav ファイルも Amazon S3 バケットに送信されます。

5. Amazon Athena と標準 SQL を使用して、Amazon S3 バケット内のデータをクエリすることができます。
6. CloudFormation テンプレートは、Athena を介してこの通話後サマリーデータをクエリするための Glue データカタログも作成します。
7. Amazon S3 バケット上のすべてのデータは、Amazon を使用して視覚化することもできます QuickSight。Amazon Athena を使用して Amazon S3 バケットとの接続 QuickSight を構築します。

Amazon Athena テーブルでは、次の機能を使用して、クエリのパフォーマンスを最適化します。

データのパーティション化

パーティショニングを行うと、テーブルが複数部分に分割され、日付、国、地域などの列の値に基づいて関連データがまとめられます。パーティションは仮想の列として機能します。この場合、CloudFormation テンプレートはテーブルの作成時にパーティションを定義します。これにより、クエリごとにスキャンされるデータ量を減らし、パフォーマンスを向上させることができます。パーティションでフィルタリングして、クエリの際にスキャンするデータの量を制限することもできます。詳細については、「Amazon Athena ユーザーガイド」の「[Athena でのデータのパーティション化](#)」を参照してください。

次の例は、2023 年 1 月 1 日の日付を指定したパーティション構造を示しています。

i.

```
s3://example-bucket/amazon_chime_sdk_data_lake
                               /serviceType=CallAnalytics/detailType={DETAIL_TYPE}/
year=2023
                               /month=01/day=01/example-file.parquet
```

ii. `DETAIL_TYPE` には次のいずれかを指定します。

- a. `CallAnalyticsMetadata`
- b. `TranscribeCallAnalytics`
- c. `TranscribeCallAnalyticsCategoryEvents`
- d. `Transcribe`
- e. `Recording`
- f. `VoiceAnalyticsStatus`
- g. `SpeakerSearchStatus`
- h. `VoiceToneAnalysisStatus`

列指向データストアの生成を最適化する

Apache Parquet では、列単位の圧縮、データ型に基づく圧縮、述語プッシュダウンを使用してデータを保存します。圧縮率を上げたり、データブロックをスキップしたりすると、Amazon S3 バケットから読み取るバイト数が減少するため、クエリのパフォーマンスが向上すると共に、コストが削減されます。こうした最適化のために、Amazon Kinesis Data Firehose では、JSON からパーケットへのデータ変換が有効になっています。

パーティション射影

Athena のこの機能では、パーティションを日ごとに自動作成することで、日付ベースのクエリのパフォーマンスを向上させます。

データレイクをセットアップする

Amazon Chime SDK コンソールを使用して、次の手順を完了します。

1. Amazon Chime SDK コンソール (<https://console.aws.amazon.com/chime-sdk/home>) を起動し、ナビゲーションペインの [通話分析] で [設定] を選択します。
2. [ステップ 1] を完了して [次へ] を選択し、[ステップ 2] ページの [音声分析] チェックボックスを選択します。
3. 出力の詳細 で、履歴分析を実行するデータウェアハウス チェックボックスを選択し、CloudFormation スタックのデプロイ リンクを選択します。

コンソールのスタックのクイック作成ページに、システムから送信されます CloudFormation 。

4. スタックの名前を入力し、次のパラメータを入力します。
 - a. DataLakeType – コール分析の作成 DataLakeを選択します。
 - b. KinesisDataStreamName — 対象のストリームを選択します。通話分析ストリーミングに使用するストリームを指定する必要があります。
 - c. S3BucketURI — Amazon S3 バケットを選択します。URI にはプレフィックス `s3://bucket-name` が必要です。
 - d. GlueDatabaseName – 一意の AWS Glue データベース名を選択します。AWS アカウント内の既存のデータベースは再利用できません。
5. 確認のチェックボックスを選択し、[データレイクを作成] を選択します。データレイクの作成には、10 分かかります。

を使用したデータレイクのセットアップ AWS CLI

を使用して AWS CLI、 の create スタックを呼び CloudFormation出すアクセス許可を持つロールを作成します。次の手順に従って IAM ロールを作成しセットアップします。詳細については、「[ユーザーガイド](#)」の「[スタックの作成AWS CloudFormation](#)」を参照してください。

1. AmazonChimeSdkCallAnalytics-Datalake-Provisioning-Role というロールを作成し、ロールを引き受けることができる信頼ポリシーをロール CloudFormation にアタッチします。

1. 次のテンプレートを使用して IAM 信頼ポリシーを作成し、そのファイルを .json 形式で保存します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

2. aws iam create-role コマンドを実行し、信頼ポリシーをパラメータとして渡します。

```
aws iam create-role \
  --role-name AmazonChimeSdkCallAnalytics-Datalake-Provisioning-Role
  --assume-role-policy-document file://role-trust-policy.json
```

3. レスポンスで返ったロール ARN を書き留めておきます。ロール ARN は次の手順で必要になります。
2. CloudFormation スタックを作成するアクセス許可を持つポリシーを作成します。
 1. 次のテンプレートを使用して IAM ポリシーを作成し、そのファイルを .json 形式で保存します。create-policy を呼び出すときに、このファイルが必要となります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DeployCloudFormationStack",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack"
      ],
      "Resource": "*"
    }
  ]
}
```

2. `aws iam create-policy` を実行し、スタック作成のポリシーをパラメータとして渡します。

```
aws iam create-policy --policy-name
testCreateStackPolicy
--policy-document file://create-cloudformation-stack-policy.json
```

3. レスポンスで返ったロール ARN を書き留めておきます。ロール ARN は次の手順で必要になります。

3. `aws iam attach-role-policy` ポリシーをロールにアタッチします。

```
aws iam attach-role-policy --role-name {Role name
created above}
--policy-arn {Policy ARN created above}
```

4. CloudFormation スタックを作成し、必要なパラメータを入力します `aws cloudformation create-stack`。

`ParameterKey` を使用して、各のパラメータ値を指定します `ParameterValue`。

```
aws cloudformation create-stack --capabilities
CAPABILITY_NAMED_IAM
--stack-name testDeploymentStack
```

```
--template-url https://chime-sdk-assets.s3.amazonaws.com/public_templates/
AmazonChimeSDKDataLake.yaml
--parameters ParameterKey=S3BucketURI,ParameterValue={S3 URI}
ParameterKey=DataLakeType,ParameterValue="Create call analytics datalake"
ParameterKey=KinesisDataStreamName,ParameterValue={Name of Kinesis Data Stream}
--role-arn {Role ARN created above}
```

データレイクのセットアップによって作成されるリソース

データレイク作成によって作成されるリソースを次の表に示します。

リソースタイプ	リソース名と説明	サービス名
AWS Glue データカタログデータベース	GlueDatabase名前 — 通話インサイトと音声分析に属するすべての AWS Glue データテーブルを論理的にグループ化します。	通話分析、音声分析
AWS Glue データカタログテーブル	amazon_chime_sdk_call_analytics_firehose_schema — Kinesis Firehose に送られる通話分析と音声分析用の複合スキーマ。	通話分析、音声分析
	call_analytics_metadata — 通話分析メタデータのスキーマ。SIPmetadataとが含まれます OneTimeMetadata。	コール分析
	call_analytics_recording_metadata — 録音および音声エンハンスメントメタデータのスキーマ。	通話分析、音声分析
	transcribe_call_analytics – TranscribeCallAnalytics ペイロードのスキーマutteranceEvent」	コール分析
	transcribe_call_analytics_category_events – TranscribeCallAnalytics ペイロードのスキーマ "categoryEvent"	コール分析
	transcribe_call_analytics_post_call – Transcribe の通話後分析概要ペイロードに使用するスキーマ。	コール分析
	transcribe – Transcribe ペイロードのスキーマ。	コール分析

リソースタイプ	リソース名と説明	サービス名
	voice_analytics_status – 音声分析の準備状況を示すイベントのスキーマ。	音声分析
	speaker_search_status – 識別一致に使用するスキーマ。	音声分析
	voice_tone_analysis_status – ボイストーン分析イベントのスキーマ。	音声分析
Amazon Kinesis Data Firehose	AmazonChimeSDK-call-analytics- UUID – 通話分析用の Kinesis Data Firehose パイプデータ	通話分析、音声分析
Amazon Athena ワークグループ	GlueDatabaseName-AmazonChimeSDKDataAnalytics – クエリの実行時にアクセスできるリソースを制御する論理的なユーザーのグループ。	通話分析、音声分析

Amazon QuickSight ダッシュボードの設定

データレイクを設定したら、データを視覚化する事前定義されたメトリクスを使用して Amazon QuickSight ダッシュボードを設定できます。利用可能なダッシュボードを次に示します。

- Transcribe Call Analytics と音声分析。メトリクスには、turn-by-turn トランスクリプト、検出された問題、結果、エンティティ検出、音声プロファイル ID の一致の概要と詳細なビジュアルが含まれます。
- Transcribe と音声分析。メトリクスには、turn-by-turn トランスクリプト、語彙一致、音声トーン、音声プロファイル ID 一致の概要と詳細ビジュアルが含まれます。

以下のトピックでは、Amazon QuickSight アカウントをまだお持ちでない場合は設定する方法と、ダッシュボードを設定する方法について説明します。

トピック

- [QuickSight アカウントの作成](#)
- [QuickSight アカウントの設定](#)
- [QuickSight ダッシュボードの作成](#)

QuickSight アカウントの作成

このセクションのステップでは、Amazon QuickSight アカウントを作成する方法について説明します。アカウントがある場合は、この手順をスキップして [QuickSight ダッシュボードの作成](#) に進んでください。

QuickSight アカウントは、次の方法で作成できます。

- Amazon CloudFormation テンプレートの使用。
- Amazon Chime SDK コンソール。

前提条件

アカウントの作成前に、次の情報を収集します。

- 通話分析に使用する Amazon S3 バケットの名前。
- 通知用のメールアドレス。システムは、このアドレス QuickSight に通知を送信します。

CloudFormation テンプレートを使用したアカウントの作成

次の手順では、Amazon CloudFormation テンプレートをデプロイして Amazon QuickSight アカウントを作成する方法について説明します。このプロセスでは、Enterprise アカウントのみ、サブスクライブします。料金の詳細については、[「Amazon の QuickSight 料金」](#)を参照してください。

テンプレートをデプロイするには

1. AWS コンソールを起動し、AWS アカウントにログインします。
2. 次の URL をブラウザのアドレスバーに貼り付けます。表示どおりに地域を入力したことを確認してください。

```
https://region.console.aws.amazon.com/cloudformation/home?
region=region#/stacks/quickcreate?templateURL=https://
chime-sdk-assets.s3.amazonaws.com/public_templates/
AmazonChimeSDKQuickSightSubscription.yaml.
```

3. [スタックのクイック作成] ページで、次のように入力します。
 - a. [スタック名] に、アカウントの名前を入力します。
 - b. 前に収集した E メールアドレス QuickSightNotificationEmail の下。

- c. でQuickSightSubscriptionForDataVisualization、新しい AWS QuickSight アカウントの作成 を選択します。
 - d. S3BucketName で、Amazon S3 バケットの名前を入力します。
 - e. 「AWS が IAM リソースを作成する CloudFormation 可能性があることを承認します。」 チェックボックスをオンにします。
4. [スタックの作成] を選択します。

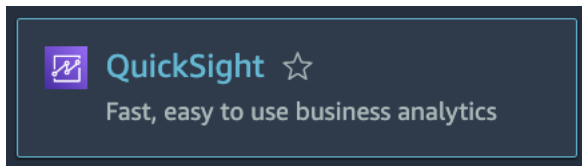
スタックの作成には約 10 分かかります。
 5. ビルドが完了したら、Amazon に移動 QuickSightを選択し、E メールアドレスを入力してアカウントにサインインします。

コンソールを使用してアカウントを作成する

次の手順では、Amazon Chime SDK コンソールを使用して Amazon QuickSight アカウントを作成する方法について説明します。ここでは、エンタープライズ版またはエンタープライズ版 + Q のアカウントを使用する必要があります。

コンソールを使用するには

1. <https://console.aws.amazon.com/chime-sdk/home> で Amazon Chime SDK コンソールを起動し、 を検索し**QuickSight**、検索結果で を選択しますQuickSight。



2. にサインアップを選択します QuickSight。
3. [エンタープライズ版] または [エンタープライズ版 + Q] を選択し、[次へ] を選択します。
4. 氏名、電話番号、先ほど収集したメールアドレスを入力して、[次へ] を選択します。
5. 以下の操作を実行します。
 - i. [認証方法] で、オプションを選択します。

Note

フェデレーションユーザーを使用するオプションの選択には、適切な IAM アクセス許可が必要です。詳細については、[「Amazon ユーザーガイド」の「Amazon](#)

[QuickSight サブスクリプションにサインアップする](#)を参照してください。

QuickSight

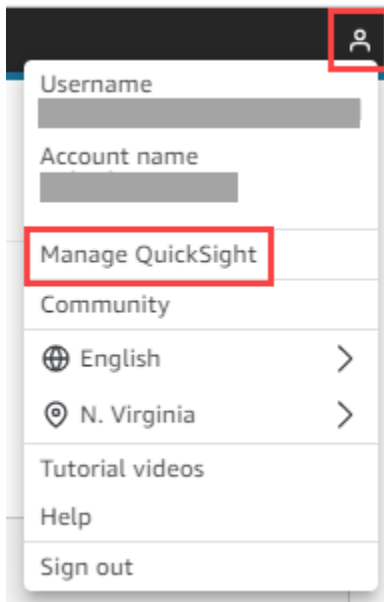
- ii. QuickSight リージョン で、リージョンを選択します。
- iii. [アカウント情報] に、アカウントの名前と、先ほど収集したメールアドレスを入力します。
- iv. QuickSight AWS Services へのアクセスで、デフォルトのロールを使用するか、既存のロールを使用してリストからロールを選択します。
- v. (オプション) 必要に応じて、[これらのリソースへのアクセスと自動検出を許可] で、追加のリソースを選択します。
- vi. 完了したら、[完了] を選択します。
- vii.ビルドが完了したら、Amazon に移動 QuickSightを選択し、E メールアドレスを入力してアカウントにサインインします。

QuickSight アカウントの設定

QuickSight アカウントにログインしたら、セキュリティを設定し、セットアッププロセスによって作成されたグループに自分自身を追加する必要があります。

セキュリティを設定するには

1. 右上隅にあるプロフィールアイコンを選択し、結果のメニューから管理 QuickSightを選択します。



2. ナビゲーションペインで、[セキュリティとアクセス権限] を選択します。

3. QuickSight AWS サービス へのアクセスで、 の管理 を選択し、次のサービスが選択されていることを確認します。
 - Amazon Redshift
 - Amazon RDS
 - Amazon S3
 - Amazon Athena
 - IAM
4. [Amazon S3 バケットを選択] リンクを選択します。
5. Amazon S3 バケットの横にあるチェックボックスを選択し、[Athena Workgroup の書き込みアクセス許可] 列の右側にあるチェックボックスを選択します。
6. [Finish] を選択します。
7. [保存] を選択します。

自分をグループに追加するには

1. ナビゲーションペインで、[グループの管理] を選択し、名前に [管理者] が含まれるグループを選択します。例えば、S3BucketName-Admins などです。
2. [ユーザーの追加] を選択し、表示されているボックスにメールエイリアスを入力します。

自分の名前が次のように表示されます: [管理者 -] 自分のエイリアス。

3. [追加] を選択します。

QuickSight ダッシュボードの作成

データレイクを作成したら、データを視覚化する QuickSight ダッシュボードを作成できます。Amazon CloudFormation テンプレートまたは Amazon Chime SDK コンソールを使用してダッシュボードを作成できます。次の手順は、その 2 つの方法を示しています。

テンプレートを使用するには

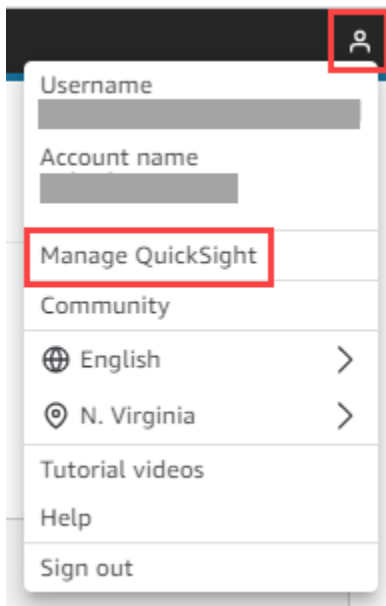
1. Amazon CloudFormation コンソールを起動します。
2. 次のリンクをブラウザのアドレスバーに貼り付けます。
`https://region.console.aws.amazon.com/cloudformation/home?region=region#/stacks/quickcreate?templateURL=https://`

