



开发人员指南

Amazon Chime SDK



Amazon Chime SDK: 开发人员指南

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

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 Amazon Chime SDK ?	1
定价	1
资源	1
使用 Amazon Chime SDK	2
Amazon Chime SDK 先决条件	2
Amazon Chime SDK 概念	3
Amazon Chime SDK 架构	3
Amazon Chime SDK 服务限额	4
Amazon Chime SDK 系统要求	5
可用区	7
控制台区域	8
调用分析区域	8
会议区域	9
媒体管道区域	11
消息收发区域	13
PSTN 区域	13
与客户端库集成	14
SIP 集成	15
Amazon Chime SDK 事件通知	17
向 EventBridge 发送通知	17
向 Amazon SQS 和 Amazon SNS 发送通知	17
授予 Amazon Chime SDK 访问 Amazon SQS 和 Amazon SNS 的权限	18
从 Amazon Chime 命名空间迁移	21
终端节点、命名空间和 CLI 命令	22
每项服务的迁移帮助	22
API 映射	22
使用 Amazon Chime SDK 会议	32
迁移到 Amazon Chime SDK 会议命名空间	32
迁移原因	33
迁移之前	33
命名空间之间的差异	34
使用会议区域	37
选择控制区域	37
选择媒体区域	38

查找最近的媒体区域	39
查找最近的 AWS GovCloud (US) 媒体区域	39
JavaScript 示例	39
检查区域状态	40
创建会议	42
选择会议功能	43
使用音频。EchoReduction	44
使用视频。MaxResolution	44
使用内容。MaxResolution	44
使用与会者。MaxCount	45
在客户端应用程序中使用会议功能	46
WebRTC 媒体	47
音频	47
视频	47
内容共享	48
数据消息	49
配置视频编解码器	49
设置视频编解码器首选项	49
网络配置	50
配置媒体和信令	51
配置 Amazon Voice Focus	52
配置回声抑制	52
配置背景替换和背景模糊	52
配置浏览器内容安全策略	53
使用 AppKeys 和租户 ID	53
会议事件	57
Amazon CloudWatch 指标	70
服务指标	70
API 使用情况指标	71
创建 Amazon Chime SDK 媒体管道	72
迁移到 ChimeSdkMediaPipelines 命名空间	75
管道创建概述	78
创建媒体捕获管道	79
创建媒体串联管道	85
创建媒体实时连接器管道	91
将音频和视频合成单一视图	92

创建媒体流管道	105
为媒体管道创建服务相关角色	122
使用媒体管道事件	125
解析转录	131
停止管道的最佳实践	131
使用 Amazon Chime SDK 实时转录	131
系统架构	132
账单和使用情况	133
配置您的账户	133
选择转录选项	133
开始和停止转录	137
转录参数	141
转录事件	141
转录消息	145
传输示例	149
使用媒体复制	152
互动参与者	153
全球参与者	154
会话生命周期	155
故障排除和调试 Amazon Chime SDK 会议	156
了解系统要求	156
设置日志记录和监控	157
自我故障排除	158
常见问题	160
使用 Amazon Chime SDK 消息传递	163
迁移到 Amazon Chime SDK 身份命名空间	163
迁移原因	164
迁移之前	164
命名空间之间的差异	165
迁移到 Amazon Chime SDK 消息传递命名空间	166
迁移原因	33
迁移之前	33
命名空间之间的差异	34
消息传递的先决条件	169
消息传递概念	170
消息传递架构	171

消息类型	171
开始使用	172
创建一个 ApplInstance	172
从后端服务发出 SDK 调用	173
对最终用户客户端应用程序进行身份验证	176
创建频道	179
发送消息	180
使用 ExpirationSettings	180
WebSockets 用于接收消息	183
配置附件	193
了解系统消息	193
IAM 角色示例	193
了解按角色划分的授权	197
ApplInstanceAdmin	198
ChannelModerator	200
成员	203
非成员	206
流式传递消息数据	209
使用弹性频道举办实时事件	213
先决条件	214
弹性频道概念	214
其他支持的功能	215
创建弹性频道	215
管理弹性频道成员	216
发送弹性频道消息	217
了解弹性频道中的 WebSocket 系统消息	217
使用 Kinesis 流接收系统消息	218
在我们的演示应用程序中测试弹性频道	218
使用移动推送通知接收消息	218
创建 Amazon Pinpoint 应用程序	219
创建服务角色	220
将移动设备终端节点注册为应用程序实例用户	221
在启用通知的情况下发送频道消息	222
接收推送通知	222
调试推送通知失败	223
使用筛选规则来筛选消息	224

使用服务相关角色	229
为数据流使用服务相关角色	229
使用频道流处理消息	232
设置频道处理器	233
创建频道流	236
关联和取消关联频道流	236
发送消息	237
使用 EventBridge 自动执行创建故障警报	238
使用机器人作为频道座席	239
创建 Amazon Lex V2 机器人	240
设置 ApplInstance 机器人	243
ApplInstanceBot 的频道成员资格	243
向 ApplInstanceBot 发送消息	244
处理来自 Amazon Lex 的消息	244
处理来自 ApplInstanceBot 的响应	245
使用规则向 Amazon EventBridge 发送事件	247
排除 ApplInstanceBots 故障	248
管理消息保留日期	249
CLI 保留命令示例	249
启用消息保留日期	250
恢复和删除消息	250
用于消息传递的 UI 组件	250
与客户端库集成	250
将 Amazon Chime 软件开发工具包消息传递与 JavaScript	251
使用 Amazon Chime SDK PSTN 音频服务	252
迁移到 Amazon Chime SDK 语音命名空间	253
迁移原因	253
迁移之前	253
命名空间之间的差异	254
了解电话号码、SIP 规则、SIP 媒体应用程序和 AWS Lambda 功能	256
PSTN 音频服务编程模型	257
将呼叫和事件路由到 AWS Lambda 函数	258
关于使用 PSTN 音频服务呼叫分支	262
调用流程示例	265
构建 PSTN 音频服务的 AWS Lambda 函数	267
了解电话事件	268

了解操作	272
调用 AWS Lambda 函数的电话事件	272
使用操作列表响应调用	297
PSTN 音频服务支持的操作	298
使用 SIP 标头	394
使用呼叫详细信息记录	398
超时和重试	399
调试和故障排除	400
VoiceFocus	408
PSTN 音频服务词汇表	413
使用呼叫分析	419
什么是 Amazon Chime SDK 呼叫分析	420
呼叫分析术语	421
创建呼叫分析配置	423
先决条件	424
使用控制台创建配置	424
使用 API 创建呼叫分析配置。	431
将配置与 Voice Connector 关联	431
使用呼叫分析配置	432
通话录音工作流程	432
基于机器学习的分析工作流程	438
管理呼叫分析管道	445
暂停和恢复呼叫分析管道	445
使用通话分析资源访问角色	446
了解通话分析状态	454
使用 Amazon 监控呼叫分析渠道 CloudWatch	456
先决条件	456
通话分析指标	456
CloudWatch 管道指标的维度	457
呼叫分析处理器和输出目的地	457
将转录与录音接收器相结合	478
使用亚马逊 EventBridge 通知	480
创建 Amazon Chime SDK 数据湖	498
配置亚马逊 QuickSight 控制面板	506
呼叫分析数据模型	511
Glue 数据目录表结构	511

Glue 数据目录表	513
示例查询	537
使用 Amazon Chime SDK 语音分析	542
语音分析架构	543
发言人搜索工作流程示例	545
语音音调分析工作流程示例	547
轮询任务结果	549
了解通知	550
数据存储、选择退出和数据留存策略	560
使用语音 API 来运行语音分析	562
呼叫分析服务限额	567
使用安卓版 Amazon Chime SDK 客户端库	569
使用 iOS 版 Amazon Chime SDK 客户端库	570
使用 Amazon Chime 软件开发工具包客户端库用于 JavaScript	571
Amazon Chime SDK 应用程序的组件	571
重要概念	572
服务架构	573
Web 应用程序架构	573
服务器应用程序架构	574
Amazon Chime SDK 媒体控制面板	574
Amazon Chime SDK 媒体数据面板	574
Web 应用程序组件架构	574
构建服务器应用程序	576
创建 IAM 用户或角色	576
配置 AWS 软件开发工具包以调用 API	577
创建会议	577
创建与会者	578
向客户端发送响应	578
构建客户端应用程序	578
将背景筛选器集成至客户端应用程序	579
关于使用背景筛选器	579
使用 content-security 策略	581
将背景筛选器添加到您的应用程序	584
背景筛选器示例	590
使用适用于 Windows 的 Amazon Chime SDK 客户端库	593
文档历史记录	594

..... **dcii**

什么是 Amazon Chime SDK ?

Amazon Chime SDK 是一组实时通信组件，可用于将消息收发、音频、视频和屏幕共享功能快速添加到他们的 Web 或移动应用程序中。

您可以使用 Amazon Chime SDK 构建用于发送和接收音频和视频并允许内容共享的实时媒体应用程序。有关 Amazon Chime SDK API 操作的详细信息，请参阅 [Amazon Chime SDK API 参考](#)。

定价

Amazon Chime SDK 提供按需付费的定价，不收取任何预付费用。您可以选择以单一速率实现部分或全部可用媒体模式（音频、视频和屏幕共享）。消息收发、媒体管道、语音增强和 PSTN 音频功能也以按需付费的定价提供。有关更多信息，请参阅 [Amazon Chime SDK 定价](#)。

资源

下列相关资源在您使用此服务的过程中会有所帮助。

- [课程和研讨会](#) – 指向基于角色的专业课程和自主进度动手实验室的链接，这些课程和实验室旨在帮助您增强 AWS 技能并获得实践经验。
- [AWS 开发人员中心](#) – 浏览教程、下载工具并了解 AWS 开发人员活动。
- [AWS 开发人员工具](#) – 指向开发人员工具、开发工具包、IDE 工具包和命令行工具的链接，这些资源用于开发和管理 AWS 应用程序。
- [入门资源中心](#) – 了解如何设置您的 AWS 账户、加入 AWS 社区和启动您的第一个应用程序。
- [动手实践教程](#) – 按照分步教程在 AWS 上启动您的第一个应用程序。
- [AWS 白皮书](#) – 指向 AWS 技术白皮书的完整列表的链接，这些资料涵盖了架构、安全性、经济性等主题，由 AWS 解决方案架构师或其他技术专家编写。
- [AWS Support 中心](#) – 用于创建和管理 AWS Support 案例的中心。还提供指向其他有用资源的链接，如论坛、技术常见问题、服务运行状况以及 AWS Trusted Advisor。
- [AWS Support](#) – 提供有关 AWS Support 的信息的主要网页，这是一个一对一的快速响应支持渠道，可以帮助您在云中构建和运行应用程序。
- [联系我们](#) – 用于查询有关 AWS 账单、账户、事件、滥用和其他问题的中央联系点。
- [AWS 网站条款](#) – 有关我们的版权和商标、您的账户、许可、网站访问和其他主题的详细信息。

使用 Amazon Chime SDK

您可以使用 Amazon Chime SDK 构建用于发送和接收音频和视频并允许内容共享的实时媒体应用程序。Amazon Chime SDK 独立于任何 Amazon Chime 管理员账户运行，并且不会影响在 Amazon Chime 上托管的会议。相反，Amazon Chime SDK 可提供生成器工具，用于构建自己的会议应用程序。

主题

- [Amazon Chime SDK 先决条件](#)
- [Amazon Chime SDK 概念](#)
- [Amazon Chime SDK 架构](#)
- [Amazon Chime SDK 服务限额](#)
- [Amazon Chime SDK 系统要求](#)
- [可用区](#)
- [与客户端库集成](#)
- [使用 Amazon Chime SDK 语音连接器进行 SIP 集成](#)
- [Amazon Chime SDK 事件通知](#)
- [从 Amazon Chime 命名空间迁移](#)

Amazon Chime SDK 先决条件

使用 Amazon Chime SDK 需要具备以下条件：

- 编程能力。
- 一个 AWS 账户。
- IAM 角色，通过其策略允许访问 Amazon Chime SDK 使用的 Amazon Chime API 操作，例如 AWS 托管的 AmazonChimeSDK 策略。有关更多信息，请参阅 Amazon Chime SDK 管理员指南中的 [Amazon Chime 如何与 IAM 配合使用](#) 和 [允许用户访问 Amazon Chime SDK 操作](#)。
- 对于大多数使用案例，您还需满足以下条件：
 - 服务器应用程序 — 管理会议和与会者资源，并将这些资源提供给客户端应用程序。在 AWS 账户中创建服务器应用程序，且必须有权访问前面提到的 IAM 角色。
 - 客户端应用程序 — 从服务器应用程序接收会议和与会者信息，并使用该信息建立媒体连接。

Amazon Chime SDK 概念

以下术语和概念可帮助您了解如何使用 Amazon Chime SDK。

会议

由唯一 MeetingId 标识的短暂资源。将 MeetingId 放置在主持活动会议的一组媒体服务上。

媒体服务组

主持活动会议的媒体服务组。

媒体置放

一组代表媒体服务组的区域化 URL。与会者通过其客户端与媒体服务组连接，发送和接收实时音频和视频，并共享屏幕。

与会者

由唯一 AttendeeId 标识的会议参与者。与会者可以使用 Amazon Chime SDK 客户端库构建的客户端应用程序自由加入和离开会议。

加入令牌

分配给每位与会者的唯一令牌。与会者使用加入令牌与媒体服务组进行身份验证。

Amazon Chime SDK 架构

下表描述了 Amazon Chime SDK 架构的不同组件如何协同工作，从而为会议和与会者、音频、视频和内容共享提供支持。

会议和与会者

当服务器应用程序创建 Amazon Chime SDK 会议时，该会议将分配给特定地区的媒体服务。服务中的主机负责在与会者客户端之间安全传输实时媒体。为每位创建的与会者分配唯一加入令牌，这是一种不透明密钥，您的服务器应用程序必须将其安全传输给有权代表与会者加入会议的客户端。每个客户端都使用加入令牌与媒体服务组进行身份验证。客户端使用安全 WebSocket 和数据报传输层安全 (DTLS) 的组合来安全地向媒体服务组发送信号，并通过媒体服务组向其他与会者发送和接收媒体。

音频

媒体服务混合每位与会者的音频，然后在从混音中减去自己的音频后，发送给每位接收者。Amazon Chime SDK 以设备和浏览器支持的最高速率采样音频，最高可达 48kHz。我们使用

Opus 编解码器对音频进行编码，默认比特率为 32kbps，可以将其增加至最高 128kbps 的立体声和 64kbps 的单声道。

视频

媒体服务使用发布和订阅模式用作选择性转发单元 (SFU)。每位与会者可以发布一个视频源，每次会议最多可同时发布 25 个视频。适用于 JavaScript 的 Amazon Chime SDK 客户端库支持最高 1280 x 720 的视频分辨率，不使用联播时，每秒 30 帧；使用联播时，每秒 15 帧。适用于 [iOS](#)、[Android](#) 和 [Windows](#) 的 Amazon Chime SDK 客户端库支持最高 1280 x 720 的视频分辨率和每秒 30 帧的帧率，但是实际的帧率和分辨率由 Amazon Chime SDK 自动管理。

激活后，视频联播会以两种不同的分辨率和比特率发送每个视频流。带宽受限的客户端会自动订阅较低比特率的视频流。视频编码和解码在可用情况下使用硬件加速来提高性能。

数据消息

除了音频和视频内容外，与会者还可以互相发送每条最大 2KB 的实时数据消息。您可以使用消息来实现自定义会议功能，例如白板、聊天、实时表情符号反应以及特定于应用程序的楼层控制信号。

内容共享

客户端应用程序可以共享音频和视频内容，例如屏幕截图或媒体文件。内容共享支持预先录制视频/音频内容。视频：最高 1280 x 720 分辨率、每秒 15 帧；音频：最高 48kHz、比特率为 64kbps。最多支持每秒 15 帧的内容共享屏幕截图，但可能会受到设备和浏览器功能的限制。

Amazon Chime SDK 服务限额

Note

服务限额视每个 API 终端节点而定。请求增加服务限额时，务必在应用程序使用的所有 API 终端节点上申请增加限额。

该表列出了适用于 Amazon Chime SDK 会议的资源 and 限额。

资源	限额	可调整
活动会议数	250	是
每次会议与会者人数	250	否

资源	限额	可调整
每次会议的音频流数	250	否
每次会议发布的视频流数	25	是，最多 250
每位与会者订阅的视频流数	25	否
每次会议的内容共享数	2	否
每次主会议的副本会议数	4	是，最多 40
每次会议的活动媒体捕获管道	1	否
每次账户的活动媒体捕获管道	us-east-1 终端节点为 100，其他终端节点为 10	是
API 速率	每秒 10 个请求 (RPS)，突增限制为 20 RPS。	是，但采用间接方式

 **Note**

当您增加活动会议限额时，API 速率限制也会增加。

Amazon Chime SDK 系统要求

以下系统要求适用于使用 Amazon Chime SDK 创建的应用程序。

支持的浏览器，适用于 JavaScript 的 Amazon Chime SDK 客户端库

操作系统	浏览器	支持的版本	注意
Windows	Mozilla Firefox	75 及更高版本	
	Google Chrome	78 及更高版本	

操作系统	浏览器	支持的版本	注意
	基于 Chromium 的 Edge	79 及更高版本	
	基于 Chromium 的 Electron	7 及更高版本	Chrome 版本 78 及更高版本。
	Opera	66 及更高版本	
macOS	Mozilla Firefox	75 及更高版本	
	Google Chrome	78 及更高版本	
	基于 Chromium 的 Edge	79 及更高版本	
	基于 Chromium 的 Electron		
	Safari	13 及更高版本	
	Opera	66 及更高版本	
iOS	Mozilla Firefox	10 及更高版本	仅限音频和视频，无共享内容。
	Google Chrome	78 及更高版本	仅限音频和视频，无共享内容。
	Safari	13 及更高版本	仅限音频和视频，无共享内容。
	WKWebView	14.3 及更高版本	仅限音频和视频，无共享内容。
Android	Google Chrome	10 及更高版本	仅限音频和视频，无共享内容。

操作系统	浏览器	支持的版本	注意
	三星	12 及更高版本	仅限音频和视频，无共享内容。
	Chromium WebView	5 及更高版本	仅限音频和视频，无共享内容。
Ubuntu LTS 16.04 及更高版本	Google Chrome	78 及更高版本	

适用于 iOS 的 Amazon Chime SDK 客户端库

- iOS 版本 13 及更高版本

适用于 Android 的 Amazon Chime SDK 客户端库

- Android OS 版本 5 及更高版本、ARM 和 ARM64 架构

可用区

下表列出了 Amazon Chime 软件开发工具包服务的功能以及提供每项服务的 AWS 区域。

Note

标有星号 (*) 的区域必须在您的 AWS 账户中启用。AWS 默认情况下会屏蔽这些区域。有关启用区域的更多信息，请参阅 AWS 一般参考中的[启用区域](#)。

主题

- [控制台区域](#)
- [调用分析区域](#)
- [会议区域](#)
- [媒体管道区域](#)
- [消息收发区域](#)

- [PSTN 区域](#)

控制台区域

您可以使用 Amazon Chime SDK 控制台配置资源并了解有关 Amazon Chime SDK 服务的更多信息。

AWS 区域	控制台
亚太地区 (首尔)	是
亚太地区 (新加坡)	
亚太地区 (悉尼)	是
亚太地区 (东京) (ap-northeast-1)	是
加拿大 (中部) (ca-central-1)	是
欧洲地区 (法兰克福) (eu-central-1)	是
欧洲地区 (爱尔兰) (eu-west-1)	是
欧洲 (伦敦) (eu-west-2)	是
美国东部 (弗吉尼亚州北部) (us-east-1)	是
美国西部 (俄勒冈州) (us-west-2)	是

调用分析区域

下表列出了可用于分析、转录和通话录音的 AWS 区域。

AWS 区域	语音分析	转录	通话录音
美国东部 (弗吉尼亚州北部) (us-east-1)	支持	是	是
美国西部 (俄勒冈州) (us-west-2)	支持	是	是

AWS 区域	语音分析	转录	通话录音
欧洲地区 (法兰克福) (eu-central-1)	否	是	是

会议区域

Amazon Chime SDK 会议具有控制区域和媒体区域。控制区域提供用于创建、更新和删除会议的 API 终端节点。控制区域还可用于接收和处理 [会议事件](#)。

媒体区域主持实际会议，客户可连接到您的媒体区域。您可以在调用 [CreateMeeting](#) API 时指定媒体区域。

控制区域可以在同一 AWS 分区中的任何媒体区域中创建会议。但是，您只能在用于创建会议的控制区域中更新会议。

有关选择控件和媒体区域的更多信息，请参阅 [使用会议区域](#)。

下表列出了提供控件、媒体或控件及媒体的区域。

AWS 区域	会议控制	会议媒体
非洲 (开普敦) (af-south-1)*		是
亚太地区 (孟买) (ap-south-1)	支持	是
亚太地区 (首尔) (ap-northeast-2)	支持	是
亚太地区 (新加坡) (ap-southeast-1)	支持	是
亚太地区 (悉尼) (ap-southeast-2)	支持	是
亚太地区 (东京) (ap-northeast-1)	支持	是
加拿大 (中部) (ca-central-1)	支持	是

AWS 区域	会议控制	会议媒体
欧洲地区 (法兰克福) (eu-central-1)	支持	是
欧洲地区 (爱尔兰) (eu-west-1)		是
欧洲 (伦敦) (eu-west-2)	支持	是
欧洲地区 (米兰) (eu-south-1)*		是
欧洲地区 (巴黎) (eu-west-3)		是
欧洲地区 (斯德哥尔摩) (eu-north-1)		是
以色列 (特拉维夫) (il-central-1)*	是*	是
南美洲 (圣保罗) (sa-east-1)		是
美国东部 (俄亥俄州) (us-east-2)		是
美国东部 (弗吉尼亚州北部) (us-east-1)	支持	是
美国西部 (北加利福尼亚) (us-west-1)		是
美国西部 (俄勒冈州) (us-west-2)	支持	是
AWS GovCloud (美国东部) (us-gov-east-1)	支持	是

AWS 区域	会议控制	会议媒体
AWS GovCloud (美国西部) (us-gov-west-1)	支持	是

* 您必须在 AWS 账户中启用这些区域。有关更多信息，请参阅AWS 一般参考中的[启用区域](#)。

** 在以色列（特拉维夫）使用会议控制功能的会议只能在以色列（特拉维夫）区域主持媒体。

Note

要在 AWS GovCloud（美国）地区创建会议，您必须使用中的控制区域 GovCloud。此外，中的控制区域 GovCloud 只能在 AWS GovCloud（美国）区域举行会议。

媒体管道区域

Amazon Chime SDK 媒体管道具有控制区域和媒体区域。控制区域提供用于创建和删除媒体管道的媒体管道 API 终端节点。您还可以使用控制区域接收和处理[媒体管道事件](#)。

媒体区域运行您的媒体管道，系统会自动选择与会议相同的媒体区域。

您可以使用控制区域在任何数据区域中创建媒体管道。媒体管道可以在任何会议媒体区域加入会议。

AWS 区域	控件	媒体
非洲（开普敦）(af-south-1)*		是
亚太地区（孟买）(ap-south-1)	支持	是
亚太地区（首尔）(ap-northeast-2)	支持	是
亚太地区（新加坡）(ap-southeast-1)	支持	是
亚太地区（悉尼）(ap-southeast-2)	支持	是

AWS 区域	控件	媒体
亚太地区 (东京) (ap-northeast-1)	支持	是
加拿大 (中部) (ca-central-1)	支持	是
欧洲地区 (法兰克福) (eu-central-1)	支持	是
欧洲地区 (爱尔兰) (eu-west-1)		是
欧洲 (伦敦) (eu-west-2)	支持	是
欧洲地区 (米兰) (eu-south-1)*		是
欧洲地区 (巴黎) (eu-west-3)		是
欧洲地区 (斯德哥尔摩) (eu-north-1)		是
南美洲 (圣保罗) (sa-east-1)		是
美国东部 (俄亥俄州) (us-east-2)		是
美国东部 (弗吉尼亚州北部) (us-east-1)	支持	是
美国西部 (北加利福尼亚) (us-west-1)		是
美国西部 (俄勒冈州) (us-west-2)	支持	是

* 您必须在 AWS 账户中启用这些区域。有关更多信息，请参阅AWS 一般参考中的[启用区域](#)。

消息收发区域

Amazon Chime SDK 消息收发具有控制区域和数据区域。控制区域公开消息收发的 API 终端节点，而数据区域存储消息。如果您使用 Amazon Kinesis 流式传输消息数据或渠道流 AWS Lambda 函数，则它们应位于控制区域。

AWS 区域	控件	数据
欧洲地区 (法兰克福) (eu-central-1)	支持	是
美国东部 (弗吉尼亚州北部) (us-east-1)	支持	是

PSTN 区域

Amazon Chime SDK SIP (会话初始协议) 功能有 API 区域和媒体区域以及 P STN 区域。API 区域提供用于创建和配置 SIP 功能的 API 终端节点。媒体区域包含 Amazon Chime SDK 语音连接器和 SIP 媒体应用程序。PSTN 区域帮助客户将本地电话系统连接至公共电话网络。此外，PSTN 区域还支持电话号码配置和管理。

AWS 区域	API	媒体	PSTN
亚太地区 (首尔) (ap-northeast-2)	支持	是	
亚太地区 (新加坡) (ap-southeast-1)	支持	是	
亚太地区 (悉尼) (ap-southeast-2)	支持	是	
亚太地区 (东京) (ap-northeast-1)	支持	是	

AWS 区域	API	媒体	PSTN
加拿大 (中部) (ca-central-1)	支持	是	
欧洲地区 (法兰克福) (eu-central-1)	支持	是	
欧洲地区 (爱尔兰) (eu-west-1)	支持	是	
欧洲 (伦敦) (eu-west-2)	支持	是	
美国东部 (弗吉尼亚州北部) (us-east-1)	支持	是	是*
美国西部 (俄勒冈州) (us-west-2)	支持	是	是*

* 有关特定 AWS 地区电话号码可用性的信息，请参阅 [Amazon Chime SDK 定价](#) 页面。

与客户端库集成

使用 Amazon Chime SDK 构建实时会议客户端之前，必须将您的客户端应用程序与 Amazon Chime SDK 客户端库集成。以下客户端库可供使用：

- [适用于 Android 的 Amazon Chime SDK 客户端库](#) — Kotlin 库，可用于在受支持的 Android 设备上构建 Amazon Chime SDK 应用程序。
- [适用于 C++ 的 Amazon Chime SDK 信令客户端库](#) — C++ 库，可用于在嵌入式设备上设置与 Amazon Chime SDK 会议的信令连接。
- [适用于 iOS 的 Amazon Chime SDK 客户端库](#) — Swift 库，可用于在受支持的 iOS 设备上构建 Amazon Chime SDK 应用程序。
- [适用于 JavaScript 的 Amazon Chime SDK 客户端库 \(NPM\)](#) — 带有 TypeScript 类型定义的 JavaScript 库，可用于在启用 WebRTC 的浏览器中构建 Amazon Chime SDK 应用程序。
- [适用于 Windows 的 Amazon Chime SDK 客户端库](#)。可用于在受支持的设备上构建 Amazon Chime SDK 应用程序的 C++ 库。