```
chime-sdk-assets.s3.amazonaws.com/public_templates/  
AmazonChimeSDKQuickSightDashboards.yaml
```

3. [スタックのクイック作成] ページの [スタック名] に、アカウントの名前を入力します。
4. でActiveQuickSightAccount、True を選択します。
5. QuicksightDashboard選択 で、通話分析 – 通話分析と音声分析ダッシュボード または通話分析 – 通話分析ダッシュボード を選択します。
6. Amazon S3BucketNameで、Amazon S3 バケットの URI を入力します。
7. GlueDatabase名前に、QuickSight ダッシュボードをデプロイする Glue データベースを入力します。
8. 「AWS が IAM リソースを作成する CloudFormation 可能性があることを承認」チェックボックスを選択し、「スタックの作成」を選択します。

QuickSight ダッシュボードを手動で設定するには

1. QuickSight アカウントに移動します。
2. 右上隅でプロフィールアイコンを選択し、 の管理を選択します QuickSight。



3. ナビゲーションペインで [グループの管理] を選択し、セットアッププロセスで作成したグループを選択します。
4. [ユーザーの追加] を選択してメールアドレスを入力し、[追加] を選択します。

ページのデプロイには 10 分かかります。

5. Amazon Chime SDK コンソールを使用して QuickSight アカウントにログインし、ダッシュボードを使用します。

通話分析のデータモデル

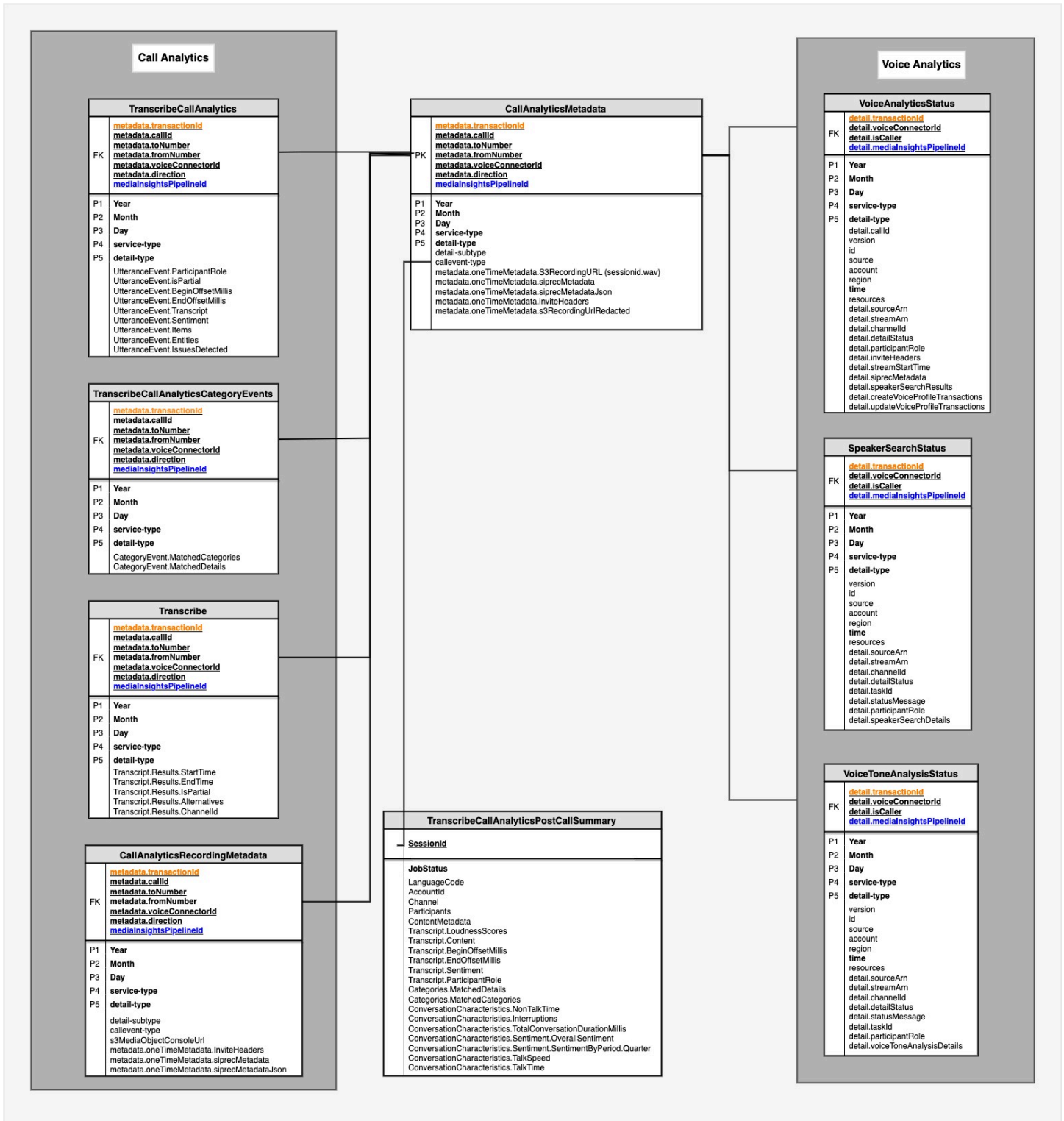
このセクションには、AWS Glue データカタログのテーブルセットである Amazon Chime SDK 通話分析データモデルの情報と、その説明が記載されています。

トピック

- [Glue データカタログのテーブル構造](#)
- [Glue データカタログのテーブル](#)
- [サンプルクエリ](#)

Glue データカタログのテーブル構造

次の図は、Amazon Chime SDK 通話分析および音声分析セッション用に作成された AWS Glue データカタログのテーブル構造を示しています。



次のセクションでは、カタログ内のテーブルとフィールドの一覧を示し、それぞれについて説明します。

Glue データカタログのテーブル

次の表には、Amazon Chime SDK 通話分析の Glue データカタログの列、データ型、要素の一覧と説明が記載されています。

トピック

- [call_analytics_metadata](#)
- [call_analytics_recording_metadata](#)
- [transcribe_call_analytics](#)
- [transcribe_call_analytics_category_events](#)
- [transcribe_call_analytics_post_call](#)
- [transcribe](#)
- [voice_analytics_status](#)
- [speaker_search_status](#)
- [voice_tone_analysis_status](#)

call_analytics_metadata

列名	データ型	[Elements] (要素)	定義
time	string		イベント生成のタイムスタンプ (ISO 8601 形式)。
detail-type (ディテールタイプ)	string		service-type に関連する特徴量タイプ。
service-type	string		AWS サービスの名前、VoiceAnalytics または CallAnalytics。
detail-subtype	string		録画タイプと CallAnalyticsMetadata

列名	データ型	[Elements] (要素)	定義
			詳細タイプに使用されます。
callevent-type	string		SIP に関連するイベントタイプ (更新、一時停止、再開など)
mediaInsightsPipelineID	string		Amazon Chime SDK メディアインサイトパイプライン ID。
metadata	string	voiceConnectorId	Amazon Chime SDK Voice Connector ID。
		callId	参加者の通話 ID (関連する使用向け)。
		transactionId	通話のトランザクション ID。
		fromNumber	発信元の電話番号 (E.164 形式)。
		toNumber	送信先の電話番号 (E.164 形式)。
		direction	通話の方向 (Outbound または Inbound)。
		oneTimeMetadata.s3RecordingUrl	Transcribe Call Analytics によって出力されたメディアオブジェクトの Amazon S3 バケット URL。

列名	データ型	[Elements] (要素)	定義
		oneTimeMetadata.s3RecordingUrlRedacted	Transcribe Call Analytics によって出力された編集済みメディアオブジェクトの Amazon S3 バケット URL。
		oneTimeMetadata.siprecMetadata	通話に関連付けられた XML 形式の SIPREC メタデータ。
		oneTimeMetadata.siprecMetadataJson	通話に関連付けられた JSON 形式の SIPREC メタデータ。
		oneTimeMetadata.InviteHeaders	招待ヘッダー。

call_analytics_recording_metadata

列名	データ型	[Elements] (要素)	定義
time	string		イベント生成のタイムスタンプ (ISO 8601 形式)。
detail-type (ディテールタイプ)	string		service-type に関連する特徴量タイプ。
service-type	string		AWS サービスの名前、VoiceAnalytics または CallAnalytics。

列名	データ型	[Elements] (要素)	定義
detail-subtype	string		録画タイプと CallAnalyticsMetadata 詳細タイプに使用されます。
callevent-type	string		SIP に関連するイベントタイプ
mediaInsightsPipelineID	string		Amazon Chime SDK メディアインサイト パイプライン ID。
s3MediaObjectConsoleUrl	string		メディアオブジェクトの S3 バケット URL。
metadata	string	voiceConnectorId	Amazon Chime SDK Voice Connector ID。
		callId	参加者の通話 ID (関連する使用向け)。
		transactionId	通話のトランザクション ID。
		fromNumber	発信元の電話番号 (E.164 形式)。
		toNumber	送信先の電話番号 (E.164 形式)。
		direction	通話の方向 (Outbound または Inbound)。

列名	データ型	[Elements] (要素)	定義
		voice enhancement	service-type に関連する特徴量サブタイプ。
		oneTimeMetadata.siprecMetadata	通話に関連付けられた XML 形式の SIPREC メタデータ。
		oneTimeMetadata.siprecMetadataJson	通話に関連付けられた JSON 形式の SIPREC メタデータ。
		oneTimeMetadata.InviteHeaders	招待ヘッダー。

transcribe_call_analytics

列名	データ型	[Elements] (要素)	定義
time	string		イベント生成のタイムスタンプ (ISO 8601 形式)。
detail-type (ディテールタイプ)	string		service-type に関連する特徴量タイプ。
service-type	string		AWS サービスの名前、VoiceAnalytics または CallAnalytics。

列名	データ型	[Elements] (要素)	定義
mediaInsightsPipelineID	string		Amazon Chime SDK メディアインサイト パイプライン ID。
metadata	string	voiceConnectorId	Amazon Chime Voice Connector ID。
		callId	参加者の通話 ID (関連する使用向け)。
		transactionId	通話のトランザクション ID。
		fromNumber	発信元の電話番号 (E.164 形式)。
		toNumber	送信先の電話番号 (E.164 形式)。
		direction	通話の方向 (Outbound または Inbound)。
UtteranceEvent	struct	UtteranceId	指定された UtteranceEvent に関連付けられている一位識別子。
		IsPartial	UtteranceEvent 内のセグメントが完全 (FALSE) か部分的 (TRUE) かを示します。

列名	データ型	[Elements] (要素)	定義
		ParticipantRole	各音声チャンネルにおけるスピーカーのロール (CUSTOMER または AGENT) を示します。
		BeginOffsetMillis	音声ストリームの開始から Utterance Event の開始までの時間 (単位: ミリ秒)。
		EndOffsetMillis	音声ストリームの開始から Utterance Event の開始までの時間 (単位: ミリ秒)。
		Transcript	文字起こしされたテキストが含まれます。
		感情	指定されたセグメントで検出された感情を示します。
		Items.beginoffsetmillis	文字起こしされた項目の開始時間 (単位: ミリ秒)。
		Items.endoffsetmillis	文字起こしされた項目の終了時間 (単位: ミリ秒)。

列名	データ型	[Elements] (要素)	定義
		Items.itemtype	識別された項目のタイプ。オプション: PRONUNCIATION (話し言葉) と PUNCTUATION。
		Items.content	文字起こしされた単語または句読点。
		Items.confidence	トランスクリプト内の単語または句に関連する信頼度スコア。スコアの値は 0~1 です。値が大きいほど、識別された項目がメディアで話されている項目と正確に一致する可能性が高くなります。
		Items.vocabularyfiltermatch	指定された項目が、リクエストに含まれる語彙フィルターの単語と一致するかどうかを示します。true の場合、語彙フィルターの単語と一致しています。

列名	データ型	[Elements] (要素)	定義
		Items.stable	部分的な結果の安定化が有効になっている場合、Stable は指定された項目が安定している (true) か、セグメントが完了すると変化する可能性がある (false) かを示します。
		IssuesDetected.characteroffsets_begin	一致が確認された最初の文字の文字数を示します。例えば、セグメントトランスクリプト内の問題やカテゴリの一致に関連する最初の文字などです。
		IssuesDetected.characteroffsets_end	一致が確認された最後の文字の文字数を示します。例えば、セグメントトランスクリプト内の問題やカテゴリの一致に関連する最後の文字などです。
		Entities.beginoffsetmillis	PII と識別された発話の開始時間 (単位: ミリ秒)。
		Entities.endoffsetmillis	PII と識別された発話の終了時間 (単位: ミリ秒)。

列名	データ型	[Elements] (要素)	定義
		Entities.category	識別された情報のカテゴリ。唯一のカテゴリは PII です。
		Entities.type	識別された PII のタイプ。例えば、NAME、CREDIT_DEBIT_NUMBER などです。
		Entities.content	PII と識別された 1 つまたは複数の単語。
		Entities.confidence	音声内の識別された PII エンティティに関連する信頼度スコア。信頼度スコアの範囲は 0~1 です。値が大きいほど、識別されたエンティティがメディアで話されているエンティティと正確に一致する可能性が高くなります。

transcribe_call_analytics_category_events

列名	データ型	[Elements] (要素)	定義
time	string		イベント生成のタイムスタンプ (ISO 8601 形式)。

列名	データ型	[Elements] (要素)	定義
detail-type (ディテールタイプ)	string		service-type に関連する特徴量タイプ。
service-type	string		AWS サービスの名前、VoiceAnalytics または CallAnalytics。
mediaInsightsPipelineID	string		Amazon Chime SDK メディアインサイトパイプライン ID。
metadata	string	voiceConnectorId	Amazon Chime Voice Connector ID。
		callId	参加者の通話 ID (関連する使用向け)。
		transactionId	通話のトランザクション ID。
		fromNumber	発信元の電話番号 (E.164 形式)。
		toNumber	送信先の電話番号 (E.164 形式)。
		direction	通話の方向 (Outbound または Inbound)。
CategoryEvent	array	MatchedCategories	ユーザーが定義したカテゴリで一致するものを一覧表示します。

transcribe_call_analytics_post_call

列名	データ型	[Elements] (要素)	定義
JobStatus	string		イベント生成のタイムスタンプ (ISO 8601 形式)。
LanguageCode	string		service-type に関連する特徴量タイプ。
Transcript	struct	LoudnessScores	<p>各参加者が話している音量を測定します。このメトリクスを使用して、発信者またはエージェントが大声で話している、または怒鳴っているかどうかを確認します。多くの場合、これは怒っていることを示します。</p> <p>このメトリクスは、0 から 100 の範囲で正規化された値 (特定のセグメントにおける音声の1秒あたりの音声レベル) として表され、値が大きいほど音声が大きいことを示します。</p>
		コンテンツ	文字起こしされたテキストが含まれます。

列名	データ型	[Elements] (要素)	定義
		ID	指定された UtteranceEvent に関連付けられている一位識別子。
		BeginOffsetMillis	音声ストリームの開始から UtteranceEvent の開始までの時間 (単位: ミリ秒)。
		EndOffsetMillis	音声ストリームの開始から UtteranceEvent の開始までの時間 (単位: ミリ秒)。
		感情	指定されたトランスクリプトセグメントで検出された感情を示します。
		ParticipantRole	各音声チャンネルにおけるスピーカーのロール (CUSTOMER または AGENT) を示します。
		IssuesDetected.CharacterOffsets.開始	一致が確認された最初の文字の文字オフセットを示します。例えば、トランスクリプトセグメント内の問題に関連する最初の文字などです。

列名	データ型	[Elements] (要素)	定義
		IssuesDetected.CharacterOffsets.終了	一致が確認された最後の文字の文字オフセットを示します。例えば、トランスクリプトセグメント内の問題に関連する最後の文字などです。
		OutcomesDetected.CharacterOffsets.開始	通話で識別された結果または解決策を示します。
		OutcomesDetected.CharacterOffsets.終了	
		ActionItemsDetected.CharacterOffsets.開始	通話で識別されたアクション項目を一覧表示します。
		ActionItemsDetected.CharacterOffsets.終了	
AccountId	string		AWS アカウント ID
カテゴリ	struct	MatchedCategories	一致したカテゴリを一覧表示します。
		MatchedDetails	音声ストリームの開始からカテゴリでの一致が検出されるまでの時間をミリ秒単位で示します。
Channel	string	Channel	音声チャンネルを示します。

列名	データ型	[Elements] (要素)	定義
参加者	array	ParticipantRole	各音声チャンネルにおけるスピーカーのロール (CUSTOMER または AGENT) を示します。
ConversationCharacteristics	struct	NonTalkTime	音声が含まれていない時間を測定します。このメトリクスを使用して、顧客を過度に長い時間待たせているなど、長い無音時間があるかどうかを確認します。
		中断	一方の参加者がもう一方の参加者の発言を途中で遮ったかどうかと、そのタイミングを測定します。頻繁な中断は無礼または怒りと関連している可能性があり、一方または両方の参加者の否定的な感情と関連していることもあります。
		TotalConversationDurationMillis	会話の合計時間。
		感情OverallSentiment 。エージェント	エージェントの OverallSentiment ラベル。

列名	データ型	[Elements] (要素)	定義
		感情OverallSentiment 。顧客	Customer の OverallSentiment ラベル。
		感情SentimentByPeriod。四半期エージェント	Agent の四半期ごとの感情ラベル。
		感情SentimentByPeriod。QUARTER.CUSTOMER	Customer の四半期ごとの感情ラベル。
		TalkSpeed	両方の参加者が話している速度を測定します。1人の参加者が話すのが速すぎると理解度に影響が出ることがあります。このメトリクスは1分あたりの単語数で測定されます。
		TalkTime	通話中に各参加者が通話した時間(ミリ秒単位)を測定します。このメトリクスを使って、1人の参加者が通話を独占しているかどうか、または会話のバランスがとれているかどうかを識別します。
SessionId	string		通話の SessionId 。

列名	データ型	[Elements] (要素)	定義
ContentMetadata	string		顧客が指定した設定に従って、未加工コンテンツと編集済みコンテンツにラベルを付けるフィールド。

transcribe

列名	データ型	[Elements] (要素)	定義
time	string		イベント生成のタイムスタンプ (ISO 8601 形式)。
detail-type (ディテールタイプ)	string		service-type に関連する特徴量タイプ。
service-type	string		AWS サービスの名前、VoiceAnalytics または CallAnalytics。
mediaInsightsPipelineID	string		Amazon Chime SDK メディアインサイトパイプライン ID。
metadata	string	voiceConnectorId	Amazon Chime Voice Connector ID。
		callId	参加者の通話 ID (関連する使用向け)。
		transactionId	通話のトランザクション ID。

列名	データ型	[Elements] (要素)	定義
		fromNumber	発信元の電話番号 (E.164 形式)。
		toNumber	送信先の電話番号 (E.164 形式)。
		direction	通話の方向 (Outbound または Inbound)。
TranscriptEvent	struct	ResultId	Result の一意の識別子。
		StartTime	Result の開始時間 (単位: ミリ秒)。
		EndTime	Result の終了時間 (単位: ミリ秒)。
		IsPartial	セグメントが完了しているかどうかを示します。IsPartial が true の場合、セグメントは完了していません。それ以外の場合、セグメントは完了しています。
		ChannelId	音声ストリームに関連付けられているチャンネルの ID。

列名	データ型	[Elements] (要素)	定義
		Alternatives.Entities	トランスクリプト出力に、個人を特定できる情報 (PII) として識別されたエンティティが含まれます。
		Alternatives.Items .Confidence	トランスクリプト内の単語または句に関連する信頼度スコア。信頼度スコアの値は 0~1 です。値が大きいほど、識別された項目がメディアで話されている項目と正確に一致する可能性が高くなります。
		Alternatives.Items .Content	文字起こしされた単語または句読点。
		Alternatives.Items .EndTime	文字起こしされた項目の終了時間 (単位: ミリ秒)。
		Alternatives.Items .Speaker	スピーカーパーティシヨニングが有効になっている場合は、Speaker は指定した項目のスピーカーにラベルを付けます。

列名	データ型	[Elements] (要素)	定義
		Alternatives.Items .Stable	部分的な結果の安定化が有効になっている場合、Stable は指定した項目が安定している (true) か、セグメントが完了すると変化する可能性がある (false) かを示します。
		Alternatives.Items .StartTime	文字起こしされた項目の開始時間 (単位: ミリ秒)。
		Alternatives.Items .Type	識別された項目のタイプ。オプション: PRONUNCIATION (話し言葉) と PUNCTUATION 。
		Alternatives.Items .VocabularyFilterMatch	指定された項目が、リクエストに含まれる語彙フィルターの単語と一致するかどうかを示します。true の場合、語彙フィルターの単語と一致しています。
		Alternatives.Transcript	文字起こしされたテキストが含まれます。

voice_analytics_status

列名	データ型	[Elements] (要素)	定義
time	string		イベント生成のタイムスタンプ (ISO 8601 形式)。
detail-type (ディテールタイプ)	string		service-type に関連する特徴量タイプ。
service-type	string		AWS サービスの名前、VoiceAnalytics または CallAnalytics。
ソース	string		イベントを生成する AWS サービス。
アカウント	string		AWS アカウント ID。
region	string		AWS アカ운트리ージョン。
version	string		イベントスキーマのバージョン。
id	string		イベントの一意的 ID。
detail	struct	taskId	タスクの一意的 ID。
		isCaller	参加者が発信者かどうかを示します。
		streamStartTime	ストリームの開始時間。

列名	データ型	[Elements] (要素)	定義
		transactionId	通話のトランザクション ID。
		voiceConnectorId	Amazon Chime Voice Connector ID。
		callId	参加者の通話 ID (関連する使用向け)。
		detailStatus	service-type に関連する詳細な特徴量タイプ。
		statusMessage	タスク ID のステータス (成功または失敗)。
		mediaInsightsPipelineId	Amazon Chime SDK メディアインサイトパイプライン ID。このフィールドは、メディアパイプライン SDK を通じて開始されたスピーカー検索タスクでのみ入力され、音声 SDK では入力されません。
		sourceArn	タスクが実行されるリソース ARN

列名	データ型	[Elements] (要素)	定義
		streamArn	タスクが実行される Kinesis Video Streams ARN。このフィールドは、メディアパイプライン SDK を通じて開始されたスピーカー検索タスクでのみ入力され、音声 SDK では入力されません。
		channelId	タスクが実行される StreamArn のチャンネル。このフィールドは、メディアパイプライン SDK を通じて開始されたスピーカー検索タスクでのみ入力され、音声 SDK では入力されません。
		speakerSearchDetails.voiceProfileId	音声埋め込みが通話中のスピーカーとほぼ一致する、登録されている音声プロファイルの ID。

列名	データ型	[Elements] (要素)	定義
		speakerSearchDetails.confidenceScore	[0, 1] の間の数値。数値が大きいほど、機械学習モデルによって音声プロファイルが一致している確率が高いと見なされていることを意味します。

speaker_search_status

列名	データ型	[Elements] (要素)	定義
time	string		イベント生成のタイムスタンプ (ISO 8601 形式)。
detail-type (ディテールタイプ)	string		service-type に関連する特徴量タイプ。
service-type	string		AWS サービスの名前、VoiceAnalytics または CallAnalytics。
ソース	string		イベントを生成する AWS サービス。
アカウント	string		AWS アカウント ID。
region	string		AWS アカ운トリージョン。
version	string		イベントスキーマのバージョン。

列名	データ型	[Elements] (要素)	定義
id	string		イベントの一意的 ID。
detail	struct	taskId	タスクの一意的 ID。
		isCaller	参加者が発信者かどうかを示します。
		transactionId	通話のトランザクション ID。このフィールドは、Voice Connector を介して行われる通話からタスクが発生した場合に入力されます。
		voiceConnectorId	Amazon Chime Voice Connector ID。このフィールドは、Voice Connector を介して行われる通話からタスクが発生した場合に入力されます。
		mediaInsightsPipelineId	メディアインサイトパイプライン ID。このフィールドは、メディアパイプライン SDK を通じて開始されたスピーカー検索タスクでのみ入力され、音声 SDK では入力されません。
		sourceArn	タスクが実行されるリソース ARN。

列名	データ型	[Elements] (要素)	定義
		streamArn	タスクが実行される Kinesis Video Streams ARN。このフィールドは、メディアパイプライン SDK を通じて開始されたスピーカー検索タスクでのみ入力され、音声 SDK では入力されません。
		channelId	タスクが実行される StreamArn のチャンネル。このフィールドは、メディアパイプライン SDK を通じて開始されたスピーカー検索タスクでのみ入力され、音声 SDK では入力されません。
		participantRole	StreamArn の channelId に関連付けられている参加者ロール。このフィールドは、メディアパイプライン SDK を通じて開始されたスピーカー検索タスクでのみ入力され、音声 SDK では入力されません。

列名	データ型	[Elements] (要素)	定義
		detailStatus	service-type に関連する詳細な特徴量タイプ。
		statusMessage	タスク ID のステータス (成功または失敗)。
		speakerSearchDetails.voiceProfileId	音声埋め込みが通話中のスピーカーとほぼ一致する、登録されている音声プロフィールの ID。
		speakerSearchDetails.confidenceScore	[0, 1] の間の数値。数値が大きいほど、機械学習モデルによって音声プロフィールが一致している確率が高いと見なされていることを意味します。

voice_tone_analysis_status

列名	データ型	[Elements] (要素)	定義
time	string		イベント生成のタイムスタンプ (ISO 8601 形式)。
detail-type (ディテールタイプ)	string		service-type に関連する特徴量タイプ。
service-type	string		AWS サービスの名前、VoiceAnalytics

列名	データ型	[Elements] (要素)	定義
			または CallAnalytics。
ソース	string		イベントを生成する AWS サービス。
アカウント	string		AWS アカウント ID。
region	string		AWS アカун トリー ジョ ン。
version	string		イベントスキーマのバージョン。
id	string		イベントの一意の ID。
detail	struct	taskId	タスクの一意の ID。
		isCaller	参加者が発信者かどうかを示します。
		transactionId	通話のトランザクション ID。このフィールドは、Voice Connector を介して行われる通話からタスクが発生した場合に入力されます。
		voiceConnectorId	Amazon Chime Voice Connector ID。このフィールドは、Voice Connector を介して行われる通話からタスクが発生した場合に入力されます。

列名	データ型	[Elements] (要素)	定義
		mediaInsightsPipelineID	メディアインサイトパイプライン ID。このフィールドは、メディアパイプライン SDK を通じて開始されたスピーカー検索タスクでのみ入力され、音声 SDK では入力されません。
		sourceArn	タスクが実行されるリソース ARN。
		streamArn	タスクが実行される Kinesis Video Streams ARN。このフィールドは、メディアパイプライン SDK を通じて開始されたスピーカー検索タスクでのみ入力され、音声 SDK では入力されません。
		channelId	タスクが実行される StreamArn のチャンネル。このフィールドは、メディアパイプライン SDK を通じて開始されたスピーカー検索タスクでのみ入力され、音声 SDK では入力されません。

列名	データ型	[Elements] (要素)	定義
		participantRole	StreamArn の channelId に関連付けられている参加者ロール。このフィールドは、メディアパイプライン SDK を通じて開始されたスピーカー検索タスクでのみ入力され、音声 SDK では入力されません。
		statusMessage	タスク ID のステータス (成功または失敗)。
		voiceToneAnalysis 詳細。startFr agmentNumber	StreamArn に関連付けられた開始フラグメント番号。
		voiceToneAnalysisD etails.currentAver ageVoiceTone.start Time	現在の平均感情の基準となる、スピーカーの通話音声の開始タイムスタンプ (ISO 8601 形式)。
		voiceToneAnalysisD etails.currentAver ageVoiceTone.endTi me	現在の平均感情の基準となる、スピーカーの通話音声の終了タイムスタンプ (ISO 8601 形式)。

列名	データ型	[Elements] (要素)	定義
		voiceToneAnalysisDetails.currentAverageVoiceTone.beginOffsetMillis	現在の平均感情の基準となる、スピーカーの通話音声の開始フラグメントからの開始オフセット (単位: ミリ)。
		voiceToneAnalysisDetails.currentAverageVoiceTone.endOffsetMillis	現在の平均感情の基準となる、スピーカーの通話音声の開始フラグメントからの終了オフセット (単位: ミリ秒)。
		voiceToneAnalysisDetails.currentAverageVoiceTone.voiceToneScore.positive	スピーカーの感情が肯定的である確率 ([0, 1] の間)。
		voiceToneAnalysisDetails.currentAverageVoiceTone.voiceToneScore.negative	スピーカーの感情が否定的である確率 ([0, 1] の間)。
		voiceToneAnalysisDetails.currentAverageVoiceTone.voiceToneScore.neutral	スピーカーの感情が中立である確率 ([0, 1] の間)。
		voiceToneAnalysisDetails.currentAverageVoiceTone.voiceToneLabel	平均音声トーンスコアの確率が最も高いラベル。

列名	データ型	[Elements] (要素)	定義
		voiceToneAnalysisDetails.overallAverageVoiceTone.startTime	全体の平均感情の基準となる、スピーカーの通話音声の開始タイムスタンプ (ISO 8601 形式)。
		voiceToneAnalysisDetails.overallAverageVoiceTone.endTime	全体の平均感情の基準となる、スピーカーの通話音声の終了タイムスタンプ (ISO 8601 形式)。
		voiceToneAnalysisDetails.overallAverageVoiceTone.beginOffsetMillis	全体の平均感情の基準となる、スピーカーの通話音声の開始フラグメントからの開始オフセット (単位: ミリ秒)。
		voiceToneAnalysisDetails.overallAverageVoiceTone.endOffsetMillis	全体の平均感情の基準となる、スピーカーの通話音声の開始フラグメントからの終了オフセット (単位: ミリ秒)。
		voiceToneAnalysisDetails.overallAverageVoiceToneScore.positive	スピーカーの感情が肯定的である確率 ([0, 1] の間)。
		voiceToneAnalysisDetails.overallAverageVoiceToneScore.negative	スピーカーの感情が否定的である確率 ([0, 1] の間)。

列名	データ型	[Elements] (要素)	定義
		voiceToneAnalysisDetails.overallAverageVoiceTone.voiceToneScore.neutral	スピーカーの感情が中立である確率 ([0, 1] の間)。
		voiceToneAnalysisDetails.overallAverageVoiceTone.voiceToneLabel	感情スコアが最も高い感情ラベル (肯定的、否定的、中立)。

サンプルクエリ

これらのサンプルクエリを使用して、Amazon Chime SDK 通話分析の Glue データカタログのデータを抽出して整理します。

Note

Amazon Athena に接続し、Glue データカタログをクエリする方法については、「[ODBC を使用した Amazon Athena への接続](#)」を参照してください。

必要に応じて各セクションを展開してください。

call_analytics_metadata テーブル内のメタデータ (STRING データ型) から値を抽出する

call_analytics_metadata には JSON 文字列形式の metadata フィールドがあります。Athena の [json_extract_scalar 関数](#) を使用して、この文字列の要素をクエリします。

```
SELECT
  json_extract_scalar(metadata, '$.voiceConnectorId') AS "VoiceConnector ID",
  json_extract_scalar(metadata, '$.fromNumber') AS "From Number",
  json_extract_scalar(metadata, '$.toNumber') AS "To Number",
  json_extract_scalar(metadata, '$.callId') AS "Call ID",
  json_extract_scalar(metadata, '$.direction') AS Direction,
  json_extract_scalar(metadata, '$.transactionId') AS "Transaction ID"
FROM
  "GlueDatabaseName"."call_analytics_metadata"
```

call_analytics_metadata テーブル内の SIPRECMetadata の更新のクエリ

call_analytics_metadata フィールドには JSON 文字列形式のメタデータフィールドがあります。metadata には oneTimeMetadata というネストされた別のオブジェクトがあります。このオブジェクトには、元の XML 形式と変換された JSON 形式の SIPRec メタデータが含まれています。Athena の `json_extract_scalar` 関数を使用して、この文字列の要素をクエリします。

```
SELECT
  json_extract_scalar(metadata, '$.voiceConnectorId') AS "VoiceConnector ID",
  json_extract_scalar(metadata, '$.fromNumber') AS "From Number",
  json_extract_scalar(metadata, '$.toNumber') AS "To Number",
  json_extract_scalar(metadata, '$.callId') AS "Call ID",
  json_extract_scalar(metadata, '$.direction') AS Direction,
  json_extract_scalar(metadata, '$.transactionId') AS "Transaction ID",

  json_extract_scalar(json_extract_scalar(metadata, '$.oneTimeMetadata'), '$.siprecMetadata')
  AS "siprec Metadata XML",

  json_extract_scalar(json_extract_scalar(metadata, '$.oneTimeMetadata'), '$.siprecMetadataJson')
  AS "Siprec Metadata JSON",

  json_extract_scalar(json_extract_scalar(metadata, '$.oneTimeMetadata'), '$.inviteHeaders')
  AS "Invite Headers"
FROM
  "GlueDatabaseName"."call_analytics_metadata"
WHERE
  callevent-type = "update";
```

call_analytics_recording_metadata テーブル内のメタデータ (STRING データ型) から値を抽出する

call_analytics_recording_metadata には JSON 文字列形式のメタデータフィールドがあります。Athena の [json_extract_scalar 関数](#)を使用して、この文字列の要素をクエリします。

```
SELECT
  json_extract_scalar(metadata, '$.voiceConnectorId') AS "VoiceConnector ID",
  json_extract_scalar(metadata, '$.fromNumber') AS "From Number",
  json_extract_scalar(metadata, '$.toNumber') AS "To Number",
  json_extract_scalar(metadata, '$.callId') AS "Call ID",
  json_extract_scalar(metadata, '$.direction') AS Direction,
  json_extract_scalar(metadata, '$.transactionId') AS "Transaction ID"
FROM
  "GlueDatabaseName"."call_analytics_recording_metadata"
WHERE
```



```
detail-subtype = "Recording"
```

voice_analytics_status テーブル内の詳細 (STRUCT データ型) から値を抽出する

voice_analytics_status には struct データ型の詳細フィールドがあります。次の例では、struct データ型フィールドをクエリする方法を示します。

```
SELECT
  detail.transactionId AS "Transaction ID",
  detail.voiceConnectorId AS "VoiceConnector ID",
  detail.siprecmetadata AS "Siprec Metadata",
  detail.inviteheaders AS "Invite Headers",
  detail.streamStartTime AS "Stream Start Time"
FROM
  "GlueDatabaseName"."voice_analytics_status"
```

voice_analytics_status テーブルと call_analytics_metadata テーブルを結合する

次のクエリ例では、call_analytics_metadata と voice_analytics_status を結合します。

```
SELECT
  a.detail.transactionId AS "Transaction ID",
  a.detail.voiceConnectorId AS "VoiceConnector ID",
  a.detail.siprecmetadata AS "Siprec Metadata",
  a.detail.inviteheaders AS "Invite Headers",
  a.detail.streamStartTime AS "Stream Start Time"
  json_extract_scalar(b.metadata, '$.fromNumber') AS "From Number",
  json_extract_scalar(b.metadata, '$.toNumber') AS "To Number",
  json_extract_scalar(b.metadata, '$.callId') AS "Call ID",
  json_extract_scalar(b.metadata, '$.direction') AS Direction
FROM
  "GlueDatabaseName"."voice_analytics_status" a
INNER JOIN
  "GlueDatabaseName"."call_analytics_metadata" b
ON a.detail.transactionId = json_extract_scalar(b.metadata, '$.transactionId')
```

transcribe_call_analytics_post_call テーブルからトランスクリプトを抽出する

transcribe_call_analytics_post_call には、ネストされた配列を含む構造体形式のトランスクリプトフィールドがあります。次のクエリを使用して配列のネストを解除します。

```
SELECT
```

```

    jobstatus,
    languagecode,
    IF(CARDINALITY(m.transcript)=0 OR CARDINALITY(m.transcript) IS NULL, NULL,
e.transcript.id) AS utteranceId,
    IF(CARDINALITY(m.transcript)=0 OR CARDINALITY(m.transcript) IS NULL, NULL,
e.transcript.content) AS transcript,
    accountid,
    channel,
    sessionid,
    contentmetadata.output AS "Redaction"
FROM
    "GlueDatabaseName"."transcribe_call_analytics_post_call" m
CROSS JOIN UNNEST
    (IF(CARDINALITY(m.transcript)=0, ARRAY[NULL], transcript)) AS e(transcript)

```

transcribe_call_analytics_post_call テーブルと call_analytics_metadata テーブルを結合する

次のクエリは、transcribe_call_analytics_post_call と call_analytics_metadata を結合します。

```

WITH metadata AS(
    SELECT
        from_iso8601_timestamp(time) AS "Timestamp",
        date_parse(date_format(from_iso8601_timestamp(time), '%m/%d/%Y %H:%i:%s') , '%m/%d/
%Y %H:%i:%s') AS "DateTime",
        date_parse(date_format(from_iso8601_timestamp(time) , '%m/%d/%Y') , '%m/%d/%Y') AS
"Date",
        date_format(from_iso8601_timestamp(time) , '%H:%i:%s') AS "Time",
        mediainsightspipelineid,
        json_extract_scalar(metadata, '$.toNumber') AS "To Number",
        json_extract_scalar(metadata, '$.voiceConnectorId') AS "VoiceConnector ID",
        json_extract_scalar(metadata, '$.fromNumber') AS "From Number",
        json_extract_scalar(metadata, '$.callId') AS "Call ID",
        json_extract_scalar(metadata, '$.direction') AS Direction,
        json_extract_scalar(metadata, '$.transactionId') AS "Transaction ID",

        REGEXP_REPLACE(REGEXP_EXTRACT(json_extract_scalar(metadata, '$.oneTimeMetadata.s3RecordingUrl')
'[^/]+(?:=\. [^.] +$)'), '\.wav$', '') AS "SessionID"
    FROM
        "GlueDatabaseName"."call_analytics_metadata"
),
transcript_events AS(
    SELECT
        jobstatus,
        languagecode,

```

```
    IF(CARDINALITY(m.transcript)=0 OR CARDINALITY(m.transcript) IS NULL, NULL,
e.transcript.id) AS utteranceId,
    IF(CARDINALITY(m.transcript)=0 OR CARDINALITY(m.transcript) IS NULL, NULL,
e.transcript.content) AS transcript,
    accountid,
    channel,
    sessionid,
    contentmetadata.output AS "Redaction"
FROM
    "GlueDatabaseName"."transcribe_call_analytics_post_call" m
CROSS JOIN UNNEST
    (IF(CARDINALITY(m.transcript)=0, ARRAY[NULL], transcript)) AS e(transcript)
)
SELECT
    jobstatus,
    languagecode,
    a.utteranceId,
    transcript,
    accountid,
    channel,
    a.sessionid,
    "Redaction"
    "Timestamp",
    "DateTime",
    "Date",
    "Time",
    mediainsightspipelineid,
    "To Number",
    "VoiceConnector ID",
    "From Number",
    "Call ID",
    Direction,
    "Transaction ID"
FROM
    "GlueDatabaseName"."transcribe_call_analytics_post_call" a
LEFT JOIN
    metadata b
ON
    a.sessionid = b.SessionID
```

音声の拡張機能の通話録音用のメディアオブジェクト URL をクエリする

次のクエリ例では、Voice enhancement call recording URL を結合します。

```
SELECT
  json_extract_scalar(metadata, '$.voiceConnectorId') AS "VoiceConnector ID",
  json_extract_scalar(metadata, '$.fromNumber') AS "From Number",
  json_extract_scalar(metadata, '$.toNumber') AS "To Number",
  json_extract_scalar(metadata, '$.callId') AS "Call ID",
  json_extract_scalar(metadata, '$.direction') AS Direction,
  json_extract_scalar(metadata, '$.transactionId') AS "Transaction ID",
  s3MediaObjectConsoleUrl
FROM
  {GlueDatabaseName}."call_analytics_recording_metadata"
WHERE
  detail-subtype = "VoiceEnhancement"
```

Amazon Chime SDK 音声分析を使用する

Amazon Chime SDK 音声分析機能を使用すると、スピーカー検索と音声トーン分析を実装できます。スピーカー検索では、新規の発信者を特定して登録したり、リポート発信者を特定してその識別情報に信頼度スコアを割り当てたりできます。音声トーン分析では、発信者の感情 (negative、neutral、または positive) を予測します。

Amazon Chime SDK 通話分析セッションのオプションコンポーネントとして音声分析を実行します。

音声分析は、メディアインサイトパイプラインまたは Amazon Chime SDK Voice Connector の通話と連携します。タスクやタスクに関する情報をよりきめ細かく制御するには、[メディアパイプライン SDK](#) を使用し、メディアインサイトパイプラインでタスクを呼び出すことをお勧めします。

Voice Connector を使用して下位互換性を確保することもできますが、更新されるのはメディアインサイトパイプライン API の新機能のみです。

Voice Connector の作成と使用の詳細については、「Amazon Chime SDK 管理者ガイド」の「[Amazon Chime Voice Connector の管理](#)」を参照してください。

音声分析には以下の機能もあります。

- 非同期のタスク処理。タスクは相互に独立して実行されます。
- インサイトを処理するタイミングの制御。

音声分析を開始するには、[StartSpeakerSearchTask](#) および [StartVoiceToneAnalysisTask](#) APIs。

次のトピックでは、音声分析の使い方について説明します。

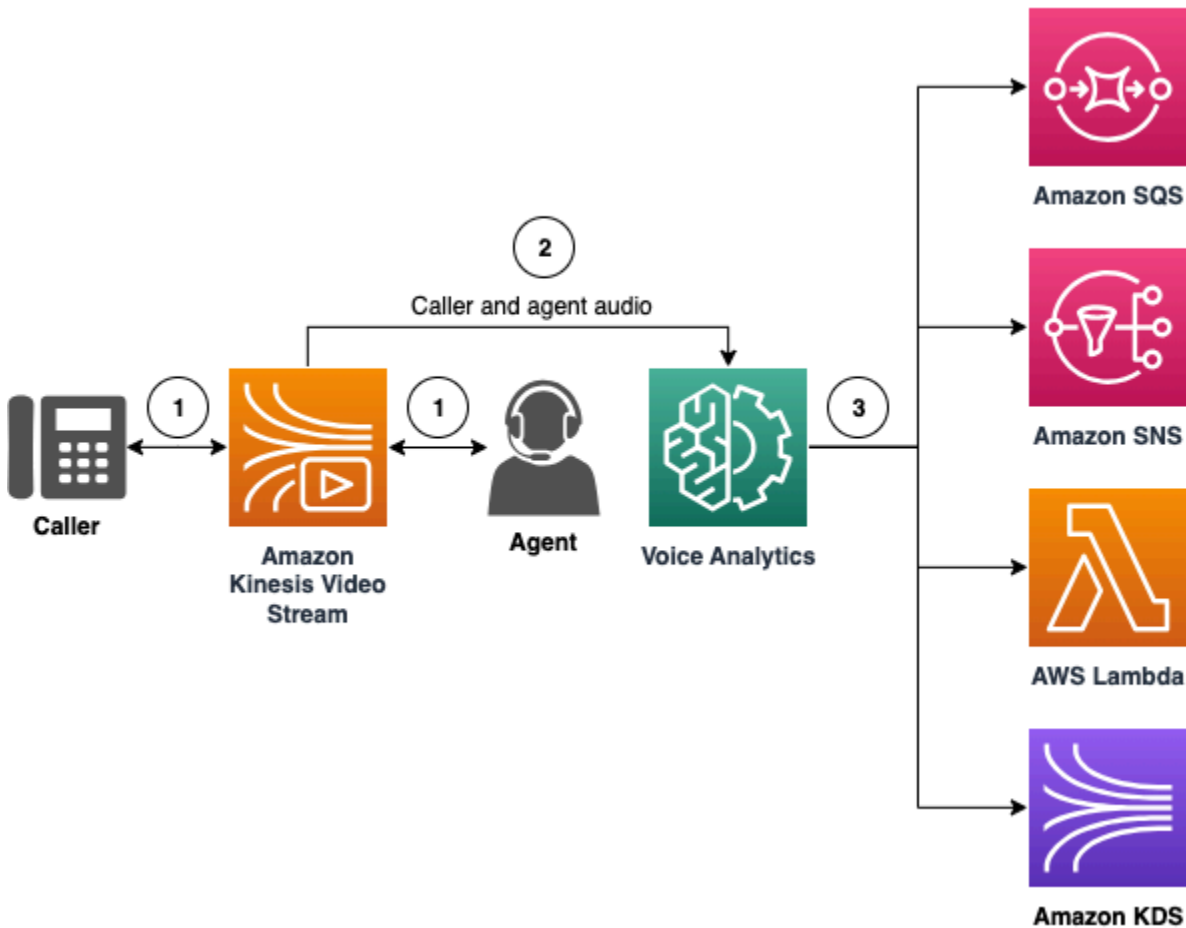
トピック

- [音声分析アーキテクチャ](#)
- [スピーカー検索ワークフローの例](#)
- [音声トーン分析のワークフローの例](#)
- [タスク結果のポーリング](#)
- [通知について](#)
- [データストレージ、オプトアウト、データ保持ポリシーについて](#)
- [音声 API を使用して音声分析を実行する](#)

音声分析アーキテクチャ

このセクションのトピックでは、各機能のデータフローを含む、Amazon Chime SDK 音声分析アーキテクチャの概要について説明します。

この図は、音声分析におけるデータフローの概要を示しています。



図中の手順を説明します。

1. 音声は、発信者とエージェントの Kinesis Video Streams にストリーミングされます。これを行うには、Kinesis Video Streams プロデューサーまたは Amazon Chime SDK Voice Connector のストリーミングを使用します。詳細については、本ガイドの「[機械学習ベースの分析を目的としたワークフロー](#)」と、「Amazon Chime SDK 管理者ガイド」の「[Amazon Chime Voice Connector から Kinesis へのメディアストリーミング](#)」を参照してください。
2. アプリケーションまたはビルダーは、発信者の同意後に音声ストリームのスピーカー検索、音声トーン分析、あるいはその両方をトリガーします。
3. 通話中、音声分析は、Amazon Simple Queue Service (SQS)、Amazon Simple Notification Service (SNS)、AWS Lambda、または Amazon Kinesis Data Streams のいずれかのターゲットに通知を送信します。

さらに、音声分析には、生成したデータを管理するためのツールも用意されています。

音声プロフィール

音声埋め込み、埋め込みの一意の ID、有効期限を組み合わせた情報です。セキュリティ上の理由から、また時間の経過と共に音声に変化することから、音声プロフィールは 3 年後に有効期限が切れます。音声プロフィールを再作成しないようにするには、[UpdateVoiceProfile](#) API を呼び出します。有効期限の詳細については、「[データ保持](#)」を参照してください。

音声埋め込みを登録したり、登録された音声埋め込みを更新したりするには、通話終了後 24 時間以内に [CreateVoiceProfile](#) または [UpdateVoiceProfile](#) APIs を呼び出す必要があります。

音声プロフィールドメイン

音声プロフィールの集合体。

スピーカー検索ワークフローの例

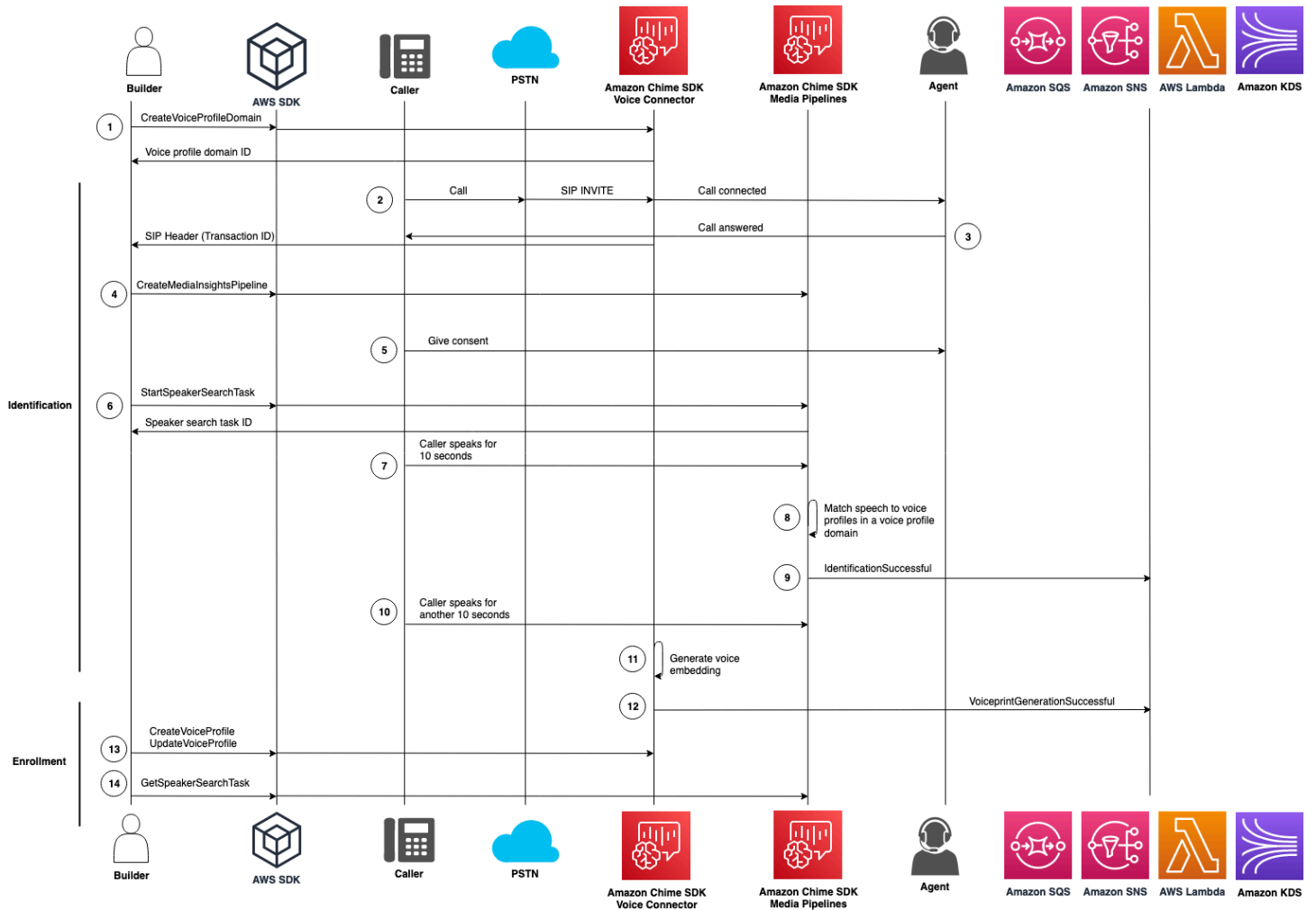
⚠ Important

スピーカー検索機能では、音声埋め込みを作成し、それを使用して発信者の声を以前に保存した音声データと比較することができます。生体認証識別子と生体認証情報をデジタル声紋の形で収集、使用、保存、保持するには、書面による発信者の同意が必要な場合があります。このような同意は、イリノイ州、テキサス州、ワシントン州の生体認証法やその他の州のプライバシー法など、さまざまな州法で義務付けられています。スピーカー検索機能を使用する前に、適用法および機能の使用を規定する [AWS サービス条件](#) に従って、すべての通知を行い、すべての同意を得る必要があります。

次の図は、スピーカー検索分析タスクでのデータフローの例を示しています。画像の下にある番号付きのテキストは、プロセスの各ステップを説明しています。

ℹ Note

この図は、Amazon Chime SDK Voice Connector を、VoiceAnalyticsProcessor のある通話分析設定で既に設定していることを前提としています。詳細については、「[Voice Connector の通話を録音する](#)」を参照してください。



図中の手順を説明します。

- ユーザーまたはシステム管理者は、音声埋め込みと音声プロファイルを保存するための音声プロファイルドメインを作成します。音声プロファイルドメインの作成の詳細については、「Amazon Chime SDK 管理者ガイド」の「[Creating voice profile domains](#)」を参照してください。[CreateVoiceProfileDomain](#) API を使用することもできます。
- 発信者は、Amazon Chime SDK Voice Connector に割り当てられた電話番号を使用してダイヤルインします。または、エージェントが Voice Connector 番号を使用して発信通話を行います。
- Amazon Chime SDK Voice Connector サービスはトランザクション ID を作成し、それを通話に関連付けます。
- アプリケーションが EventBridge イベントをサブスクライブしていると仮定すると、アプリケーションはメディアインサイトパイプライン設定と Voice Connector コールの Kinesis Video Stream ARNs を使用して [CreateMediaInsightsPipeline](#) API を呼び出します。

の使用の詳細については EventBridge、「」を参照してください [機械学習ベースの分析を目的としたワークフロー](#)。

- アプリケーション (自動音声応答システムなど) またはエージェントは、通話の録音や音声分析のための音声埋め込みの使用について発信者に通知し、参加の同意を求めます。
- 発信者が同意すると、[Voice Connector とトランザクション ID がある場合は、アプリケーションまたはエージェントが Voice SDK を介して StartSpeakerSearchTask API を呼び出すことができます](#)。または、トランザクション ID の代わりにメディアインサイトパイプライン ID がある場合は、メディアパイプライン SDK で [StartSpeakerSearchTask API を呼び出します](#)。 https://docs.aws.amazon.com/chime-sdk/latest/APIReference/API_Operations_Amazon_Chime_SDK_Media_Pipelines.html

発信者が同意すると、アプリケーションまたはエージェントは StartSpeakerSearchTask API を呼び出します。Voice Connector ID、トランザクション ID、音声プロファイルドメイン ID を API に渡す必要があります。非同期タスクを識別するためのスピーカー検索タスク ID が返されません。

Note

いずれかの SDK で StartSpeakerSearchTask API を呼び出す前に、法律および [AWS サービス条件](#) に従って必要な通知を行い、必要な同意を得る必要があります。

- システムには、発信者の音声は 10 秒分蓄積されます。発信者は少なくともその時間だけ話さなければなりません。システムは無音部分をキャプチャしたり分析したりしません。
- メディアインサイトパイプラインは、音声をドメイン内の音声プロファイルと比較し、一致率の高い上位 10 件を一覧表示します。一致する音声プロファイルが見つからない場合、Voice Connector は音声プロファイルを作成します。
- メディアインサイトパイプラインサービスは、設定された通知ターゲットに通知イベントを送信します。
- 発信者は話し続け、無音以外の音声をさらに 10 秒間提供します。
- メディアインサイトパイプラインは、音声プロファイルの作成や既存の音声プロファイルの更新に使用できる登録音声埋め込みを生成します。
- メディアインサイトパイプラインは、設定された通知ターゲットに VoiceprintGenerationSuccessful 通知を送信します。
- アプリケーションは [CreateVoiceProfile](#) または [UpdateVoiceProfile](#) APIs を呼び出して、プロファイルを作成または更新します。

14.アプリケーションは、必要に応じて [GetSpeakerSearchTask](#) API を呼び出して、発話者検索タスクの最新のステータスを取得します。

音声トーン分析のワークフローの例

Important

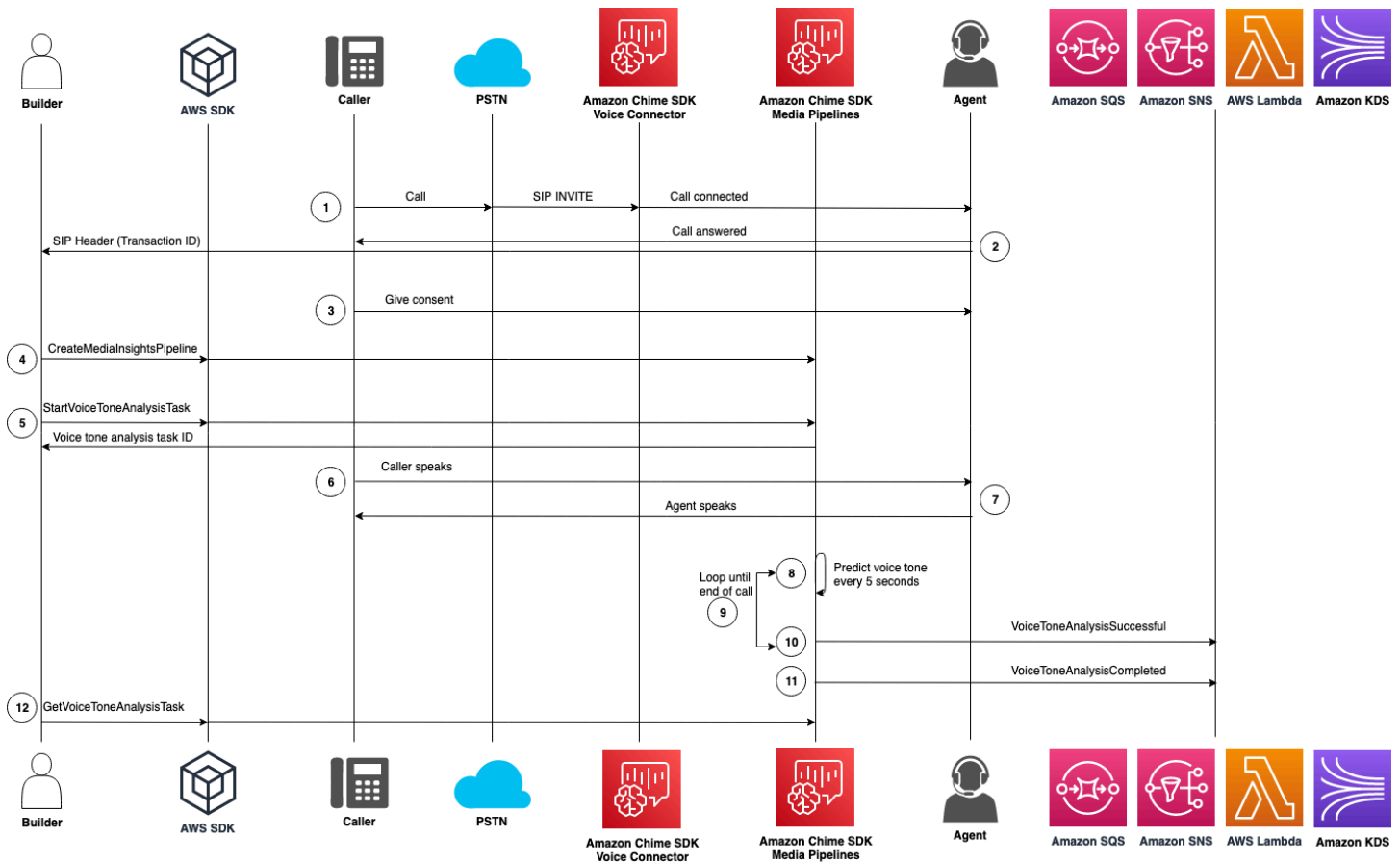
音声トーン分析では、言語およびトーンに関する情報に基づいてスピーカーの感情を予測します。感情分析は、法律で禁止されているいかなる方法でも使用してはなりません。これには、個人に対して法的な影響、または同様の重大な影響を与えるような意思決定 (雇用、住宅、信用、金銭的な申し出などに関するもの) における使用が含まれます。

音声トーン分析では、通話中のスピーカーの声を分析し、positive、negative、neutral のいずれかの感情を予測します。

次の図は、音声トーン分析のワークフローの例を示しています。画像の下にある番号付きの項目は、プロセスの各ステップを説明しています。

Note

この図は、Amazon Chime SDK Voice Connector を、VoiceAnalyticsProcessor のある通話分析設定で既に設定していることを前提としています。詳細については、「[Voice Connector の通話を録音する](#)」を参照してください。



図中の手順を説明します。

1. 発信者は、Amazon Chime SDK Voice Connector に割り当てられた電話番号を使用してダイヤルインします。または、エージェントが Voice Connector 番号を使用して発信通話を行います。
2. Voice Connector サービスはトランザクション ID を作成し、それを通話に関連付けます。
3. アプリケーション (自動音声応答システムなど) またはエージェントは、通話の録音や音声分析のための音声埋め込みの使用について発信者に通知し、参加の同意を求めます。
4. アプリケーションが EventBridge イベントをサブスクライブしていると仮定すると、アプリケーションはメディアインサイトパイプライン設定と Voice Connector コールの Kinesis Video Stream ARNs を使用して [CreateMediaInsightsPipeline](#) API を呼び出します。


の使用の詳細については EventBridge、「」を参照してください [機械学習ベースの分析を目的としたワークフロー](#)。

5. 発信者が同意すると、Voice Connector とトランザクション ID [がある場合は、アプリケーションまたはエージェントが Voice SDK を介して StartSpeakerSearchTask API を呼び出すことができます。または、トランザクション ID の代わりにメディアインサイトパイプライン ID がある場合は、メディアインサイトパイプライン SDK で StartSpeakerSearchTask](#)


API を呼び出します。 https://docs.aws.amazon.com/chime-sdk/latest/APIReference/API_Operations_Amazon_Chime_SDK_Media_Pipelines.html

発信者が同意すると、アプリケーションまたはエージェントは StartSpeakerSearchTask API を呼び出します。Voice Connector ID、トランザクション ID、音声プロファイルドメイン ID を API に渡す必要があります。非同期タスクを識別するためのスピーカー検索タスク ID が返されます。

6. ユーザーは通話中ずっと話します。
7. エージェントは通話中ずっと話します。
8. メディアインサイトパイプラインは 5 秒ごとに、機械学習モデルを使用して過去 30 秒間の音声を分析し、その時間における、および最初に StartVoiceToneAnalysisTask が呼び出された時点からの通話全体における発信者のトーンを予測します。
9. メディアインサイトパイプラインは、設定された通知ターゲットにその情報を含む通知を送信します。ストリーム ARN とチャンネル ID に基づいて通知を識別できます。詳細については、このセクションで後述する「[通知について](#)」を参照してください。
10. 通話が終了するまでステップ 9 と 10 を繰り返します。
11. 通話が終了すると、メディアインサイトパイプラインは、過去 30 秒間の最新の平均トーン予測と、通話全体の平均トーンを含む最後の通知を 1 つ送信します。
12. アプリケーションは、必要に応じて [GetVoiceToneAnalysisTask](#) API を呼び出して、音声トーン分析タスクの最新のステータスを取得します。

 Note

GetVoiceToneAnalysisTask API はトーンデータをストリーミングしません。

 Note

[GetVoiceToneAnalysisTask](#) API は音声トーンデータを返しません。

タスク結果のポーリング

⚠ Important

デフォルトでは、音声分析の結果は 7 日間有効で、その後データは自動的に削除されます。タスクデータを長期間使用したい場合や、データ保持法を遵守したい場合は、それらのデータを保存する必要があります。詳細については、このガイドで後述する「[データ保持](#)」を参照してください。

音声分析では通常、各タスク結果が少なくとも 1 回は配信されます。ただし、ネットワークに問題がある場合、レイテンシーが増加する可能性があります。潜在的な問題を回避したい場合や、同期処理を行いたい場合は、[メディアパイプライン SDK](#) または [音声 SDK](#) で次の API を使用できます。

- [GetSpeakerSearchTask](#)
- [GetVoiceToneAnalysisTask](#)

⚠ Important

`GetVoiceToneAnalysisTask` API はタスクのステータスのみを返します。タスクの結果は返しません。結果を表示するには、Amazon SQS、Amazon SNS、または AWS Lambda 通知ターゲットを使用します。

`GetSpeakerSearchTask` API は、タスク ID、遅延メッセージ、または順不同で届いたメッセージについて、最新の結果を同期的に取得します。ただし、通知ターゲットと非同期処理を使用することをお勧めします。そうすることで、コンピューティングリソースの消費量を抑えることができます。

通知について

音声分析では、スピーカー検索または音声トーン分析のタスクの開始時、実行中、および終了時に、自動的にターゲットにイベントが送信されます。通知ターゲットを使用してこれらのイベントを受信します。ワークフローやアプリケーションに高可用性が必要な場合は、複数の通知ターゲットを使用することをお勧めします。

また、通知ターゲットへのアクセスに必要なポリシーには IAM ロールを使用する必要があります。詳細については、「[通話分析のリソースアクセスロールを使用する](#)」を参照してください。

Note

Amazon SQS および Amazon SNS では、first-in-first-out キューはサポートされていません。そのため、メッセージが順不同で届く可能性があります。タイムスタンプを確認して必要に応じてメッセージを順序付け、メッセージを Amazon DynamoDB などのデータストアに保持することをお勧めします。「[タスク結果のポーリング](#)」で説明されている Get API を使用して最新の結果を取得することもできます。

次の表に、イベントとそれに対応する詳細タイプを示します。

通知イベント	Detail-type
音声分析メタデータ	VoiceAnalyticsStatus
発話者検索	SpeakerSearchStatus
音声トーン分析	VoiceToneAnalysisStatus

通知ターゲットの IAM ポリシー

Amazon SQS、Amazon SNS、AWS Lambda、または Amazon KDS の通知ターゲットへのアクセスを許可する、通話分析設定の IAM ロールのポリシーを使用する必要があります。詳細については、このガイドの「[通話分析のリソースアクセスロールを使用する](#)」を参照してください。

スピーカー検索イベント

スピーカー検索イベントの詳細タイプは SpeakerSearchStatus です。

Amazon Chime SDK Voice Connector は、以下のスピーカー検索イベントを送信します。

- ID の一致
- 音声埋め込みの生成

イベントには以下のステータスがあります。

- IdentificationSuccessful - 特定の音声プロファイルドメイン内で、高い信頼度スコアで一致する音声プロファイル ID を少なくとも 1 つ正常に識別しました。

- `IdentificationFailure` - 識別を実行できませんでした。原因: 発信者が 10 秒以上話していないか、音質が良くありません。
- `IdentificationNoMatchesFound` - 指定された音声プロファイルドメインで、高い確率で一致する音声プロファイルが見つかりませんでした。発信者が新規であるか、声が変わっている可能性があります。
- `VoiceprintGenerationSuccessful` - システムは、20 秒間の無音ではない音声を使用して音声埋め込みを生成しました。
- `VoiceprintGenerationFailure` - システムは音声埋め込みを生成できませんでした。原因: 発信者が 20 秒以上話していないか、音質が良くありません。

ID の一致

特定の `StartSpeakerSearchTask` API が呼び出されると `transactionId`、`Voice Connector` サービスは 10 秒間無音でない音声の後に識別一致通知を返します。このサービスは、上位 10 件の一致を、音声プロファイル ID と [0, 1] の範囲の信頼スコアと共に返します。信頼度スコアが高いほど、通話中のスピーカーが音声プロファイル ID と一致する可能性が高くなります。機械学習モデルで一致が見つからなかった場合、通知の `detailStatus` フィールドには `IdentificationNoMatchesFound` が含まれます。

次の例は、一致が見つかった場合の通知を示しています。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-111122223333",
  "detail-type": "SpeakerSearchStatus",
  "service-type": "VoiceAnalytics",
  "source": "aws.chime",
  "account": "111122223333",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "taskId": "uuid",
    "detailStatus": "IdentificationSuccessful",
    "speakerSearchDetails" : {
      "results": [
        {
          "voiceProfileId": "vp-505e0992-82da-49eb-9d4a-4b34772b96b6",
          "confidenceScore": "0.94567856",
```

```

    },
    {
      "voiceProfileId": "vp-fba9cbfa-4b8d-4f10-9e41-9dfdd66545ab",
      "confidenceScore": "0.82783350",
    },
    {
      "voiceProfileId": "vp-746995fd-16dc-45b9-8965-89569d1cf787",
      "confidenceScore": "0.77136436",
    }
  ]
},
"mediaInsightsPipelineId": "87654321-33ca-4dc6-9cdf-abcde6612345",
"sourceArn": "arn:aws:chime:us-east-1:111122223333:media-
pipeline/87654321-33ca-4dc6-9cdf-abcde6612345",
"streamArn": "arn:aws:kinesisvideo:us-east-1:111122223333:stream/my-
stream/0123456789012",
"channelId": 0
}
}