要了解如何将您的客户端应用程序与 Amazon Chime SDK 集成，请参阅客户端库 README.md 文件中的操作。使用演示学习如何为您的应用程序构建特定的媒体组件。

使用 Amazon Chime SDK 语音连接器进行 SIP 集成

将兼容 SIP 的语音基础设施与 Amazon Chime SDK 语音连接器集成以进行 SIP 语音呼叫。您必须使用 us-east-1 或 us-east-2 区域。您必须具有 IP 专用小交换机 (PBX)、会话边界控制器 (SBC) 或其他具有互联网访问权限且支持会话启动协议 (SIP) 的语音基础设施。有关更多信息，请参阅 Amazon Chime SDK 管理员指南中的[开始前的准备工作](#)。

集成您的语音基础设施

1. 在您的 AWS 账户下创建 Amazon Chime SDK 语音连接器。有关更多信息，请参阅《Amazon Chime SDK 管理员指南》中的[创建 Amazon Chime SDK 语音连接器](#)。
2. 编辑您的 Amazon Chime SDK 语音连接器设置，允许从您的语音基础设施呼叫 AWS。有关更多信息，请参阅《Amazon Chime SDK 管理员指南》中的[编辑 Amazon Chime SDK 语音连接器](#)。
 - a. 对于终止设置，选择已启用。
 - b. 对于允许列表，选择新建。
 - c. 输入内部 SIP 基础设施的 IP 地址的 CIDR 表示法。这允许您的基础设施访问 Amazon Chime SDK 语音连接器。例如，要允许来自 IP 地址 10.24.34.0 的流量，将 CIDR 表示法 10.24.34.0/32 列入许可名单。
 - d. 选择 Add (添加)。
 - e. 对于 Calling plan (呼叫计划)，选择要为其添加呼叫计划的国家/地区。
 - f. 根据需要编辑任何其他设置，然后选择保存。
3. 在 Amazon Chime SDK 控制台的语音连接器下，查看 Amazon Chime SDK 语音连接器的出站主机名。例如，`abcdefghijklmno3pqr4.voiceconnector.chime.aws`。
4. 要使用 Amazon Chime SDK 加入会议，通过 SIP URI 向 Amazon Chime SDK 语音连接器的出站主机名发送 SIP 请求。使用 SIP URI 中的电话号码 **+17035550122**。设置 transport 参数，使用 TLS 协议。最后，使用通过调用 [CreateAttendee](#) API 操作生成的唯一加入令牌。有关更多信息，请参阅以下示例。

Example 示例：SIP 请求

以下示例显示用于向 Amazon Chime SDK 语音连接器发送 SIP 请求的 SIP URI 的内容。

```
sip:+17035550122@abcdefghijklmnopqr4.voiceconnector.chime.aws;transport=tls;X-chime-join-token=join-token
```

以下示例显示加入 Amazon Chime SDK 会议的示例 SIP 邀请消息。

```
INVITE sip:
+17035550122@abcdefghijklmnopqr4.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@abcdefghijklmnopqr4.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 只能识别 E.164 格式的电话号码。确保 From 标头中有 E.164 格式的电话号码。

Amazon Chime SDK 事件通知

Amazon Chime SDK 支持发送事件通知到 Amazon EventBridge、Amazon Simple Queue Service (Amazon SQS) 和 Amazon Simple Notification Service (Amazon SNS)。

Note

Amazon Chime SDK 会议命名空间默认使用 ChimeSDKMeetings 终端节点。传统的 Chime 命名空间使用单个终端节点。有关命名空间和终端节点的更多信息，请参阅 [迁移到 Amazon Chime SDK 会议命名空间](#)。

向 EventBridge 发送通知

您可以向 EventBridge 发送 Amazon Chime SDK 事件通知。有关使用 Amazon Chime SDK 与 EventBridge 的详细信息，请参阅《Amazon Chime SDK 管理员指南》中的[使用 EventBridge 自动运行 Amazon Chime SDK](#)。有关 EventBridge 的信息，请参阅《Amazon EventBridge 用户指南》<https://docs.aws.amazon.com/eventbridge/latest/userguide/>。

向 Amazon SQS 和 Amazon SNS 发送通知

您可以使用 Amazon Chime SDK API 参考中的 [CreateMeeting](#) API 将 Amazon Chime SDK 会议事件通知发送到每个会议的一个 Amazon SQS 队列和一个 Amazon SNS 主题。这可以帮助减少通知延迟。有关 Amazon SQS 权限的更多信息，请参阅 [Amazon 简单队列服务开发人员指南](#)。有关 Amazon SNS 的更多信息，请参阅 [Amazon Simple Notification Service 开发人员指南](#)。

发送到 Amazon SQS 和 Amazon SNS 的通知包含的信息与 Amazon Chime SDK 发送到 EventBridge 的通知相同。Amazon Chime SDK 支持向用于创建会议的 API 区域中的队列和主题发送会议事件通知。事件通知可能未按发生顺序传递。

授予 Amazon Chime SDK 访问 Amazon SQS 和 Amazon SNS 的权限

在 Amazon Chime SDK 通过 Amazon SQS 队列或 Amazon SNS 主题向您发送通知之前，您必须授予 Amazon Chime SDK 向队列或主题的 Amazon 资源名称 (ARN) 发布消息的权限。为此，将 AWS Identity and Access Management (IAM) 策略附加到队列或主题，以授予对 Amazon Chime SDK 的相应权限。有关更多信息，请参阅《Amazon Simple Queue Service 开发人员指南》中的 [Amazon SQS 中的身份识别与访问管理](#) 以及《Amazon Simple Notification Service 开发人员指南》中的 [Amazon SNS 访问控制示例案例](#)。

Note

您的亚马逊 SQS 队列或亚马逊 SNS 主题必须使用与您的 Amazon Chime SDK API 终端节点相同的 AWS 区域。

Example 允许 Amazon Chime SDK 将事件发布到 Amazon SQS 队列

以下示例 IAM policy 授予 Amazon Chime SDK 向指定的 Amazon SQS 队列发布会议活动通知的权限。请注意 `aws:SourceArn` 和 `aws:SourceAccount` 的条件语句。他们解决了潜在的[混淆代理](#)问题。

Note

- 创建以下策略时，您可以使用 `aws:SourceArn` 或 `aws:SourceAccount`。您不需要同时使用这两种方法。
- 这些示例使用 `ChimeSDKMeetings` 命名空间及相应的终端节点。如果使用 `Chime` 命名空间，则必须使用 `chime.amazonaws.com` 终端节点。

```
{
  "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"
      }
    }
  }
]
}

```

此示例显示了一项 Amazon SNS 策略，允许 Amazon Chime SDK 向您的 SNS 主题发送会议事件通知。

```

{
  "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"
    }
  }
}
```

如果为服务器端加密 (SSE) 启用 Amazon SQS 队列，则必须执行额外步骤。将 IAM policy 附加到相关 AWS KMS 密钥，授予 Amazon Chime SDK 执行加密添加到队列的数据所需 AWS KMS 操作的权限。

```
{
  "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 允许 Amazon Chime SDK 将事件发布到 Amazon SNS 主题

以下示例 IAM policy 授予 Amazon Chime SDK 向指定的 Amazon SNS 主题发布会议事件通知的权限。

```
{
  "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"
      }
    }
  }
}
```

从 Amazon Chime 命名空间迁移

Amazon Chime SDK 在一组终端节点上公开 API。尽管您可以直接向终端节点发出 HTTPS 请求，但许多客户在其应用程序中使用 AWS SDK 调用服务 API。AWS SDK 有不同的语言版本，其通过封装请求签名和重试逻辑来简化 API 调用。AWS SDK 包括每个服务终端节点的命名空间。

首次启动时，Amazon Chime SDK 会与 Amazon Chime 应用程序共享一个终端节点。因此，解决方案使用 AWS SDK 中的 Chime 命名空间调用 Amazon Chime 应用程序和 Amazon Chime SDK API。

Amazon Chime SDK 现在为每项子服务（例如会议和 PSTN 音频）提供专用的终端节点。每个终端节点均可通过 AWS SDK 中的专用命名空间寻址。

以下主题列出了服务、命名空间和终端节点，并描述了如何在代码的 AWS CLI 中使用。

主题

- [终端节点、命名空间和 CLI 命令](#)
- [每项服务的迁移帮助](#)
- [API 映射](#)

终端节点、命名空间和 CLI 命令

下表列出了专用 Amazon Chime SDK 命名空间、终端节点和 CLI 命令。这些链接可提供每项服务的更多信息。

端点	AWS SDK 命名空间	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
voice-chime	ChimeSDKVoice	chime-sdk-voice

每项服务的迁移帮助

所有客户都应考虑使用专用的 Amazon Chime SDK 终端节点访问最新的 Amazon Chime SDK 功能、API 和 AWS 区域。如果您将共享终端节点与 Chime 命名空间一起使用，请在迁移之前参阅以下迁移指南了解技术差异。

- [迁移到 Amazon Chime SDKIdentity 命名空间](#)
- [迁移到 Amazon Chime SDKMediaPipelines 命名空间](#)
- [迁移到 Amazon Chime SDK 会议命名空间](#)
- [迁移到 Amazon Chime SDKMessaging 命名空间](#)
- [迁移到 Amazon Chime SDKVoice 命名空间](#)

API 映射

下表列出了 Chime 命名空间中的 API 及其相应的专用命名空间和 API。部分专用 API 与 Chime API 有所不同，下表显示了这些实例。

Chime 命名空间 API	专用命名空间	专用命名空间 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
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*	不适用	
CreateProxySession	voice-chime	CreateProxySession
CreateSipMediaApplication	voice-chime	CreateSipMediaApplication

Chime 命名空间 API	专用命名空间	专用命名空间 API
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
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

Chime 命名空间 API	专用命名空间	专用命名空间 API
DeleteVoiceConnector	voice-chime	DeleteVoiceConnector
DeleteVoiceConnectorEmergencyCallingConfiguration	voice-chime	DeleteVoiceConnectorEmergencyCallingConfiguration
DeleteVoiceConnectorGroup	voice-chime	DeleteVoiceConnectorGroup
DeleteVoiceConnectorOrigin	voice-chime	DeleteVoiceConnectorOrigin
DeleteVoiceConnectorProxy	voice-chime	DeleteVoiceConnectorProxy
DeleteVoiceConnectorStreamingConfiguration	voice-chime	DeleteVoiceConnectorStreamingConfiguration
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

Chime 命名空间 API	专用命名空间	专用命名空间 API
DescribeChannelModerator	messaging-chime	DescribeChannelModerator
DisassociatePhoneNumbersFromVoiceConnector	voice-chime	DisassociatePhoneNumbersFromVoiceConnector
DisassociatePhoneNumbersFromVoiceConnectorGroup	voice-chime	DisassociatePhoneNumbersFromVoiceConnectorGroup
GetAppInstanceRetentionSettings	identity-chime	GetAppInstanceRetentionSettings
GetAppInstanceStreamingConfigurations	messaging-chime	GetMessagingStreamingConfigurations
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

Chime 命名空间 API	专用命名空间	专用命名空间 API
GetVoiceConnectorGroup	voice-chime	GetVoiceConnectorGroup
GetVoiceConnectorLoggingConfiguration	voice-chime	GetVoiceConnectorLoggingConfiguration
GetVoiceConnectorOrigination	voice-chime	GetVoiceConnectorOrigination
GetVoiceConnectorProxy	voice-chime	GetVoiceConnectorProxy
GetVoiceConnectorStreamingConfiguration	voice-chime	GetVoiceConnectorStreamingConfiguration
GetVoiceConnectorTermination	voice-chime	GetVoiceConnectorTermination
GetVoiceConnectorTerminationHealth	voice-chime	GetVoiceConnectorTerminationHealth
ListAppInstanceAdmins	identity-chime	ListAppInstanceAdmins
ListAppInstances	identity-chime	ListAppInstances
ListAppInstanceUsers	identity-chime	ListAppInstanceUsers
ListAttendees	meetings-chime	ListAttendees
ListAttendeeTags*	不适用	
ListChannelBans	messaging-chime	ListChannelBans
ListChannelMemberships	messaging-chime	ListChannelMemberships
ListChannelMembershipsForAppInstanceUser	messaging-chime	ListChannelMembershipsForAppInstanceUser
ListChannelMessages	messaging-chime	ListChannelMessages
ListChannelModerators	messaging-chime	ListChannelModerators

Chime 命名空间 API	专用命名空间	专用命名空间 API
ListChannels	messaging-chime	ListChannels
ListChannelsModeratedByAppInstanceUser	messaging-chime	ListChannelsModeratedByAppInstanceUser
ListMediaCapturePipelines	media-pipelines-chime	ListMediaCapturePipelines
ListMeetings*	不适用	
ListMeetingTags+	meetings-chime	ListTagsForResource
ListProxySessions	voice-chime	ListProxySessions
ListSipMediaApplications	voice-chime	ListSipMediaApplications
ListSipRules	voice-chime	ListSipRules
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

Chime 命名空间 API	专用命名空间	专用命名空间 API
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
SendChannelMessage	messaging-chime	SendChannelMessage
StartMeetingTranscription	meetings-chime	StartMeetingTranscription
StopMeetingTranscription	meetings-chime	StopMeetingTranscription
TagAttendee*	不适用	
TagMeeting+	meetings-chime	TagResource
TagResource	identity-chime	TagResource
	media-pipelines-chime	TagResource
	meetings-chime	TagResource
	messaging-chime	TagResource
	voice-chime	TagResource
UntagAttendee*	不适用	

Chime 命名空间 API	专用命名空间	专用命名空间 API
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
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 已被另一个名称的 API 所取代。

* API 不再可用。

使用 Amazon Chime SDK 会议

本部分中的主题介绍如何使用 Amazon Chime SDK 会议创建自定义会议应用程序。我们建议按所列顺序查看以下主题。

主题

- [迁移到 Amazon Chime SDK 会议命名空间](#)
- [使用会议区域](#)
- [创建会议](#)
- [选择会议功能](#)
- [WebRTC 媒体](#)
- [配置视频编解码器](#)
- [网络配置](#)
- [会议事件](#)
- [Amazon CloudWatch 指标](#)
- [创建 Amazon Chime SDK 媒体管道](#)
- [使用 Amazon Chime SDK 实时转录](#)
- [使用媒体复制](#)
- [故障排除和调试 Amazon Chime SDK 会议](#)

迁移到 Amazon Chime SDK 会议命名空间

[Amazon Chime SDK 会议命名空间](#)是 API 的专用位置，用于创建和管理 Amazon Chime SDK 会议资源。您可以使用命名空间在 Amazon Chime SDK 会议 API 终端节点所在的任何 AWS 区域对其进行寻址。如果您刚开始使用 Amazon Chime SDK，则使用此命名空间。有关“区域”的更多信息，请参阅本指南中的 [可用区](#)。

使用 [Amazon Chime](#) 命名空间的现有应用程序应计划迁移到专用命名空间，以使用最新 API 和功能。

主题

- [迁移原因](#)
- [迁移之前](#)
- [命名空间之间的差异](#)

迁移原因

出于以下原因，我们鼓励您迁移到 [Amazon Chime SDK 会议](#) 命名空间：

选择 API 终端节点

Amazon Chime SDK 会议命名空间是唯一可以在任何[提供 API 终端节点的区域](#)中使用 API 终端节点的 API 命名空间。如果您想使用 us-east-1 以外的 API 终端节点，则必须使用 Amazon Chime SDK 会议命名空间。

有关 Amazon Chime SDK 会议如何使用 AWS 区域的更多信息，请参阅本指南中的[会议区域](#)。

更新和新的会议 API

我们仅在 Amazon Chime SDK 会议命名空间中添加或更新会议 API。

迁移之前

在迁移之前，记下命名空间之间的差异。下表列出并描述了以上差异。

	Amazon Chime SDK 会议命名空间	Amazon Chime 命名空间
AWS SDK 命名空间	ChimeSDKMeetings	Chime
区域	多个	仅限 us-east-1
终端节点	https://meetings-chime.region.amazonaws.com	https://service.chime.amazonaws.com
服务主体	meetings.chime.amazonaws.com	chime.amazonaws.com
API	仅用于会议的 API	用于会议和 Amazon Chime 其他部分的 API
CreateMeeting	ExternalMeetingId 和 MediaRegion 为必填。	ExternalMeetingId 和 MediaRegion 是可选的。
CreateMeetingWithAttendees	ExternalMeetingId 和 MediaRegion 为必填。	ExternalMeetingId 和 MediaRegion 是可选的。

	Amazon Chime SDK 会议命名空间	Amazon Chime 命名空间
ListMeetings	不可用	可用
ExternalMeetingId	验证包括模式匹配	可用
ExternalUserId	验证包括模式匹配	可用
会议标签 API	TagResource、UntagResource、ListTagsForResource	TagMeeting、UntagMeeting、ListMeetingTags
与会者标签	不可用	可用
回声抑制	可用	不可用
实时转录语言识别	可用	不可用
与会者能力	可用	不可用
媒体复制	可用	不可用
AppKeys 和 TenantIds	可用	不可用
媒体管道	媒体管道支持 Amazon Chime SDK 会议命名空间中的多个区域。有关更多信息，请参阅 迁移到 ChimeSdkMediaPipelines 命名空间 。	通过 us-east-1 终端节点可用
SIP 媒体应用程序	JoinChimeMeeting 操作需要 MeetingId	JoinChimeMeeting 操作不需要 MeetingId
直接 SIP 集成	不可用	可用

命名空间之间的差异

以下各节解释了 Amazon Chime 与 Amazon Chime SDK Meetings 命名空间之间的差异。

AWS SDK 命名空间

Amazon Chime SDK 命名空间使用 Chime 正式名称。Amazon Chime SDK 会议命名空间使用 ChimeSDKMeetings 正式名称。名称的确切格式因平台而异。

例如，如果您在 Node.js 中使用 AWS SDK 来创建会议，则使用一行代码来寻址命名空间。

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

要迁移到 Amazon Chime 会议 SDK，使用新的命名空间和终端节点区域更新这行代码。

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

区域

[Amazon Chime](#) 命名空间只能寻址 us-east-1 区域的 API 终端节点。[Amazon Chime SDK 会议](#)命名空间可以在任何可用区域寻址 Amazon Chime SDK 会议 API 终端节点。有关会议区域的最新列表，请参阅本指南中的 [可用区](#)。

终端节点

[Amazon Chime SDK 会议](#)命名空间使用不同于 [Amazon Chime](#) 命名空间的 API 终端节点。

只能使用用于创建会议的终端节点对其进行修改。这意味着通过 EU-CENTRAL-1 中的终端节点创建的会议只能通过 EU-CENTRAL-1 进行修改。这也意味着您无法使用 US-EAST-1 中的 ChimeSDKMeetings 命名空间寻址通过 Chime 命名空间创建的会议。有关当前终端节点的更多信息，请参阅本指南中的 [API 映射](#)。

服务主体

[Amazon Chime SDK 会议](#)命名空间使用新的服务主体：meetings.chime.amazonaws.com。如果您有授予服务访问权限的 SQS、SNS 或其他 IAM 访问策略，则需要更新这些策略以授予新服务主体访问权限。

API

[Amazon Chime SDK 会议](#)命名空间仅包含用于创建和管理会议的 API。[Amazon Chime](#) 命名空间包括用于会议和 Amazon Chime 服务其他部分的 API。

CreateMeeting 必填字段

在 Amazon Chime SDK 会议命名空间中，[CreateMeeting](#) 和 [CreateMeetingWithAttendees](#) API 需要指定 ExternalMeetingId 和 MediaRegion 字段。

外部 ID 值

[Amazon Chime SDK 会议](#)命名空间强制对可用于 ExternalMeetingId 和 ExternalUserId 的值进行额外验证。

回声抑制

[Amazon Chime SDK 会议](#)命名空间提供基于机器学习的回声抑制功能，以帮助防止噪音和来自本地扬声器的声音重新传入会议。有关更多信息，请参阅 GitHub 上的指南。

与会者能力

[Amazon Chime SDK 会议](#)命名空间提供对会议中与会者发送和接收音频、视频和内容等功能的精细控制。

媒体复制

[Amazon Chime SDK 会议](#)命名空间提供媒体复制，可将主会议链接到副本会议，使多达 10,000 人参与实时会话。连接到副本会话的参与者会接收连接至主会话的主讲人的媒体，但他们可以被升级到主会议。有关更多信息，请参阅本指南中的 [使用媒体复制](#)。

AppKeys 和 TenantIds

[Amazon Chime SDK 会议](#)命名空间提供了一种限制网络对特定 Amazon Chime SDK 会议访问的方法。有关更多信息，请参阅本指南中的 [使用 AppKeys 和租户 ID](#)。

媒体管道

[Amazon Chime SDK 媒体管道](#)可与任何会议终端节点创建的会议配合使用，无论是使用 [Amazon Chime SDK 会议](#)，还是使用 [Amazon Chime](#) 命名空间。有关媒体管道区域的最新列表，请参阅[可用区域](#)。

SIP 媒体应用程序

[Amazon Chime SDK SIP 媒体应用程序](#)可与任何会议终端节点创建的会议配合使用，无论是使用 [Amazon Chime SDK 会议](#)，还是使用 [Amazon Chime](#) 命名空间。在通过 Amazon Chime SDK 会议命名空间创建的会议中使用 SIP 媒体应用程序时，[JoinChimeMeeting](#) 操作需要 MeetingId 参数。

其它 API

会议命名空间的 API 列表越来越多，而 Chime 命名空间没有这些 API。如果您刚开始使用 Amazon Chime SDK，请使用会议命名空间访问最新功能。

使用会议区域

Amazon Chime SDK 会议具有控制区域和媒体区域。控制区域提供用于创建、更新和删除会议的 API 终端节点。媒体区域主持实际会议。

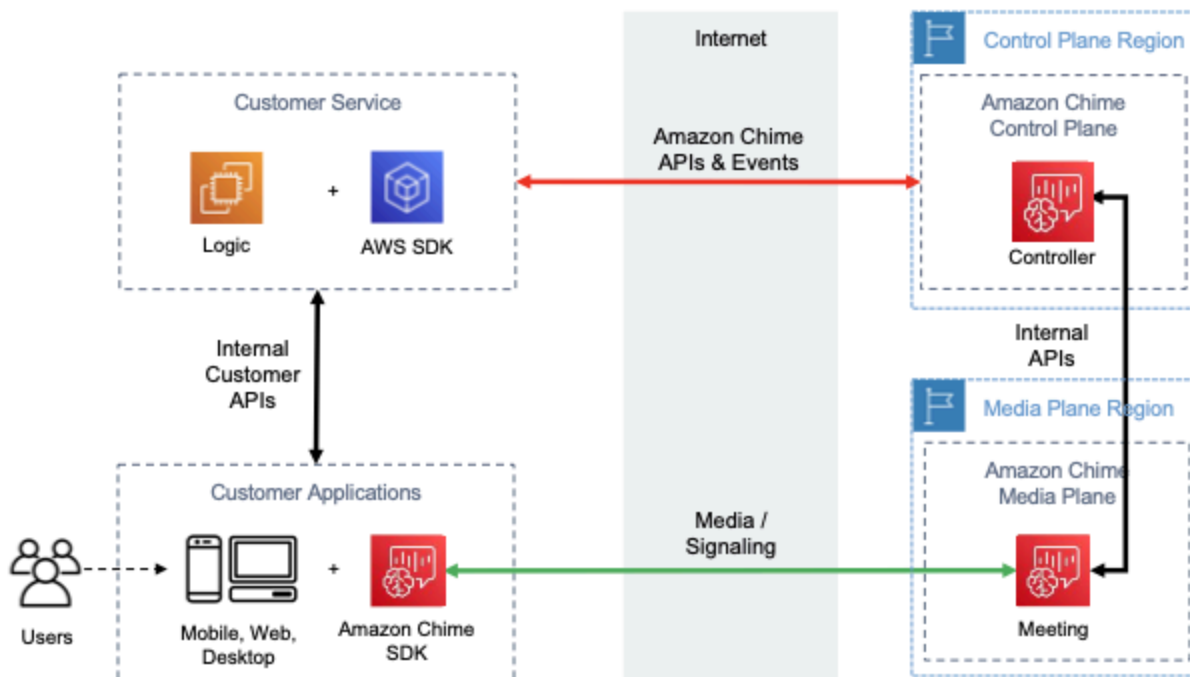
通常，您的应用程序服务使用 [AWS SDK](#) 对控制区域中的 API 进行[签名和调用](#)。您的应用程序客户端使用适用于 [JavaScript](#)、[iOS](#) 或 [Android](#) 的 Amazon Chime SDK 客户端库来连接媒体区域中的会议。

控制区域可以在同一 AWS 分区的任何媒体区域中创建会议。但是，您只能在用于创建会议的控制区域中更新会议。要查找离客户最近的媒体区域，请登录 <https://nearest-media-region.l.chime.aws>。

会议控制区域中的会议事件，例如 AttendeeJoined 调用 [EventBridge](#)、[Amazon Simple Queue Service \(SQS\)](#) 或 [Amazon Simple Notification Service \(SNS\)](#)。

有关可用的 Amazon Chime SDK 会议控制和媒体区域的列表，请参阅本指南中的 [可用区](#)。

此图表显示了通过控制区域和媒体区域的典型数据流。



选择控制区域

在为 Amazon Chime SDK 会议选择控制区域时，请记住以下因素：

- 监管要求。您的应用程序是否需要在地缘政治边界内，或者使用带有 FIPS 140-2 验证的加密模块的终端节点？

- API 延迟。使用距离应用程序服务的 AWS 区域最近的控制区域可以帮助缩短 API 的网络延迟。反过来，这有助于减少创建会议所需的时间，让用户更快地加入会议。
- 高可用性。您可以使用多个控制区域来实现高可用性架构。但是，每个控制区域都是独立运行的。而且，您只能在用于创建会议的控制区域中更新会议。此外，您必须使用相同的区域通过 [EventBridge](#)、[Amazon Simple Queue Service \(SQS\)](#) 或 [Amazon Simple Notification Service \(SNS\)](#) 使用会议事件。

选择媒体区域

Note

我们建议您始终在 [CreateMeeting](#) API 操作的 `MediaRegion` 参数中指定一个值。有关“区域”的更多信息，请参阅 [可用区](#)。

在为您的 Amazon Chime SDK 会议选择媒体区域时，考虑以下常见因素：

监管要求

如果您的 Amazon Chime SDK 会议需要根据法规要求在地缘政治边界内托管，则考虑根据固定的应用程序逻辑对会议区域进行硬编码。

例如，远程医疗应用程序可能需要将所有会议托管在执业医生的管辖范围内。如果应用程序支持位于欧洲和美国的诊所，则您可以通过每家诊所的地址选择其管辖范围内的区域。

会议质量

在媒体区域托管 Amazon Chime SDK 会议时，每个与会者的音频和视频都从该地区发送和接收。随着与会者与区域之间距离的增加，会议质量可能会受到网络延迟的影响。为您的 Amazon Chime SDK 会议指定区域有助于提高与会者的会议质量，无论他们彼此靠近还是分布在不同的地理位置。

您可以使用以下任一方法为 Amazon Chime SDK 会议选择媒体区域：

对媒体区域进行硬编码

如果您的 Amazon Chime SDK 会议全部在特定 AWS 区域托管，则建议采用这种方法。

选择最近的媒体区域

如果您的 Amazon Chime SDK 会议与会者处于同一 AWS 区域，但您的会议在不同区域托管，则建议采用这种方法。

查找最近的媒体区域

要查找能够托管 Amazon Chime SDK 会议的最近媒体区域，请登录 <https://nearest-media-region.l.chime.aws>。此终端节点返回单个区域，例如 {"region": "us-west-2"}。从您的客户端应用程序中调用 URL 标识距离用户最近的区域，然后使用 [CreateMeeting](#) API `MediaRegion` 参数中的结果在该区域创建会议。

通常在客户端应用程序启动或其网络连接更改时调用 URL。通过预先确定最近的区域，可以避免在创建会议时增加调用的延迟。

查找最近的 AWS GovCloud (US) 媒体区域

要查找能够托管 Amazon Chime SDK 会议的最近 AWS GovCloud (US) 媒体区域，请登录 <https://nearest-us-gov-media-region.l.chime.aws>。此终端节点返回最近区域，例如 {"region": "us-gov-west-1"}。从您的客户端应用程序中调用 URL 标识距离用户最近的 AWS GovCloud (US)，然后使用 [CreateMeeting](#) API `MediaRegion` 参数中的结果在该区域创建会议。

通常在客户端应用程序启动或其网络连接更改时调用 URL。通过预先确定最近的区域，可以避免在创建会议时增加调用的延迟。

JavaScript 示例

以下示例使用 HTML 和 JavaScript 返回最近的媒体区域和 AWS GovCloud (US) 媒体区域。

```
<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>

```

检查区域状态

登录 <https://region.status.chime.aws/> 检索每个区域中 Amazon Chime SDK 服务的运行状况。结果显示推荐区域。如果媒体区域的状态不是推荐，则最近的媒体区域终端节点将不会返回该区域。

以下示例显示典型结果。

```

{
  "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"
}
}
```

创建会议

以下程序演示如何为服务器和客户端应用程序创建带有音频和视频的会议。开始之前，您必须将您的客户端应用程序与 Amazon Chime SDK 客户端库集成。有关更多信息，请参阅[与客户端库集成](#)。

创建带有音频和视频的会议

1. 通过服务器应用程序完成以下步骤：
 - a. 使用 Amazon Chime SDK API 参考中的 [CreateMeeting](#) API 操作创建会议。使用 `MediaRegion` 参数指定 AWS 区域。有关选择会议区域的更多信息，请参阅 [会议区域](#)。
 - b. 使用 [CreateAttendee](#) API 操作或 [BatchCreateAttendee](#) API 操作将与会者添加到会议中。将会议和与会者从您的服务器应用程序安全转移至经授权的客户端，作为相应与会者。有关会议和与会者的更多信息，请参阅 Amazon Chime SDK API 参考中的 [会议](#) 和 [与会者](#)。
2. 通过客户端应用程序完成以下步骤：
 - a. 使用 Amazon Chime SDK 客户端库构造 `MeetingSessionConfiguration` 对象。使用前面步骤中的会议和与会者信息。
 - b. 实施 `AudioVideoObserver` 接口。
 - c. 使用 `MeetingSessionConfiguration` 创建 `MeetingSession`。
 - d. 使用 `MeetingSession` 中的 `AudioVideoFacade` 控制实时媒体。
 - i. 注册 `AudioVideoObserver` 接口的实例。这使您能够在会议状态发生变化时接收事件。
 - ii. 为音频输入、音频输出和视频输入选择初始设备。
 - iii. 开始视听会话。
 - iv. 当用户想要共享视频时，开始本地视频捕获。
 - v. 要显示视频磁贴，在客户端应用程序中管理视频磁贴事件，并将这些磁贴绑定至视频表面。

- vi. 管理其他用户互动，例如静音和取消静音，或者开始和停止本地视频捕获。
 - vii. 要离开会议，停止视听会话。
- e. (可选) 使用 `MeetingSession` 中的 `AudioVideoFacade` 与其他客户端共享媒体内容，例如屏幕截图等。
- i. 启动屏幕共享会话。内容以其他与会者形式加入会议。
 - ii. 要显示共享内容，在客户端应用程序中管理视频磁贴事件，并将这些磁贴绑定至表面。
 - iii. 管理其他互动，例如暂停、重启或停止内容共享。

当您运行 [DeleteMeeting](#) API 操作时，会议结束。此外，会议会在以下情况下自动结束：

- 会议时间超过 24 小时。
- 会议属于[副本会议](#)，且主会议结束。
- 在非副本会议中，五分钟内没有与会者连接。

选择会议功能

在调用 [CreateMeeting](#) API 时，您可以指定向加入会话的客户端提供的功能。请注意，某些功能选项会产生额外费用。

会话具有以下功能：

- `Audio.EchoReduction` - 机器学习回声抑制。
- `Video.MaxResolution` - 最大网络摄像头视频分辨率。
- `Content.MaxResolution` - 最大内容共享分辨率。
- `Attendees.MaxCount` - 最大参与者人数。

主题

- [使用音频。EchoReduction](#)
- [使用视频。MaxResolution](#)
- [使用内容。MaxResolution](#)
- [使用与会者。MaxCount](#)
- [在客户端应用程序中使用会议功能](#)

使用音频。EchoReduction

可使用 `Audio.EchoReduction` 帮助防止用户扬声器发出的声音传回会议中。

回声抑制非常适合用户的扬声器将成为会议音频的主要输出设备的情况。例如，当多个用户在会议室中使用同一台设备参加会议时，或者当单个远程参与者没有佩戴耳机时。

回声减少功能可在 JavaScript 和 React 客户端库中使用。有关更多信息，请参阅中的 [文档 GitHub](#)。需要支付额外费用，有关详细信息，请参阅 [Amazon Chime SDK 定价页面](#)。

使用视频。MaxResolution

可使用 `Video.MaxResolution` 指定会议的最大网络摄像头视频分辨率。此功能提供以下选项：

- None：不允许摄像头视频
- HD：高清摄像头视频 (1280x720p)
- FHD: full-high-definition 摄像机视频 (1920x1080)

如果请求 FHD (1080p) 视频，则会创建高清 WebRTC 会话。有关详细信息，请参阅 [Amazon Chime SDK 定价页面](#)。

如果客户端尝试发送超过指定最大值的网络摄像头视频，则该服务会拒绝此视频并发送以下错误：

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

使用内容。MaxResolution

可使用 `Content.MaxResolution` 指定会议的最大内容共享分辨率。此功能提供以下选项：

- None：不允许内容共享
- FHD: full-high-definition 内容共享 (1920x1080)
- UHD: ultra-high-definition 内容共享 (3840x2160)

如果请求 UHD (4K) 内容，则会创建高清 WebRTC 会话。

如果客户端尝试发送超过最大分辨率的内容共享，则共享内容的分辨率将被缩减至指定的最大分辨率。可以通过将 `MediaTrackConstraints` 应用于内容共享轨道进行扩缩。以下示例显示了如何扩缩共享轨道。

```

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})`
  );
}

```

下表显示了预期的内容共享行为。

内容功能	内容共享原生分辨率	扩展	内容编码分辨率
FHD	1280x720	否	1280x720
FHD	1920x1080	否	1920x1080
FHD	3840x2160	是	1920x1080
UHD	1920x1080	否	1920x1080
UHD	3840x2160	否	3840x2160
UHD	4200x2400	是	3780x2160

使用与会者。MaxCount

可使用 `Attendee.MaxCount` 指定允许加入会议的最大参与者人数。参加者的上限。MaxCount 取决于会话类型。对于标准会话，最多可以选择 250 名参与者。对于高清会话，最多只能选择 25 名参与者。

如果请求 FHD (1080p) 视频或 UHD (4K) 内容，会话将是高清会话。

高清会话需要支付参与者容量费用。有关详细信息，请参阅 [Amazon Chime SDK 定价页面](#)。

在客户端应用程序中使用会议功能

创建具有指定功能的会议

要创建会议，请调用 [CreateMeeting](#) API 并指定所需的会议功能。以下示例演示了如何指定所有功能。

```
// 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();
```

在客户端使用会议功能

创建具有所需功能的会议后，可以在创建 `MeetingSessionConfiguration` 对象时将这些功能传入 `joinInfo` 中。会议功能在创建 `meetingSession` 时用于设置网络摄像头视频分辨率和比特率，以及内容共享分辨率和比特率。

```
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)
);
```

WebRTC 媒体

Amazon Chime SDK 支持两种类型的 WebRTC 会话：标准和高清。以下主题介绍在使用 JavaScript、React、iOS 和 Android 的 Amazon Chime SDK 客户端库时，每种类型会话中可用的媒体。

主题

- [音频](#)
- [视频](#)
- [内容共享](#)
- [数据消息](#)

音频

每个 Amazon Chime 客户端向会话发送一个音频流，并从会话接收一个音频流。通常，本地设备上的麦克风会生成音频。接收到的音频是从其他会话客户端发送的音频的混合。

两个会话都支持高达 48 kHz 的采样率和最多 2 个通道（立体声），使用 Opus 编解码器进行编码，比特率最高可达 128 Kbps。但是，发送和接收的音频流因客户端库类型而有所区别：

- 适用于 JavaScript 和 React 的 Amazon Chime SDK 客户端库支持以设备和浏览器支持的最高采样率（最高 48 kHz）发送和接收单声道和立体声音频。
- 适用于 iOS 和安卓的 Amazon Chime SDK 客户端库支持发送高达 48kHz 的单声道音频和接收 48kHz 的立体声音频。

视频

每个 Amazon Chime 客户端都可向会话发送一个视频流，并从会话接收最多 25 个视频流。发送的视频通常来自本地设备的摄像头。每个客户端最多可以选择接收 25 个视频流，并可在会话期间随时更改选择。

标准会话支持分辨率高达 1280 x 720 的视频以每秒 30 帧速度使用 H.264、VP8、VP9 和 AV1 进行编码，其比特率最高可达 1500 Kbps。

高清会话支持分辨率高达 1920 x 1080 的视频以每秒 30 帧速度使用 H.264、VP8、VP9 和 AV1 进行编码，其比特率最高可达 2500 Kbps。

适用于 JavaScript 和 React 的 Amazon Chime SDK 客户端库支持以每秒 15 帧的速度在联播中发送视频，或者使用可扩展视频编码 (SVC) 发送视频。SVC 以 100%、50% 和 25% 的目标值对具有三个空间层和三个时间层的单个视频流进行编码。此服务会根据观看者的可用带宽自动选择要发送给每位观看者的层。

适用于 iOS 和 Android 的 Amazon Chime SDK 客户端库支持每秒最多发送 15 帧。但是，实际的帧速率和分辨率由 Amazon Chime SDK 自动管理。

视频编码和解码在可用情况下使用硬件加速来提高性能。

如果客户端发送其比特率大于允许的最大比特率的视频，则会话将首先通过实时控制协议开始向客户端发送接收器估计的最大比特率消息。如果客户端继续发送比特率大于允许的最大比特率的视频，会话将丢弃传入视频流数据包。

内容共享

最多两个客户端可以向会议共享内容。内容共享可以包括视频轨道和/或音频轨道。内容共享的一个常见示例是屏幕共享，即使用屏幕截图作为内容的来源。另一示例是将预先录制的内容与视频和音频轨道共享。

会话将内容音频混合到所发送的音频流中。内容音频支持高达 48kHz 的采样率和最多 2 个通道（立体声），使用 Opus 编解码器进行编码，比特率最高可达 128kbps。

视频内容通过单独的视频流发送到会话并转发至客户端。标准会话支持高达 1920 x 1080、每秒 30 帧的内容视频。高清会话支持高达 3840 x 2160、每秒 30 帧的内容视频。

用于内容共享的屏幕截图使用正在捕获的屏幕或窗口的分辨率、会话类型的最大内容分辨率，以及最高每秒 30 帧的速率。但是，设备和浏览器的容量可能会限制这些值。

适用于 JavaScript 和 React 的 Amazon Chime SDK 客户端库支持从屏幕截图和其他来源共享内容。

适用于 iOS 和安卓的 Amazon Chime SDK 客户端库支持从屏幕截图共享内容。

数据消息

数据消息为客户端提供了一种向会话中的其他客户端广播信息的方式。例如，应用程序可能会在会话期间使用数据消息共享表情符号反应。

每条数据消息包括：

- 一个主题，最长 64 个字符的字符串。
- 最高 2KB 的数据，包括主题。

客户端会向会话发送数据消息，而会话会将数据消息发送给所有连接的客户端。

会话可以选择缓存数据消息，最长五分钟。如果客户端加入或重新连接到会话，该会话将自动向客户端发送此前尚未发送的所有缓存数据消息。会话缓存最多可存储 1024 条数据消息。

一个会话支持每秒最多发送 100 条数据消息。使用[实时转录](#)时，每个客户端都通过数据消息接收[转录消息](#)，这些消息计入每秒发送的消息总数。

配置视频编解码器

客户端设备使用视频编解码器在原始视频发送到服务之前对其进行压缩，并在渲染接收到的视频之前对其进行解压缩。

使用适用于 JavaScript 的 Amazon Chime SDK 客户端库时，可以指定发送视频的编解码器首选项。

适用于 iOS 和 Android 的 Amazon Chime SDK 客户端库会根据设备的功能自动为您选择编解码器。

设置视频编解码器首选项

在适用于 JavaScript 的 Amazon Chime SDK 客户端库中，可以为网络摄像头和内容视频指定独立的视频编解码器首选项。

使用 [AudioVideoControllerFacade.setVideoCodecSendPreferences](#) 函数设置发送网络摄像头视频的编解码器首选项。此链接会将您转至 GitHub。

您的首选项以有序数组的形式传递，首先传递您最希望使用的编解码器，最后传递您最不希望使用的编解码器。

当提供多个编解码器首选项时，该服务会自动选择所有会话参与者都可以解码的首选编解码器。

以下示例演示如何为具有 VP8 回退选项的 VP9 设置视频编解码器首选项：

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

以下情况适用于这些首选项：

- 最佳 - 客户端使用 VP9 编解码器对视频进行编码。
- 本地回退 - 如果客户端不支持 VP9 编码，则会转为使用 VP8 编码。如果客户端不支持 VP8 编码，则会转为使用浏览器和服务支持的任何编解码器。
- 远程回退 - 如果会话中的另一个客户端没有 VP9 解码器，则该客户端会转为使用 VP8 编码。
- 本地故障 - 如果客户端不支持 VP9 或 VP8 编码，将不会发送视频。

要设置内容共享首选项，请使用

[ContentShareControllerFacade.setContentShareVideoCodecPreferences](#) 函数设置发送内容视频的编解码器首选项。此链接会将您转至 GitHub。

以下示例为具有 VP8 回退选项的 VP9 设置内容视频编解码器首选项。

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

网络配置

当您将 Amazon Chime SDK 集成到客户端应用程序中时，该 SDK 会连接到其后端服务，以发送和接收音频、视频、内容共享和数据消息。如果用户的网络阻止了 Amazon Chime SDK 服务的流量，其使用该服务的能力将会受到影响。网络管理员可以使用这些信息重新配置其网络，以允许访问 Amazon Chime SDK 服务。

Note

配置网络时，必须默认启用 DNS (EDNS0) 扩展机制。这可以确保主机信息的大小与 UDP 数据包的大小一致，以使您的应用程序能够访问 Amazon Chime SDK 服务。

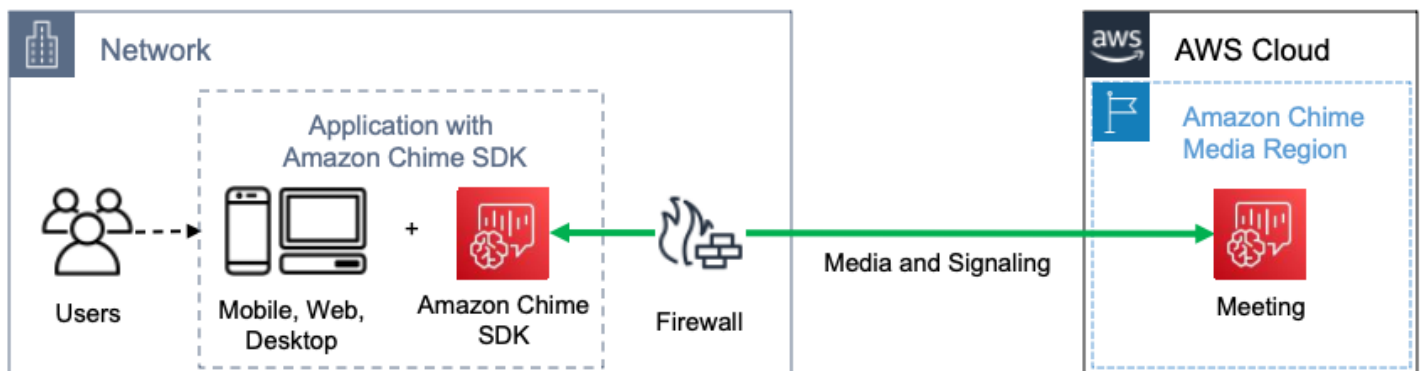
主题

- [配置媒体和信令](#)
- [配置 Amazon Voice Focus](#)
- [配置回声抑制](#)
- [配置背景替换和背景模糊](#)
- [配置浏览器内容安全策略](#)
- [使用 AppKeys 和租户 ID](#)

配置媒体和信令

Amazon Chime SDK 音频、视频和内容尽可能使用用户数据报协议 (UDP) 传输。如果 UDP 被阻止，Amazon Chime SDK 会尝试为双向媒体传输建立传输层安全性协议 (TLS) 连接。Amazon Chime SDK 信令和消息使用传输控制协议 (TCP) 和 WebSocket 连接。

下图显示了一个典型的网络，其中包含运行 Amazon Chime SDK 的应用程序。



Amazon Chime SDK 针对媒体和信令使用以下目标和端口。

Domain	子网	端口
*.chime.aws	99.77.128.0/18	TCP:443

Domain	子网	端口
		UDP:3478

此子网是 [AWS IP 地址范围](#) 内的 CHIME_MEETINGS 服务。

配置 Amazon Voice Focus

适用于 iOS 和 Android 的 Amazon Chime SDK 客户端库包括 Amazon Voice Focus 模块。适用于 JavaScript 的 Amazon Chime SDK 客户端库可从 Amazon CloudFront 下载 Amazon Voice Focus 模块。适用于 Windows 的 Amazon Chime SDK 客户端库不支持 Voice Focus。

Amazon Voice Focus 使用以下目标和端口。

Domain	端口
*.sdkassets.chime.aws	TCP:443

此子网是 [AWS IP 地址范围](#) 内的 CLOUDFRONT 服务。

配置回声抑制

适用于 JavaScript 的 Amazon Chime SDK 客户端库可从 Amazon CloudFront 下载回声抑制模块。

回声抑制使用以下目标和端口。

Domain	端口
*.sdkassets.chime.aws	TCP:443

此子网是 [AWS IP 地址范围](#) 内的 CLOUDFRONT 服务。

配置背景替换和背景模糊

适用于 JavaScript 的 Amazon Chime SDK 客户端库可从 Amazon CloudFront 下载背景替换和背景模糊模块。

背景替换和背景模糊使用以下目标和端口。

Domain	端口
*.sdkassets.chime.aws	TCP:443

此子网是 [AWS IP 地址范围](#) 内的 CLOUDFRONT 服务。

配置浏览器内容安全策略

使用适用于 JavaScript 的 Amazon Chime SDK 客户端库构建应用程序时，您需要在应用程序中配置浏览器内容安全策略。有关更多信息，请参阅 GitHub 上的 [内容安全策略指南](#)。

使用 AppKeys 和租户 ID

您可以使用 AppKeys 和 TenantID 来限制从网络访问特定应用程序的 Amazon Chime SDK WebRTC 媒体会话。

开发人员使用 Amazon Chime SDK 创建通过 UDP 发送和接收实时视频的应用程序。应用程序用户需要对 [CHIME_MEETINGS](#) 子网的 UDP 访问权限。组织（网络所有者）可以使用 AppKeys 和 TenantIds 来限制从其网络上访问特定应用程序的 WebRTC 媒体会话。

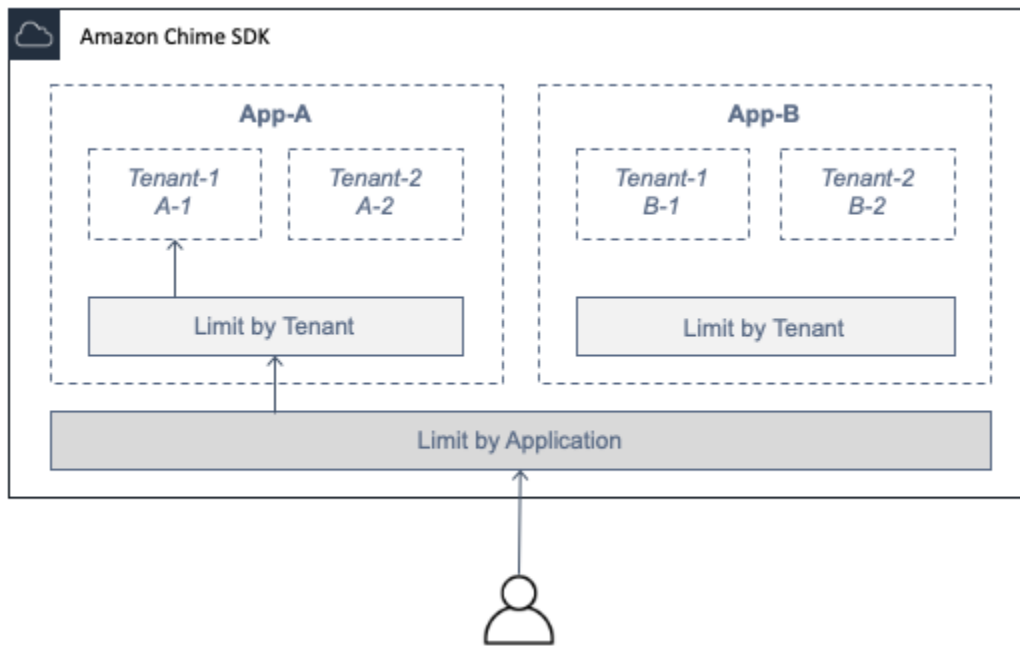
示例 1：使用 AppKeys

如果应用程序 A 和 App-B 使用 Amazon Chime SDK，则组织可以允许应用程序 A 从其网络访问 WebRTC 媒体会话，但可以屏蔽应用程序 B 和使用 Amazon Chime SDK 的任何其他应用程序。Organizations 可以使用 App-A AppKey 和 HTTPS 代理来做到这一点。更多信息，请参阅该主题后面的 [限制对特定应用程序的访问](#)。

示例 2：使用 AppKeys 和租户标识

如果 App-A 公开可用并被许多客户使用，组织可能希望仅当其用户参与会话时才允许 App-A 从其网络访问 WebRTC 媒体会话，并阻止访问所有其他 App-A 会话。组织可以通过使用应用程序 AppKey、组织的 TenantId 和 HTTPS 代理来做到这一点。更多信息，请参阅该主题后面的 [限制对特定租户的访问](#)。

要使用 AppKeys 和 TenantIds，你必须有一个允许向请求添加 HTTPS 标头的 HTTPS 代理服务器。下图显示了 TenantID 的工作 AppKeys 原理。



在图中，App-A 有租户 A-1 和 A-2，App-B 有租户 B-1 和 B-2。在这种情况下，AppKey 仅允许App-A 连接到WebRTC媒体会话，而租户ID仅允许租户A-1加入该会话。

主题

- [限制对特定应用程序的访问](#)
- [限制对特定租户的访问](#)
- [HTTPS 标头示例](#)

限制对特定应用程序的访问

A AppKey是 Amazon Chime 为每个账户创建的一致且唯一的 256 位值。AWS如果您没有 AppKey，可以向 Amazon Support 申请一个。如果您有多个AWS账户，则可以 AppKey 申请所有账户的通用账户。

Note

您可以安全地 AppKeys 公开共享您的信息，并允许其他组织限制来自其网络的访问。

Amazon Chime SDK 会 AppKey 根据用于创建会话的AWS账户 ID 自动将每个 WebRTC 媒体会话与关联起来。要限制从您的网络访问特定应用程序，请执行以下操作：

1. 通过 HTTPS 代理服务器将所有出站请求路由到 CHIME_MEETINGS 子网。
2. 配置代理服务器，向 CHIME_MEETINGS 子网的所有出站请求添加以下标头：

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

例如，X-Amzn-Chime-App-Keys:*AppKey-A, AppKey-B, AppKey-C*允许与这些应用程序关联的应用程序 AppKeys 访问子网。

Amazon Chime SDK 会检查入站 WebRTC 媒体会话连接的 X-Amzn-Chime-App-Keys 标头并应用以下逻辑：

1. 如果标X-Amzn-Chime-App-Keys头存在并且包含会话的标头 AppKey，请接受连接。
2. 如果标X-Amzn-Chime-App-Keys头存在但不包括会话的标头 AppKey，则以 403 错误拒绝连接。
3. 如果 X-Amzn-Chime-App-Keys 标头不存在，则接受连接。如果用户能够从组织网络外部访问应用程序，他们也可以访问该会话。

限制对特定租户的访问

TenantID 是由开发人员创建的不透明标识符。请记住以下关于 TenantID 的信息：

- 不能保证 TenantID 在应用程序之间是唯一的，因此您必须 AppKey为每个 TenantID 列表指定一个。
- TenantID 区分大小写。根据开发人员的规定准确输入。
- 一个组织可以限制对多个应用程序的访问权限，但只能为部分应用程序指定 TenantID。没有 TenantID 的应用程序可以连接所有 WebRTC 媒体会话。

要将媒体会话与 TenantIDs 关联，开发者必须先将该TenantIds属性和 TenantID 列表添加到或请求中。[CreateMeetingCreateMeetingWithAttendees](#)

例如：

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

要在特定应用程序中限制从组织网络访问其 WebRTC 媒体会话，执行以下操作：

1. 按照 [限制对特定应用程序的访问](#) 中的步骤操作。

2. 将 HTTPS 代理服务器配置为在出站连接上添加 X-Amzn-Chime-Tenants 标头。包括 AppKeys 和 TenantId 的列表，分隔如以下示例所示：`X-Amzn-Chime-Tenants: AppKey-A:tenantId-A-1,tenantId-A-2;AppKey-B:tenantId-B-1,tenantId-B-2`

Amazon Chime SDK 会检查入站 WebRTC 媒体会话连接的 X-Amzn-Chime-Tenants 标头并应用以下逻辑：

- 如果标头包含该会话的 AppKey:tenantId，则接受连接。
- 如果标头包含会话的 AppKey，但与 tenantId 不匹配，则以 403 错误拒绝连接。
- 如果标头不包含会话的 AppKey，则接受连接。
- 如果标头包含会话 AppKey，但会话并无至少一个受允许的 tenantId，则以 403 错误拒绝连接。这可能是开发人员的错误。
- 如果标头不存在，则接受连接。如果用户能够从组织网络外部访问应用程序，他们也可以访问所有会话。