```

音声埋め込みの生成

無音でない音声 さらに 10 秒間続くと、Voice Connector は通知ターゲットに音声埋め込み生成通知を送信します。音声プロファイルに新しい音声埋め込みを登録したり、音声プロファイルに既に登録されている声紋を更新したりできます。

次の例は、一致が見つかった場合の通知を示しています。この場合、関連する音声プロファイルを更新できます。

```

{
  "version": "0",
  "id": "12345678-1234-1234-1234-111122223333",
  "detail-type": "SpeakerSearchStatus",
  "service-type": "VoiceAnalytics",
  "source": "aws.chime",
  "account": "111122223333",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "taskId": "guid",
    "detailStatus": "VoiceprintGenerationSuccess",
    "mediaInsightsPipelineId": "87654321-33ca-4dc6-9cdf-abcde6612345",
  }
}

```



```
    "sourceArn": "arn:aws:chime:us-east-1:111122223333:media-  
pipeline/87654321-33ca-4dc6-9cdf-abcde6612345",  
    "streamArn": "arn:aws:kinesisvideo:us-east-1:111122223333:stream/my-  
stream/0123456789012",  
    "channelId": 0  
  }  
}
```

音声トーン分析イベント

音声トーン分析イベントの詳細タイプは `VoiceToneAnalysisStatus` です。分析では次のステータスが返されます。

- `VoiceToneAnalysisSuccessful` — 発信者とエージェントの声を分析し、感情 (肯定的、否定的、または中立的) を予測できました。
- `VoiceToneAnalysisFailure` - トーン分析を実行できませんでした。これは、発信者が 10 秒間話さずに電話を切ったり、音質が悪くなりすぎたりした場合に発生する可能性があります。
- `VoiceToneAnalysisCompleted` - ユーザーとエージェントの声を分析して、通話全体の感情を予測できました。これは音声トーン分析が終了したときに送信される最後のイベントです。

次の例は、典型的な音声トーン分析イベントを示しています。

```
{  
  "detail-type": "VoiceToneAnalysisStatus",  
  "service-type": "VoiceAnalytics",  
  "source": "aws.chime",  
  "account": "216539279014",  
  "time": "2022-08-26T17:55:15.563441Z",  
  "region": "us-east-1",  
  "detail": {  
    "taskId": "uuid",  
    "detailStatus": "VoiceToneAnalysisSuccessful",  
    "voiceToneAnalysisDetails": {  
      "currentAverageVoiceTone": {  
        "startTime": "2022-08-26T17:55:15.563Z",  
        "endTime": "2022-08-26T17:55:45.720Z",  
        "voiceToneLabel": "neutral",  
        "voiceToneScore": {  
          "neutral": "0.83",  
          "positive": "0.13",  
          "negative": "0.04"  
        }  
      }  
    }  
  }  
}
```

```
    }
  },
  "overallAverageVoiceTone": {
    "startTime": "2022-08-26T16:23:13.344Z",
    "endTime": "2022-08-26T17:55:45.720Z",
    "voiceToneLabel": "positive",
    "voiceToneScore": {
      "neutral": "0.25",
      "positive": "0.65",
      "negative": "0.1"
    }
  }
},
"startFragmentNumber": "01234567890123456789",
"mediaInsightsPipelineId": "87654321-33ca-4dc6-9cdf-abcde6612345",
"sourceArn": "arn:aws:chime:us-east-1:111122223333:media-
pipeline/87654321-33ca-4dc6-9cdf-abcde6612345",
"streamArn": "arn:aws:kinesisvideo:us-east-1:111122223333:stream/my-
stream/0123456789012",
"channelId": 0
},
"version": "0",
"id": "Id-f928dfe3-f44b-4965-8a17-612f9fb92d59"
}
```

通話後のサマリーイベント

通話後のサマリーイベントは、通話が終了した 5 分後に送信されます。これらのサマリーには、通話中に発生したスピーカー検索タスクの概要が記載されています。

次の例は、通話後のサマリーを示しています。これには、最も一致した音声プロファイル、確認済みのスピーカー ID、通話中に行われた CreateVoiceProfile および UpdateVoiceProfile API コールを通じて作成または更新された音声プロファイルのリストが含まれます。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-111122223333",
  "detail-type": "VoiceAnalyticsStatus",
  "service-type": "VoiceAnalytics",
  "source": "aws.chime",
  "account": "111122223333",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
```

```
"resources": [],
"detail": {
  "detailStatus": "PostCallVoiceAnalytics",
  "callId": "22e8dee8-bbd7-4f94-927b-2d0ebaeddc1c",
  "transactionId": "daaeb6bf-2fe2-4e51-984e-d0fbf2f09436",
  "voiceConnectorId": "abcdef1ghij2klmno3pqr4",
  "isCaller": true | false,
  "speakerSearchResults": {
    "bestMatchedVoiceProfileId": "vp-04c25ba1-a059-4fd3-8495-4ac91b55e2bf",
    "customerValidatedCallerIdentity": "vp-04c25ba1-
a059-4fd3-8495-4ac91b55e2bf",
    "createVoiceProfileTransactions": [
      {
        "voiceProfileId": "vp-04c25ba1-a059-4fd3-8495-4ac91b55e2bf",
        "requestTimestamp": "2022-12-14T18:38:38.796Z"
      },
      {
        "voiceProfileId": "vp-04c25ba1-a059-4fd3-8495-4ac91b55e2bf",
        "requestTimestamp": "2022-12-14T18:38:38.796Z",
      }
    ],
    "updateVoiceProfileTransactions": [
      {
        "voiceProfileId": "vp-04c25ba1-a059-4fd3-8495-4ac91b55e2bf",
        "requestTimestamp": "2022-12-14T18:38:38.796Z",
      },
      {
        "voiceProfileId": "vp-04c25ba1-a059-4fd3-8495-4ac91b55e2bf",
        "requestTimestamp": "2022-12-14T18:38:38.796Z",
      }
    ]
  }
}
```

音声分析の Lambda 関数の例

次の例の Python コードは、Voice Connector から受信した通知を処理します。AWS Lambda 関数にコードを追加できます。また、これを使用して Amazon SQS キュー、Amazon SNS トピック、または Amazon Kinesis Data Streams をトリガーすることもできます。その後、通知を EventTable に保存して今後の処理に備えることができます。正確な通知形式については、「[通知について](#)」を参照してください。

```
import base64
import boto3
import json
import logging
import time

from datetime import datetime
from enum import Enum

log = logging.getLogger()
log.setLevel(logging.INFO)

dynamo = boto3.client("dynamodb")

EVENT_TABLE_NAME = "EventTable"

class EventType(Enum):
    """
    This example code uses a single Lambda processor to handle either
    triggers from SQS, SNS, Lambda, or Kinesis. You can adapt it to fit your
    desired infrastructure depending on what you prefer. To distinguish
    where we get events from, we use an EventType enum as an
    example to show the different ways of parsing the notifications.
    """
    SQS = "SQS"
    SNS = "SNS"
    LAMBDA = "LAMBDA"
    KINESIS = "KINESIS"

class AnalyticsType(Enum):
    """
    Define the various analytics event types that this Lambda will
    handle.
    """
    SPEAKER_SEARCH = "SpeakerSearch"
    VOICE_TONE_ANALYSIS = "VoiceToneAnalysis"
    ANALYTICS_READY = "AnalyticsReady"
    UNKNOWN = "UNKNOWN"

class DetailType(Enum):
    """
```

```
Define the various detail types that Voice Connector's voice
analytics feature can return.
"""
SPEAKER_SEARCH_TYPE = "SpeakerSearchStatus"
VOICE_TONE_ANALYSIS_TYPE = "VoiceToneAnalysisStatus"
ANALYTICS_READY = "VoiceAnalyticsStatus"

def handle(event, context):
    """
    Example of how to handle incoming Voice Analytics notification messages
    from Voice Connector.
    """
    logging.info(f"Received event of type {type(event)} with payload {event}")
    is_lambda = True

    # Handle triggers from SQS, SNS, and KDS. Use the below code if you would like
    # to use this Lambda as a trigger for an existing SQS queue, SNS topic or Kinesis
    # stream.
    if "Records" in event:
        logging.info("Handling event from SQS or SNS since Records exists")
        is_lambda = False
        for record in event.get("Records", []):
            _process_record(record)

    # If you would prefer to have your Lambda invoked directly, use the
    # below code to have the Voice Connector directly invoke your Lambda.
    # In this scenario, there are no "Records" passed.
    if is_lambda:
        logging.info(f"Handling event from Lambda")
        event_type = EventType.LAMBDA
        _process_notification_event(event_type, event)

def _process_record(record):
    # SQS and Kinesis use eventSource.
    event_source = record.get("eventSource")

    # SNS uses EventSource.
    if not event_source:
        event_source = record.get("EventSource")

    # Assign the event type explicitly based on the event source value.
    event_type = None
```

```

if event_source == "aws:sqs":
    event = record["body"]
    event_type = EventType.SQS
elif event_source == "aws:sns":
    event = record["Sns"]["Message"]
    event_type = EventType.SNS
elif event_source == "aws:kinesis":
    raw_data = record["kinesis"]["data"]
    raw_message = base64.b64decode(raw_data).decode('utf-8')
    event = json.loads(raw_message)
    event_type = EventType.KINESIS
else:
    raise Exception(f"Event source {event_source} is not supported")