HTTPS 标头示例

以下示例显示了在 HTTPS 标头中使用 AppKeys 和 TenantIds 的一些方法。

一个应用程序只有一个租户

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

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

用户只能在指定应用程序中访问组织的 WebRTC 媒体会话。已阻止所有其他的应用程序。

一个应用程序有两个租户

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

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

用户只能在指定应用程序中访问工程和销售的媒体会话。已阻止所有其他的应用程序。

两个应用程序，一个仅限租户使用

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

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

用户只能在应用程序 1 中访问组织的媒体会话，而在应用程序 2 中可以访问任何会话。已阻止所有其他的应用程序。

会议事件

Amazon Chime SDK 发送会议生命周期事件，用于触发通知和启动下游工作流程。使用会议事件的部分示例包括：

- 与会者加入或离开 Amazon Chime SDK 会议时更新元数据。
- 为 Amazon Chime SDK 会议实施推送通知或花名册。
- 测量 Amazon Chime SDK 会议中视频和内容共享的使用情况。

您可以向亚马逊 EventBridge、亚马逊简单通知服务 (SNS) 和亚马逊简单队列服务 (SQS) 发送事件。有关更多信息，请参阅 Amazon EventBridge 用户指南中的[来自AWS服务的事件](#)。

Amazon Chime SDK 会议开始

新会议开始时 Amazon Chime SDK 会发送此事件。

Example 事件数据

以下示例显示此事件的数据。

```
{
  "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 会议结束

活动会议结束时 Amazon Chime SDK 会发送此事件。

Note

为了提高效率，该服务还会在您调用 [DeleteMeeting](#) API 时发送此事件。

Example 事件数据

以下示例显示此事件的数据。

```
{
  "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 与会者已添加

新与会者添加至活动会议时，Amazon Chime SDK 会发送此事件。

Example 事件数据

以下示例显示此事件的数据。

```
{
  "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 与会者已删除

当您使用 [DeleteAttendee](#) API 将与会者从正在进行的会议中移除时，Amazon Chime SDK 会发送此事件。

Example 事件数据

以下示例显示此事件的数据。

```
{
  "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 与会者已授权

当已加入会议的用户使用相同的加入令牌再次加入会议时，Amazon Chime SDK 会发送此事件。例如，用户可以从台式机切换到移动设备。这实际上是将会议“移交”到新设备上。

Example 事件数据

以下示例显示此事件的数据。

```
{  
  "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 与会者加入会议

现有与会者加入使用指定网络传输的 Amazon Chime SDK 会议时，Amazon Chime SDK 会发送此事件。

Example 事件数据

以下示例显示此事件的数据。

```
{  
  "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 与会者离开会议

现有与会者离开使用指定网络传输 Amazon Chime SDK 会议时，Amazon Chime SDK 会发送此事件。

Note

该服务从不为同一“离开”操作发送 `chime:AttendeeLeft` AND `chime:AttendeeDropped` 事件。丢弃和离开是不同的动作，系统会发送与每个动作对应的事件。

例如，假设一位联系不佳的与会者在上午 11 点加入会议。您可以期待以下操作：

```
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 事件数据

以下示例显示此事件的数据。

```
{
  "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 与会者已退出会议

当当前的与会者退出亚马逊 Chime SDK 会议时，Amazon Chime SDK 会发送此事件，通常是因为连接不畅。当服务在 10-15 秒内没有收到数据包时，它会考虑客户端已丢弃并发出事件。

该服务通常会触发丢弃操作，但客户端也可以触发它们。例如，假设用户将笔记本电脑从 Wi-Fi 切换到以太网。这构成了网络适配器的更改，连接被重置。反过来，这会重置 websocket 并触发组合的删除加入操作。

Note

该服务从不为同一“离开”操作发送 `chime:AttendeeLeft` AND `chime:AttendeeDropped` 事件。丢弃和离开是不同的动作，系统会发送与每个动作对应的事件。

例如，假设一位联系不佳的与会者在上午 11 点加入会议。您可以期待以下操作：

```
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 事件数据

以下示例显示此事件的数据。

```
{
  "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 与会者开始流视频

现有与会者开始流视频时，Amazon Chime SDK 会发送此事件。

Example 事件数据

以下示例显示此事件的数据。

```
{
  "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 与会者停止流视频

现有与会者停止流视频时，Amazon Chime SDK 会发送此事件。

Example 事件数据

以下示例显示此事件的数据。

```
{
  "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 与会者开始共享屏幕

现有与会者开始共享其屏幕时，Amazon Chime SDK 会发送此事件。

Example 事件数据

以下示例显示此事件的数据。

```
{
  "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 与会者停止共享屏幕

现有与会者停止共享其屏幕时，Amazon Chime SDK 会发送此事件。

Example 事件数据

以下示例显示此事件的数据。

```
{
  "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 与会者容量已更新

当现有与会者的容量更新时，Amazon Chime SDK 会发送此事件。

Example 事件数据

```
{
  "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 与会者内容加入会议

内容共享加入使用指定网络传输的 Amazon Chime SDK 会议时，Amazon Chime SDK 会发送此事件。

Example 事件数据

以下示例显示此事件的数据。

```
{
  "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 与会者内容离开会议

内容共享离开使用指定网络传输的 Amazon Chime SDK 会议时，Amazon Chime SDK 会发送此事件。

Example 事件数据

以下示例显示此事件的数据。

```
{
  "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 与会者内容失去会议连接

当内容共享从亚马逊 Chime SDK 会议中掉线时，Amazon Chime SDK 会发送此事件，这通常是因为带宽过低。

Example 事件数据

以下示例显示此事件的数据。

```
{
  "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 与会者内容开始流视频

内容共享开始流视频时，Amazon Chime SDK 会发送此事件。

Example 事件数据

以下示例显示此事件的数据。

```
{
  "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 与会者内容停止流视频

内容共享停止流视频时，Amazon Chime SDK 会发送此事件。

Example 事件数据

以下示例显示此事件的数据。

```
{
  "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 指标

当您使用 Amazon Chime SDK 时，它会向 CloudWatch 发送服务和使用情况指标。这些指标可用于通过 CloudWatch 图表和控制面板监控您使用 Amazon Chime SDK 服务的情况。这些指标会捕获您调用的每个 API 的数据。

以下部分列出并描述指标。

主题

- [服务指标](#)
- [API 使用情况指标](#)

服务指标

Amazon Chime SDK 向 AWS/ChimeSDK 命名空间发布以下服务指标：

指标	单位	描述
AttendeeAuthorizationSuccess	计数	成功授权尝试的总计数。成功意味着允许与会者加入会议。
AttendeeAuthorizationError	计数	授权失败总计数，表示与会者无法加入会议。
AttendeeAudioDrops	计数	音频掉线的总计数。
AttendeeContentDrops	计数	内容共享掉线总计数。
MeetingSQSNotificationErrors	计数	SQS 通知错误总计数。
MeetingSNSNotificationErrors	计数	SNS 通知错误总计数。

API 使用情况指标

API 使用情况指标与 AWS 服务限额对应。您可以配置警报，以在用量接近服务限额时向您发出警报。有关 CloudWatch 与服务限额集成的更多信息，请参阅《Amazon CloudWatch 用户指南》中的 [AWS 使用情况指标](#)。

Amazon Chime SDK 在 AWS/Usage 命名空间中使用 ChimeSDK 服务名称发布以下 API 指标。

指标	描述
CallCount	对 Amazon Chime SDK 中 API 的调用总数。SUM 表示在指定时间段内对 API 的调用总数。
ErrorCount	Amazon Chime SDK 中由 API 引发的错误总数。SUM 表示在指定时间段内对 API 的调用总数。

指标	描述
ThrottleCount	Amazon Chime SDK 中由 API 引发的节流错误总数。SUM 表示在指定时间段内对 API 的调用总数。

Amazon Chime SDK 通过以下维度向 AWS/Usage 命名空间发布使用情况指标：

维度	描述
服务	包含该资源的 AWS 服务的名称。对于 Amazon Chime SDK 使用情况指标，此维度的值为 ChimeSDK。
类型	正在报告的实体的类型。Amazon Chime SDK 使用情况指标的唯一有效值为 API。
资源	报告指标的资源类型。对于 Amazon Chime SDK 使用情况指标，此维度的值为 API。
类	所跟踪的资源的类。Amazon Chime SDK 指标的唯一有效值为 None。

创建 Amazon Chime SDK 媒体管道

Important

您和您的最终用户必须明白，录制 Amazon Chime SDK 会议可能需要遵守有关电子通信录制的法律或法规。您和您的最终用户有责任遵守与录制有关的所有适用法律，包括适当通知录制会话或通信中的所有参与者正在录制会话或通信，并获得他们的许可。

您和您的最终用户应对使用媒体实时连接器服务进行的所有流媒体内容负责，并且必须确保此类内容不会违反法律、侵犯或盗用任何第三方的权利，也不会以其他方式违反您与 Amazon 签订的协议中的重要条款。

要捕获或流式传输 Amazon Chime SDK 会议，您可以创建一个媒体管道。媒体管道可以由以下管道之一组成：

- **媒体捕获** — 您可以使用媒体捕获管道捕获音频、视频和内容共享流，以及会议活动和数据消息。所有媒体捕获管道都将其数据保存到您创建的 [Amazon Simple Storage Service \(S3\)](#) 存储桶中。您可以为每次 Amazon Chime SDK 会议创建一个媒体捕获管道。有关更多信息，请参阅本节后文中的 [管道创建概述](#)。
- **媒体串联** — 您可以使用媒体串联管道来连接媒体捕获管道中的构件。串联管道独立于媒体捕获和实时连接器管道工作。有关更多信息，请参阅本节后文中的 [创建媒体串联管道](#)。
- **媒体实时连接器** — 您可以使用媒体实时连接器管道连接到服务，这些服务使您能够将 Amazon Chime SDK 会议流式传输到 RTMP 终端节点。您可以为每次 Amazon Chime SDK 会议创建最多一个媒体实时连接器管道。有关更多信息，请参阅本节后文中的 [创建媒体实时连接器管道](#)。
- **媒体流** — 您可以使用媒体流管道捕获会议中所有与会者的个人音频，以及媒体连接管道生成的混合音频。所有媒体流管道会将其数据保存至 [Amazon Kinesis Video Streams \(KVS\)](#)。有关更多信息，请参阅本节后文中的 [创建媒体流管道](#)。

您创建的管道取决于您使用的命名空间。如果使用 Chime 命名空间，则只能创建媒体捕获管道。如果您使用 ChimeSdkMediaPipelines 命名空间，则还可以创建媒体串联和媒体实时连接器管道，并使用合成功能。如果要迁移到 ChimeSdkMediaPipelines 命名空间，请参阅 [迁移到 ChimeSdkMediaPipelines 命名空间](#)。

下表列出每个区域中活动媒体管道的默认限制。每种类型的管道都计入限制。

区域	默认活动管道限制
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

区域	默认活动管道限制
ca-central-1	10
eu-central-1	10
eu-west-2	10

Note

如果您超过任何区域的限制，[CreateMediaCapturePipelineCreateMediaConcatenationPipeline](#)、和 [CreateMediaLiveConnectorPipeline](#) API 将抛出超出资源限制异常。您可以使用 AWS 控制台中的 [Service Quotas](#) 页面来调整您的有效渠道限制，也可以联系您的 [客户支持代表](#)。有关 Amazon Chime SDK 会议限制的更多信息，请参阅 [Amazon Chime SDK 服务限额](#)。

开始之前，您必须将您的客户端应用程序与 Amazon Chime SDK 客户端库集成。有关更多信息，请参阅 [与客户端库集成](#)。有关媒体管道的更多信息，请参阅 [使用媒体管道捕获 Amazon Chime SDK 会议](#)。

主题

- [迁移到 ChimeSdkMediaPipelines 命名空间](#)
- [管道创建概述](#)
- [创建媒体捕获管道](#)
- [创建媒体串联管道](#)
- [创建媒体实时连接器管道](#)
- [将音频和视频合成单一视图](#)
- [创建媒体流管道](#)
- [为媒体管道创建服务相关角色](#)
- [使用媒体管道事件](#)
- [解析转录](#)
- [停止管道的最佳实践](#)

迁移到 ChimeSdkMediaPipelines 命名空间

您可以使用 ChimeSdkMediaPipelines 命名空间对任何可用的 AWS 区域中的媒体管道 API 终端节点进行寻址。如果您刚开始使用 Amazon Chime SDK，则使用此命名空间。有关“区域”的更多信息，请参阅本指南中的 [可用区](#)。

使用 [Amazon Chime](#) 命名空间的现有应用程序应计划迁移到专用命名空间。

主题

- [迁移管道的原因](#)
- [在迁移管道之前](#)

迁移管道的原因

出于以下原因，我们鼓励您迁移到 ChimeSdkMediaPipelines 命名空间：

选择 API 终端节点

Amazon Chime SDK 媒体捕获命名空间是唯一可以在任何可用区域中使用 API 终端节点的 API 命名空间。有关“区域”的更多信息，请参阅 [可用区](#)。如果您想使用 us-east-1 以外的 API 终端节点，则必须使用 ChimeSdkMediaPipelines 命名空间。有关当前终端节点的更多信息，请参阅本指南中的 [API 映射](#)。

更新和新的媒体管道 API

我们仅在 ChimeSdkMediaPipelines 命名空间中添加或更新媒体管道 API。

在迁移管道之前

在迁移之前，记下命名空间之间的差异。下表列出并描述了以上差异。

项目	媒体管道命名空间	Chime 命名空间
命名空间名称	ChimeSdkMediaPipelines	Chime
区域	多个	仅限 us-east-1
终端节点	https://media-pipelines-chime.region.amazonaws.com	https://service.chime.aws.amazon.com

项目	媒体管道命名空间	Chime 命名空间
服务主体	mediapipelines.chime.amazonaws.com	chime.amazonaws.com
API	仅适用于媒体管道的 API	用于媒体管道和 Amazon Chime 其他部分的 API
会议	us-west-2 、 ap-southeast-1 、 和 eu-central-1 区域中的媒体管道仅适用于在 Amazon Chime SDK 会议命名空间中创建的会议。 us-east-1 区域中的媒体管道适用于由任一命名空间中的任何会议终端节点创建的会议。	媒体管道适用于由任一命名空间中的任何会议终端节点创建的会议。
默认活动媒体管道	us-east-1 区域中的 100 以及 us-west-2、ap-southeast-1 和 eu-central-1 区域中的 10。	仅适用于 us-east-1 中的 100。
服务相关角色	AWSServiceRoleForAmazonChimeSDKMediaPipelines	
标签	可用	不适用于媒体管道 API。
CloudTrail 事件源	chime-sdk-media-pipelines.amazonaws.com	chime.amazonaws.com.
媒体直播连接器	可用	不适用于媒体管道 API。
合成	可用	不适用于媒体管道 API。
联接	可用	不可用。

以下列表提供了有关 Chime 和 AWS ChimeSdkMediaPipelines 命名空间之间差异的更多信息。

命名空间名称

Amazon Chime SDK 命名空间使用 `AWS.Chime` 正式名称。Amazon Chime SDK 媒体管道命名空间使用 `AWS.ChimeSDKMediaPipelines` 正式名称。名称的确切格式因平台而异。

例如，此行 Node.js 代码寻址 `chime` 命名空间：

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

要迁移到媒体管道 SDK 命名空间，使用新的命名空间和终端节点区域更新该代码。

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

区域

Amazon Chime 命名空间只能寻址 `US-EAST-1` 区域的 API 终端节点。Amazon Chime SDK 媒体管道命名空间可在任何拥有 Amazon Chime SDK 媒体管道 API 终端节点的区域中对其终端节点进行寻址。有关媒体管道区域的最新列表，请参阅本指南中的 [可用区](#)。

终端节点

要修改媒体捕获管道，必须使用与创建该管道相同的终端节点。例如，如果您通过 `eu-central-1` 中的终端节点创建管道，则必须使用 `eu-central-1` 与该管道进行交互。有关当前终端节点的更多信息，请参阅本指南中的 [API 映射](#)。

服务主体

[Amazon Chime SDK 媒体管道](#)命名空间使用新的服务主体：

`mediapipelines.chime.amazonaws.com`。如果您有 Amazon S3 存储桶或授予服务访问权限的其他 IAM 策略，则需要更新这些策略以授予新服务主体访问权限。

例如，创建媒体管道时，必须将 [创建 Amazon S3 存储桶](#) 中列出的策略权限添加到新的服务主体中。有关策略的更多信息，请参阅《IAM 用户指南》中的 [AWS JSON 策略元素：主体](#)。

API

Amazon Chime SDK 媒体管道命名空间仅包括用于创建和管理媒体管道的 API。Amazon Chime 命名空间包括用于媒体管道、会议和 Amazon Chime 服务其他部分的 API。

会议

IAD 区域中的媒体管道适用于由任一命名空间中的任何会议终端节点创建的会议。

服务相关角色

仅适用于 Amazon Chime SDK 媒体管道命名空间。创建 `AWSServiceRoleForAmazonChimeSDKMediaPipelines` 角色。

标签

[Amazon Chime SDK 媒体管道](#)命名空间支持标签。使用一个或多个标签调用 [CreateMediaCapturePipeline](#) 或 [CreateMediaLiveConnectorPipeline](#) API 时，角色必须拥有调用 `TagResource` 操作的权限。

管道创建概述

您可以按照多步骤流程创建 Amazon Chime SDK 媒体管道，也可以创建多种类型的管道。以下列表概述了创建过程，并提供了指向有关创建各种类型管道的更多信息的链接。

- 创建 Amazon S3 存储桶。您必须在与会议相同的 AWS 区域创建存储桶。有关更多信息，请参阅[创建 Amazon S3 存储桶](#)。
- 创建服务相关角色并命名为 `AWSServiceRoleForAmazonChimeSDKMediaPipelines`。这允许媒体管道代表您访问会议。有关更多信息，请参阅[为媒体管道创建服务相关角色](#)。
- 创建具有足够权限的 IAM 角色来与 [Amazon Chime SDK 媒体管道 API](#) 进行交互。要创建该角色，我们建议从 IAM 控制台添加[AmazonChime软件开发工具包](#)托管策略。该策略包含必要的 API。

您的 IAM 角色还必须有权在所有资源上调用 S3 [GetBucketPolicy](#)API。以下示例显示典型策略。

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

获得这些项目后，请参阅这些主题以获取有关创建管道的信息。

- [创建媒体捕获管道](#)

- [创建媒体串联管道](#)
- [创建媒体实时连接器管道](#)
- [创建媒体流管道](#)

创建媒体捕获管道

媒体捕获管道可捕获音频、视频和内容共享流，以及会议活动和数据消息。所有媒体捕获管道都将其数据保存到您创建的 [Amazon Simple Storage Service \(S3\)](#) 存储桶中。您可以为每次 Amazon Chime SDK 会议创建一个媒体捕获管道。

以下部分介绍如何创建媒体捕获管道。按照列出的顺序操作。

主题

- [创建 Amazon S3 存储桶](#)
- [为 Amazon S3 存储桶启用服务器端加密](#)
- [创建媒体捕获管道](#)
- [使用媒体捕获构件](#)
- [配置音频文件夹](#)
- [配置视频文件夹](#)
- [了解数据通道文件夹中的消息](#)
- [了解 S3 存储桶的文件夹结构](#)
- [了解会议活动文件](#)
- [了解转录文件](#)
- [串联数据流](#)

创建 Amazon S3 存储桶

您可以使用 Amazon S3 控制 AWS 台、软件开发工具包或 AWS CLI 来创建 Amazon S3 存储桶。有关更多信息，请参阅 Amazon Simple Storage Service (S3) 用户指南中的[创建存储桶](#)。

用于您的媒体捕获管道的 Amazon S3 存储桶必须与 Amazon Chime SDK 会议属于同一个 AWS 账户。此外，您必须向 Amazon Chime SDK 服务主体授权 `s3:PutObject` 和 `s3:PutObjectAcl` 的访问权限，即 mediapipelines.chime.amazonaws.com。您可以使用 Amazon S3 控制台或 AWS Command Line Interface (AWS CLI) 来做到这一点。S3 存储桶必须属于可用的 [Amazon Chime SDK 媒体区域](#) 之一。

Note

确保向您的 IAM 用户添加策略以授予对您的存储桶的访问权限。此外，如果您使用默认 AWS 禁用的区域，则该区域中必须有 Amazon S3 存储桶。

默认情况下，AWS 会禁用以下区域，在启用这些区域之前，您无法在其中托管会议资源：

- 非洲 (开普敦)
- 亚太地区 (香港)
- 亚太地区 (雅加达)
- 欧洲地区 (米兰)
- Middle East (Bahrain)

如果您使用其中一个区域，该区域必须具有 Amazon S3 存储桶。即使您使用 Amazon S3 API 与默认情况下未被屏蔽且已有存储桶的区域进行通信，也适用此操作。有关启用阻止区域的更多信息，请参阅 AWS 一般参考中的[管理 AWS 区域](#)。

创建存储桶后，记录其 ARN。您可以使用它创建媒体捕获管道。

以下代码示例显示 Amazon S3 存储桶策略。

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

```
        }  
    }  
]  
}
```

为 Amazon S3 存储桶启用服务器端加密

要为 Amazon Simple Storage Service (Amazon S3) 存储桶启用服务器端加密，您可以使用以下类型的加密密钥：

- Amazon S3 托管式密钥
- 密钥管理服务 (KMS) 中的 AWS 客户托管密钥

Note

密钥管理服务支持两种类型的密钥，即客户托管密钥和 AWS 托管密钥。Amazon Chime SDK 会议仅支持客户托管密钥。

使用 Amazon S3 托管密钥

使用 Amazon S3 控制台、CLI 或 REST API 为 Amazon S3 存储桶启用服务器端加密。在这两种情况下，都选择 Amazon S3 密钥作为加密密钥类型。无需进一步操作。当您使用存储桶进行媒体捕获时，会在服务器端上传和加密构件。有关更多信息，请参阅 Amazon S3 用户指南中的[指定 Amazon S3 加密](#)。

使用您拥有的密钥

要使用您管理的密钥启用加密，您需要使用客户托管密钥启用 Amazon S3 存储桶的服务器端加密，然后将语句添加至密钥策略，允许 Amazon Chime 使用该密钥并加密任何上传构件。

1. 在 KMS 中创建客户托管密钥。有关执行此操作的信息，请参阅 Amazon S3 用户指南中的[使用 AWS KMS \(SSE-KMS\) 指定服务器端加密](#)。
2. 在密钥策略中添加语句，允许该 `GenerateDataKey` 操作生成密钥由 Amazon Chime SDK 服务主体使用的密钥，`mediapipelines.chime.amazonaws.com`。

此示例显示一个典型语句。

```
...  
{
```

```
"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. 如果您使用媒体串联管道，在密钥策略中添加语句，允许 Amazon Chime SDK 服务主体 `mediapipelines.chime.amazonaws.com` 使用 `kms:Decrypt` 操作。
4. 配置 Amazon S3 存储桶以启用使用密钥的服务器端加密。

创建媒体捕获管道

创建并配置一个或多个 Amazon S3 存储桶后，即可创建媒体捕获管道。

创建媒体捕获管道

- 调用 [CreateMediaCapturePipelineAPI](#)。

将存储桶 ARN 用作 `SinkArn` 参数。

成功后，Amazon Chime SDK 会创建参加并捕获会议的与会者。

创建媒体捕获管道并设置其权限后，您可以创建媒体连接管道，将 5 秒钟的媒体区块串联成一个文件。有关更多信息，请参阅本节后文中的 [创建媒体串联管道](#)。

使用媒体捕获构件

在 Amazon Chime SDK 会议期间，媒体捕获管道会创建以下类型的构件。

- 音频

- 视频
- 数据通道消息
- 会议事件
- 转录消息

管道在您的 Amazon S3 存储桶中的一组文件夹中创建构件，您可以配置音频和视频文件夹以限制某些类型的构件。以下各节说明文件夹结构、如何配置文件夹、如何为 Amazon S3 存储桶设置权限以及如何连接构件文件。

配置音频文件夹

音频文件夹包含混合音频流的 5 秒 MP4 文件，即包含所有与会者的音频以及当前发言人的视频。该文件夹包含整个会议的文件。您可以根据需要将该文件夹配置为仅包含音频构件。每个文件名都包含一个 yyyy-mm-dd-hour-min-seconds-milleseconds 时间戳。时间戳采用 UTC，标记开始时间。您可以将该文件夹配置为仅包含音频构件。

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

配置视频文件夹

视频文件夹由包含视频流的 5 秒 MP4 文件，以及内容共享流（如果在 API 请求中指定）组成。每个文件名都包含一个 < yyyy-mm-dd-hour-min-seconds-milleseconds >-<attendeelD>时间戳和与会者 ID。内容共享视频块附加为 < yyyy-mm-dd-hour-min-seconds-milleseconds >-<attendeelD>#content .mp4。您可以将该文件夹配置为仅包含视频构件。

```
"ArtifactsConfiguration": {
  "Audio": {
    "MuxType": "AudioOnly"
  },
```

```

    "Content": {
      "State": "Disabled"
    },
    "Video": {
      "MuxType": "VideoOnly"
      "State": "Enabled"
    }
  }
}

```

了解数据通道文件夹中的消息

数据通道文件夹包含 .txt 格式的数据消息，每条消息都是一个 JSON 对象。消息在所有配置选项均可见。文件名包含 yyyy-mm-dd-hour-min-seconds-milleseconds 时间戳。此示例显示消息中的数据字段。

```

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

```

了解 S3 存储桶的文件夹结构

媒体捕获管道的 S3 存储桶使用该文件夹结构。

```

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

```

了解会议活动文件

会议事件文件夹包含 .txt 格式的会议事件，每个事件都是一个 JSON 对象。消息在所有配置选项均可见。文件名包含 < yyyy-mm-dd-hour-min-seconds-milleseconds > 时间戳。此示例显示典型事件文件中的字段和数据。