_process_notification_event(event_type, event)

def _process_notification_event(
    event_type: EventType,
    event: dict
):
    """
    Extract the attributes from the Voice Analytics notification message
    and store it as a DynamoDB item to process later.
    """
    message_id = event.get("id")
    analytics_type = _get_analytics_type(event.get("detail-type"))
    pk = None
    if analytics_type == AnalyticsType.ANALYTICS_READY.value or analytics_type ==
AnalyticsType.UNKNOWN.value:
        transaction_id = event.get("detail").get("transactionId")
        pk =
f"transactionId#{transaction_id}#notificationType#{event_type.value}#analyticsType#{analytics_
    else:
        task_id = event.get("detail").get("taskId")
        pk =
f"taskId#{task_id}#notificationType#{event_type.value}#analyticsType#{analytics_type}"
        logging.info(f"Generated PK {pk}")
        _create_request_record(pk, message_id, json.dumps(event))

def _create_request_record(pk: str, sk: str, body: str):
    """
    Record this notification message into the Dynamo db table

```

```
"""
try:
    # Use consistent ISO8601 date format.
    # 2019-08-01T23:09:35.369156 -> 2019-08-01T23:09:35.369Z
    time_now = (
        datetime.utcnow().isoformat()[:-3] + "Z"
    )
    response = dynamo.put_item(
        Item={
            "PK": {"S": pk},
            "SK": {"S": sk},
            "body": {"S": body},
            "createdOn": {"S": time_now},
        },
        TableName=EVENT_TABLE_NAME,
    )
    logging.info(f"Added record in table {EVENT_TABLE_NAME}, response :
{response}")
except Exception as e:
    logging.error(f"Error in adding record: {e}")
```

```
def _get_analytics_type(detail_type: str):
    """
    Get analytics type based on message detail type value.
    """
    if detail_type == DetailType.SPEAKER_SEARCH_TYPE.value:
        return AnalyticsType.SPEAKER_SEARCH.value
    elif detail_type == DetailType.VOICE_TONE_ANALYSIS_TYPE.value:
        return AnalyticsType.VOICE_TONE_ANALYSIS.value
    elif detail_type == DetailType.ANALYTICS_READY.value:
        return AnalyticsType.ANALYTICS_READY.value
    else:
        return AnalyticsType.UNKNOWN.value
```

Important

[StartSpeakerSearchTask](#) または [StartVoiceToneAnalysis](#) APIs を呼び出す前に、同意を取得する必要があります。同意が得られるまでは、Amazon DynamoDB などの保持エリアでイベントを保持することをお勧めします。

データストレージ、オプトアウト、データ保持ポリシーについて

Amazon Chime SDK は、音声データを使用してスピーカー検索サービスを提供および改善しています。その一環として、登録オーディオ (音声埋め込みの作成に使用される録音されたスニペット) を使用して、機械学習モデルと人工知能モデルをトレーニングしています。モデルのトレーニングにデータが使用されることをオプトアウトすることができます。このセクションのトピックでは、その方法を説明します。

トピック

- [スピーカー検索用のデータストレージ](#)
- [スピーカー検索のオプトアウト処理](#)
- [データ保持](#)

スピーカー検索用のデータストレージ

Amazon Chime SDK は、スピーカー検索用に以下のデータを保存します。

- 音声プロファイルに添付されている音声埋め込み。スピーカー検索機能を提供するために使用されます。
- 登録オーディオ。録音された音声スニペットで、各音声プロファイルの音声埋め込みに使用されます。登録オーディオの録音は、以下の目的で使用されます。
 - スピーカー検索モデルを最新の状態に保ちます。これは、スピーカー検索機能を提供する上で極めて重要です。
 - 機械学習モデルをトレーニングして、サービスを開発および改善します。トレーニングでの登録オーディオの使用は任意です。次のセクションで説明するように、オプトアウトポリシーを選択することで、この使用をオプトアウトできます。

スピーカー検索のオプトアウト処理

エンドユーザーと組織全体のオプトアウトを処理できます。オプトアウトには次のような効果があります。

- オプトアウトすると、音声分析のモデルトレーニングで新しい登録オーディオが使用されなくなり、オプトアウト前に収集、保存された登録音声も使用されなくなります。
- オプトアウトすると、音声分析ではスピーカー検索サービスを提供するために登録オーディオが保存および使用されます。

⚠ Warning

以下のオプトアウトアクションは元に戻すことはできません。また、削除したデータを復元することもできません。

エンドユーザーのオプトアウト処理

エンドユーザーがスピーカー検索をオプトアウトする場合は、[DeleteVoiceProfile](#) API を呼び出します。このアクションにより、音声プロファイルに加えて、音声埋め込みと登録オーディオが削除されます。

音声埋め込みのグループを削除するには、[DeleteVoiceProfileDomain](#) API を呼び出してドメインを削除します。このアクションにより、ドメイン内のすべての音声プロファイルが削除されます。

組織レベルでのオプトアウト処理

組織全体のオプトアウトを処理するには、AWS Organizations のオプトアウトポリシーを使用します。chimesdkvoiceanalytics サービス名を使用します。ポリシーの詳細については、「AWS Organizations ユーザーガイド」の「[AI サービスオプトアウトポリシー](#)」を参照してください。

ℹ Note

オプトアウトポリシーを使用するには、AWS アカウントが AWS Organizations によって一元管理されている必要があります。AWS アカウントでまだ組織を作成していない場合は、「AWS Organizations ユーザーガイド」の「[組織の作成と管理](#)」を参照してください。

データ保持

デフォルトでは、Amazon Chime SDK の音声分析は 3 年後に音声埋め込みを削除します。これは、人々の声が時間とともに変化するためであり、セキュリティを確保するためでもあります。[UpdateVoiceProfile](#) API を使用して、期限切れの音声埋め込みを更新できます。

[StartSpeakerSearchTask](#) および [StartVoiceToneAnalysisTask](#) は、それぞれの [GetSpeakerSearchTask](#) および [GetVoiceToneAnalytisTask](#) APIs から最大 7 日間取得できます。

から生成された音声埋め込み [StartSpeakerSearchTask](#) は、[CreateVoiceProfile](#) および [UpdateVoiceProfile](#) APIs を介して 24 時間永続的に利用でき、その後は削除され、使用できなくなります。

結果を削除する方法、および顧客からの同意の取り消しを処理する方法については、前のセクションを参照してください。

音声 API を使用して音声分析を実行する

下位互換性を保つため、Amazon Chime SDK 音声 API を使用して音声分析を開始および管理することは可能です。ただし、音声分析用のメディアインサイトパイプライン API のみが新機能を提供するため、代わりにこの API を使用することを強くお勧めします。

以下のセクションでは、音声 API とメディアインサイトパイプライン API の違いについて説明します。

タスクの停止

Voice Connector を使用して音声分析タスクを開始し、[UpdateMediaInsightsPipelineStatus](#) API を使用してパイプラインを一時停止すると、タスクは引き続き実行されます。タスクを停止するには、[StopSpeakerSearchTask](#) および [StopVoiceToneAnalysisTask](#) APIs を呼び出す必要があります。

通知の違いについて

音声 API を使用して音声分析を実行する場合、通知はメディアインサイトパイプラインによって生成される通知とは異なります。

- 音声分析準備完了イベントは、音声 API を使用して開始されたタスクでのみ使用できます。
- 音声分析タスクを通話に関連付けるには、voiceConnectorId、transactionId、または callId フィールドを使用する必要があります。メディアインサイトパイプラインを使用して音声分析を実行する場合は、mediaInsightsPipelineId および streamArn または channelId フィールドを使用してタスクを通話に関連付けます。

以下のトピックでは、音声 API で通知を使用する方法を説明します。

トピック

- [音声分析準備完了イベント](#)
- [スピーカー検索イベント](#)
- [音声トーン分析イベント](#)

音声分析準備完了イベント

音声分析準備完了イベントの詳細タイプは VoiceAnalyticsStatus です。

Amazon Chime SDK Voice Connector を使用して分析タスクを開始します。音声分析準備完了イベントを受信したら、次のプロパティで識別される通話のスピーカー検索または音声トーン分析タスクをトリガーできます。

- `voiceConnectorId`
- `transactionId`

Note

この通知は、音声分析が有効で、Voice Connector に関連付けられているメディアインサイトパイプライン設定がある場合にのみ提供されます。この通知は、顧客が `CreateMediaInsightsPipeline` API を呼び出し、メディアパイプライン SDK を介してスピーカー検索タスクまたは音声トーン分析タスクを起動した場合には提供されません。

Voice Connector から返される SIP ヘッダーには、`transactionId` が含まれます。SIP ヘッダーにアクセスできない場合、`AnalyticsReady` 通知イベントには `voiceConnectorId` と `transactionId` も含まれます。これにより、プログラムで情報を受信し [StartSpeakerSearchTask](#)、または [StartVoiceToneAnalysisTask](#) APIs を呼び出すことができます。

音声分析の処理準備が整うと、Voice Connector は `"detailStatus": "AnalyticsReady"` のイベントを JSON 本文として通知ターゲットに送信します。Amazon SNS または Amazon SQS を使用している場合、その本文は Amazon SNS または Amazon SQS ペイロードの「Records」フィールドに表示されます。

次の例は一般的な JSON 本文を示しています。

```
{
  "detail-type": "VoiceAnalyticsStatus",
  "version": "0",
  "id": "Id-f928dfe3-f44b-4965-8a17-612f9fb92d59",
  "source": "aws.chime",
  "account": "123456789012",
  "time": "2022-08-26T17:55:15.563441Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "detailStatus": "AnalyticsReady",
```

```
    "callDetails": {
      "isCaller": false,
      "transactionId": "daaeb6bf-2fe2-4e51-984e-d0fbf2f09436",
      "voiceConnectorId": "fuiopl1fsv9caobmqf2vy7"
    }
  }
}
```

この通知により、音声分析タスク API を呼び出す前に、アプリケーションへの追加のコールバックをトリガーしたり、通知や同意などの法的要件を処理したりできます。

スピーカー検索イベント

スピーカー検索イベントの詳細タイプは `SpeakerSearchStatus` です。

Amazon Chime SDK Voice Connector は、以下のスピーカー検索イベントを送信します。

- ID の一致
- 音声埋め込みの生成

イベントには以下のステータスがあります。

- `IdentificationSuccessful` - 特定の音声プロファイルドメイン内で、高い信頼度スコアで一致する音声プロファイル ID を少なくとも 1 つ正常に識別しました。
- `IdentificationFailure` - 識別を実行できませんでした。原因: 発信者が 10 秒以上話していないか、音質が良くありません。
- `IdentificationNoMatchesFound` - 指定された音声プロファイルドメインで、高い確率で一致する音声プロファイルが見つかりませんでした。発信者が新規であるか、声が変わっている可能性があります。
- `VoiceprintGenerationSuccessful` - システムは、20 秒間の無音ではない音声を使用して音声埋め込みを生成しました。
- `VoiceprintGenerationFailure` - システムは音声埋め込みを生成できませんでした。原因: 発信者が 20 秒以上話していないか、音質が良くありません。

ID の一致

特定の `transactionId` に対して [StartSpeakerSearchTask](#) API が呼び出されると `transactionId`、Voice Connector サービスは 10 秒間無音でない音声の後に識別一致通知を返します。このサービスは、

上位 10 件の一致を、音声プロファイル ID と [0, 1] の範囲の信頼スコアと共に返します。信頼度スコアが高いほど、通話中のスピーカーが音声プロファイル ID と一致する可能性が高くなります。機械学習モデルで一致が見つからなかった場合、通知の `detailStatus` フィールドには `IdentificationNoMatchesFound` が含まれます。

次の例は、一致が見つかった場合の通知を示しています。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-111122223333",
  "detail-type": "SpeakerSearchStatus",
  "service-type": "VoiceAnalytics",
  "source": "aws.chime",
  "account": "111122223333",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "taskId": "uuid",
    "detailStatus": "IdentificationSuccessful",
    "speakerSearchDetails" : {
      "results": [
        {
          "voiceProfileId": "vp-505e0992-82da-49eb-9d4a-4b34772b96b6",
          "confidenceScore": "0.94567856",
        },
        {
          "voiceProfileId": "vp-fba9cbfa-4b8d-4f10-9e41-9dfdd66545ab",
          "confidenceScore": "0.82783350",
        },
        {
          "voiceProfileId": "vp-746995fd-16dc-45b9-8965-89569d1cf787",
          "confidenceScore": "0.77136436",
        }
      ]
    },
    "isCaller": false,
    "voiceConnectorId": "abcdef1ghij2klmno3pqr4",
    "transactionId": "daaeb6bf-2fe2-4e51-984e-d0fbf2f09436"
  }
}
```

音声埋め込みの生成

無音でない音声さらに 10 秒間続くと、Voice Connector は通知ターゲットに音声埋め込み生成通知を送信します。音声プロファイルに新しい音声埋め込みを登録したり、音声プロファイルに既に登録されている声紋を更新したりできます。

次の例は、一致が見つかった場合の通知を示しています。この場合、関連する音声プロファイルを更新できます。

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-111122223333",
  "detail-type": "SpeakerSearchStatus",
  "service-type": "VoiceAnalytics",
  "source": "aws.chime",
  "account": "111122223333",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "taskId": "guid",
    "detailStatus": "VoiceprintGenerationSuccess",
    "isCaller": false,
    "transactionId": "12345678-1234-1234",
    "voiceConnectorId": "abcdefghijklmno3pqr"
  }
}
```

音声トーン分析イベント

音声トーン分析イベントの詳細タイプは VoiceToneAnalysisStatus です。分析では次のステータスが返されます。

- VoiceToneAnalysisSuccessful — 発信者とエージェントの声を分析し、感情 (肯定的、否定的、または中立的) を予測できました。
- VoiceToneAnalysisFailure - トーン分析を実行できませんでした。これは、発信者が 10 秒間話さずに電話を切ったり、音質が悪くなりすぎたりした場合に発生する可能性があります。
- VoiceToneAnalysisCompleted - ユーザーとエージェントの声を分析して、通話全体の感情を予測できました。これは音声トーン分析が終了したときに送信される最後のイベントです。

次の例は、典型的な音声トーン分析イベントを示しています。

```
{
  "detail-type": "VoiceToneAnalysisStatus",
  "service-type": "VoiceAnalytics",
  "source": "aws.chime",
  "account": "216539279014",
  "time": "2022-08-26T17:55:15.563441Z",
  "region": "us-east-1",
  "detail": {
    "taskId": "uuid",
    "detailStatus": "VoiceToneAnalysisSuccessful",
    "voiceToneAnalysisDetails": {
      "currentAverageVoiceTone": {
        "startTime": "2022-08-26T17:55:15.563Z",
        "endTime": "2022-08-26T17:55:45.720Z",
        "voiceToneLabel": "neutral",
        "voiceToneScore": {
          "neutral": "0.83",
          "positive": "0.13",
          "negative": "0.04"
        }
      },
      "overallAverageVoiceTone": {
        "startTime": "2022-08-26T16:23:13.344Z",
        "endTime": "2022-08-26T17:55:45.720Z",
        "voiceToneLabel": "positive",
        "voiceToneScore": {
          "neutral": "0.25",
          "positive": "0.65",
          "negative": "0.1"
        }
      }
    },
    "isCaller": true,
    "transactionId": "daaeb6bf-2fe2-4e51-984e-d0fbf2f09436",
    "voiceConnectorId": "fuiopl1fsv9caobmqf2vy7"
  },
  "version": "0",
  "id": "Id-f928dfe3-f44b-4965-8a17-612f9fb92d59"
}
```

通話分析のサービスクォータ

Amazon Chime SDK 通話分析のサービスクォータを次の表に示します。

通話分析のリージョンについては、このガイドで前述した「[利用できるリージョン](#)」で詳しく確認できます。

Amazon Chime SDK の通話分析と音声分析のサービスクォータは、次のとおりです。

リソース	デフォルトの制限	調整可能
リージョンあたりのメディアインサイトパイプライン設定	100	はい
リージョンあたりのアクティブなメディアインサイトパイプライン	20	はい
リージョンあたりの音声プロファイルドメイン	3	はい
音声プロファイルドメインあたりの音声プロファイル	20	はい
リージョンあたりのアクティブなスピーカー検索タスク	25	はい
リージョンあたりのアクティブな音声トーン分析タスク	25	はい
リージョンあたりの音声分析機能を備えたアクティブな Voice Connector コール	25	はい
トランザクション ID あたりの各 Voice Connector 通話に対するアクティブなスピーカー検索タスク	1	いいえ

リソース	デフォルトの制限	調整可能
トランザクション ID あたりの各 Voice Connector コールに対するアクティブな音声トーン分析タスク	1	いいえ
音声プロフィールドメインあたりの同時 API コールの最大数	1	はい
音声プロフィールあたりの同時 API コールの最大数	1	はい
スピーカー検索タスクあたりの同時 API コールの最大数	1	はい
音声トーン分析タスクあたりの同時 API コールの最大数	1	はい

API のレートおよびクォータの詳細については、「AWS 全般リファレンス」の「[Amazon Chime SDK エンドポイントとクォータ](#)」を参照してください。

Note

いずれかのリージョンでクォータを超えると、リソースの制限を超えました例外メッセージが表示されます。AWS コンソールの Service Quotas ページを使用して引き上げをリクエストするか、[カスタマーサポート担当者](#) にお問い合わせください。

一部のコール分析 APIs 他 AWS サービスのリソースと API リクエストを作成します。これらの追加分はアカウントのクォータにカウントされます。クォータをリクエストしたり、通話分析から transactions-per-second 引き上げたりする場合は、それらの他の AWS サービスの引き上げもリクエストする必要があります。そうしなかった場合、リクエストがスロットリングされて失敗する可能性があります。

Android 用 Amazon Chime SDK クライアントライブラリを使用する

現在、Android 用の Amazon Chime SDK クライアントライブラリは GitHub で提供されています。<https://github.com/aws/amazon-chime-sdk-android> にアクセスしてください。

iOS 用の Amazon Chime SDK クライアントライブラリを使用する

現在、iOS 用の Amazon Chime SDK クライアントライブラリは GitHub で提供されています。<https://github.com/aws/amazon-chime-sdk-ios> にアクセスしてください。

での Amazon Chime SDK クライアントライブラリの使用 JavaScript

このガイドでは、用の Amazon Chime SDK クライアントライブラリの概念的な概要と JavaScript、重要なサーバーおよびクライアントコンポーネントのサンプルコードについて説明します。

トピック

- [Amazon Chime SDK アプリケーションのコンポーネント](#)
- [主要なコンセプト](#)
- [サービスアーキテクチャ](#)
- [ウェブアプリケーションのアーキテクチャ](#)
- [サーバーアプリケーションのアーキテクチャ](#)
- [Amazon Chime SDK メディアコントロールプレーン](#)
- [Amazon Chime SDK メディアデータプレーン](#)
- [ウェブアプリケーションコンポーネントのアーキテクチャ](#)
- [サーバーアプリケーションを構築する](#)
- [クライアントアプリケーションをビルドする](#)
- [背景フィルタをクライアントアプリケーションに統合する](#)

Amazon Chime SDK アプリケーションのコンポーネント

Amazon Chime SDK アプリケーションにリアルタイムの音声、動画、画面共有機能を埋め込むには、以下のコンポーネントを使用します。

- 用の Amazon Chime SDK クライアントライブラリ JavaScript。ブラウザまたは Electron ウェブアプリケーションに統合するクライアント側の SDK。そのためには、[Amazon Chime SDK for JavaScript NPM パッケージ](#)を依存関係として追加します。このパッケージでは、[MediaDevices](#)および [WebRTC](#) APIsを活用して会議に参加し、音声、動画、コンテンツを他の参加者と共有します。これは、さまざまな種類のメディアを管理したり、それらのリソースをアプリケーションのユーザーインターフェイスにバインドしたりするためのコントロールサーフェスになります。
- AWS SDK は、サーバーアプリケーションがウェブアプリケーションからの会議リクエストを認証および承認するために使用する Amazon Chime SDK API です。AWS SDK では、会

議[chime:CreateMeeting](#)や参加者のリソースを作成および管理[chime:CreateAttendee](#)するための やなどの API アクションが用意されています。

他の AWS リソースと同様に、AWS Identity and Access Management (IAM) サービスはこれらのアクションへのアクセスを設定します。AWS SDK は[複数のプログラミング言語](#)で使用でき、サーバーアプリケーションから AWS SDK Chime API を呼び出す手間が省けます。アプリケーションが現在サーバーアプリケーションを使用していない場合は、[デモ/サーバーレス](#)フォルダに含まれている AWS CloudFormation テンプレートから開始できます。このデモでは、AWS SDK Chime API を使用する AWS Lambdaベースのサーバーレスアプリケーションを構築する方法を示します。

- Amazon Chime SDK メディアサービスは、用の Amazon Chime SDK クライアントライブラリが会議に接続 JavaScript するために使用するオーディオ、ビデオ、シグナリングを提供します。メディアサービスは世界中で利用でき、音声ミキシング、動画転送、TURN リレーを使用した NAT トラバーサルをサポートしています。Amazon Chime サービスチームは、これらのサービスをデプロイ、監視、管理します。メディアサービスは 99.77.128.0/18 という単一の IP アドレス範囲でホストされ、ポート TCP/443 と UDP/3478 を使用して IT 管理者のファイアウォール設定を簡素化します。最後に、これらのサービスでは [AWS グローバルクラウドインフラストラクチャ](#)が活用されています。

主要なコンセプト

会議とユーザーを作成および管理する方法を完全に把握するには、以下の概念を理解する必要があります。

[Meeting](#) – マルチパーティーメディアセッション。すべての会議には一意の会議識別子が付与されます。サポートされている AWS リージョンのいずれかで会議を作成できます。会議を作成すると、メディア URL のリストが返されます。これらは会議に参加するうえで必要なデータの重要な部分であり、会議に参加しようとしているすべてのユーザーにそのデータを配布する必要があります。

[Attendee](#) – マルチパーティーメディアセッションに参加しようとしているユーザー。すべての参加者には、一意の識別子、開発者のシステムでユーザーを参加者に割り当てるために渡せる外部ユーザー識別子、会議へのアクセスを許可する署名済みの参加トークンが付与されます。

[MeetingSession](#) および [\(DefaultMeetingSession\)](#) — 会議内の各ユーザーのセッション JavaScript を表すの Amazon Chime SDK クライアントライブラリのルートオブジェクト。ウェブアプリケーションは、まず適切な会議 MeetingSession と参加者の情報を使用してインスタンス化して設定します。

[MeetingSessionConfiguration](#) – 会議セッションへの参加に必要な会議と参加者のデータを保存します。このデータは、サーバーアプリケーションによって行われた `CreateMeeting` および `CreateAttendee` API コールのレスポンスです。サーバーアプリケーションはこのデータをウェブアプリケーションに渡します。ウェブアプリケーションはそれを使用して `MeetingSession` をインスタンス化します。

[DeviceController](#) (`DefaultDeviceController`) – ユーザーのシステムで使用可能なオーディオおよびビデオデバイスのリストを列挙するために使用されます。会議中にデバイスコントローラーを使用して、アクティブなデバイスを切り替えることもできます。

[AudioVideoFacade](#) (`DefaultAudioVideoFacade`) – 会議を強化する主要なインターフェイス。会議の開始、制御、終了を行う API を提供します。さらに、参加または退出するユーザー、ミュートまたはミュート解除されているユーザー、活発に発言しているユーザー、接続状態が悪いユーザーを追跡することにより、ユーザーエクスペリエンスの変化を促す重要なイベント (参加者の一覧など) をリッスンする API も提供します。これらの API を使用して、音声コントロール HTML 要素を会議の音声出力にバインドし、選択した音声出力デバイスで再生することもできます。

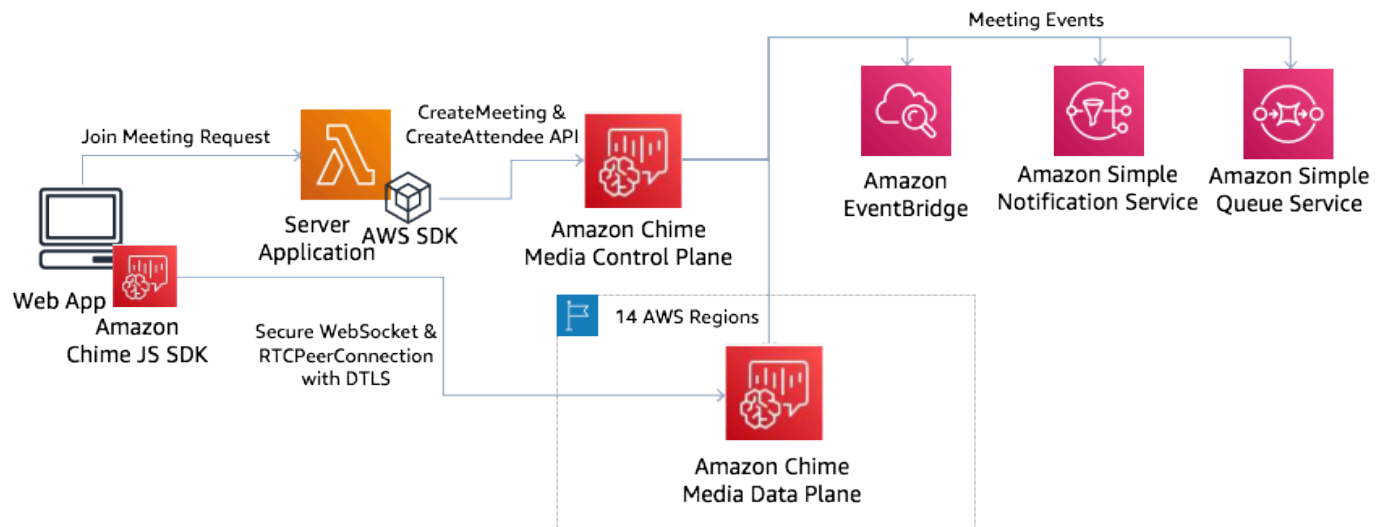
[ActiveSpeakerDetectorFacade](#) (`DefaultActiveSpeakerDetector`) – アクティブなスピーカーイベントをサブスクライブする API。時間の経過と共に、マイクの音量順に参加者のリストを定期的に返します。発言中のスピーカーのポリシーは、必要に応じて上書きしたり、微調整したりできます。

[ContentShareController](#) (`DefaultContentShareController`) – APIs。また、ライフサイクルイベントをリッスンしてコンテンツ共有のステータスを追跡する API としても利用できます。

[Logger](#) ([ConsoleLogger](#)) – コンソールログを活用するために使用されるインターフェイス、またはロガーオブジェクトを渡して現在のログ記録実装を上書きし、Amazon Chime SDK からさまざまなレベルのログを取得します。

サービスアーキテクチャ

この大まかなアーキテクチャ図は、「」にリストされているコンポーネントが他の AWS サービスとどのように [主要なコンセプト](#) 相互作用し、連携するかを示しています。



ウェブアプリケーションのアーキテクチャ

コンテンツ配信ネットワークからウェブアプリケーションを提供し、ユーザーがブラウザで URL に移動した際に読み込むことができます。また、ユーザーがマシンにインストールするプラットフォームネイティブな Electron アプリケーションにラップすることもできます。

新規または既存の会議に参加するために、ウェブアプリケーションでサーバーアプリケーションに REST リクエストを出します。通常、リクエストには、アプリケーションで他の API リクエストに使用する認証トークンまたは Cookie が含まれます。ウェブクライアントを設計して、リージョンヒントをサーバーに送信することもできます。後者は、MediaRegion パラメータをに提供するとき使用できます [chime:CreateMeeting](#)。ウェブアプリケーションは、<https://l.chimenearest-media-region.aws> エンドポイントに **HTTP GET** リクエストを行うことで、最も近いメディアサービスリージョンを特定できます。

サーバーアプリケーションのアーキテクチャ

サーバーでクライアントからリクエストを受け取ったら、まず、ユーザーに会議を開始または会議に参加する権限があることを確認します。サーバーは、選択した言語で埋め込み AWS SDK を使用して、グローバルメディアコントロールプレーンに対して [chime:CreateMeeting](#) および [chime:CreateAttendee](#) API コールを行います。これにより、サポートされている AWS リージョンのいずれかで会議と参加者が作成されます。これらのリクエストを行うには、サービスに適切な IAM ユーザーまたはロールが必要です。次に、IAM ユーザーとロールには [AmazonChimeSDK](#) ポリシーが必要です。

Amazon Chime SDK メディアコントロールプレーン

Amazon Chime SDK メディアコントロールプレーンはグローバルで、us-east-1 からホストされており [chime:CreateMeeting](#)、データプレーン全体で会議と参加者のリソースを作成および管理するために使用する および [chime:CreateAttendee](#) APIs を提供します。認証情報を検証し、セッションがリクエストされたリージョンのデータプレーンでブートストラップされるようにします。

また、コントロールプレーンは、[Amazon Chime SDK Events](#) を Amazon EventBridge、Amazon Simple Queueing Service (SQS)、Amazon Simple Notification Service (SNS) などの通知メカニズムにトリガーします。AWS は、サービスを常にモニタリングし、負荷の増加に応じて自動的にスケールアップします。API は、不透明なユーザー識別子のみを受け入れ、ユーザーデータを受け入れないように設計されているため、データ主権の要件に準拠しています。

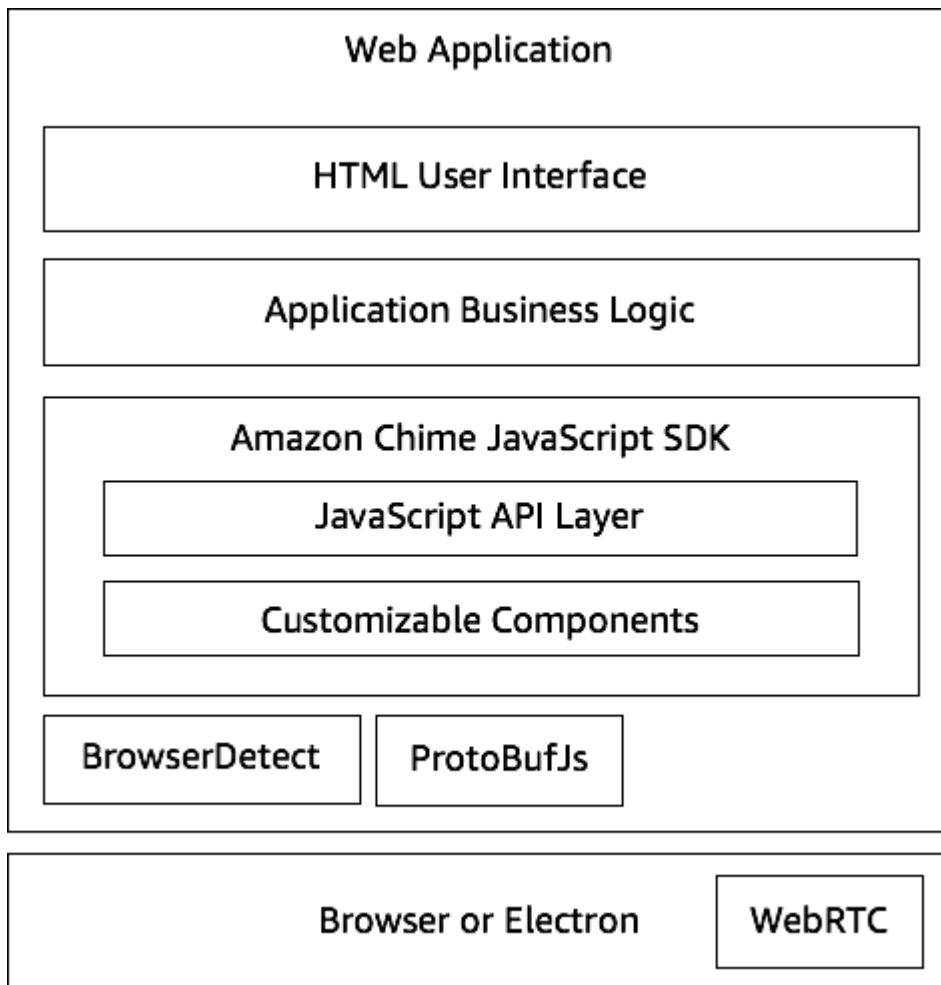
Amazon Chime SDK メディアデータプレーン

どのコントロールプレーンリージョンでも、すべての AWS リージョンで会議を作成できます。メディアデータプレーンは、すべての AWS リージョンで使用できます。これには音声ミキシングサービス、動画転送サービス、TURN サービス、セッション初期化プロトコル (SIP) 相互運用サービスが含まれます。サービスは常に監視されており、負荷が増えると自動的にスケールするように設計されています。詳細については、「[Amazon Chime SDK メディアリージョン](#)」を参照してください。

[リージョンとアベイラビリティゾーン](#)の詳細については、リージョンとアベイラビリティゾーンのリストを参照してください。

ウェブアプリケーションコンポーネントのアーキテクチャ

この図は、Amazon Chime SDK ウェブクライアントアプリケーションのアーキテクチャを示しています。



ウェブアプリケーションは通常、アプリケーションのビジネスロジックレイヤーを基盤とする HTML および CSS のユーザーインターフェイスレイヤーで構成されています。ウェブアプリケーションは、プレーン HTML および で構築することも JavaScript、React や Angular などの UI フレームワークを使用することもできます。

ウェブアプリケーションのビジネスロジックレイヤーは、一連の API JavaScript を介しての Amazon Chime SDK クライアントライブラリとやり取りします。JavaScript APIs [DefaultMeetingSession](#) は SDK のルートオブジェクトです。サーバーアプリケーションを構築するときは [MeetingSessionConfiguration](#)、会議と参加者の情報を使用してサーバーアプリケーションを初期化し、会議に参加するために使用します。DefaultMeetingSession また、は を公開します。これにより [AudioVideoFacade](#)、ビジネスロジックレイヤーはアクションを実行し、セッションの基盤となる状態が変更されたときにユーザーインターフェイスを更新するコールバックを登録できます。

の Amazon Chime SDK クライアントライブラリ JavaScript はオープンソースで、必要に応じて書きできるカスタマイズ可能なコンポーネントのセットがあります。デフォルトの実装では、デモ用の MeetingV2 アプリケーションなど、完全なユニファイドコミュニケーションアプリケーションを構

築できます。の Amazon Chime SDK クライアントライブラリ JavaScript は、他の 2 つのライブラリに依存します。

- [Browser-Detect](#) : ブラウザのタイプと機能を識別します。
- [ProtoBufJs](#) は、メディアセッションへの参加に必要なシグナリングコマンドとレスポンスをエンコードおよびデコードします。

また、Amazon Chime SDK では、ブラウザまたは Electron アプリケーションを活用して、音声および動画セッション用のデバイス管理 API と WebRTC 実装を提供しています。

のソース Amazon Chime SDK クライアントライブラリ JavaScript は にありますが TypeScript、コン TypeScript パイラを使用して にコンパイルできます JavaScript。その後、Webpack などのモジュールバンドラーを使用してバンドルできます。ベストプラクティスとして、NPM レジストリ JavaScript から 用の Amazon Chime SDK CommonJS 環境で使用します。 は、[HTML にスクリプトタグとして直接含める場合に備えて、Amazon Chime SDK をミニファイド JS ファイルにバンドルするためのロールアップスクリプト](#) AWS も提供します。

サーバーアプリケーションを構築する

以下のセクションの情報では、Amazon Chime SDK サーバーアプリケーションを構築する方法について説明します。各セクションでは必要に応じてコード例が提供されており、ニーズに合わせてそのコードを適応させることができます。

トピック

- [IAM ユーザーまたはロールを作成する](#)
- [APIs を呼び出すように AWS SDK を設定する](#)
- [会議を作成する](#)
- [参加者を作成する](#)
- [クライアントにレスポンスを送信する](#)

IAM ユーザーまたはロールを作成する

IAM ユーザーとして、またはユースケースに適したロールでユーザーを作成します。その後、作成したユーザーに以下のポリシーを割り当てます。これにより、サーバーアプリケーションに埋め込まれた AWS SDK に必要なアクセス許可が確実に付与されます。次に、それによって、会議および参加者のリソースに対してライフサイクル操作を実行できるようになります。

```
// Policy ARN:      arn:aws:iam::aws:policy/AmazonChimeSDK
// Description:    Provides access to Amazon Chime SDK operations
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "chime:CreateMeeting",
        "chime>DeleteMeeting",
        "chime:GetMeeting",
        "chime:ListMeetings",
        "chime:CreateAttendee",
        "chime:BatchCreateAttendee",
        "chime>DeleteAttendee",
        "chime:GetAttendee",
        "chime:ListAttendees"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

APIs を呼び出すように AWS SDK を設定する

このコードサンプルは、AWS SDK に認証情報を渡し、リージョンとエンドポイントを設定する方法を示しています。

```
AWS.config.credentials = new AWS.Credentials(accessKeyId, secretAccessKey, null);
const chime = new AWS.Chime({ region: 'us-east-1' });
chime.endpoint = new AWS.Endpoint('https://service.chime.aws.amazon.com/console');
```

会議を作成する

[CreateMeeting](#) API コールは、開発者ClientRequestTokenが一意性コンテキストを渡すことができる必須パラメータであるを受け入れます。また、会議用に選択するメディアサービスデータプレーンのリージョンを表す MediaRegion、会議の主催者を表す不透明な識別子を渡すために必要に応じて使用する MeetingHostId、会議のライフサイクルイベント

を受信するための `NotificationsConfiguration` などのオプションパラメータも受け入れます。デフォルトでは、Amazon はイベントを `EventBridge` 配信します。オプションで、`NotificationsConfiguration` で `SQS` キューの ARN または `SNS` トピックの ARN を渡してイベントを受信することもできます。API は、一意の `MeetingId` を含む `Meeting` オブジェクト、`MediaRegion`、一連のメディアの URL を含む `MediaPlacement` オブジェクトを返します。

```
meeting = await chime.createMeeting({
  ClientRequestToken: clientRequestToken,
  MediaRegion: mediaRegion,
  MeetingHostId: meetingHostId,
  NotificationsConfiguration: {
    SqsQueueArn: sqsQueueArn,
    SnsTopicArn: snsTopicArn
  }
}).promise();
```

参加者を作成する

会議を作成した後、メディアセッションに参加しようとする各ユーザーを表した参加者リソースを作成します。[CreateAttendee](#) API は以下を実行します。

- ユーザーを追加する会議の `MeetingId`。
- `ExternalUserId`。ID システムからの不透明なユーザー識別子であればどれでも構いません。

例えば、`Active Directory (AD)` を使用する場合、これは `AD` 内のユーザーのオブジェクト ID になります。`ExternalUserId` は、クライアント SDK から参加者イベントを受信するとクライアントアプリケーションに返されるため、有用です。これにより、クライアントアプリケーションで会議に参加または退席した人物を把握し、表示名、メールアドレス、画像など、そのユーザーに関する追加情報をサーバーアプリケーションから取得することができます。

`CreateAttendee` API を呼び出すと、`Attendee` オブジェクトが生成されます。オブジェクトには、サービスによって `AttendeeId` 生成された一意の、渡 `ExternalUserId` された、および参加者がその期間中、または [DeleteAttendee](#) API が参加者を削除するまで会議に `アクセスJoinToken` できるようにする署名が含まれます。

```
attendee = await chime.createAttendee({
```

```
MeetingId: meeting.MeetingId,  
ExternalUserId: externalUserId,  
}).promise();
```

クライアントにレスポンスを送信する

会議および参加者リソースを作成したら、サーバーアプリケーションで会議および参加者オブジェクトをエンコードしてクライアントアプリケーションに送り返す必要があります。クライアントは、の Amazon Chime SDK クライアントライブラリをブートストラップし JavaScript、参加者がウェブまたは Electron ベースのアプリケーションから会議に正常に参加できるようにするために、これらの情報を必要とします。

クライアントアプリケーションをビルドする

クライアントアプリケーションを構築するには、「」の「[Amazon Chime JavaScript SDK API の概要](#)」に記載されているステップに従います GitHub。概要では、必要に応じてコード例が提供されています。

背景フィルタをクライアントアプリケーションに統合する

このセクションでは、背景ぼかし 2.0 と背景置換 2.0 を使用して動画の背景をプログラムでフィルタリングする方法について説明します。動画ストリームに背景フィルタを追加するには、VideoFxConfig オブジェクトを含む VideoFxProcessor を作成します。次に、そのプロセッサを VideoTransformDevice に挿入します。

背景フィルタープロセッサは、TensorFlow Lite 機械学習モデル、JavaScript ウェブワーカー、および WebAssembly を使用して、ビデオストリーム内の各フレームの背景にフィルターを適用します。これらのアセットは、VideoFxProcessor を作成すると実行時にダウンロードされます。

[のブラウザデモアプリケーション GitHub](#)は、新しい背景ぼかしフィルターと置換フィルターを使用します。これらを試すには、npm run start でデモを起動して、会議に参加し、カメラをクリックして動画を有効にします。[フィルタを適用] メニュー



(を開き、[背景ぼかし 2.0] または [背景置換 2.0] オプションのいずれかを選択します。

トピック

- [背景フィルタの使用について](#)
- [コンテンツセキュリティポリシーを使用する](#)
- [アプリケーションを背景フィルタに追加する](#)
- [背景フィルタの例](#)

背景フィルタの使用について

背景フィルタは、CPU に負荷がかかる場合と GPU に負荷がかかる場合があります。一部のモバイルデバイスと低スペックのラップトップまたはデスクトップコンピュータでは、複数の動画ストリームで同時にフィルタを実行できない場合があります。

SIMD のサポート

背景フィルタは、SIMD (単一命令・複数データ) をサポートする環境で効率が向上します。SIMD を有効にすると、特定の複雑度レベルでフィルタの CPU 使用量が少なくなります。SIMD をサポートしていないブラウザを実行中の低消費電力デバイスでは、背景フィルタが実行されない場合があります。

WebGL2 のサポート

`VideoFxProcessor` オブジェクトでは、クライアントデバイスの GPU にアクセスするために WebGL2 をサポートしているブラウザが必要です。

コンテンツ配信と帯域幅

Amazon コンテンツ配信ネットワークは、実行時にバックグラウンドフィルタ用の machine-learning-model ファイルをロードします。これにより、アプリケーションの一部としてファイル一式を提供することなく、低レイテンシーのグローバル配信が可能になります。ただし、モデルファイルを読み込むと、アプリケーションの一部にレイテンシーが追加される可能性があります。その影響を軽減するために、ブラウザではモデルファイルを無期限にキャッシュします。そのキャッシュにより、後続の読み込みが大幅に迅速化されます。ベストプラクティスとして、サポートされているブラウザを確認した後、ユーザーがレイテンシーに気付いていないと思われるときに背景フィルタリソースを作成することが挙げられます。例えば、ユーザーがロビーで待っている間や、デバイスピッカーを使用している間にモデルファイルをダウンロードすることができます。

アプリケーションを以下に接続する必要があります。

- Amazon Chime SDK メディアサービス。
- HTTPS CloudFront 経由の Amazon (ポート 443)。

すべてのリクエストは `sdkassets.chime.aws` のサブドメインに送信されます。コンテンツ配信ネットワークにアクセスできないか、[コンテンツセキュリティポリシー](#) に正しいドメインを含めていないアプリケーションでは、サポートのチェックに失敗し、フィルタを使用できなくなります。

CloudFrontの IP アドレス範囲の詳細については、「[Amazon CloudFront デベロッパーガイド](#)」の [CloudFront 「エッジサーバーの場所と IP アドレス範囲」](#) を参照してください。

ブラウザの互換性

次の表では、背景フィルタをサポートするブラウザとバージョンの一覧が表示されています。

ブラウザ	サポートされている最小バージョン
Firefox	76 以降
Chromium ベースのブラウザと環境 (Edge や Electron を含む)	78 以降
Android Chrome	110 以降
macOS 版 Safari	16.3 以降

ブラウザ	サポートされている最小バージョン
iOS 版 Safari (iPhone、iPad)	16.x
iOS 版 Chrome	110.0.0.x.x
iOS 版 Firefox (iPhone iPad)	16.x

VideoFxProcessor オブジェクトのバージョン 3.14 は Android をサポートしています。3.14 より前のバージョンの Android デバイスをサポートにするには、BackgroundBlurVideoFrameProcessor および BackgroundReplacementVideoFrameProcessor オブジェクトを使用します。使用の詳細については、「」の[backgroundfilter_video_processor](#)ページを参照してください GitHub。

コンテンツセキュリティポリシーを使用する

最新のウェブアプリケーションでは、特定の種類の攻撃からユーザーを保護するために、コンテンツセキュリティポリシーが使用されています。を使用するアプリケーションには、次のポリシーディレクティブを含める VideoFxProcessor 必要があります。これらのディレクティブにより、Amazon Chime SDK で実行時に必要なリソースにアクセスできます。

トピック

- [必須のコンテンツセキュリティポリシーディレクティブ](#)
- [クロスオリジンオープナーポリシーを使用する](#)

必須のコンテンツセキュリティポリシーディレクティブ

次のコンテンツセキュリティポリシーディレクティブを使用する必要があります。

- `script-src:` では、動画処理コードを読み込むために `blob: https://*.sdkassets.chime.aws` を追加し、それを実行できるようにするために `wasm-unsafe-eval` を追加します。
- `script-src-elem:` では、ソースから動画処理コードを読み込むために `blob: https://*.sdkassets.chime.aws` を追加します。
- `worker-src:` オリジン JavaScript 間でワーカーをロード `blob: https://*.sdkassets.chime.aws` するには、 を追加します。

これらのエントリのいずれかを省略したり、HTTP ヘッダーと `http-equiv` メタタグを使用してポリシーを指定し、これらのいずれかを交差によって誤って除外したりすると、背景フィルタは初期化できなくなります。背景フィルタはサポートされていないように表示されます。または、動作しない動画フレームプロセッサが作成されます。ブラウザコンソールに次のようなエラーが表示されます。

```
Refused to connect to
'https://static.sdkassets.chime.aws/bgblur/workers/worker.js...'
because it violates the document's content security policy.
```

必須のスクリプトポリシーディレクティブ

機能するには、`VideoFxProcessor` クラスは実行時に Amazon コンテンツ配信ネットワークから JavaScript クラスをロードする必要があります。これらのクラスでは `WebGL2` を使用して動画の後処理を実装します。アプリケーションでこれらのクラスを取得して実行できるようにするには、次のディレクティブを含める必要があります。

- `script-src 'self' blob: https://*.sdkassets.chime.aws`
- `script-src-elem 'self' blob: https://*.sdkassets.chime.aws`

Note

Safari と Firefox を完全にサポートするには、`script-src` および `script-src-elem` ディレクティブを使用する必要があります。

ワーカーポリシーディレクティブ

は、ウェブワーカースレッドを実行するための BLOB として JavaScript クラスを `VideoFxProcessor` ロードします。このスレッドでは、機械学習モデルを使用して動画を処理します。このワーカーを取得して使用するためのアクセス権をアプリケーションに付与するには、次のディレクティブを含めてください。

```
worker-src 'self' blob: https://*.sdkassets.chime.aws
```

WebAssembly ポリシー

は、同じ Amazon 所有のコンテンツ配信ネットワークから WebAssembly (WASM) モジュールを `VideoFxProcessor` ロードします。Chrome 95 以降では、コンパイルされた WASM モジュール

を複数のモジュール境界をまたいで渡すことはできません。これらのモジュールの取得とインスタンス化を許可するには、`script-src`ディレクティブ'`wasm-unsafe-eval`'に を含めます。