```

{
  "Timestamp": "string",

```

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

了解转录文件

转录消息文件夹包含 .txt 格式的转录文件。但是，只有在启用实时转录时，该文件夹才会接收文件。有关启用实时转录的更多信息，请参阅 [使用 Amazon Chime SDK 实时转录](#)。

该文件夹包含所有部分和完整的转录消息，每条消息都是一个 JSON 对象。文件名包含 < yyyy-mm-dd-hour-min-seconds-milleseconds > 时间戳。您可以在 [传输示例](#) 查看转录文件示例。

串联数据流

Note

要自动执行媒体捕获构件串联的过程，请参阅本指南中的 [创建媒体串联管道](#)。

此示例使用 ffmpeg 将视频或音频文件串联成一个 mp4 文件。首先，创建一个包含所有输入文件的 filelist.txt 文件。使用以下格式：

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

接下来，使用该命令串联输入文件：

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

有关媒体串联管道的更多信息，请参阅本指南中的 [创建媒体串联管道](#)。

创建媒体串联管道

您可以使用媒体串联管道连接媒体捕获管道生成的构件（文件）。

媒体捕获管道通过对媒体流进行分块并将这些构件存储在 S3 存储桶中来捕获会议内容。媒体捕获管道会创建以下类型的构件：

- 音频
- 视频
- 内容共享
- 数据通道消息
- 转录消息
- 会议事件
- 合成视频，即内容共享和以视频磁贴形式在网格中显示的多个视频流。

媒体串联管道允许您将每种类型的构件串联成一个文件，然后将这些较大的文件存储在 S3 存储桶中。无需等待媒体捕获事件结束即可创建媒体串联管道，但是只有在捕获管道停止时，串联管道才会开始串联。

Note

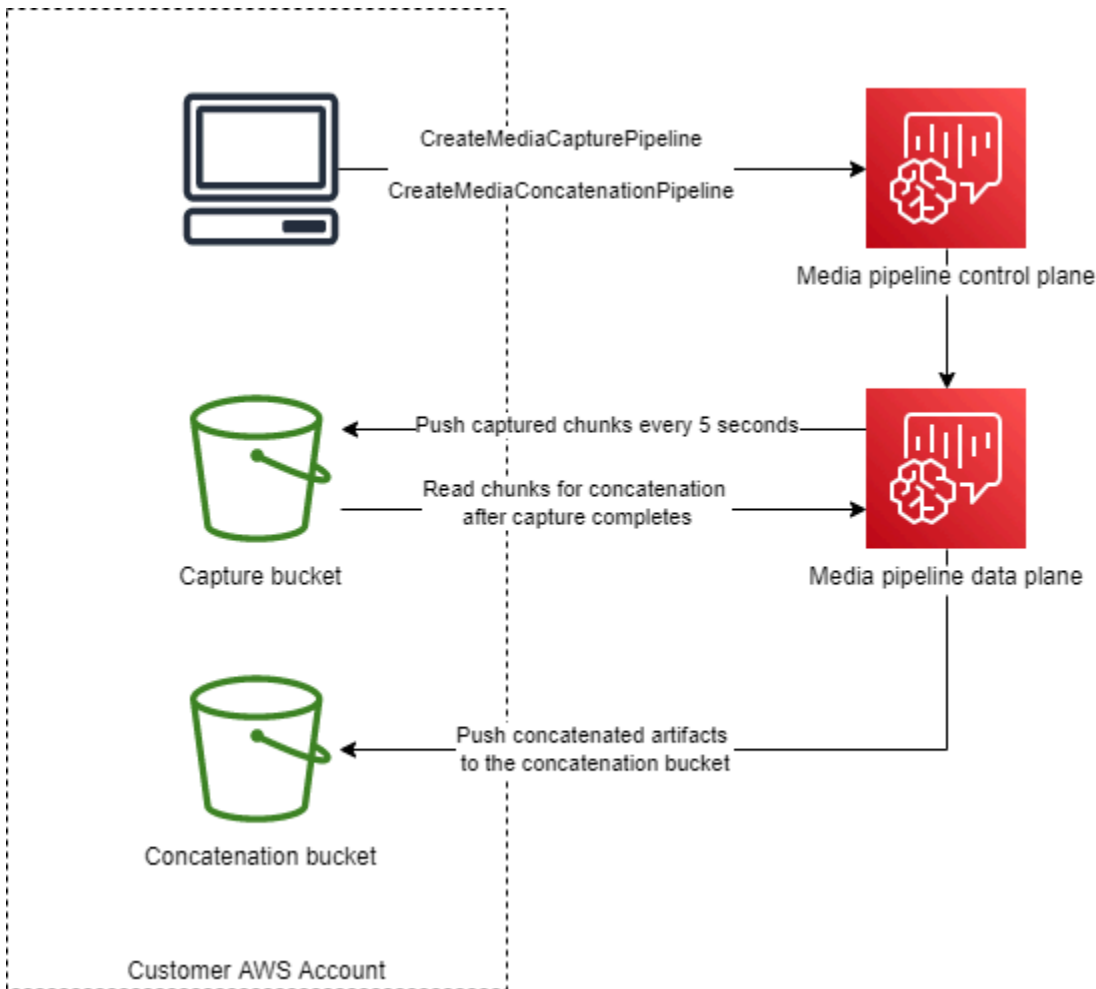
媒体捕获管道、媒体连接管道和 S3 存储桶必须位于同一个账户中。AWS

主题

- [串联管道架构](#)
- [构建媒体串联管道](#)
- [了解 S3 存储桶的文件夹结构](#)

串联管道架构

下图显示媒体串联管道的架构。



在图中，收到[CreateMediaCapturePipeline](#)请求后，媒体管道控制平面在媒体管道数据平面中启动媒体捕获管道。然后，数据面板每 5 秒钟将捕获的数据块推送到捕获桶。收到[CreateMediaConcatenationPipeline](#)请求后，媒体管道控制平面等待指定的媒体捕获管道完成，然后在媒体管道数据平面中启动媒体连接管道。然后，数据面板读取存储桶中捕获的数据块，并将串联的构件推送到串联存储桶。

构建媒体串联管道

您需要按照多步骤流程创建 Amazon Chime SDK 媒体串联管道。以下步骤描述了此过程。

1. 创建 Amazon S3 存储桶以用作媒体捕获管道的数据接收器，然后配置存储桶策略。有关为 S3 存储桶启用服务器端加密的信息，请参阅本指南中的[为 Amazon S3 存储桶启用服务器端加密](#)。如果您已创建用于媒体捕获管道的 Amazon S3 存储桶，则必须将 `s3:GetObject` 和 `s3:ListBucket` 操作添加到该存储桶的策略中。`s3:ListBucket` 操作需要对存储桶本身的访问权限。其他操作需要存储桶中对象的访问权限。您必须使用两个不同的 Amazon 资源名称 (ARN) 指定存储桶级和对象级的访问权限。

以下示例显示了存储桶策略。根据需要复制并粘贴此示例。

```
{
  "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. 创建 Amazon S3 存储桶以用作媒体串联管道的数据接收器，然后配置存储桶策略。有关为 S3 存储桶启用服务器端加密的信息，请参阅本指南中的[为 Amazon S3 存储桶启用服务器端加密](#)。

以下示例显示了该策略。

```
{
  "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

您可以将单个 S3 存储桶用于媒体捕获和媒体串联管道。但是，如果您这样做，则必须将 `s3:GetObject` 和 `s3:ListBucket` 的访问权限添加到步骤 2 中所示的媒体串联存储桶策略中。如果您不希望串联存储桶策略拥有这些权限，则为每个管道创建单独的存储桶。

- 使用 [CreateMediaCapturePipeline](#) API 创建媒体捕获管道。作为其中的一部分，获取管道的 ARN。有关获取 ARN 的信息，请参阅 [管道创建概述](#)。您将在下一个步骤中使用 ARN。
- 使用 [CreateMediaConcatenationPipeline](#) API 创建串联管道。

以下示例显示请求主体。## 字段是可选的，默认为串联管道的 ID。

Note

您必须使用最近 30 天内创建的 `MediaPipelineArn`。

```
{
  "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"
      }
    }
  ]
}
```

```
}
```

每当捕获管道停止时，串联就会开始。串联管道在完成串联后停止。

了解 S3 存储桶的文件夹结构

媒体串联管道的 Amazon S3 存储桶使用该文件夹结构。

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

Note

如果您在创建媒体管道时指定前缀，则文件夹的路径将变为存储桶名称/前缀。如果没有前缀，则路径将变为存储桶名称/媒体管道 ID。您可以在 `S3BucketSinkConfiguration` 对象的 `Destination` 字段中指定前缀。串联后的文件名由媒体管道 ID.mp4（媒体文件）和媒体管道 ID.txt（用于文本文件）组成。

创建媒体实时连接器管道

以下各节列出并描述媒体实时连接器管道的实时消息协议 (RTMP)、音频和视频设置。

RTMP 设置

媒体实时连接器管道支持通过 TLS/SSL 连接进行的 RTMP。接收器 URL 由流 URL 和流密钥组成。URL 遵循以下格式：

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

以下示例显示如何连接到常见的流传输平台。

- 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 使用加密帮助确保数据流不会被未经授权的实体拦截。作为最佳实践，当您需要更高的数据安全性时，使用 RTMPS。

音频设置

媒体实时连接器管道支持以下音频设置：

- 编解码器 — AAC
- 采样率 — 44100Hz 或 48000Hz。默认值为 44100Hz。
- 声道 — 单声道或立体声。默认值为单声道。

视频设置

媒体实时连接器管道使用 H264 编码器。您可以使用 1280 x 720 的高清或 1920 x 1080 的全高清。两种分辨率都使用每秒 30 帧，每两秒钟有一个关键帧。

停止媒体实时连接器管道

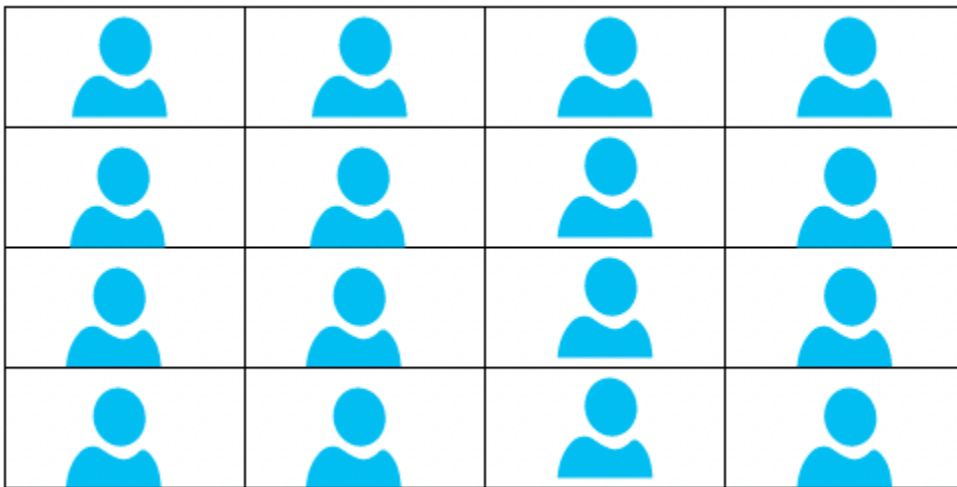
作为停止媒体直播连接器管道的最佳实践，请调用 [DeleteMediaPipeline](#) API。在 IVS 等流传输平台上终止数据流并不会停止媒体实时连接器管道。

将音频和视频合成单一视图

Amazon Chime SDK 媒体管道支持将音频、网络摄像头视频和内容共享流合成到单一视图中。然后，您可以使用直播连接器将该单一视图发送到流媒体服务，例如亚马逊互动视频服务、Twitch 或 YouTube Live。合成视频也可以捕获到 Amazon Simple Storage Service 中进行存储或进一步消耗。

合成使用名为 GridView 的默认屏幕布局，该布局具有以下行为。

- 当只有网络摄像头视频处于活动状态时，GridView 按以下网格模式组织视频流：



网格最多显示 25 个网络摄像头视频流，并根据用户打开摄像头的的时间对磁贴进行排序。

- GridView 提供了两个画布方向，Landscape 和 Portrait。横向是默认方向，全高清支持 1280 x 720 和 1920 x 1080 的视频分辨率。横向支持 720 x 1280 和 1080 x 1920 的全高清分辨率。
- 您可以配置视频磁贴的顺序、位置、总数、磁贴纵横比、角半径、边框颜色、边框厚度和高光颜色。
- 当会议期间有人共享屏幕时，网络摄像头视频磁贴会动态切换，以便为内容共享腾出空间。您可以使用下一节中描述的布局配置控制这些过渡以及视频磁贴的位置。

关于布局配置

当有人开始内容共享时，您可以使用以下布局配置选择如何合成内容共享和网络摄像头视频流。

- `ActiveSpeakerOnlyConfiguration` 全屏合成内容视频，将当前发言人的网络摄像头视频叠加在角落里。您可以指定角落。
- `PresenterOnlyConfiguration` 全屏合成内容视频，将主讲人的网络摄像头视频叠加在角落里。您可以指定角落。
- `VerticalLayoutConfiguration` 在相邻的垂直列中将内容视频与网络摄像头视频合成。您可以将该列显示在内容共享的右侧或左侧。
- `HorizontalLayoutConfiguration` 在相邻水平行中将内容视频与网络摄像头视频合成。您可以显示内容共享上方或下方的行。

根据内容共享是否处于活动状态，合成布局会自动在 GridView 和您选择的布局之间切换。

以下主题介绍如何使用全局 GridView 设置及每种配置布局。

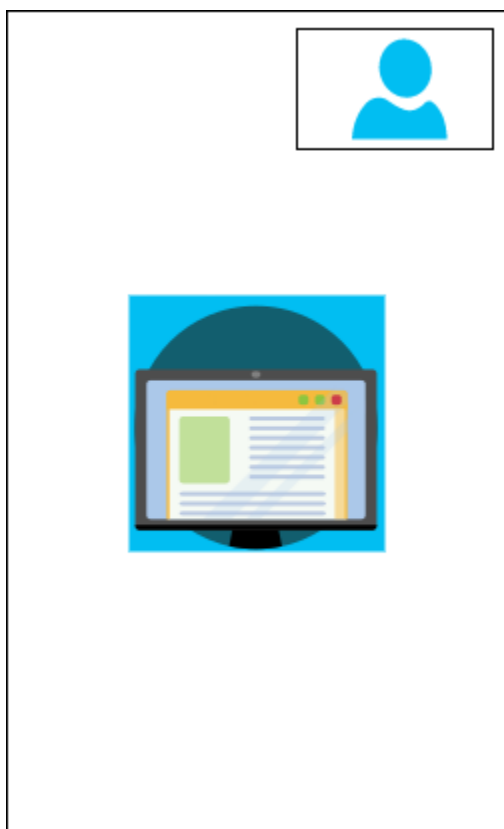
主题

- [设置画布方向](#)
- [设置边框和角属性](#)
- [使用布局配置](#)

设置画布方向

在合成中，画布包含您的所有视频流。您可以为画布指定 Landscape 或 Portrait 方向。横向提供 16:9 的纵横比。纵向提供 9:16 的纵横比。

下图显示了纵向方向。



以下示例显示如何使用右上角的视频磁贴完成纵向画布。在此示例中，活跃发言人显示在磁贴中。有关更多信息，请参阅 [ActiveSpeakerOnlyConfiguration](#)。

```
{
  "CompositedVideo": {
    "Layout": "GridView",
    "Resolution": "FHD",
    "GridViewConfiguration": {
```



```
    "ContentShareLayout": "ActiveSpeakerOnly",
    "ActiveSpeakerOnlyConfiguration": {
        "ActiveSpeakerPosition": "TopRight"
    }
},
"CanvasOrientation": "Portrait"
}
```

下图显示了横向方向。



CanvasOrientation

描述 — 方向设置，横向或纵向。

允许的值 — Landscape | Portrait

必填 — 否

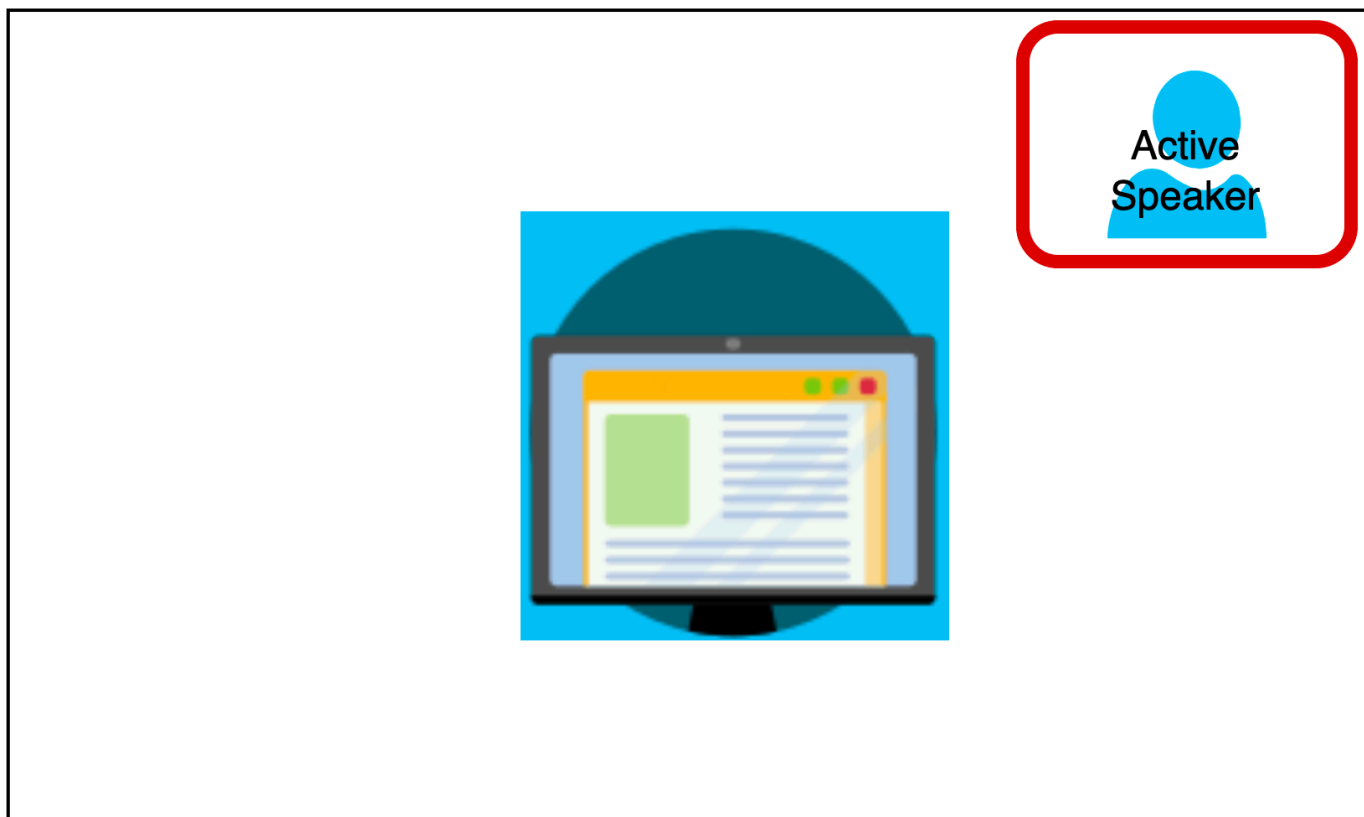
默认值 — 横向

设置边框和角属性

根据需要，您可以使用 `VideoAttribute` 参数为视频磁贴指定边框和角设置。您可以指定宽度、颜色和圆角。您也可以指定突出显示颜色，即当有人讲话时，边框会变为该颜色。

无论内容共享如何，您的属性设置都适用于所有布局。

下图显示应用了边框颜色和角半径的视频磁贴。



以下示例显示如何使用每个属性。在这种情况下，视频磁贴的圆角为五个像素半径。磁贴有绿色边框，宽度也为五个像素。当说话者说话时，该HighlightColor属性会将边框颜色更改为红色。

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

```
}
```

VideoAttribute

描述 — 指定视频磁贴边框和圆角的设置

允许的值 — `BorderColor` | `BorderThickness` | `CornerRadius` | `HighlightColor`

必填 — 否

VideoAttribute.BorderColor

描述 — 定义所有视频磁贴的边框颜色

允许的值 — 颜色名称，例如红色、绿色或蓝色

必填 — 否

VideoAttribute.BorderThickness

描述 — 定义所有视频磁贴的边框厚度（以像素为单位）

类型 — 整数

允许的值 — 1—20

必填 — 否

VideoAttribute.CornerRadius

描述 — 定义所有视频磁贴的角半径（以像素为单位）

类型 — 整数

允许的值 — 1—20

必填 — 否

VideoAttribute.HighlightColor

描述 — 定义主讲人或发言人讲话时显示的边框颜色

允许的值 — 颜色名称，例如红色、绿色或蓝色

必填 — 否

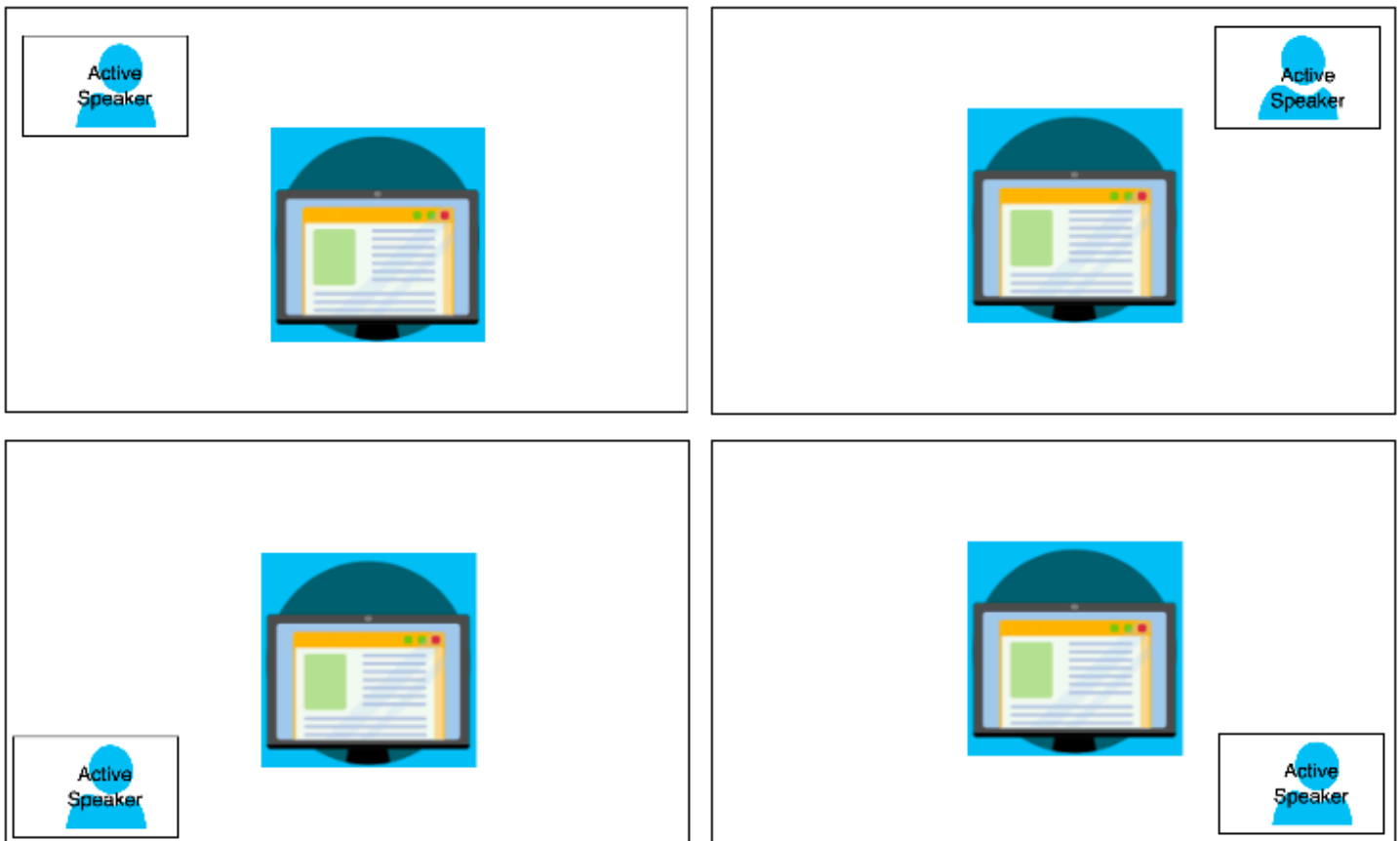
使用布局配置

以下主题介绍如何使用不同的配置布局。只有当有人开始共享内容时，布局才会生效。展开每个部分以了解更多信息。

ActiveSpeakerOnlyConfiguration

ActiveSpeakerOnlyConfiguration 显示内容共享和当前发言人的视频，这意味着讲话的人会出现覆盖内容共享流的小视频磁贴中。

下图显示发言人磁贴的配置和可用位置。



以下示例说明如何以编程方式实现 ActiveSpeakerOnly 布局。在这种情况下，主讲人磁贴会显示在左上角。

```
{  
  "CompositedVideo":{
```

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

ActiveSpeakerOnlyConfiguration

描述 — ActiveSpeakerOnly 视频磁贴的配置设置

类型- ActiveSpeakerOnlyConfiguration 对象

必填 — 否

ActiveSpeakerOnlyConfiguration.ActiveSpeakerPosition

描述 — 当前发言人视频磁贴的位置

类型 — 字符串

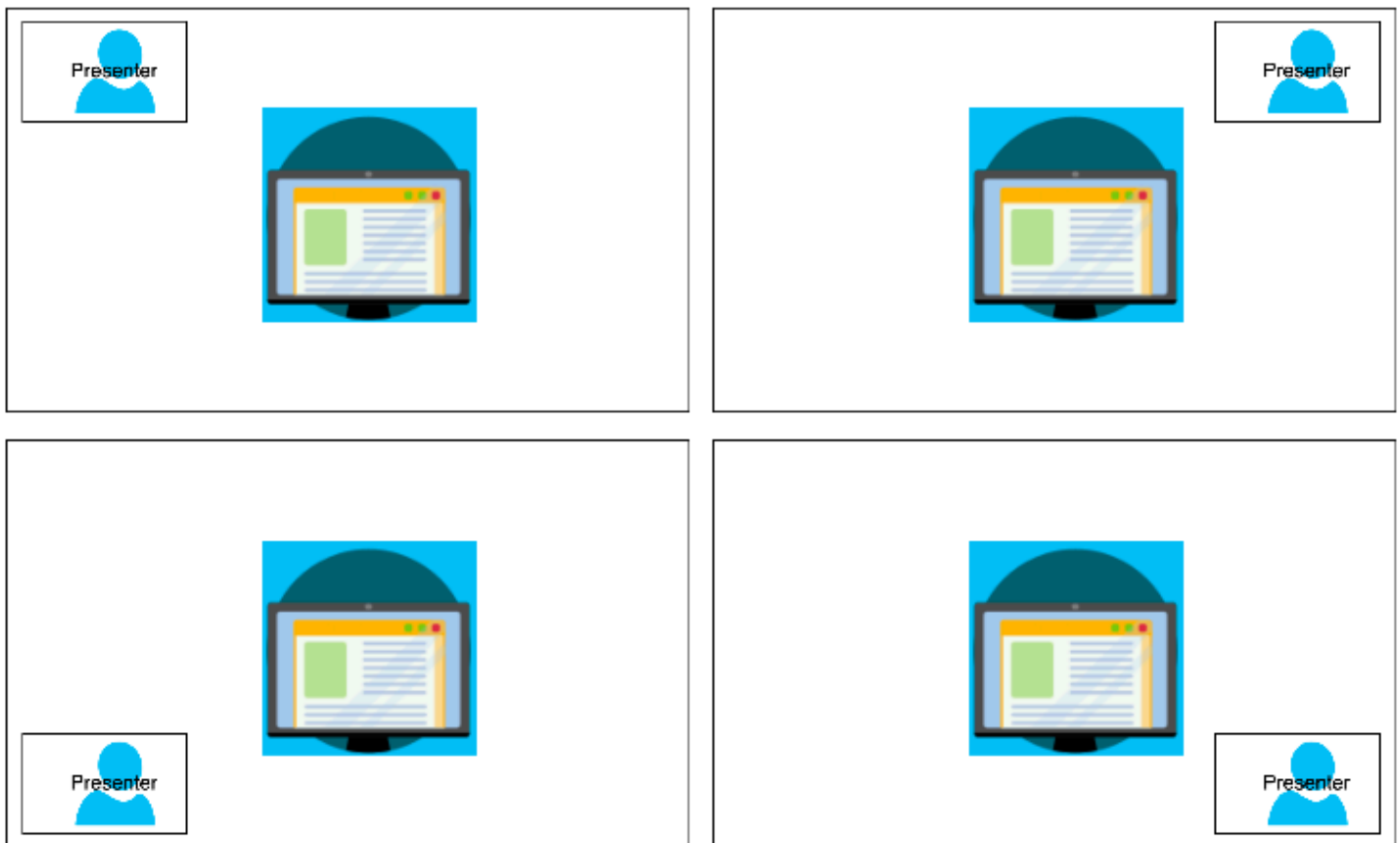
有效值 — TopLeft | TopRight | BottomLeft | BottomRight

必填 — 否

默认值 — TopRight

PresenterOnlyConfiguration

无论谁在讲话，PresenterOnlyConfiguration 都会显示内容共享以及仅主讲人的视频。下图显示配置。



以下示例说明如何以编程方式实现布局，并将主讲人放置在右上角。

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

PresenterOnlyConfiguration

描述 — PresenterOnly 布局的配置设置

类型 — PresenterOnlyConfiguration 对象

必填 — 否

PresenterOnlyConfiguration.PresenterPosition

描述 — 主讲人视频磁贴的位置

类型 — 字符串

有效值 — TopLeft | TopRight | BottomLeft | BottomRight

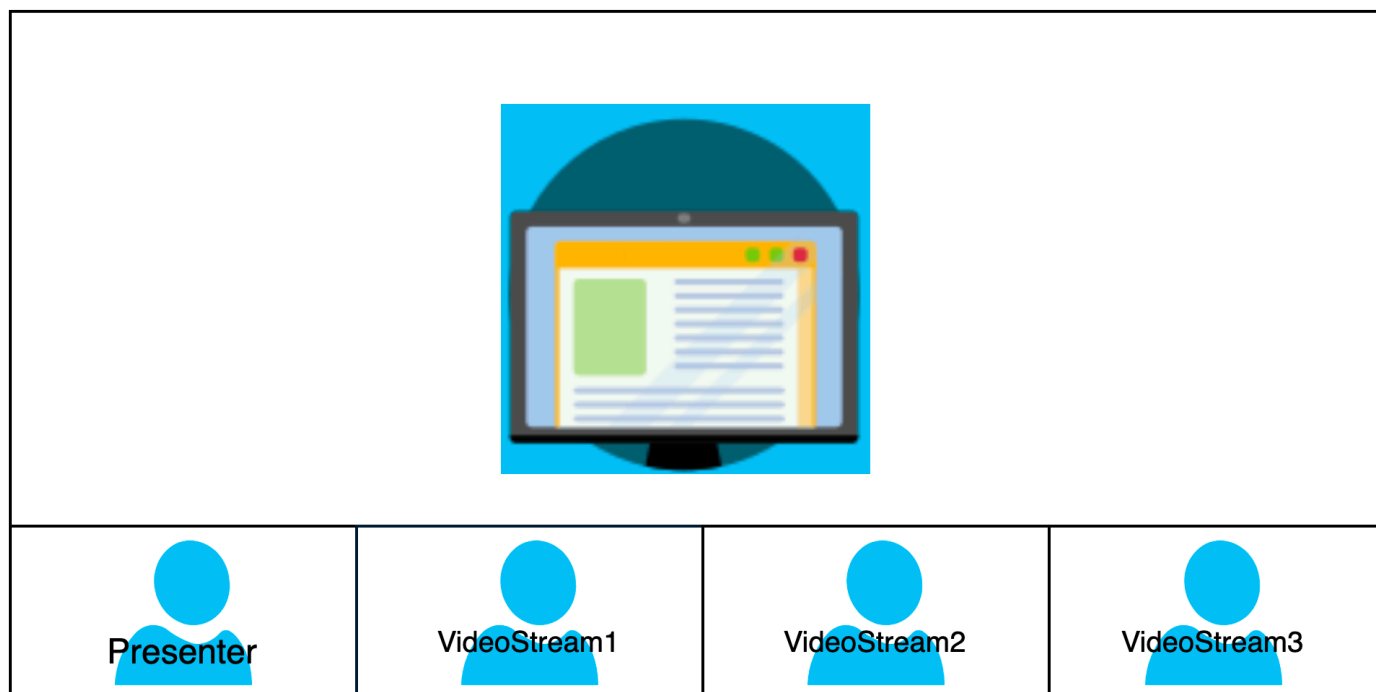
必填 — 否

默认值 — TopRight

HorizontalLayoutConfiguration

HorizontalLayoutConfiguration 水平显示内容共享和视频流。您可以使用 TilePosition 设置显示内容共享流上方或下方的磁贴。主讲人始终显示在左侧。其他磁贴根据 JoinSequence 所指示的顺序显示。

下图显示内容共享流下方的磁贴。



以下示例说明如何以编程方式实现水平布局。在这种情况下，布局会根据 SpeakerSequence 对磁贴进行排序，并将其放在屏幕共享的下方。该布局最多允许四个磁贴并应用 16/9 的纵横比。

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

HorizontalLayoutConfiguration

描述 — 水平布局的配置设置

类型- HorizontalLayoutConfiguration 对象

必填 — 否

HorizontalLayoutConfiguration.TilePosition

描述 — 将磁贴放在内容共享的上方或下方。

类型 — 字符串

有效值 — Bottom|Top

必填 — 否

默认值 — 底部

HorizontalLayoutConfiguration.TileOrder

描述 — 根据用户何时加入或何时讲话对磁贴进行排序

类型 — 字符串

有效值 — JoinSequence|SpeakerSequence

必填 — 否

默认- JoinSequence

HorizontalLayoutConfiguration.TileCount

描述 — 指定屏幕共享期间保持可见的磁贴数量

类型 — 整数

有效值 — 1—10

必填 — 否

默认值 — 4

HorizontalLayoutConfiguration.TileAspectRatio

描述 — 指定磁贴的纵横比

类型 — 整数

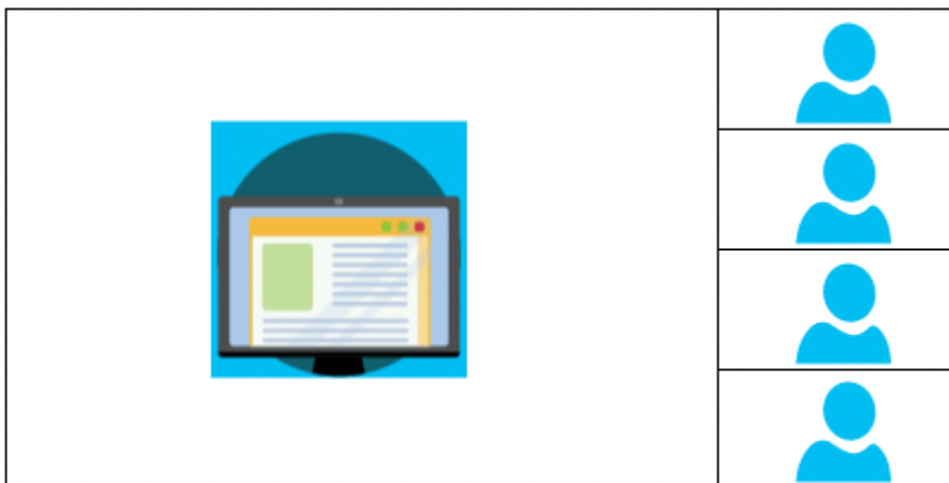
有效值 — n/n

必填 — 否

默认值 — 16/9，该值适用于所有磁贴

VerticalLayoutConfiguration

`VerticalLayoutConfiguration` 在右侧显示内容共享和四个最新的视频。主讲人始终显示在顶部。其他与会者按以下顺序出现。 `TileOrder`



以下示例说明如何以编程方式实现垂直布局。在这种情况下，布局会按图块排序 JoinSequence 并将其放置在屏幕共享的右侧。该布局最多允许四个磁贴并应用 16/9 的纵横比。

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

VerticalLayoutConfiguration

描述 — 垂直布局的配置设置

类型- VerticalLayoutConfiguration 对象

必填 — 否

VerticalLayoutConfiguration.TilePosition

描述 — 将磁贴放在内容共享的右侧或左侧。

类型 — 字符串

有效值 — Bottom|Top

必填 — 否

默认值 — 底部

VerticalLayoutConfiguration.TileOrder

描述 — 根据用户何时加入或何时讲话对磁贴进行排序

类型 — 字符串

有效值 — JoinSequence|SpeakerSequence

必填 — 否

默认- JoinSequence

VerticalLayoutConfiguration.TileCount

描述 — 指定磁贴的数量

类型 — 整数

有效值 — 1—10

必填 — 否

默认值 — 4

VerticalLayoutConfiguration.TileAspectRatio

描述 — 指定磁贴的纵横比

类型 — 整数

有效值 — n/n

必填 — 否

默认值 — 9/16，该值适用于所有磁贴

创建媒体流管道

媒体流管道捕获会议中所有与会者的个人音频，以及媒体连接管道生成的混合音频。所有媒体流管道会将其数据保存至 [Amazon Kinesis Video Streams](#) (KVS)。

您可以通过调用 [CreateMediaPipelineKinesisVideoStreamPool](#) API 来创建视频流。您可以为每次 Amazon Chime SDK 会议创建一个媒体流管道。

以下部分介绍如何创建媒体流管道。按照列出的顺序操作。

主题

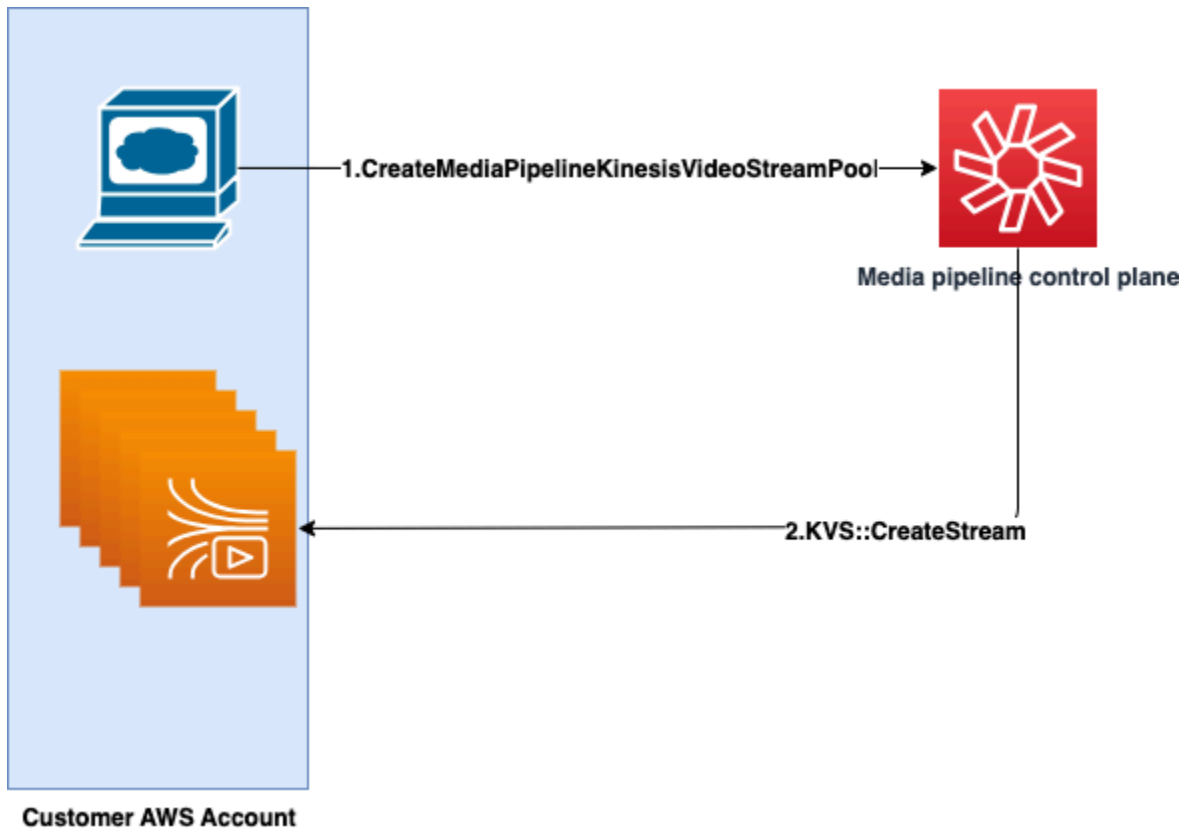
- [创建 Kinesis Video Streams 池](#)
- [Kinesis Video Streams 池的示例代码](#)
- [创建媒体流管道](#)
- [媒体流管道的示例代码](#)

- [使用事件总线通知](#)
- [使用媒体流管道数据](#)

创建 Kinesis Video Streams 池

媒体流管道的 Kinesis Video Streams 池必须与 Amazon Chime SDK 会议属于同一个 AWS 账户。您可以通过调用 API 来创建 Kinesis Video Streams 池 [CreateMediaPipelineKinesisVideoStreamPool](#)。

下图显示 Kinesis Video Streams 池的媒体管道架构。图像中的数字对应于以下带编号的文本：



在图中：

1. 您调用 [CreateMediaPipelineKinesisVideoStreamPool](#) API。
2. 媒体管道控制面板可代表您在您的账户中创建和管理 Kinesis Video Streams (KVS) 和池。

KVS 池操作，即在池中创建、更新和删除流的过程，是异步的。因此，事件总线通知使用 Chime Media Pipeline Kinesis Video Pool State Change 详细信息类型传达池中流的状态。

您可以创建一次池，然后在不同会议中重复使用。您也可以根据需要创建不同的池，并在不需要时将其删除。

根据您的并发调用突增量，池会自动纵向扩展。您可以删除任何不需要的池。

Note

删除池时，必须等待池完全删除，才能删除池中的 KVS 流。事件总线通知将指示池何时完全删除。这种情况发生在使用池进行所有会议结束之后。您也可以调用 [GetMediaPipelineKinesisVideoStreamPool](#) API 来查看 PoolId 给定 KVS 池的。当你调用 Kinesis Video [DeleteStream](#) Streams API 时，你可以使用该命名字符串在池中搜索和删除直播。您也可以调用 [GetMediaPipelineKinesisVideoStreamPool](#) API 来查看 PoolId 给定 KVS 池的。以下部分中的相关示例将说明如何操作。

Kinesis Video Streams 池的示例代码

以下示例显示如何创建、更新、获取、列出和删除 Kinesis Video Streams (KVS) 池。展开每个部分以了解更多信息。

导入和常用变量

```
...
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'])

```

输出：

```

{
  "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'])

```

输出：

```

{
  "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'])

```

输出：

```

{
  "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)

```

输出：

```

[
  {
    "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
)

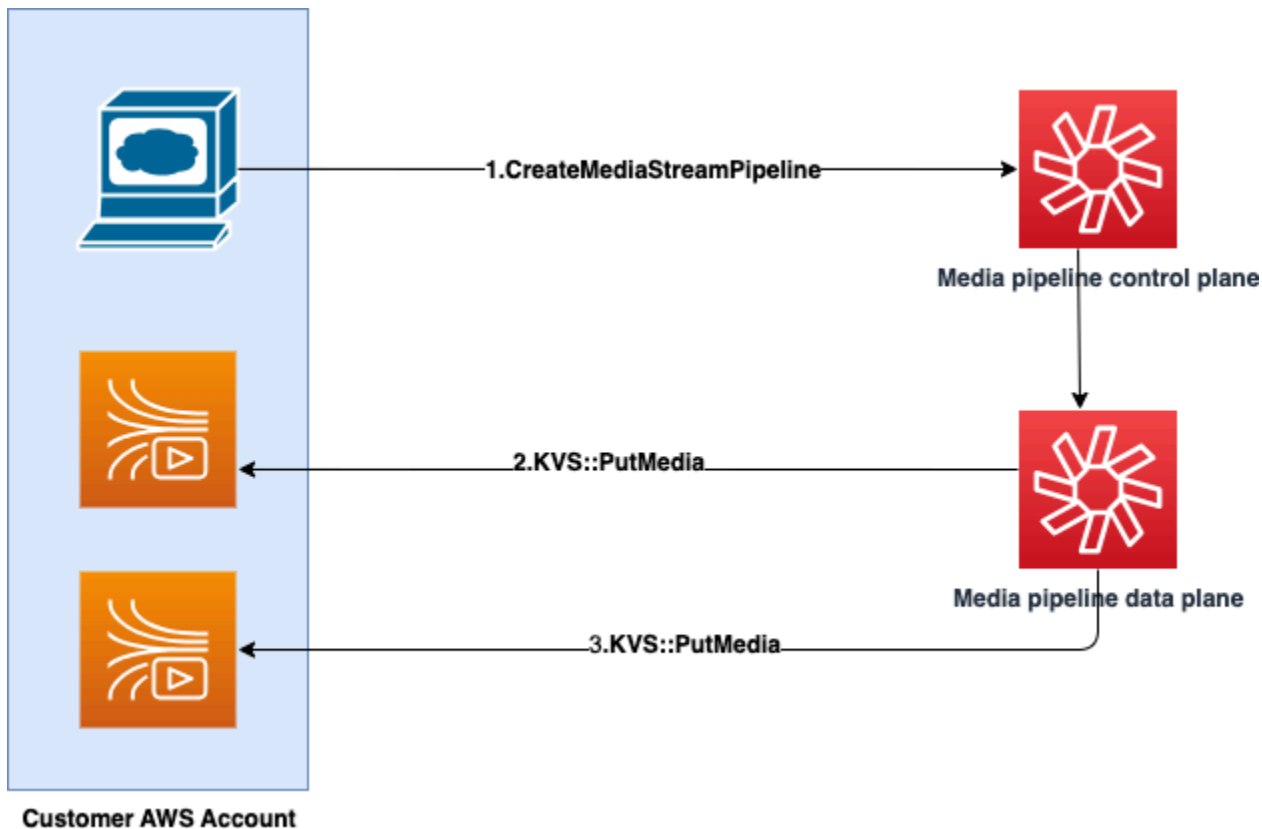
```

输出：成功的 `delete_media_pipeline_kinesis_video_stream_pool` 请求没有正文。

创建媒体流管道

Chime 媒体流管道必须与 Amazon Chime SDK 会议属于同一个 AWS 账户。您可以通过调用 [CreateMediaStreamPipeline](#) API 并指定来源和接收器来创建 Amazon Chime SDK 媒体流管道。

下图显示 Amazon Chime SDK 媒体流管道的架构。图表中的数字对应于以下带编号的文本。



在图中：

- 您调用 `CreateMediaStreamPipeline` API。在请求中，指定流的来源和接收器。是否想要捕获单个音频和/或混合音频。请求中包含 KVS 池的 ARN。
 - 来源数组由 `SourceType` 和 `SourceArn` 组成。您必须使用 `ChimeSdkMeeting` `SourceType`。`SourceArn` 是 `ChimeSdkMeeting` 的 ARN。
 - 接收器数组由 `SinkType`、`SinkArn`、`ReservedStreamCapacity` 和 `MediaStreamType` 组成。我们仅支持 `KinesisVideoStreamPoolSinkType`。`SinkArn` 是 `KinesisVideoStreamPool` 的 ARN。`MediaStreamType` 控制流式传输到接收器的媒体类型，`MixedAudio` 或 `IndividualAudio`。`ReservedStreamCapacity` 设置为 `KinesisVideoStreamPool` 的 `MediaStreamType` 分配的流数量。
 - 如果要同时流式传输 `IndividualAudio` 和 `MixedAudio`，则在 `Sinks` 数组中创建两个接收器对象，一个用于 `IndividualAudio`，另一个用于 `MixedAudio`。`SinkArn` (`KinesisVideoStreamPool` 的 ARN) 可能因每个接收器的不同而有所不同。
 - 要仅流式传输单个音频或混合音频，创建一个包含所需 `MediaStreamType` 的接收器对象。
 - 注意以下几点：

- 使用KinesisVideoStreamPool作为调用 [CreateMediaStreamPipelineAPI](#) 时SinkType，SinkARN必须属于CreateMediaStreamPipeline正在调用的控制平面区域。

例如，如果您在 us-east-1 中创建媒体流管道，则必须使用 us-east-1 中的 KinesisVideoStreamPool。

- ReservedStreamCapacity 应在您指定 MixedAudio MediaStreamType 时为 **1**，并且在指定 IndividualAudio MediaStreamType 时介于 **1-10** 之间。
2. 媒体管道数据平面调用 KVS [PutMediaAPI](#) 将单个音频存储在属于您指定的 KVS 池的 KVS 流中。
 3. 媒体管道数据面板调用 KVS PutMedia API 将混合音频存储在属于您指定 KVS 池的流中。

Note

调用 [CreateMediaStreamPipelineAPI](#) 后，生成器可以使用[媒体管道事件](#)或调用 [GetMediaPipelineAPI](#) 来确定管道状态是否为InProgress。

管道状态达到 InProgress 后，媒体（IndividualAudio 和 MixedAudio 的任意组合）就会流式传输到 KVS。

对于 IndividualAudio 流类型，与会者 ID 和从 KinesisVideoStreamPool 中分配的 KVS 流之间存在 1:1 映射。此映射适用于媒体管道的整个生命周期。

要知道哪个 KVS 直播映射到出席者 ID 或为其分配了哪个 KVS 直播 MixedAudio，请使用以下方法之一：

- 使用[事件总线通知](#)。每条通知提供诸如与会者 ID 以及流式传输至与会者音频的 KVS ARN 之类的信息。当 IndividualAudio 或 MixedAudio 流会话开始时，我们会发送一个 chime:MediaPipelineKinesisVideoStreamStart 事件。当与会者离开调用（针对 IndividualAudio）或会议结束时，流会话就会结束。
- 使用 Kinesis Video Streams 随每个片段发送的持久性元数据。元数据包含与事件总线发送的内容相似的信息。构建者需要KinesisVideoStreamPool通过此解决方案在 [ListStreams](#)Kinesis Video Streams API 中将池名称指定为前缀，从而解析所有直播流。

删除会议或为该媒体流管道调用 [DeleteMediaPipelineAPI](#) 时，媒体流管道就会终止。此外，还会发送[事件总线通知](#)以指示媒体管道终止。

媒体流管道的示例代码

以下示例显示如何为混合音频和/或单个音频创建媒体流管道。展开每个部分以了解更多信息。

CreateMediaStreamPipeline 用于混合音频

```
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'  
        },  
    ],  
)
```

响应：

```
{  
    '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 用于个人音频

```

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'
    },
  ],
)

```

```
]
)
```

响应：

```
{
  '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 用于混合和单独音频

```
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'
    },
  ],
]
)

```

响应：

```

{
  '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'
    },
]
}
}

```

使用事件总线通知

除了 [使用媒体管道事件](#) 之外，媒体流管道还会在开始和停止流式传输至 KVS 以及视频池状态发生变化时发送事件总线通知。

主题

- [媒体流管道事件](#)
- [媒体管道 Kinesis 视频池事件](#)

媒体流管道事件

媒体流管道发送以下事件。展开每个部分以了解更多信息。

Amazon Chime 媒体流管道 Kinesis 视频流开始

当媒体流管道开始接收来自会议的音频并将音频流式传输到 KVS 时，Amazon Chime SDK 媒体管道会发送此事件。空白的 `AttendeeId` 和 `ExternalUserId` 字段表示媒体管道将混合音频发送到 KVS 流。

```

{
  "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 媒体流管道 Kinesis 视频流结束

当流式传输至 KVS 结束时，媒体管道会将此事件发送到事件总线。

```

{
  "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"
  }
}

```

媒体管道 Kinesis 视频池事件

当池的状态发生变化时，媒体管道会将以下事件发送到事件总线。展开每个部分以了解更多信息。

Amazon Chime 媒体管道 Kinesis 视频池已激活

媒体管道发送此事件是在 [CreateMediaPipelineKinesisVideoStreamPoolAPI](#) 创建池之后发送的。

```

{
  "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 媒体管道 Kinesis 视频池已更新

在 [UpdateMediaPipelineKinesisVideoStreamPoolAPI](#) 更新池后，媒体管道会发送此事件。

```

{
  "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 媒体管道 Kinesis 视频池已删除

[DeleteMediaPipelineKinesisVideoStreamPool](#) 删除池时，媒体管道会将此事件发送到事件桥。

有关删除池的更多信息，请参阅本节中的 [创建 Kinesis Video Streams 池](#)。

```

{
  "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 媒体管道 Kinesis 视频池临时故障

当视频池暂时出现故障时，媒体管道会向事件总线发送以下事件。

```

{
  "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 媒体管道 Kinesis 视频池永久故障

当视频池出现永久故障时，媒体管道会向事件总线发送以下事件。

```
{
  "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"
  }
}
```

使用媒体流管道数据

您可以使用通知中的元数据获取 KVS ARN、片段编号和片段时间戳。这些信息可以帮助您处理 KVS 流中的音频数据。

此外，您还可以将 KVS ARN 与 KVS API 配合使用，以从流中读取数据。根据用例，您可以调用 [GetMedia](#) 和 [GetMediaForFragmentList](#) API。通常，在 [GetMediaForFragmentList](#) 调用之前会调

用 [ListFragments](#) API。有关更多信息，请参阅 Amazon Kinesis Video Streams 常见问题解答中的 [从流中读取数据](#)。

根据用例，构建者可以使用 Kinesis Video Streams 解析器库，该库反过来又使用 KVS API。 [GetMedia](#)

媒体流管道会将以下会议和与会者元数据添加到每个片段。

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

媒体数据以 MKV 格式存储。所有 MKV 音频数据均采用 AAC 编码。有关更多信息，请参阅《Amazon Kinesis Video Streams 开发人员指南》中的 [Kinesis Video Streams 数据模型](#)。

为媒体管道创建服务相关角色

以下各节中的信息说明如何创建服务相关角色以授予媒体管道访问您的 Amazon Chime SDK 会议的权限。

主题

- [设置角色权限](#)
- [创建 服务相关角色](#)
- [编辑 服务相关角色](#)
- [删除 服务相关角色](#)
- [服务相关角色的受支持区域](#)

设置角色权限

媒体管道使用名为 `AWSServiceRoleForAmazonChimeSDKMediaPipelines` 的服务相关角色。该角色允许捕获渠道代表您访问 Amazon Chime SDK 会议并向亚马逊 CloudWatch 发布指标。此角色信任 `mediapipelines.chime.amazonaws.com` 服务。