のコンテンツセキュリティポリシーのドキュメントの詳細については WebAssembly、「」の[WebAssembly「コンテンツセキュリティポリシー」](#)を参照してください GitHub。

(オプション) 背景画像ポリシー

動的に読み込まれる背景画像を背景置換フィルタと共に使用するには、`VideoFxProcessor`でその画像にアクセスできる必要があります。そのためには、イメージをホストするドメインに `connect-src` ディレクティブを含めてください。

コンテンツセキュリティポリシーの例

次のポリシーの例では、`VideoFxProcessor` を使用できます。`connect-src` の定義は `VideoFxProcessor` に固有のものではありません。代わりに、Amazon Chime SDK ミーティングの音声と動画に関連付けられています。

```
<head>
  <meta http-equiv="Content-Security-Policy"
        content="base-uri 'self';
        connect-src      'self' https://*.chime.aws wss://*.chime.aws https://
*.amazonaws.com wss://*.chime.aws https://*.ingest.chime.aws;
        script-src      'self' blob: 'wasm-unsafe-eval' https://
*.sdkassets.chime.aws;
        script-src-elem 'self' blob: https://*.sdkassets.chime.aws;
        worker-src      'self' blob: https://*.sdkassets.chime.aws;">
</head>
```

コンテンツセキュリティポリシーのエラー

必要なディレクティブのいずれかを省略すると、`VideoFxProcessor` はインスタンス化されず、サポートされなくなります。その場合、ブラウザコンソールに次の (または同様の) エラーが表示されます。

```
Refused to connect to
'https://static.sdkassets.chime.aws/ml_media_fx/otherassets/worker.js'
because it violates the document's content security policy.
```

クロスオリジンオープナーポリシーを使用する

メモリの使用量を制限するには、モジュールで処理に `SharedArrayBuffer` を優先して使用します。ただし、このためにはウェブセキュリティを慎重に設定する必要があります。アプリケーション HTML を配信する際には、次のヘッダーを設定する必要があります。

```
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Embedder-Policy: require-corp
```

これらには対応するメタタグがないため、サーバーでこれらを設定する必要があります。これらのヘッダーを設定しない場合、背景フィルタによる RAM の使用量が少し増加する可能性があります。

背景フィルタは、CPU に負荷がかかる場合と GPU に負荷がかかる場合があります。一部のモバイルデバイスと低スペックのラップトップまたはデスクトップコンピュータでは、複数の動画ストリームで同時にフィルタを実行できない場合があります。

アプリケーションを背景フィルタに追加する

背景フィルタを追加する手順には、以下の大まかな手順に従います。

- サポートされているブラウザを確認します。
- 使用したい設定を備えた `VideoFxConfig` オブジェクトを作成します。
- その設定オブジェクトを使用して `VideoFxProcessor` オブジェクトを作成します。
- その `VideoFxProcessor` オブジェクトを `VideoTransformDevice` に含めます。
- `VideoTransformDevice` を使用して動画の入力を開始します。

Note

これらのステップを完了するには、まず以下を実行する必要があります。

- `Logger` を作成します。
- `MediaDeviceInfo` クラスの動画デバイスを選択します。
- `MeetingSession` に正常に参加します。

次のセクションの手順では、このプロセスを完了する方法について説明します。

トピック

- [フィルタを提供する前にサポート状況をチェックする](#)
- [VideoFxConfig オブジェクトの作成](#)
- [VideoFxProcessor オブジェクトの作成](#)
- [VideoFxProcessor オブジェクトの設定](#)
- [VideoTransformDevice オブジェクトの作成](#)
- [動画入力を開始する](#)
- [リソース使用率を調整する](#)

フィルタを提供する前にサポート状況をチェックする

Amazon Chime SDK には、サポートされているブラウザをチェックし、必要なアセットのダウンロードを試みる非同期の静的メソッドが用意されています。ただし、デバイスのパフォーマンスのチェックは行われません。ベストプラクティスとして、フィルタを提供する前に、ユーザーのブラウザとデバイスでフィルタのサポートが可能なことを常に確認することが挙げられます。

```
import {
  VideoFxProcessor
} from 'amazon-chime-sdk-js';

if (!await VideoFxProcessor.isSupported(logger)) {
  // logger is optional for isSupported
}
```

VideoFxConfig オブジェクトの作成

同一のオブジェクト内で `backgroundBlur` と `backgroundReplacement` の構成を定義できます。ただし、両方のフィルタに対して同時に `isEnabled` を `true` に設定することはできません。それは無効な設定です。

`VideoFxConfig` クラスでそれ自体の検証を行うことはありません。検証は次のステップで行われます。

次の例は、有効な `VideoFxConfig` を示しています。

```
const videoFxConfig: VideoFxConfig = {
  backgroundBlur: {
    isEnabled: false,
    strength: 'medium'
  },
};
```

```
backgroundReplacement: {
  isEnabled: false,
  backgroundImageURL: 'space.jpg',
  defaultColor: undefined,
}
}
```

次の表には、VideoFxConfig オブジェクトで指定可能な VideoFxProcessor プロパティが一覧表示されています。

背景ぼかしフィルタのプロパティ

プロパティ	型	説明
isEnabled	boolean	true の場合、フィルタで背景をぼかします。
strength	string	ぼかしの度合いを決定します。有効な値: low medium high

背景置換フィルタのプロパティ

プロパティ	型	説明
isEnabled	boolean	true の場合、フィルタで背景が置換されます。
backgroundImageURL	string	背景画像の URL。フィルタでは、現在の画面の大きさに合わせて画像のサイズを動的に変更します。https://... などの文字列や、data:image/jpeg;base64 などのデータ URL を使用できます。
defaultColor	string	000000 や FFFFFFFF などの 16 進数のカラー文字列、ま

プロパティ	型	説明
		たは black や white などの文字列。画像 URL を指定しない場合、プロセッサでは defaultColor を背景として使用します。defaultColor を指定しない場合、プロセッサはデフォルトの黒になります。

VideoFxProcessor オブジェクトの作成

VideoFxProcessor オブジェクトを作成すると、AWS サーバーはランタイムアセットをダウンロードするか、ブラウザキャッシュがアセットをロードします。ネットワークまたは CSP の設定によってアセットにアクセスできない場合、VideoFx.create オペレーションで例外をスローします。その結果 VideoFxProcessor、ビデオストリームには影響しない no-op プロセッサとして設定されます。

```
let videoFxProcessor: VideoFxProcessor | undefined = undefined;
try {
  videoFxProcessor = await VideoFxProcessor.create(logger, videoFxConfig);
} catch (error) {
  logger.warn(error.toString());
}
```

VideoFxProcessor.create では backgroundReplacement.backgroundImageUrl からの画像の読み込みも試行します。画像の読み込みに失敗すると、プロセッサで例外をスローします。プロセッサでは、構成が無効である、ブラウザがサポートされていない、ハードウェアの性能が低いなど、その他の理由でも例外をスローします。

VideoFxProcessor オブジェクトの設定

次の表では、設定できる VideoFxProcessor プロパティが一覧表示されています。表の下にある例は、一般的なランタイム設定を示しています。

背景ぼかし

背景ぼかしでは、以下のプロパティを取得します。

プロパティ	型	説明
isEnabled	boolean	true の場合、フィルタで背景をぼかします。
strength	string	ぼかしの度合いを決定します。有効な値: low medium high

背景置換

背景置換では、以下のパラメータを取得します。

プロパティ	型	説明
isEnabled	boolean	true の場合、フィルタで背景が置換されます。
backgroundImageURL	string	背景画像の URL。フィルタでは、現在の画面の大きさに合わせて画像のサイズを動的に変更します。https://... などの文字列や、data:image/jpeg;base64 などのデータ URL を使用できます。
defaultColor	string	000000 や FFFFFFFF などの 16 進数のカラー文字列、または black や white などの文字列。画像 URL を指定しない場合、プロセッサでは defaultColor を背景として使用します。defaultColor を指定しない場合、プロセッサはデフォルトの黒になります。

実行時に設定を変更する

`videoFxProcessor.setEffectConfig` パラメータを使用して、実行時に `VideoFxProcessor` 設定を変更できます。次の例は、背景置換を有効にして、背景ぼかを無効にする方法を示しています。

Note

一度に1つのタイプの背景置換のみを指定できます。値を `backgroundImageURL` または `defaultColor` のどちらかに指定してください。ただし、両方を指定することはできません。

```
videoFxConfig.backgroundBlur.isEnabled = false;
videoFxConfig.backgroundReplacement.isEnabled = true;
try {
  await videoFxProcessor.setEffectConfig(videoFxConfig);
} catch(error) {
  logger.error(error.toString())
}
```

`setEffectConfig` で例外をスローしても、以前の設定は引き続き有効です。`setEffectConfig` では、`VideoFxProcessor.create` で例外をスローした際と同様の条件の下で例外をスローします。

次の例は、動画の実行中に背景画像を変更する方法を示しています。

```
videoFxConfig.backgroundReplacement.backgroundImageURL = "https://my-domain.com/my-  
other-image.jpg";
try {
  await videoFxProcessor.setEffectConfig(videoFxConfig);
} catch(error) {
  logger.error(error.toString())
}
```

VideoTransformDevice オブジェクトの作成

次の例は、`VideoFxProcessor` を含む `VideoTransformDevice` オブジェクトの作成方法を示しています。

```
// assuming that logger and videoInputDevice have already been set
```



```
const videoTransformDevice = new DefaultVideoTransformDevice(  
  logger,  
  videoInputDevice,  
  [videoFxProcessor]  
);
```

動画入力を開始する

次の例は、VideoTransformDevice オブジェクトを使用して動画入力を開始する方法を示しています。

```
// assuming that meetingSession has already been created  
await meetingSession.audioVideo.startVideoInput(videoTransformDevice);  
meetingSession.audioVideo.start();  
meetingSession.audioVideo.startLocalVideoTile();
```

リソース使用率を調整する

VideoFxProcessor を作成する際に、オプションの processingBudgetPerFrame パラメータを指定し、フィルタで使用する CPU と GPU の量を制御できます。

```
let videoFxProcessor: VideoFxProcessor | undefined = undefined;  
const processingBudgetPerFrame = 50;  
try {  
  videoFxProcessor = await VideoFxProcessor.create(logger, videoFxConfig,  
    processingBudgetPerFrame);  
} catch (error) {  
  logger.warn(error.toString());  
}
```

VideoFxProcessor ではフレームの処理に時間がかかります。所要時間は、デバイス、ブラウザ、およびブラウザやデバイスで他に実行されている処理によって異なります。プロセッサではバジエットの概念を利用して、各フレームの処理とレンダリングに使用する時間の目安を定めます。

処理時間はミリ秒単位です。バジエットの使い方の一例として、1 秒は 1000 ミリ秒です。15 フレーム/秒の動画キャプチャを目安に設定すると、合計のバジエットは $1000\text{ms}/15\text{fps} = 66$ ミリ秒になります。上の例で示されているように、processingBudgetPerFrame パラメータに値 50 を指定することで、その 50% (33 ミリ秒) のバジエットを設定できます。

次に、VideoFxProcessor では指定されたバジエット内でフレームの処理を試みます。処理がバジエットを超えて実行されると、プロセッサではバジエット内に収まるように画質を下げます。プロ

セッサで画質を最低限まで下げ続けると、その時点で画質の低下は停止されます。この処理時間は継続的に測定されるため、(別のアプリケーションが終了して CPU が解放されるなどして) 利用可能なリソースが増えると、プロセッサではバジェットに達するまで、または画質が最大限になるまで、再び画質を上げます。

`processingBudgetPerFrame` に対して値を指定しない場合、`VideoFxProcessor` はデフォルトの 50 になります。

背景フィルタの例

次の例は、フィルタを実装する方法を示しています。

```
import {
  VideoFxConfig,
  VideoFxTypeConversion,
  VideoTransformDevice,
  DefaultVideoTransformDevice,
  Logger,
  VideoFxProcessor,
  MeetingSession
} from 'amazon-chime-sdk-js';

let videoTransformDevice: VideoTransformDevice | undefined = undefined;
let videoFxProcessor: VideoFxProcessor | undefined = undefined;

const videoFxConfig: VideoFxConfig = {
  backgroundBlur: {
    isEnabled: false,
    strength: "medium"
  },
  backgroundReplacement: {
    isEnabled: false,
    backgroundImageURL: 'space.jpg',
    defaultColor: undefined,
  }
}

export const addEffectsToMeeting = async (videoInputDevice: MediaDeviceInfo,
  meetingSession: MeetingSession, logger: Logger): Promise<void> => {
  try {
    videoFxProcessor = await VideoFxProcessor.create(logger, videoFxConfig);
  } catch (error) {
    logger.error(error.toString());
  }
}
```

```
        return;
    }

    videoTransformDevice = new DefaultVideoTransformDevice(
        logger,
        videoInputDevice,
        [videoFxProcessor]
    );

    await meetingSession.audioVideo.startVideoInput(videoTransformDevice);
}

export const enableReplacement = async (logger: Logger) => {
    videoFxConfig.backgroundBlur.isEnabled = false;
    videoFxConfig.backgroundReplacement.isEnabled = true;
    await updateVideoFxConfig(videoFxConfig, logger);
}

export const enableBlur = async (logger: Logger) => {
    videoFxConfig.backgroundReplacement.isEnabled = false;
    videoFxConfig.backgroundBlur.isEnabled = true;
    await updateVideoFxConfig(videoFxConfig, logger);
}

export const pauseEffects = async (logger: Logger) => {
    videoFxConfig.backgroundReplacement.isEnabled = false;
    videoFxConfig.backgroundBlur.isEnabled = false;
    await updateVideoFxConfig(videoFxConfig, logger);
}

export const setReplacementImage = async (newImageUrl: string, logger: Logger) => {
    videoFxConfig.backgroundReplacement.backgroundImageUrl = newImageUrl;
    videoFxConfig.backgroundReplacement.defaultColor = undefined;
    await updateVideoFxConfig(videoFxConfig, logger);
}

export const setReplacementDefaultColor = async (newHexColor: string, logger: Logger)
=> {
    videoFxConfig.backgroundReplacement.defaultColor = newHexColor;
    videoFxConfig.backgroundReplacement.backgroundImageUrl = undefined;
    await updateVideoFxConfig(videoFxConfig, logger);
}
```

```
export const setBlurStrength = async (newStrength: number, logger: Logger) => {
  videoFxConfig.backgroundBlur.strength =
  VideoFxTypeConversion.useBackgroundBlurStrengthType(newStrength);
  await updateVideoFxConfig(videoFxConfig, logger);
}

export const updateVideoFxConfig = async (config: VideoFxConfig, logger: Logger) => {
  try {
    await videoFxProcessor.setEffectConfig(videoFxConfig);
  } catch (error) {
    logger.error(error.toString())
  }
}

export const turnOffEffects = () => {
  const innerDevice = await videoTransformDevice?.intrinsicDevice();
  await videoTransformDevice?.stop();
  videoTransformDevice = undefined;
  videoFxProcessor = undefined;
  await meetingSession.audioVideo.startVideoInput(innerDevice);
}
```

Windows 用の Amazon Chime SDK クライアントライブラリの使用

現在、GitHub では、C++ で記述された Windows 用 Amazon Chime SDK クライアントライブラリが提供されています。<https://github.com/aws/amazon-chime-sdk-cpp> にアクセスしてください。

よくある質問

以下のセクションのトピックでは、Amazon Chime SDK に関するよくある質問への回答を提供します。詳細については、トピックを展開してください。

トピック

- [会議FAQs](#)
- [メディアパイプラインFAQs](#)
- [PSTN オーディオFAQs](#)

会議FAQs

トピック

- [参加者](#)
- [セキュリティと暗号化](#)
- [オーディオ/ビデオ](#)
- [ライブ文字起こし](#)
- [Service Quotas](#)
- [名前空間の移行](#)
- [モニタリング](#)
- [ログ記録](#)
- [エラーメッセージ](#)

参加者

Amazon Chime SDK 会議に参加できるユーザー

必要な参加トークンを持つ参加者のみ。[CreateAttendee](#)、[BatchCreateAttendee](#)、または [CreateMeetingWithAttendees](#) APIsを使用する場合、クライアントに渡す結合トークンを作成し、会議への参加を有効にします。これらの APIs によって生成されたトークンは サービスによって認証され、は会議に参加するアクセス許可を付与します。

Note

Amazon Chime SDK は、会議参加者の会議 IDs や参加 URLs を作成しません。

会議参加者のクォータは何ですか？

参加者のクォータは会議ごとです。Amazon Chime SDK は、標準セッションで 250 人の参加者、高解像度セッションで 100 人の参加者をサポートします。さらに参加者が必要な場合は、メディアレプリケーションの使用を検討してください。これにより、[AWS サポートセンターコンソールから制限の引き上げをリクエストした後、最大 10,000 人の参加者が許可されます](#)。メディアレプリケーションの詳細については、このガイドの前半の[メディアレプリケーションの使用](#)「」を参照してください。

会議に参加者がいない場合は課金されますか？

いいえ。Amazon Chime SDK は、参加者が会議に参加した場合にのみ課金します。また、会議は、最後のアクティブな参加者が会議を退席または退席してから 5 分後に自動的に終了します。

AttendeeDeleted、AttendeeLeftAttendeeDropped 会議イベントの違いは何ですか？

AttendeeLeft は、参加者が会議から退出することを決定したときにトリガーされます。AttendeeDropped は、通常、ネットワークの問題により参加者が会議から切断されたときにトリガーされます。AttendeeDeleted は [DeleteAttendee](#) API が呼び出されたときにトリガーされます。

AttendeeLeft は、次の場合にもトリガーされます。

- DeleteAttendee API がサーバー側の会議ハンドラーから とともに呼び出されると AttendeeDeleted、
- クライアントは、[のクライアントライブラリ](#) [JavaScript](#) meetingSession.audioVideo.Stop、[iOS](#) および [Android](#) SDK の対応する APIs、または会議終了時に API を呼び出します。SDKs

会議イベントの詳細については、このガイドの前半の[会議イベント](#)「」を参照してください。

参加者が参加しても接続が悪いために退席し、誰も会議を終了しない場合、会議はどのくらいの期間実行されますか？

会議は、次の場合に自動的に終了します。

- 会議時間が 24 時間を超えた場合。
- その会議がレプリカ会議であり、プライマリ会議が終了した場合。
- レプリカ以外の会議では、5 分間連続して接続する参加者はいません。

Amazon Chime SDK が参加者との再接続を試みる期間はどのくらいですか？

デフォルトでは、[の Amazon Chime SDK クライアントライブラリ](#)

[JavaScript](#)は、MeetingSessionConfiguration会議イベントで指定されているように、2 分間再接続を試みます。また、参加者が会議から削除され、セッションに再接続されない場合、Amazon Chime SDK はAttendeeDroppedイベントを送信します。

会議イベントの詳細については、このよくある質問の後半にある[モニタリング](#)「」を参照してください。

セキュリティと暗号化

Amazon Chime SDK は end-to-end AWS 256 ビット暗号化をサポートしていますか？

はい。すべてのメディアは転送中に暗号化され、サービスを通過します。メディアは、クライアントと会議をホストする特定のメディアインスタンスの間で暗号化されます。メディアインスタンスは、ミキシングのために音声を復号し、クライアントへの送信のために混合音声を暗号化します。メディアがメディアキャプチャを介して記録されている場合、メディアはメディアインスタンスとキャプチャインスタンスの間で暗号化されます。

オーディオ/ビデオ

トピック

- [全般](#)
- [コーデックとサイマルキャスト](#)
- [エコーリダクション](#)
- [ノイズ抑制](#)
- [背景ぼかし](#)
- [画面共有](#)

全般

Amazon Chime SDK は、バックグラウンドでビデオを一時停止しますか？

いいえ。ただし、帯域幅が制約されている場合、ビデオストリームを一時停止できます。

会議中にビデオストリームとタイトルを優先するにはどうすればよいですか？ 会議中に特定のビデオストリームを停止できますか？

各クライアントがサブスクライブするビデオストリームをプログラムで制御できます。これにより、「プレゼンターは常に表示される」や「会議ホストは常に表示される」などのロジックをページ分割されたディスプレイに実装できます。クライアントがリソースに制約がある場合は、優先順位が最も低いストリームをオフにできます。詳細については、「[での優先度ベースのダウンリンクポリシーのユーザーガイド](#)」を参照してください GitHub。

コーデックとサイマルキャスト

どのビデオコーデックとオーディオコーデックをサポートしていますか？

ビデオコーデック

H.264、VP8、VP9、および AV1。

オーディオコーデック

Opus、16 kHz/48 kHz、および 48 kHz ステレオ。

Amazon Chime SDK が複数の解像度をサポートする方法

VP8 と H.264 によるビデオサイマルキャストと、VP9 によるスケーラブルなビデオエンコーディングをサポートしています。[用の Amazon Chime SDK クライアントライブラリ JavaScript](#)を使用すると、ビデオを送信するためのコーデック設定を指定できます。iOS および Android クライアントライブラリは、デバイスの機能に基づいて自動的にコーデックを選択します。ビデオコーデックの詳細については、このガイドの前半の[動画コーデックの設定](#)「」を参照してください。アダプティブサイマルキャストとさまざまなポリシーの詳細については、「」の[「Video Simulcast](#)」を参照してください GitHub。

エコーリダクション

会議の特定の参加者にエコーリダクションを選択的に適用できますか？

いいえ。 [CreateMeeting](#) または [CreateMeetingWithAttendees](#) APIs が有効になります。エコーリダクションの使用の詳細については、の「[アプリケーションへのエコーリダクションの追加](#)」と「[エコーリダクションによる Voice Focus の有効化](#)」を参照してください GitHub。

ノイズ抑制

Amazon Chime SDK はどのノイズ抑制プロバイダーを使用していますか？

Amazon Voice Focus は、によって構築されたノイズ抑制テクノロジーです AWS。Amazon Voice Focus の詳細については、このガイドの前半の[Amazon Voice Focus の設定VoiceFocus](#) 「」および「」を参照してください。

ノイズ抑制のオンとオフを切り替えることができるのは誰ですか？

ソリューションのコーディング方法に応じて、会議の参加者は通常、ノイズ抑制をオンまたはオフにします。Amazon Chime SDK クライアントライブラリは、ノイズ抑制のプログラムによるコントロールを公開し、実装方法を選択します。例えば、トグルボタンや、ノイズ抑制を制御するためのアプリレベルの設定を指定できます。詳細については、「[Amazon Voice Focus on](#)」を参照してください GitHub。

背景ぼかし

背景ぼかしはどのくらいの CPU を使用しますか？

v1 と v2 のアルゴリズムがあります。v1 アルゴリズムには、CPU 使用率 (10% ~ 40%) に基づく 4 つのオプションがあります。v2 アルゴリズムは、高、中、低のぼかしを効果的に処理します。の[JavaScript ブラウザベースのデモ](#) GitHub は、実例を示しています。

画面共有

ウェブクライアントでの画面共有の解像度はどのくらいですか？

JavaScript クライアントの場合、ブラウザはクライアントライブラリの画面フレームを提供します。解像度は共有画面のネイティブ解像度で、会議でサポートされる最大解像度で上限が設定されています。フレームレートを設定できますが、フレームレートが高いほど CPU 負荷が増加することに注意してください。

共有用のコーデックを選択することもできます。標準定義会議では、解像度は 1080p で、1.5 Mbps でエンコードされます。高解像度会議では、解像度は 4K エンコードで 2.5 Mbps です。

モバイルデバイスブラウザで画面を共有できないのはなぜですか？

モバイルデバイスブラウザは、画面キャプチャや画面共有をサポートしていません。画面共有をサポートするアプリケーション GitHub を開発するには、で [iOS](#) または [Android](#) SDKs を使用する必要があります。詳細については、の以下のトピックを参照してください GitHub。