角色权限策略允许 Amazon Chime 软件开发工具包对所有 AWS 资源完成以下操作：

- 操作：`all AWS resources` 上的 `cloudwatch:PutMetricData`

- 操作 : `chime:CreateAttendee` 上的 all AWS resources
- 操作 : `chime>DeleteAttendee` 上的 all AWS resources
- 操作 : `chime:GetMeeting` 上的 all AWS resources
- 操作 : `kinesisvideo:CreateStream` 上的
`arn:aws:kinesisvideo:*:111122223333:stream/ChimeMediaPipelines-*`
- 操作 : `kinesisvideo:PutMedia` 上的 `arn:aws:kinesisvideo:*:111122223333:stream/ChimeMediaPipelines-*`
- 操作 : `kinesisvideo:UpdateDataRetention` 上的
`arn:aws:kinesisvideo:*:111122223333:stream/ChimeMediaPipelines-*`
- 操作 : `kinesisvideo:DescribeStream` 上的
`arn:aws:kinesisvideo:*:111122223333:stream/ChimeMediaPipelines-*`
- 操作 : `kinesisvideo:GetDataEndpoint` 上的
`arn:aws:kinesisvideo:*:111122223333:stream/ChimeMediaPipelines-*`
- 操作 : `kinesisvideo:ListStreams` 上的
`arn:aws:kinesisvideo:*:111122223333:stream/*`

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关权限的更多信息，请参阅 IAM 用户指南中的[服务相关角色权限](#)。

创建 服务相关角色

使用 IAM 控制台为 Amazon Chime SDK 媒体管道创建服务相关角色。您必须具有 IAM 管理权限才能完成这些步骤。如果没有，请联系系统管理员。

创建角色

1. 登录 AWS 管理控制台，然后打开 IAM 控制台，[网址为 https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/)。
2. 在 IAM 控制台的导航窗格中，选择角色，然后选择创建角色。
3. 选择 AWS 服务角色类型，然后选择 Chime SDK 媒体管道。

IAM 策略随即出现。

4. 勾选策略旁边的复选框，然后选择下一步: 标签。
5. 选择 下一步: 审核。
6. 根据需要编辑描述，然后选择创建角色。

您也可以使用 AWS CLI 或 AWS API 创建名为 `mediapipelines.chime.amazonaws.com` 的服务相关角色。在 AWS CLI 中，运行以下命令：

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

有关创建角色的更多信息，请参阅 IAM 用户指南中的[创建服务相关角色](#)。如果您已删除此角色，则可以重复此过程再次创建。

编辑 服务相关角色

您无法编辑 `AWSServiceRoleForAmazonChimeSDKMediaPipelines` 服务相关角色。创建角色后，您无法更改该角色的名称，因为可能有不同的实体引用该角色。但是，您可以使用 IAM 编辑角色描述。有关更多信息，请参阅《IAM 用户指南》中的[编辑服务相关角色](#)。

删除 服务相关角色

如果不需要服务相关角色，我们建议您将其删除。为此，您需要先删除使用该角色的媒体管道。您可以使用 AWS CLI 或 [DeleteMediaCapturePipeline](#) API 来删除管道。

使用 CLI 删除管道

在 AWS CLI 中使用此命令删除您账户中的媒体管道。

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

使用 API 删除管道

使用 [DeleteMediaCapturePipeline](#) API 删除您账户中的媒体管道。

删除角色

删除管道后，您可以使用 IAM 控制台、AWS CLI 或 AWS API 删除角色。有关删除角色的更多信息，请参阅 IAM 用户指南中的[删除服务相关角色](#)。

服务相关角色的受支持区域

Amazon Chime SDK 支持在提供服务的所有 AWS 地区使用服务相关角色。有关更多信息，请参阅 Amazon Web Services 一般参考中的 [Amazon Chime SDK 终端节点和限额](#)。

使用媒体管道事件

每种类型的媒体管道会发送生命周期事件，用于触发通知和启动下游工作流程。使用会议管道事件的部分示例包括：

- 媒体管道完成后处理捕获的媒体。
- 如果媒体管道出现暂时故障，则通知会议参与者。
- 如果媒体管道出现永久故障，则停止会议。

您可以向亚马逊 EventBridge、亚马逊简单通知服务 (SNS) 和亚马逊简单队列服务 (SQS) 发送事件。有关更多信息，请参阅 Amazon EventBridge 用户指南中的[来自 AWS 服务的事件](#)。

已创建的 Amazon Chime SDK 媒体管道

创建媒体管道时，Amazon Chime SDK 会发送此事件。

示例：事件数据

以下是此事件的示例数据。

```
{
  "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 媒体管道已删除

当媒体管道被删除后，Amazon Chime SDK 会发送此事件。

示例：事件数据

以下是此事件的示例数据。

```
{
  "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 媒体管道出现暂时故障

当媒体管道出现暂时故障时，Amazon Chime SDK 会发送此事件。

示例：事件数据

以下是此事件的示例数据。

```
{
  "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 媒体管道出现暂时故障后恢复

当媒体管道出现暂时故障后恢复时，Amazon Chime SDK 会发送此事件。

示例：事件数据

以下是此事件的示例数据。

```
{
  "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:MediaPipelineResumed",
    "timestamp": 1627503710485?,
    "meetingId": "1e6bf4f5-f4b5-4917-b8c9-bda45c340706",
    "externalMeetingId": "Meeting_Id",
    "mediaPipelineId": "e40ee45e-2ed1-408e-9156-f52b8208a491",
    "mediaRegion": "ap-southeast-1"
  }
}
```

Amazon Chime SDK 媒体管道出现永久故障

当媒体管道出现永久故障时，Amazon Chime SDK 会发送此事件。

示例：事件数据

以下是此事件的示例数据。

```
{
  "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"
  }
}
```

设置 Amazon S3 存储桶权限

如果您尚未创建 Amazon S3 存储桶，请确保在您主持会议的账户和区域中创建您的存储桶。另外，请确保授予服务足够的权限。有关 Amazon S3 存储桶名称的更多信息，请参阅 [创建 Amazon S3 存储桶](#)。

向 CloudTrail 发送媒体管道事件

在创建您的 AWS 账户时，AWS 启用 CloudTrail。当用户在媒体管道 SDK 中调用受支持的 API 时，CloudTrail 会在事件历史记录中记录该 API 的活动以及其他 AWS 事件。您可以在 AWS 账户中查看、搜索和下载媒体管道事件。有关更多信息，请参阅《CloudTrail 用户指南》中的 [使用 CloudTrail 事件历史记录查看事件](#)。

要持续记录媒体管道事件，可创建跟踪。通过跟踪记录，CloudTrail 可将日志文件传送至 Simple Storage Service (Amazon S3) 存储桶。以下示例显示媒体管道跟踪。数据包括调用 API 的用户、用于调用 API 的 IAM 角色和时间戳。有关使用 CloudTrail 的更多信息，请参阅《Amazon Chime SDK 管理员指南》中的 [日志和监控](#)。

```
{
  "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": "ABCDEFGHJKLMNOPQRSTUVWXYZ",
            "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"
    }
},
]
}
```

解析转录

使用以下命令解析转录消息中的转录内容。该命令解析 transcript-message.txt 文件中的完整句子。

```
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"])
```

停止管道的最佳实践

作为停止媒体管道的最佳实践，请调用 [DeleteMediaPipeline](#) API。API 允许您删除媒体捕获和媒体实时连接器管道。您也可以调用 [DeleteMediaCapturePipeline](#) API 来删除媒体捕获管道。会议结束后，所有媒体管道都会停止。

使用 Amazon Chime SDK 实时转录

您可以使用 Amazon Chime SDK 实时转录生成用户归属的实时会议记录。Amazon Chime SDK 实时转录与 Amazon Transcribe 和 Amazon Transcribe Medical 服务集成，可在会议进行期间生成 Amazon Chime SDK 会议记录。

Amazon Chime SDK 实时转录分别处理每个用户的音频，以提高多人场景中的准确性。Amazon Chime SDK 使用其主动说话者算法来选择前两个主动说话者，然后通过单个流将其音频以不同的频道发送到 Amazon Transcribe。会议参与者通过 Amazon Chime SDK 数据消息接收用户归属的转录。您可以通过多种方式使用转录，例如显示字幕、创建会议记录或使用转录进行内容分析。

在会议转录期间，实时转录使用一个流到 Amazon Transcribe。适用标准 Amazon Transcribe 和 Amazon Transcribe Medical 费用。有关更多信息，请参阅 [Amazon Transcribe 定价](#)。有关使用量或账单的问题，请联系您的 AWS 客户经理。

⚠ Important

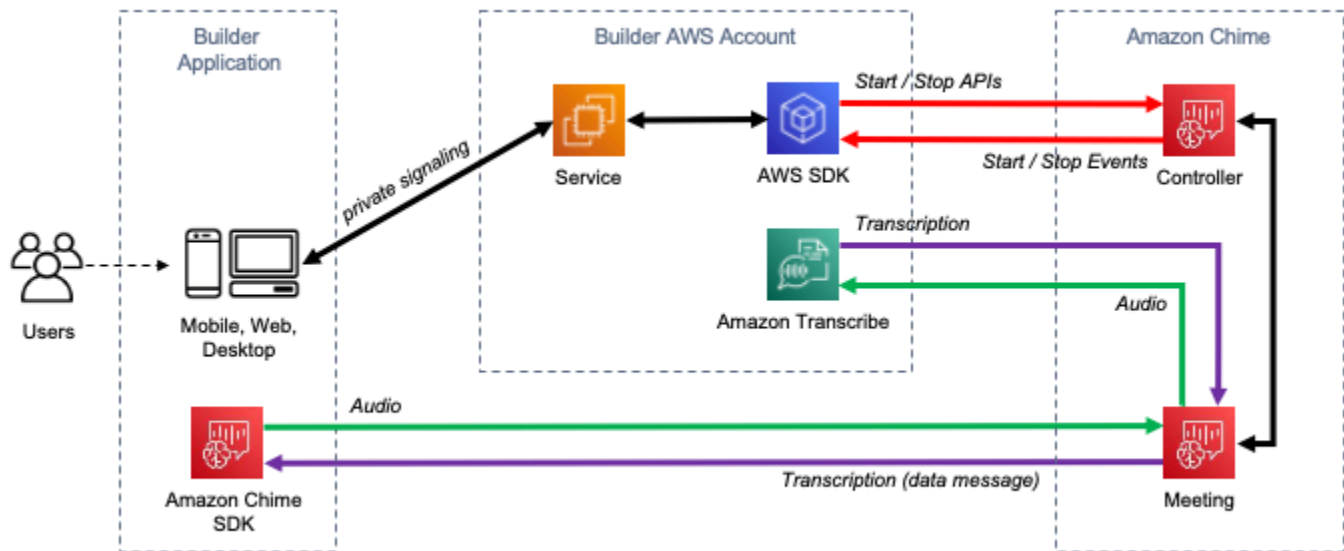
默认情况下，Amazon Transcribe 可能会存储该服务处理的音频内容，并使用这些内容来开发和改进 AWS AI/ML 服务，有关详细信息，请参阅 [AWS 服务条款](#) 中的第 50 条。使用 Amazon Transcribe 时可能需要遵守联邦和州有关录制或拦截电子通信的法律或法规。您和您的最终用户有责任遵守与录音有关的所有适用法律，包括适当通知录制会话或通信中的所有参与者正在录制会话或通信，并获得他们的同意。通过使用 AWS Organizations 配置 AI 服务选择退出策略，可以选择退出 AWS，不使用音频内容来开发和改进 AWS AI/ML 服务。

主题

- [系统架构](#)
- [账单和使用情况](#)
- [配置您的账户](#)
- [选择转录选项](#)
- [开始和停止转录](#)
- [转录参数](#)
- [转录事件](#)
- [转录消息](#)
- [传输示例](#)

系统架构

Amazon Chime SDK 通过与您的 Amazon Transcribe 或 Amazon Transcribe Medical 账户的服务端集成，创建实时会议转录，无需音频离开 AWS 网络。为了提高准确性，用户的音频会被单独处理，然后混合到会议中。Amazon Chime SDK 使用其主动说话者算法来选择前两个主动说话者，然后通过单个流将其的音频以不同的频道发送到 Amazon Transcribe 或 Amazon Transcribe Medical。为了减少延迟，用户归属的转录将通过数据消息直接发送给每位会议参与者。使用媒体管道捕获会议音频时，还会捕获会议的转录信息。



A diagram showing the data flow of meeting transcription.

账单和使用情况

在会议转录期间，实时转录使用一个流到 Amazon Transcribe 或 Amazon Transcribe Medical。适用标准 Amazon Transcribe 和 Amazon Transcribe Medical 费用。有关更多信息，请参阅 [Amazon Transcribe 定价](#)。有关使用量或账单的问题，请联系您的 AWS 客户经理。

配置您的账户

在使用 Amazon Chime SDK 实时转录前，您必须授予 Amazon Chime SDK 在您的 AWS 账户中调用 Amazon Transcribe 和 Amazon Transcribe Medical 的权限。为此，您可以将 Chime 转录服务相关角色添加到您的账户。有关为实时转录创建服务相关角色的信息，请参阅 Amazon Chime SDK 管理指南中的 [使用带实时转录的角色](#)。有关 IAM 服务相关角色的更多信息，请参阅 IAM 用户指南中的 [服务相关角色](#)。

选择转录选项

当您使用 Amazon Chime SDK 实时转录时，您使用您的 AWS 账户中的 [Amazon Transcribe](#) 或 [Amazon Transcribe Medical](#)。您可以访问 [Amazon Transcribe 支持的所有流传输语言](#)，以及 [自定义词汇表](#) 和 [词汇筛选器](#) 等功能。使用 Amazon Transcribe Medical 时，您可以选择医学专业、对话类型，也可以选择提供任何自定义词汇。适用标准 Amazon Transcribe 和 Amazon Transcribe Medical 费用。

选择转录选项的过程要遵循以下步骤。

第 1 步：选择转录服务

您需要决定使用哪种转录服务，[Amazon Transcribe](#) 或者 [Amazon Transcribe Medical](#)。

如果您的用例需要医疗语音转文本功能，则可能需要使用 Amazon Transcribe Medical。对于所有其他使用案例，您可能需要使用 Amazon Transcribe。

您可以在调用 StartMeetingTranscription API 时指定要使用的转录服务：

- 要使用 Amazon Transcribe，请使用 EngineTranscribeSettings 指定 TranscriptionConfiguration。
- 要使用 Amazon Transcribe Medical，请使用 EngineTranscribeMedicalSettings 指定 TranscriptionConfiguration。

第 2 步：选择转录区域

您需要为转录服务选择 AWS 区域。有关 Amazon Transcribe 和 Amazon Transcribe Medical 的可用 AWS 区域的信息，请参阅[AWS 区域服务表](#)。

通常，会议的媒体区域和转录区域之间的最低延迟可提供最佳的用户体验。为了实现最低延迟，请尽可能将相同区域用于媒体和转录。但是，在选择区域时，您可能需要考虑其他因素，例如监管要求或您配置 Amazon Transcribe 或 Amazon Transcribe Medical 的区域。

Amazon Transcribe 和 Amazon Transcribe Medical 功能，例如自定义词汇表或词汇筛选器，因地区而异。如果您配置这些功能中的任何一个，则应在打算使用实时转录的所有 AWS 区域中以相同的方式进行配置。或者，您可以为所有会议使用相同的 Amazon Transcribe 区域。

您可以指定转录服务使用的区域。为此，您可以在调用 StartMeetingTranscription API 时将区域名称添加到转录引擎设置的 Region 字段中。如果您未指定区域，Amazon Chime SDK 会尝试在会议的媒体区域使用转录服务。要让 Amazon Chime SDK 为您选择转录服务的区域，请在 Region 字段中指定 auto。这样，Amazon Chime 会根据会议的媒体区域选择转录服务区域，如下表所示。有关 StartMeetingTranscription API 的更多信息，请参阅本指南中的[开始和停止转录](#)。

Note

Amazon Chime SDK 选择的转录区域可能随着 AWS，Amazon Chime SDK、Amazon Transcribe 和 Amazon Transcribe Medical 提供更多可用区域而发生变化。

Amazon Transcribe 的自动区域选择

Amazon Chime SDK 媒体区域	区域代码	转录区域
美国东部 (俄亥俄州)	us-east-2	us-east-2
美国东部 (弗吉尼亚州北部)	us-east-1	us-east-1
美国西部 (北加利福尼亚)	us-west-1	us-west-2
美国西部 (俄勒冈州)	us-west-2	us-west-2
非洲 (开普敦) *	af-south-1	eu-west-2
亚太地区 (孟买)	ap-south-1	eu-west-2
亚太地区 (首尔)	ap-northeast-2	ap-northeast-2
亚太地区 (新加坡)	ap-southeast-1	ap-northeast-1
亚太地区 (悉尼)	ap-southeast-2	ap-southeast-2
亚太地区 (东京)	ap-northeast-1	ap-northeast-1
加拿大 (中部)	ca-central-1	ca-central-1
欧洲地区 (法兰克福)	eu-central-1	eu-central-1
欧洲地区 (爱尔兰)	eu-west-1	eu-west-1
欧洲地区 (伦敦)	eu-west-2	eu-west-2
欧洲地区 (米兰) *	eu-south-1	eu-central-1
欧洲地区 (巴黎)	eu-west-3	eu-central-1
欧洲地区 (斯德哥尔摩)	eu-north-1	eu-central-1
南美洲 (圣保罗)	sa-east-1	sa-east-1
GovCloud (美国东部)	us-gov-east-1	us-gov-west-1
GovCloud (美国西部)	us-gov-west-1	us-gov-west-1

Amazon Transcribe Medical 的自动区域选择

Amazon Chime SDK 媒体区域	区域代码	转录区域
美国东部 (俄亥俄州)	us-east-2	us-east-2
美国东部 (弗吉尼亚州北部)	us-east-1	us-east-1
美国西部 (北加利福尼亚)	us-west-1	us-west-2
美国西部 (俄勒冈州)	us-west-2	us-west-2
非洲 (开普敦) *	af-south-1	eu-west-1
亚太地区 (孟买)	ap-south-1	eu-west-1
亚太地区 (首尔)	ap-northeast-2	us-west-2
亚太地区 (新加坡)	ap-southeast-1	ap-southeast-2
亚太地区 (悉尼)	ap-southeast-2	ap-southeast-2
亚太地区 (东京)	ap-northeast-1	us-west-2
加拿大 (中部)	ca-central-1	ca-central-1
欧洲地区 (法兰克福)	eu-central-1	eu-west-1
欧洲地区 (爱尔兰)	eu-west-1	eu-west-1
欧洲地区 (伦敦)	eu-west-2	us-east-1
欧洲地区 (米兰) *	eu-south-1	eu-west-1
欧洲地区 (巴黎)	eu-west-3	eu-west-1
欧洲地区 (斯德哥尔摩)	eu-north-1	eu-west-1
南美洲 (圣保罗)	sa-east-1	us-east-1

Note

要在标有星号 (*) 的区域中使用实时转录，必须先在其您的 AWS 账户中启用该区域。有关更多信息，请参阅 AWS 一般参考中的[启用区域](#)。

有关区域和每个服务终端节点的更多信息，请参阅：

- [Amazon Chime SDK 媒体区域](#)
- [Amazon Transcribe 终端节点和限额](#)
- [Amazon Transcribe Medical 终端节点和限额](#)

第 3 步：查看服务限额

每个带有实时转录功能的 Amazon Chime SDK 会议只需要一个 HTTP/2 流到 Amazon Transcribe 或 Amazon Transcribe Medical。这两项服务都有区域服务限额，包括并发 HTTP/2 流的数量和每秒的始流交易量。有关限额的更多信息，请参阅《Amazon Transcribe 开发人员指南》中的[准则和限额](#)。有关增加限额的信息，请参阅 AWS 控制台中的服务限额。

开始和停止转录

您可以使用 Amazon Chime SDK [StartMeetingTranscription](#) API 通过将 `TranscriptionConfiguration` 应用至会议来启动会议转录。Amazon Chime SDK 控制器将配置异步转发给会议。启动会议转录的成功或失败通过 Amazon Simple Notification Service (Amazon SNS) 和 Amazon EventBridge 发送消息进行通知的。

开始转录

此示例显示如何使用 Amazon Transcribe 开始实时转录。

```
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"
  }
}
}

```

此示例显示如何使用 Amazon Transcribe Medical 开始实时转录。

```

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 — 开始会议转录。

MeetingId — 会议的 ID，由 [CreateMeeting API](#) 返回。

TranscriptionConfiguration — 封装实时转录的参数。您必须精确指定一个配置，EngineTranscribeSettings 或 EngineTranscribeMedicalSettings。

EngineTranscribeSettings — [指定 Amazon Transcribe 的使用并将其设置传递给 Amazon Transcribe。](#)

LanguageCode — 必填。

VocabularyFilterMethod — 可选。

VocabularyFilterName — 可选。

VocabularyName — 可选。

Region — 可选。

EnablePartialResultsStabilization — 可选。

PartialResultsStability — 可选。

ContentIdentificationType — 可选。

ContentRedactionType — 可选。

PiiEntityTypes — 可选。

LanguageModelName — 可选。

EngineTranscribeMedicalSettings — 指定 Amazon Transcribe Medical 的使用并将其设置传递给 [Amazon Transcribe Medical](#)。

LanguageCode — 必填。

Speciality — 必填。

Type — 必填。

VocabularyName — 可选。

Region — 可选。

ContentIdentificationType — 可选。

响应

Amazon Transcribe 和 Amazon Transcribe Medical 会作出以下响应：

- 带空主体的 OK (200)，如果将 TranscriptionConfiguration 成功应用于会议。

错误消息

Amazon Transcribe 和 Amazon Transcribe Medical 会显示以下错误消息：

- BadRequestException (400)：输入参数与服务的限制不匹配。
- ForbiddenException (403)：永久禁止客户端提出请求。

- `NotFoundException (404)`: `meetingId` 不存在。
- `ResourceLimitExceededException (400)`: 请求超出了资源限制。例如，启用实时转录功能的会议太多。
- `ServiceFailureException (500)`: 服务遇到意外错误。
- `ServiceUnavailableException (503)`: 服务当前不可用。
- `ThrottledClientException (429)`: 客户端超出请求速率限制。
- `UnauthorizedClientException (401)`: 当前未授权客户端提出请求。

第二次呼叫 `StartMeetingTranscription` 会更新应用于会议的 `TranscriptionConfiguration`。

停止转录

您可以使用 [StopMeetingTranscription](#) API 删除用于给定 `meetingId` 和结束会议转录的 `TranscriptionConfiguration`。结束会议会自动停止转录。

此示例显示调用的 `StopMeetingTranscription` 请求语法。

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

响应

Amazon Transcribe 和 Amazon Transcribe Medical 会作出以下响应：

- 带空主体的 `OK (200)`，如果将 `TranscriptionConfiguration` 成功从会议中删除。

错误消息

Amazon Transcribe 和 Amazon Transcribe Medical 会显示以下错误消息：

- `BadRequestException (400)`: 输入参数与服务的限制不匹配。
- `ForbiddenException (403)`: 永久禁止客户端提出请求。
- `NotFoundException (404)`: `meetingId` 不存在。
- `ServiceFailureException (500)`: 服务遇到意外错误。
- `ServiceUnavailableException (503)`: 服务当前不可用。
- `ThrottledClientException (429)`: 客户端超出请求速率限制。

- `UnauthorizedClientException (401)`: 当前未授权客户端提出请求。

转录参数

Amazon Transcribe 和 Amazon Transcribe Medical API 在启动流传输转录时提供了许多参数，例如 [StartStreamTranscription](#) 和 [StartMedicalStreamTranscription](#)。您可以在 `StartMeetingTranscription` API 中使用这些参数，除非 Amazon Chime SDK 预先确定参数的值。例如，`MediaEncoding` 和 `MediaSampleRateHertz` 参数不可用，因为 Amazon Chime SDK 会自动进行设置。

Amazon Transcribe 和 Amazon Transcribe Medical 对参数进行验证，以允许您在新的参数值可用后立即使用。例如，如果 Amazon Transcribe Medical 启动对新语言的支持，则您只需要在 `LanguageCode` 参数中指定新的语言值。

转录事件

Amazon Chime SDK 发送生命周期事件，用于触发通知和启动下游工作流程。使用转录事件的部分示例包括：

- 衡量 Amazon Chime SDK 会议中实时转录的采纳情况
- 跟踪语言首选项

您可以将事件发送到 Amazon EventBridge、Amazon Simple Notification Service (SNS) 和 Amazon Simple Queue Service (SQS)。有关更多信息，请参阅 Amazon EventBridge 用户指南中的 [AWS 服务中的事件](#)。

Amazon Chime SDK 会议转录开始

当会议转录开始或 [TranscriptionConfiguration](#) 更新时，Amazon Chime SDK 会发送此事件。

示例：事件数据

以下是此事件的示例数据。

```
{
  "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 会议转录停止

会议转录停止时，Amazon Chime SDK 会发送此事件。

示例：事件数据

以下是此事件的示例数据。

```
{
  "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 会议转录中断

会议转录中断时，Amazon Chime SDK 会发送此事件。

示例：事件数据

以下是此事件的示例数据。

```
{
  "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 会议转录恢复

如果会议转录中断后恢复，Amazon Chime SDK 会发送此事件。

示例：事件数据

以下是此事件的示例数据。

```
{
  "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 会议转录失败

如果会议转录未能开始或中断后恢复失败，Amazon Chime SDK 会发送此事件。

示例：事件数据

以下是此事件的示例数据。

```

{
  "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": "{...}"
  }
}

```

```
}
```

转录消息

Amazon Chime SDK 服务通过在数据消息中发送 `TranscriptEvent` 对象，与与会者共享转录信息。`TranscriptEvent` 传递 `Transcript` 或 `TranscriptionStatus`。

`Transcript` 包括带有时间戳、用户归属单词和标点符号的结果。结果可能是“部分的”，在这种情况下，系统通常会在后续 `TranscriptEvent` 中对其进行更新。这使您可以快速查看转录，并在以后根据需要应用内联更新。

`TranscriptionStatus` 可能会传递其中一个 `TranscriptionStatusType` 事件，如以下部分示例中所列。

较新版本的 Amazon Chime SDK 包括用于常见处理 `TranscriptEvent` 的其他数据类型和辅助函数。

TranscriptEvent

此示例显示了一个典型转录事件。

```
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;  
}
```

数据准则

请记住这些准则。

1. `transcription.results` 可能有多个结果。
2. 如果为 `transcription.results[i].isPartial = true`，则可能会更新整个结果。可能会更新，但不能保证。更新具有相同的 `transcript.result[i].resultId`。如果您要避免低置信度转录，则可以完全跳过部分结果。如果您要低延迟结果，则可以显示部分结果，然后在更新时完全覆盖。
3. `transcription.results[i].alternatives` 始终至少包含一个条目。如果包含多个条目，则最确定的条目位于列表的第一位。在大多数情况下，您可以选择 `transcription.results[i].alternatives` 中的第一个条目而忽略其他条目。
4. `transcription.results[i].alternatives[j].items` 包括每个单词或标点符号的条目。
5. `transcription.results[i].alternatives[j].items[k]`。内容为所说的内容。
6. `transcription.results[i].alternatives[j].items[k].attendee` 为内容的用户归属（谁）。
7. `transcription.results[i].alternatives[j].items[k].startTimeMs` 为内容的“时间”。这样就可以按说出单词的顺序逐字呈现不同用户的用户属性转录。
8. 通常可以忽略 `transcription.results[i].alternatives[j].items[k].endTimeMs` 字段，但该字段旨在完整说明谁在何时说了什么内容。
9. 如果内容与过滤器中的单词匹配，则 `transcription.results[i].alternatives[j].items[k].vocabularyFilterMatch` 为真，否则为假。
10. `transcription.results[i].alternatives[j].items[k].confidence` 是 0 到 1 之间的值。它表示引擎对项目内容与所说单词正确匹配的置信度，其中 0 表示置信度最低，1 表示置信度最高。