- [コンテンツ共有 \(JavaScript\)](#)。
- [コンテンツ共有 \(iOS\)](#)。
- [コンテンツ共有 \(Android\)](#)

ライブ文字起こし

文字起こしから PII を編集するにはどうすればよいですか？

Amazon Transcribe を使用して PII を編集します。[StartMeetingTranscription](#) API を使用して会議を文字起こしする場合、コンテンツ編集タイプと編集するさまざまな PII エンティティを指定できます。

Note

機械学習の予測性により、Amazon Transcribe は機密データのすべてのインスタンスを特定して削除することはできず、1996 年の米国医療保険の相互運用性と説明責任に関する法律 (HIPAA) などの医療プライバシー法に準拠していない可能性があります。詳細については、Amazon Transcribe デベロッパーガイドの [「個人を特定できる情報の編集または識別」](#) を参照してください。

会議中に文字起こしが開始または終了するタイミングを追跡できますか？

はい。をサブスクライブすると `transcribeEvent`、すべてのクライアントがそのイベントを受信し、エンドユーザーにクライアントに表示することができます。詳細については、このガイドの前半の [文字起こしイベント](#) 「」を参照してください。

次の例は、をサブスクライブする方法の 1 つを示しています `transcribeEvent`。

```
useEffect(() => {
```

```
        if (audioVideo) {  
  
            audioVideo.transcriptionController?.subscribeToTranscriptEvent((transcriptEvent) => {  
                setTranscripts(transcriptEvent);  
            });  
        }  
    }, [audioVideo]);
```

の使用の詳細についてはtranscribeEvent、このガイドの前半の[文字起こしイベント](#)「」を参照してください。

冒濫的な言葉を除外する方法を教えてください。

Amazon Transcribe を使用してカスタム語彙と語彙フィルターを作成し、[StartMeetingTranscription](#)API を呼び出すときに、不要な単語をマスクするための VocabularyFilterName と VocabularyFilterMethod の値を指定します。詳細については、「[Amazon Transcribe デベロッパーガイド](#)」の「[カスタム語彙](#)」および「[語彙フィルターの作成](#)」を参照してください。 Amazon Transcribe

ライブ文字起こしはどの言語で会議をサポートしていますか？

リアルタイムライブ文字起こしの場合、Amazon Transcribe は以下をサポートします。

- 簡体字中国語 (zh-CN)
- 英語 (オーストラリア語 (en-AU))
- 英国語 (en-GB)
- 米国 (en-US))
- フランス語 (フランス (fr-FR) およびカナダ (fr-CA))
- ドイツ語 (de-DE)
- ヒンディー語 (hi-IN)
- イタリア語 (it-IT)
- 日本語 (jp-JP)
- 韓国語 (ko-KR)
- ポルトガル語 (ブラジル (pt-BR))
- スペイン語 (米国 (es-US))
- タイ語 (th-TH)

リアルタイムまたはバッチで文字起こしに使用できる言語の詳細については、「Amazon Transcribe デベロッパーガイド」の「[サポートされている言語と言語固有の機能](#)」を参照してください。

Service Quotas

US-EAST-1 (バージニア北部) でクォータを更新しました。更新は米国東部のエンドポイントにのみ適用されますか？

はい。サービスクォータは API エンドポイントごとに適用されます。別の API エンドポイントに切り替えると、デフォルトの制限が適用されます。

名前空間の移行

チャイム名前空間からチャイムディスク名前空間への移行に関する情報はどこにありますか？

このガイドの以下のトピックを参照してください。

- [Amazon Chime 名前空間からの移行](#).
- [Amazon Chime SDK ミーティング名前空間への移行](#).
- [Amazon Chime SDK Identity 名前空間への移行](#).
- [Amazon Chime SDK Voice 名前空間への移行](#).

Amazon Chime SDK の CloudWatch イベントは、専用エンドポイントと名前空間でのみ使用できますか？

はい。イベントを使用するには、chime名前空間からchimesdk名前空間に移行する必要があります。詳細については、このガイドの以下のトピックを参照してください。

- [Amazon Chime 名前空間からの移行](#).
- [Amazon Chime SDK ミーティング名前空間への移行](#).
- [Amazon Chime SDK Identity 名前空間への移行](#).
- [Amazon Chime SDK Voice 名前空間への移行](#).

モニタリング

日付、時刻、通話時間、参加者などの会議データをどのように追跡しますか？

Amazon、Amazon SNS EventBridge、または Amazon SQS を介して会議イベントと参加者イベントを送信します。Amazon SNS イベントには、会議の開始時刻と停止時刻、参加者の参加、削除、退出アクションなどの情報が含まれます。会議イベントとその使用方法の詳細については、以下のトピックを参照してください。

- [Amazon Chime SDK イベント通知](#)、このガイドの前半。
- [Amazon Chime SDK イベントのサーバー側のログ記録とモニタリングに関する](#) ブログ記事。
- 「[Amazon Chime SDK ミーティングイベントのモニタリングとトラブルシューティング](#)」 ブログ記事。
- での[会議イベント](#) GitHub。

どの CloudWatch メトリクスを利用できますか？

メトリクスには、AttendeeAuthorizationSuccess、AttendeeAuthorizationErrors、および `AttendeeAudioDrops` が含まれます。メトリクスの詳細については、このガイドの前半の[Amazon CloudWatch メトリクス](#)「」を参照してください。

ログ記録とモニタリング用のダッシュボードを構築するにはどうすればよいですか？

Amazon Chime SDK は、オーディオ、ビデオ、画面共有、参加者のアクティビティなど、クライアントアプリケーションのコンポーネントのさまざまな状態に基づいて会議イベントを生成します。これらのイベントを CloudWatch ログに書き込み、それらのログにダッシュボードを構築できます。さまざまなイベント、エラーメッセージ、ステータスコードを含めることで、データからインサイトを得ることができます。

Amazon Chime SDK は Amazon EventBridge、Amazon SQS、Amazon SNS とも統合され、会議、参加者、メディアパイプラインの作成や削除のリクエストなどのサーバー側のイベントを追跡できます。関心のあるイベントをフィルタリングし、イベントを CloudWatch ログに書き込むようにルールを設定できます。

会議イベントとそのイベントを使用してダッシュボードを作成する方法の詳細については、以下を参照してください。

- での[会議イベント](#) GitHub。

- [会議イベント](#)、このガイドの前半。
- [Amazon CloudWatch メトリクス](#)、このガイドの前半。
- [Amazon Chime SDK イベント通知](#)、このガイドの前半。

これを試すには、以下のブログ記事の指示に従ってください。

- [Amazon Chime SDK イベントのサーバー側のログ記録とモニタリング](#)。
- [Amazon Chime SDK 会議イベントによるモニタリングとトラブルシューティング](#)

会議が自動的に終了するか、DeleteMeeting API が呼び出されたときに終了するかをモニタリングするにはどうすればよいですか？

どちらの方法でも、会議を終了するとMeetingFailedイベントがトリガーされます。[DeleteMeeting](#) API の Cloud Trail または EventBridge エントリがない場合は、会議が自動的に終了したと仮定できます。

ログ記録

Google Chrome で WebRTC デバッグログを有効にするにはどうすればよいですか？

次のコマンドとフラグを実行します: `chrome --enable-logging --vmodule=*/webrtc/*=1`。これにより、WebRTC の INFO との VERBOSE ログ記録が有効になります。結果のログの名前は Chrome ユーザーデータディレクトリに `chrome_debug.log` 保存されます。

macOS で Safari の WebRTC デバッグログ記録を有効にする方法

以下のステップに従います。

1. Safari で、**設定** を選択します。
2. **詳細オプション** を選択し、**ウェブデベロッパー向けの機能を表示** を選択します。

ブラウザに **開発** メニューが表示されます。

3. **開発** メニューで、**JavaScript コンソールを表示** を選択します。
4. JavaScript コンソールで、**設定** を選択し、**WebRTC ログ記録を有効にします**。必要に応じて、**基本的なログ記録** または **詳細なログ記録** を選択できます。

エラーメッセージ

「セッションが停止しました - 理由 - ICE GatheringTimeout Workground」エラーのトラブルシューティング方法を教えてください。

以下の操作を実行します。

- IP 範囲 99.77.128.0/18 の UDP ポート 3478 の出力が有効になっていることを確認します。詳細については、このガイドの前半の[ネットワーク構成](#)「」を参照してください。
- ウイルス対策ブラウザの拡張機能がリソースのロードを妨げていないことを確認します。UDP 3478 は TURN 用であり、ローカルコンピュータファイアウォールまたは企業ネットワークファイアウォールのいずれかでエンドユーザー側でブロックを解除する必要があります。
- 接続の再試行はポート 443 経由で TLS にフォールバックするため、ドメインまたはサブネットがブロックされていないことを確認してください。

「エラー: 無効なキャプチャパイプライン ARN」メッセージは何を意味しますか？

このエラーは通常、サービスがメディアパイプライン ARN を解決できない場合に発生します。ARN が会議ではなくメディアパイプラインに属していることを確認します。MediaPipelineArnは[CreateMediaCapturePipeline](#) API レスポンスの一部です。

AudioJoinedFromAnotherDevice 「」エラーの意味と回避方法

このエラーは、同じ参加者が 2 つのデバイスから参加した場合に発生します。エラーはイベントの meetingErrorMessage 属性で返されますmeetingFailed。これを回避するには、各参加者に一意の `ExternalUserId`、2 つ以上の会議で[CreateAttendee](#)、[BatchCreateAttendee](#)、または [CreateMeetingWithAttendees](#) APIs からの同じ参加者の応答を同時に使用しないようにします。

「禁止: アカウント ID 111122223333 で Chime SDK を呼び出す権限がない」と解決するにはどうすればよいですか？

非推奨の Amazon Chime API を呼び出しています。この問題を解決するには、Amazon Chime SDK 名前空間に移行します。詳細については、このガイドの前半にある以下のトピックを参照してください。

- [Amazon Chime 名前空間からの移行](#).
- [Amazon Chime SDK ミーティング名前空間への移行](#).

- [Amazon Chime SDK Identity 名前空間への移行.](#)
- [Amazon Chime SDK Voice 名前空間への移行.](#)

「禁止: アカウント ID 111122223333 は、チャイムエンドポイントで非推奨の Amazon Chime SDK API を呼び出す権限がありません」と解決するにはどうすればよいですか？

非推奨の Amazon Chime API を呼び出しています。この問題を解決するには、Amazon Chime SDK 名前空間に移行します。詳細については、このガイドの前半にある以下のトピックを参照してください。

- [Amazon Chime 名前空間からの移行.](#)
- [Amazon Chime SDK ミーティング名前空間への移行.](#)
- [Amazon Chime SDK Identity 名前空間への移行.](#)
- [Amazon Chime SDK Voice 名前空間への移行.](#)

メディアパイプラインの場合、「ランタイムエラー: Chime への問い合わせに関する問題: アクティブなリソースがないクライアントリクエストトークンが存在するため、クライアントリクエストトークンを再生成してください」のトラブルシューティング方法を教えてください。

クライアントリクエストトークンは、API リクエストをべき等にする一意の識別子です。このエラーは、トークンが非アクティブなメディアパイプラインに関連付けられている場合に発生します。この問題を解決するには、新しい一意のトークンを生成し、API リクエストとともに送信します。

メディアパイプラインFAQs

メディアキャプチャは 5 秒のセグメントにどの形式を使用しますか？

メディアキャプチャは MP4 形式を使用します。これには、5 秒のセグメントと、録画ファイルと複合ファイルの組み合わせが含まれます。

メディアキャプチャパイプラインによって作成された参加者を削除するにはどうすればよいですか？

メディアキャプチャ参加者を削除するには、パイプラインを終了するか、[DeleteMediaCapturePipeline](#) API を呼び出します。

録画はクラウドで行われますか、それともローカルで行われますか？

メディアキャプチャパイプラインは Amazon S3 バケットに直接記録します。メディアキャプチャでは、クライアントに帯域幅や接続要件が課されることはありません。

会議に関連するメディアキャプチャはどこで作成できますか？

メディアキャプチャを作成するリージョンを選択するには、まず、利用可能な会議コントロールプレーンリージョンから API エンドポイントを選択します。次に、そのリージョンに会議とメディアキャプチャパイプラインを作成します。メディアキャプチャは、任意の Amazon Chime SDK メディアリージョンのアカウントの Amazon S3 バケットに書き込むことができます。使用可能なリージョンとエンドポイント、メディアパイプラインコントロールプレーン、およびメディアリージョンの詳細については、このガイドの前[利用できるリージョン](#)半の「」、およびAWS リファレンスガイドの「[Amazon Chime SDK エンドポイントとクォータ](#)」を参照してください。

メディアキャプチャでは、会議内の参加者 250 件のビデオがすべて記録されますか？

いいえ。パイプラインは最初の 25 個のビデオストリームのみをキャプチャします。

会議の進行中に録音を停止できますか？

指定された分数後に [DeleteMediaCapturePipeline](#) API を呼び出すメカニズムを作成できます。例えば、メディアキャプチャが開始され、事前に定義された待機時間を持つときに開始するステップ関数を作成できます。

録音がオンになっている間に会議を停止できますか？

[DeleteMediaCapturePipeline](#) API を呼び出して録音を終了するか、会議の終了が予定されているときに [DeleteMeeting](#) API を呼び出すことができます。会議中、メディアキャプチャ参加者が唯一の残った参加者である場合、会議は 5 分後に自動的に終了します。

PSTN オーディオFAQs

PSTN オーディオを使用して、米国以外の番号からのインバウンドコールを Voice Connector にルーティングできますか？

いいえ。PSTN Audio を使用して、米国以外の番号を Voice Connector にルーティングすることはできません。

参加者が PSTN オーディオ経由で接続する場合、現在の会議から新しい会議に移動できますか？

はい。まず、会議に接続されたレグの [Hangup](#) アクションを呼び出します。これにより、着信通話を終了せずに参加者を会議から切断します。次に、[JoinChimeMeeting](#) アクションを呼び出して、参加者を新しい会議に参加させます。

ドキュメント履歴

次の表は、2019年9月以降の「Amazon Chime Developer Guide」の重要な変更点を示しています。このドキュメントの更新に関する通知をするために、RSS フィードをサブスクライブすることができます。

変更	説明	日付
Alexa のスキル内通話が削除されました	Amazon Alexa チームによる変更により、SIP メディアアプリケーションに Alexa コールを追加できなくなります。詳細については、 「Alexa スマートプロパティ」 ページを参照してください。	2024 年 4 月 1 日
新しい会議リージョン	複数の新しい会議リージョンを使用できるようになりました。詳細については、本ガイドの「 Available Regions 」と、「AWS 全般リファレンス」の「 Amazon Chime SDK エンドポイントとクォータ 」を参照してください。	2023 年 9 月 25 日
音声エンハンスメント	通話録音を有効にし、録音した通話を Amazon S3 バケットに保存できるようになりました。詳細については、本ガイドの「 Understanding voice enhancement 」を参照してください。	2023 年 8 月 31 日
リージョンの更新	Amazon Chime SDK で、これまでよりも多くのリージョンを使用できるようになりました。詳細については、	2023 年 8 月 29 日

「[Available regions](#)」を参照してください。

[通話分析と音声分析](#)

ローコード分析と文字起こしの機能をソリューションに追加できるようになりました。詳細については、本ガイドの「[Using Amazon Chime SDK call analytics](#)」を参照してください。

2023 年 3 月 27 日

[Windows 用クライアントライブラリ](#)

C++ で記述された Windows 用の Amazon Chime SDK クライアントライブラリを使用できるようになりました。詳細については、本ガイドの [Amazon Chime SDK client library for Windows](#) を参照してください。

2023 年 2 月 2 日

[リージョンの更新](#)

Amazon Chime SDK で、これまでよりも多くのリージョンを使用できるようになりました。詳細については、「[Available regions](#)」を参照してください。

2022 年 11 月 18 日

[の C++ クライアントライブラリ GitHub](#)

Amazon Chime SDK ミーティングを使用するデベロッパーは、で C++ シグナリングクライアントライブラリと統合できるようになりました GitHub。詳細については、「[Integrating with a client library](#)」を参照してください。

2022 年 8 月 19 日

メディアパイプライン

Amazon Chime SDK ミーティングで、メディアパイプラインを作成できるようになりました。メディアパイプラインは、メディアキャプチャパイプライン、メディア連結パイプライン、ライブコネクタパイプラインで構成されます。詳細については、「[Creating Amazon Chime SDK media pipelines](#)」を参照してください。

2022 年 8 月 18 日

Elastic チャネル

Amazon Chime SDK メッセージングでは、Elastic チャネルをチャットソリューションで使用できるようになりました。Elastic チャネルにより、最大 100 万人のユーザーをホストできます。詳細については、「[Using elastic channels to host live events](#)」を参照してください。

2022 年 8 月 12 日

911 緊急時の住所検証

Amazon Chime SDK ミーティングでは、緊急通報の発信元住所をプログラムで検証できます。詳細については、「Amazon Chime API リファレンス [ValidateE911Address](#)」の「」および「[Amazon Chime SDK 管理ガイド](#)」の「[緊急通報のアドレスの検証](#)」を参照してください。

2022 年 8 月 11 日

[会議タグの更新](#)

Chime と ChimeSDK Meetings の名前空間で、会議タグを使用できるようになりました。詳細については、「[Amazon Chime SDK ミーティング名前空間への移行](#)」を参照してください。

2022 年 8 月 4 日

[CallAndBridge Voice Connector および Voice Connector グループのアクション](#)

Amazon Chime SDK Audio サービスでは、CallAndBridge アクションを使用して、Voice Connector または Voice Connector グループとして設定した SIP トランクに発信できません。詳細については、「[CallAndBridge](#)」を参照してください。

2022 年 7 月 14 日

[AppKeys および TenantIDs](#)

Amazon Chime SDK ミーティングでは、AppKeys と TenantIDs を使用して、顧客ネットワークから WebRTC メディアセッションへのアクセスを制御できるようになりました。詳細については、「[AppKeys とテナント IDs](#)」を参照してください。

2022 年 7 月 7 日

[接続 API](#)

Amazon Chime SDK Messaging を使用するデベロッパーは、WebSockets を使用してバックエンドサーバーに接続し、 のメッセージを受信できるようになりましたAppInstanceUser 。詳細については、[「Connect API の使用」](#)および [「WebSockets を使用してメッセージを受信する」](#)を参照してください。

2022 年 6 月 6 日

[参加者の機能](#)

Amazon Chime SDK ミーティングで、音声、動画、コンテンツへの参加者アクセスを制御する機能を使用できるようになりました。詳細については、「Amazon Chime SDK API リファレンス[AttendeeCapabilities](#)」の「」を参照してください。

2022 年 6 月 2 日

[Amazon CloudWatch メトリクス](#)

デベロッパーは、Amazon Chime SDK が に発行するサービスと使用状況のメトリクスを利用できるようになりました CloudWatch。メトリクスを使用すると、CloudWatch グラフとダッシュボードを使用して、Amazon Chime SDK サービスの使用状況をモニタリングできます。詳細については、[「Amazon CloudWatch メトリクス」](#)を参照してください。

2022 年 6 月 1 日

エコーリダクション

エコーリダクションを実装できるようになりました。この機能があれば、エコー (ユーザーのスピーカーが発する音をマイクが拾っている状態) が会議の音声に混じり、ディスカッションが中断されるのを防ぐことができます。詳細については、「[Using echo reduction](#)」を参照してください。

2021 年 11 月 23 日

通話録音

Amazon Chime SDK SIP メディアアプリケーション呼び出しの 1 つ以上のレッグに音声録音を実装できるようになりました。詳細については、「[Using call recording](#)」を参照してください。さらに、RecordAudio アクションに、SilenceDurationInSeconds や RecordingTerminatorUsed などの新しいパラメータが含まれるようになりました。詳細については、「[RecordAudio](#)」を参照してください。

2021 年 10 月 28 日

背景ぼかし

Amazon Chime SDK アプリケーションで、背景のぼかしを行えるようになりました。詳細については、「[Using background blur](#)」を参照してください。

2021 年 10 月 21 日

IAM ポリシーの更新	更新された IAM ポリシーを利用できるようになりました。これにより、Amazon Chime SDK ミーティングのライブ文字起こしが可能になります。詳細については、「 Creating IAM users or roles with the Chime SDK policy 」を参照してください。	2021 年 9 月 22 日
SIP ヘッダー	開発者は、AWS Lambda 関数で User-to-User ヘッダー、Diversion ヘッダー、カスタム SIP ヘッダーを送受信できるようになりました。詳細については、「 Using SIP headers 」を参照してください。	2021 年 9 月 13 日
Amazon Chime SDK ミーティング	Amazon Chime SDK のライブ文字起こしを使用できるようになりました。詳細については、「 Using Amazon Chime SDK live transcription 」を参照してください。	2021 年 8 月 11 日
Amazon Chime SDK ミーティング	メディアパイプラインを作成できるようになりました。詳細については、「 Creating Amazon Chime SDK media capture pipelines 」を参照してください。	2021 年 7 月 7 日

[AWS Lambda 関数での SIP メディアアプリケーションの使用](#)

タイトルが「Lambda 関数 SDK」から変更され、すべてのトピックの内容が正確性のために改訂されました。CallAndBridge セクションを追加しました。詳細については、[「PSTN オーディオサービスの使用」](#)および「」を参照してください。[CallAndBridge](#)。

2021 年 6 月 17 日

[Lambda 関数 SDK](#)

Amazon Chime SDK 管理者が作成した Amazon Chime SDK SIP メディアアプリケーションで使用するカスタム Lambda 関数を構築できます。詳細については、「Amazon Chime [デベロッパーガイド](#)」の「[PSTN オーディオサービスの使用](#)」を参照してください。

2020 年 11 月 17 日

[JavaScript SDK](#)

デベロッパーは JavaScript を使用して Amazon Chime SDK アプリケーションを構築できます。詳細については、「[Amazon Chime デベロッパーガイド](#)」の「[用の Amazon Chime SDK JavaScript の使用](#)」を参照してください。

2020 年 11 月 17 日

[Android および iOS クライアントライブラリ](#)

Android、iOS、Windows 用のクライアントライブラリを簡単な操作ですぐに見つけられます。詳細については、「Amazon Chime Developer Guide」の「[Using the Amazon Chime SDK client library for Android](#)」と「[Using the Amazon Chime SDK client library for iOS](#)」を参照してください。

2020 年 11 月 17 日

[プロキシ電話セッション](#)

Amazon Chime SDK Voice Connector で使用するプロキシ電話セッションを作成できます。詳細については、「[Amazon Chime デベロッパーガイド](#)」の「[用の Amazon Chime SDK JavaScript の使用](#)」を参照してください。

2020 年 4 月 7 日

[Amazon Chime SDK コンテンツ共有](#)

Amazon Chime SDK は、コンテンツ共有に対応しています。詳細については、「Amazon Chime Developer Guide」の「[Amazon Chime SDK architecture](#)」を参照してください。

2020 年 3 月 31 日

[Android および iOS 用 Amazon Chime SDK](#)

Android および iOS 用 Amazon Chime SDK がリリースされました。詳細については、「Amazon Chime Developer Guide」の「[Integrating with a client library](#)」を参照してください。

2020 年 3 月 24 日

[Amazon Chime SDK](#)

Amazon Chime SDK がリリースされました。詳細については、「Amazon Chime Developer Guide」の「[Using the Amazon Chime SDK](#)」を参照してください。

2019 年 11 月 20 日

[Amazon Chime Developer Guide](#)

「Amazon Chime Developer Guide」がリリースされました。

2019 年 9 月 11 日

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。