- 11.transcription.results[i].alternatives[j].items[k].stable 表示当前单词是否会在后续部分结果更新中发生变化。仅当您在请求中将 EnablePartialResultsStabilization 设置为 true 以启用部分结果的稳定性功能时，此值才可为真。
- 12.transcription.results[i].alternatives[j].entities 包括内容识别或编修功能检测到的每个实体的条目。只有启用内容识别或编修后，才会填充该列表。实体可以是个人身份信息或个人健康信息等数据。在转录过程中，您可以使用实体来突出显示感兴趣的单词或对其采取行动。
- 13.transcription.results[i].alternatives[j].entities[k].category 是实体的类别。它等同于请求中提供的内容识别或编修类型，例如“PII”或“PHI”。
- 14.transcription.results[i].alternatives[j].entities[k].confidence 衡量特定内容真正成为实体的引擎有多强大。请注意，这与项目级别的置信度不同，后者衡量引擎对单词本身正确性的信心。
- 15.transcription.results[i].alternatives[j].entities[k].content 是构成实体的实际文本。这可以是多个项目，例如地址。
- 16.transcription.results[i].alternatives[j].entities[k].startTimeMs 捕获实体开始讲话的时间。
- 17.transcription.results[i].alternatives[j].entities[k].endTimeMs 捕获实体结束讲话的时间。
- 18.transcription.results[i].alternatives[j].entities[k].type 仅支持 Transcribe 引擎并提供实体的子类型。这些是诸如“地址”、“CREDIT_DEBIT_NUMBER”等值。

为 TranscriptEvents 注册事件处理程序

以下示例使用适用于 JavaScript 的 Amazon Chime SDK 客户端库。但是，该模式在所有 Amazon Chime SDK 中都是一致的。

RealtimeController 和 RealtimeControllerFacade 中的 TranscriptionController 包括用于添加处理 TranscriptionEvents 的处理程序的特定函数：

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

TranscriptionController 有两个函数，用于管理对 TranscriptionEvent 回调的订阅和取消订阅：

```
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;
}
```

使用可选 **TranscriptionController**

我们提供了名为 `DefaultTranscriptionController` 的 `TranscriptionController` 接口的默认实现。`DefaultRealtimeController` 和 `DefaultAudioVideoFacade` 中的默认实现会返回一个 `DefaultTranscriptionController` 对象：

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

`DefaultRealtimeController` 也在其构造函数中使用一个可选 `TranscriptionController` 对象。这允许您覆盖 `DefaultTranscriptionController` 行为。开发人员应用程序通过 `AudioVideoFacade` 对象的 `TranscriptionController` 对象订阅和取消订阅一个或多个回调：

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

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

传输示例

以下示例显示如何处理收到的 `TranscriptEvent`。

Note

确切输出取决于几个因素，包括个人说话的速度和停顿时间。

示例 1 : StartMeetingTranscription

此示例显示了典型的 StartMeetingTranscription 操作。

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

该操作会生成一个 TranscriptEvent。

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

示例 2 : 部分转录结果

在这个示例中，一位与会者说：“那只快速的棕色狐狸跳过那只懒狗。”在示例中，isPartial 值为 true。如果更深入地研究消息，您会发现系统将“狐狸”一词处理为“事实”。系统使用相同的 resultId 更新转录。

```
{  
  transcript: {  
    results: [{  
      resultId: "1",  
      startTimeMs: 1620118800000,  
      alternatives: [{  
        items: [{  
          type: 'pronunciation',  
          startTimeMs: 1620118800000,  
          attendee: { attendeeId: "1",  
            isPartial: true,  
            endTimeMs: 1620118801000,  
            endTimeMs: 1620118800200,  
            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:"facts",
        endTimeMs: 1620118801000,
        externalUserId: "A" },
        vocabularyFilterMatch: false
    },
    {
        type:'punctuation',
        startTimeMs: 1620118801000,
        attendee:{ attendeeId: "1",
        content:  ", ",
        endTimeMs: 1620118801500,
        externalUserId: "A" },
        vocabularyFilterMatch: false
    }
    ]
    ]
    ]
}

```

示例 3：最终转录结果

如果转录不完整，系统会再次处理该短语。此示例的 `false` 值为 `isPartial`，消息包含“狐狸”而不是“事实”。系统使用相同的 ID 重新发出消息。

```

{
  transcript: {
    results: [{
      resultId:"1",
      startTimeMs: 1620118800000,
      alternatives: [{
        items:[{
          isPartial: false,
          endTimeMs: 1620118801000,

```

```

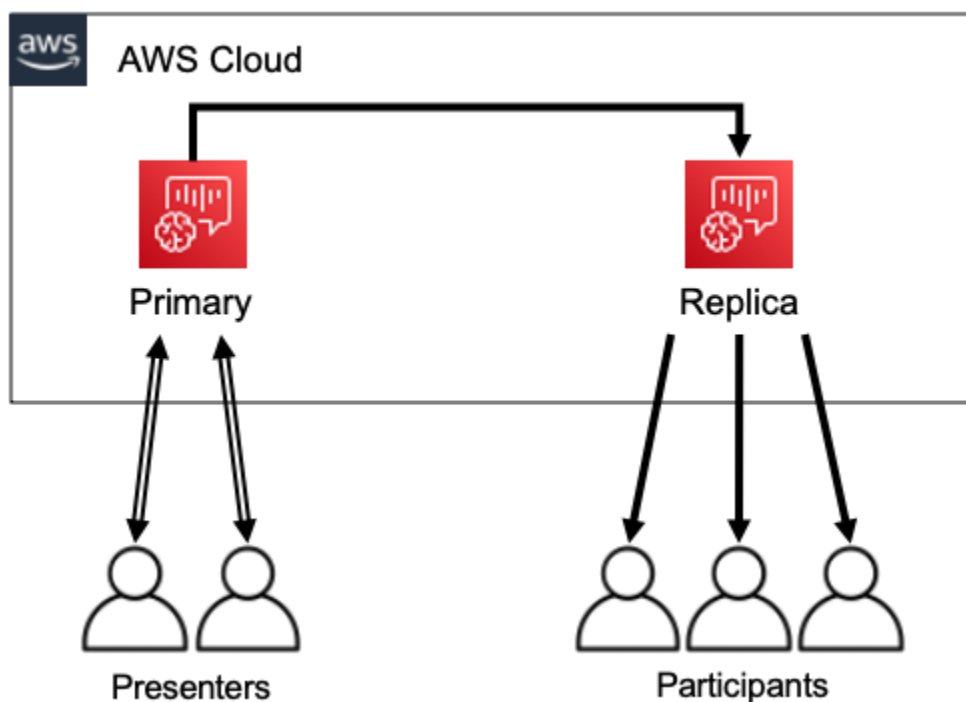
    type:      'pronunciation',
    startTimeMs: 1620118800000,
    attendee: { attendeeId: "1",
                content: "the",
                endTimeMs: 1620118800200,
                externalUserId: "A"},
                vocabularyFilterMatch: false
  },
  {
    type:      'pronunciation',
    startTimeMs: 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
  }}
  ]]
  ]]
}
}
}

```

使用媒体复制

您可以使用媒体复制将主 WebRTC 会话与多个副本会话关联，以覆盖更多的受众。每个 WebRTC 媒体会话支持 250 个连接，并且您可以将一个主会话复制到多个副本会话。连接到副本会话的参与者只能接收连接到主会话的主讲人的音频和视频。他们对连接到复制会话的参与者一无所知，这使得媒体复制非常适合网络研讨会和其他需要隐私的使用案例。

下图显示了带有主讲人共享音频和视频的主会话与参与者使用媒体的副本会话之间的媒体复制。



Note

Chime SDK Meetings — 每个主会议的副本会议的服务限额默认值为 4，您可以根据请求提高该限制。有关限额的更多信息，请参阅 AWS 一般参考中的 [AWS 服务限额](#)。

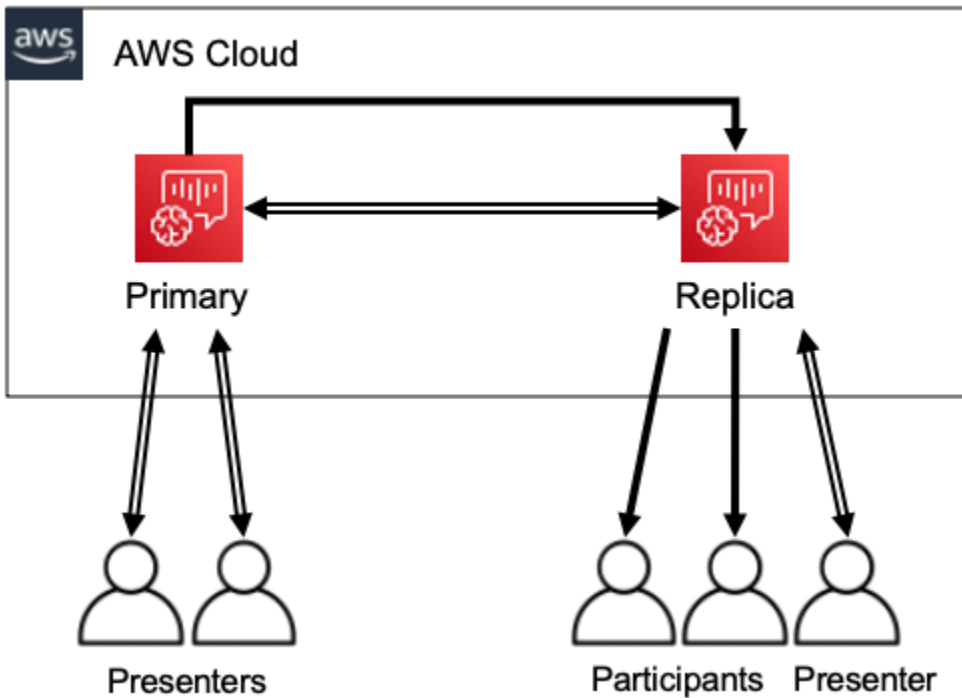
主题

- [互动参与者](#)
- [全球参与者](#)
- [会话生命周期](#)

互动参与者

可向连接到副本会话的参与者授予加入主会话的访问权限。由于每个人都使用 WebRTC 连接，因此主讲人和参与者不会遇到转码延迟。当参与者在主会话和副本会话之间切换时，他们会重复使用自己的 WebRTC 连接，因此切换速度非常快。这使参与者能够在不错过任何内容的情况下仍能进行实时对话。

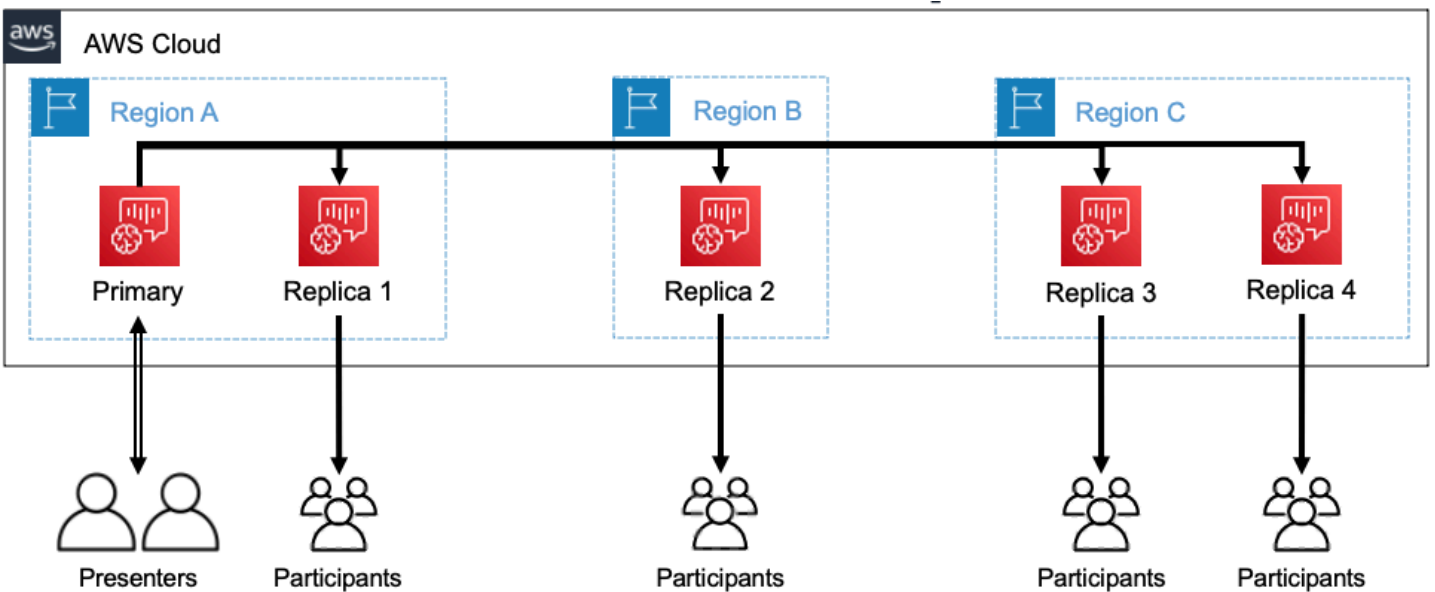
下图显示了副本会话的参与者使用其 WebRTC 连接切换到主会话。



全球参与者

您可以为每次 WebRTC 媒体会话选择 AWS 区域。这使您能够在比主会话所在区域更靠近参与者的区域中创建副本会话。执行此操作时，媒体会通过 AWS 网络从主会话流向副本会话，然后通过 Internet 从副本会话流向参与者。向全球受众演示时，将副本会话创建于参与者附近可以帮助确保媒体通过 AWS 网络而不是互联网在全球传输，从而获得更好的会议体验。

下图显示了不同区域的主会话和副本会话。



会话生命周期

创建会话

您可以使用 [CreateMeeting](#) 或 [CreateMeetingWithAttendees](#) API 创建 WebRTC 媒体会话。默认情况下，API 会创建主会话，除非专门创建副本会话。

您可以通过在 [CreateMeeting](#) 或 [CreateMeetingWithAttendees](#) API 调用中将主会话的 MeetingId 指定为 PrimaryMeetingId 来创建副本会话。

Note

如果您将副本会话的 MeetingId 指定为 PrimaryMeetingId，则 API 调用将失败。

创建与会者

要创建加入 WebRTC 媒体会话所需的与会者凭证，您可以使用 [CreateMeetingWithAttendees](#)、[CreateAttendees](#) 或 [BatchCreateAttendees](#) API。

Note

为大量与会者创建会话时，使用 [CreateMeetingWithAttendees](#) 或 [BatchCreateAttendee](#) 以最大限度地减少所需的 API 调用次数。

删除与会者

您可以使用 [DeleteAttendee](#) API 撤消与会者参加 WebRTC 媒体会话的凭证。如果与会者已连接到会议，他们将断开连接并且无法重新加入。

使用 [DeleteMeeting](#) API 删除 WebRTC 媒体会话时，该 API 会自动删除所有与会者，并且您无需调用 [DeleteAttendee](#)。

切换会话

要允许参与者从副本会话切换到主会话，您必须在主会议中为他们创建凭证。请参阅此列表前面的创建与会者。使用 Amazon Chime SDK 客户端库中带有 `promoteToPrimaryMeeting` 方法的凭证切换到主会话。

要将参与者切换回副本会话，使用 Amazon Chime SDK 客户端库中的 `demoteToPrimaryMeeting` 方法，或者使用 [DeleteAttendee](#) API 使其主会话凭证失效。

Note

直接连接到主会话的主讲人无法切换到副本会话。

有关在会话之间切换的更多信息，请参阅客户端库文档：

- GitHub 上的 [Amazon Chime SDK 安卓版](#)。
- GitHub 上的 [Amazon Chime SDK iOS 版](#)。
- GitHub 上的 [适用于 JavaScript 的 Amazon Chime SDK 客户端库](#)。

删除会话

您可以使用 [DeleteMeeting](#) API 删除 WebRTC 媒体会话。

如果您删除主会话，DeleteMeeting API 会自动删除所有附加的副本会话。因此，要删除所有会话，只需删除主会话即可。

如果连续 5 分钟内没有与会者连接，则该服务会自动删除主会话。该服务仅在删除主会话时自动删除副本会话。也就是说，您可以在创建主会话时创建副本会话，并且副本将在主会话期间可用。

故障排除和调试 Amazon Chime SDK 会议

使用以下主题帮助诊断和排除使用 Amazon Chime SDK 时遇到的问题。

主题

- [了解系统要求](#)
- [设置日志记录和监控](#)
- [自我故障排除](#)
- [常见问题](#)

了解系统要求

作为故障排除的一部分，确保您的代码适用于受支持的浏览器。有关受支持的浏览器、版本和操作系统的当前列表，请参阅 [Amazon Chime SDK 系统要求](#)。[Github 上的开发人员指南和常见问题解答](#)可以解决浏览器和其他兼容性问题。另外，请熟悉[已知的浏览器问题](#) GitHub和任何解决方法。

如果你刚开始使用 Amazon Chime SDK Meetings，那么 [Amazon Chime SDK 的 Builder Journey](#) 提供了 step-by-step 使用亚马逊 Chime SDK 进行构建的指南，以及故障排除所需的工具。

设置日志记录和监控

日志记录可帮助您收集诸如服务器端会议事件和客户端浏览器控制台日志等信息。

Amazon Chime SDK 提供服务器端会议活动，您可以将其发送到亚马逊和 EventBridge 亚马逊 CloudWatch 活动日志。您可以创建 CloudWatch 指标和见解，并在控制面板中使用它们进行监控。[Amazon Chime SDK 事件的服务器端日志和监控](#) 博客文章介绍了如何启用 CloudWatch 指标、见解和控制面板。

Amazon Chime SDK 提供客户端事件，用于监控音频和视频质量、网络带宽和连接问题。[使用 Amazon Chime SDK 会议事件进行监控和故障排除](#) 博客文章介绍了如何针对加入失败、音频质量问题以及麦克风和摄像头设置失败启用 CloudWatch 指标、见解和控制面板。有关会议活动的更多信息，请参阅 Github 上的[会议事件](#)。

用于对指标进行故障排除的选项

您可以使用以下选项来收集故障排除事件。

- 在每次活动时发送指标
- 每 N 秒批处理一次事件
- 在会议结束时发送指标
- 浏览器控制台日志的日志记录级别

推荐的指标

您至少应收集和记录以下指标。

- SDK 平台和版本
- 浏览器和版本
- 操作系统
- 逻辑函数
- 会议开始
- 会议结束
- 已加入与会者
- 已离开与会者

- 已掉线与会者

此外，根据您遇到的问题，以下指标可以提供有关连接、带宽和质量问题的信息。您可以将这些指标每一次出现的情况记录下来，也可以只计算数量。计数可以提供对潜在问题的总结视图：

- connectionDidSuggestStopVideo
- connectionDidBecome不错
- connectionDidBecome可怜的
- Attendee join time > t seconds
- MeetingStartFailed
- MeetingFailed

启用客户端日志记录

您可以通过将 `LogLevel.INFO` 传递给 `ConsoleLogger` 对象来启用 `INFO` 级别的浏览器日志。

```
const logger = new ConsoleLogger('MyLogger', LogLevel.INFO);const meetingSession = new DefaultMeetingSession(configuration, logger, deviceController);
```

您还可以使用 Amazon Chime 软件开发工具包中的 `POSTLogger` 组件 JavaScript 来捕获后端的浏览器日志，例如亚马逊 CloudWatch 日志。 `POSTLoggerHTTP` `POST` 请求将浏览器日志上传到 `POSTLogger` 构造函数中的给定网址。例如，[上的 Amazon Chime SDK 无服务器演示 GitHub](#) 使用将浏览器日志发送 `POSTLogger` 到 Amazon Logs 以 CloudWatch 供将来调查。

启用服务器端日志记录

适用的 Amazon Chime SDK JavaScript 还会使用关键会议事件（例如 `MeetingStartFailed` 和 `MeetingFailed`）调用 `eventDidReceive` 旁听者方法。 `MeetingFailed` 会议活动通常包括故障的具体原因。例如，假设一大群客户遇到故障。您的 Web 应用程序可收集这些会议事件，然后与我们共享以排查根本原因。有关会议活动的更多信息，请参阅 [上的 GitHub 会议活动指南](#) 以及 [使用 Amazon Chime SDK 会议事件进行监控和故障排除](#) 博客文章。

自我故障排除

本主题中的各部分说明几种对 Amazon Chime SDK 会议进行自我故障排除的方法。

主题

- [查看常见问题解答和已知问题](#)
- [验证网络访问](#)

查看常见问题解答和已知问题

查看这些常见问题解答和已知问题列表，GitHub 以获取故障排除和调试建议。

- [亚马逊 Chime 软件开发工具包 —— 会议 JavaScript](#)
- [亚马逊 Chime 软件开发工具包 —— 媒体 JavaScript](#)
- [亚马逊 Chime 软件开发工具包 —— 联网 JavaScript](#)
- [Amazon Chime SDK —— 音频和视频](#)

验证网络访问

企业通常会设置网络防火墙，限制对特定端口的访问或连接到其网络之外的 IP 地址范围。以下各部分说明验证网络访问的一些方法。

主题

- [验证 AWS SDK 和 Amazon Chime SDK 子网和端口](#)
- [使用演示应用程序重现问题](#)
- [使用会议准备情况检查程序](#)

验证 AWS SDK 和 Amazon Chime SDK 子网和端口

使用 Amazon Chime SDK 的应用程序使用两个层级，即服务器和客户端。服务器层使用 AWS SDK 并具有服务器端会议处理程序。客户端层使用客户端 SDK。

S AWS DK 用于调用服务器 API，例如[CreateMeeting](#)。这些 API 会连接到 us-east-1、us-west-2、ap-southeast-1、eu-central-1、us-gov-east-1 和 us-gov-west-1 区域的 AWS 全球服务终端节点。AWS 一般参考中的 [AWS IP 地址范围](#) 页面列出了每个区域的 IP 地址范围。有关服务终端节点和限额的信息，请参阅 [Amazon Chime SDK 终端节点和限额](#)

客户端软件开发工具包（例如适用的 Amazon Chime 软件开发工具包 JavaScript）连接到域中的 *.chime.aws 服务终端节点。

使用以下验证确保您拥有网络权限：

- 运行 [Amazon Chime SDK 会议准备情况检查器](#)，验证您是否可以访问您的网络和端口。GitHub
- 确认您可以从您的网络或最终用户的网络解析 *.chime.aws 域。
- 确保您的防火墙允许通过 TCP 端口 443 (用于控制命令) 和 UDP 端口 3478 (用于媒体) 连接至 AWS IP 范围。

使用演示应用程序重现问题

作为最佳实践，通过尝试在一个演示应用程序中重现您的问题开始调试过程。这使服务团队能够找到问题可能出在哪里。如果您无法在演示应用程序中重现问题，可以查看该应用程序的代码，以了解其如何实现相关使用案例。

Amazon Chime SDK	功能	演示应用程序资源
JavaScript SDK	会议	演示说明 、 源代码
React 组件	会议	演示说明 、 源代码
会议聊天	消息收发	博客文章 、 演示说明 、 源代码
iOS/Android	会议	(博客文章) 使用 Amazon Chime SDK 在安卓系统上构建会议应用程序 (博客文章) 使用 Amazon Chime SDK 在 iOS 系统上构建会议应用程序
PSTN 音频	入站调用	博客文章 源代码

使用会议准备情况检查程序

使用 [Amazon Chime SDK 会议准备情况检查器](#)。GitHub 检查程序可帮助验证音频和视频设备以及用户连接情况。您可以使用通过/失败状态向最终用户展示结果，这些状态可以显示任何问题的根本原因。

常见问题

以下各部分提供常见会议问题的故障排除方法。

主题

- [连接问题](#)
- [音频和视频质量问题](#)
- [验证 SDK 限额和 API 节流](#)
- [打开支持案例](#)

连接问题

有关连接问题，请参阅 [验证网络访问](#)。

音频和视频质量问题

音频和视频质量问题可能有多种原因。音频/视频质量不理想的两大主要原因是网络带宽和设备性能。有关不同挑战以及这些挑战如何影响音频/视频质量的详细信息，请查看质量、带宽和连接 (https://aws.github.io/amazon-chime-sdk-js/modules/qualitybandwidth_connectivity.html)。本文描述可以监控的不同的事件和指标，以检测带宽问题和潜在缓解措施。

您可以选择距离目标会议会话的受众更近的媒体区域。要了解如何选择最佳媒体区域，请参阅使用会议区域 (<https://docs.aws.amazon.com/chime-sdk/latest/dg/chime-sdk-meetings-regions.html>)。

Amazon Chime SDK 会根据会议与会者的可用带宽调整正在接收/上传的视频的视频质量。要了解如何控制不同视频布局的视频质量，请访问管理不同视频布局的视频质量 (<https://aws.github.io/amazon-chime-sdk-js/modules/videolayout.html>)。本文描述视频生命周期管理和上行/下行链路策略。

视频分辨率注意事项

- 上传视频的默认分辨率为 540p，在 1400kbps 时为 15fps。您可以根据带宽降低分辨率和帧速率。
- 根据可用的接收器带宽，确定要显示视频磁贴的数量。所有视频磁贴和内容共享不要超过 6Mbps。当最终用户没有足够带宽时，会看到黑色视频磁贴。

使用视频上行链路和下行链路带宽策略

Amazon Chime SDK 提供以下带宽策略。

- N ScaleVideoUplinkBandwidthPolicy — 实现的捕获和编码参数几乎等于桌面、Web 和移动客户端使用的参数。
- AllHighestVideoBandwidthPolicy — 始终订阅最高质量的视频流。

- `NoVideoDownlinkBandwidthPolicy` — 当带宽降至给定阈值以下时，禁用视频。
- `VideoPriorityBasedPolicy` — 在带宽较低的情况下，优先考虑音频而不是视频。
- `VideoAdaptiveProbePolicy`

验证 SDK 限额和 API 节流

[Amazon Chime SDK 终端节点和限额](#)页面列出服务限额、API 速率以及您是否可以进行调整。使用[AWS 控制台服务限额](#)页面请求限额调整。

微调您的 API 速率

超过其 API 速率的应用程序会收到 HTTP 状态码 429 和 `ThrottledClientException` 消息。您可以调整您的 API 速率，但在调整之前，请检查您的应用程序中是否存在可能耗尽这些速率的错误。例如，您可以在循环中创建会议，或者创建会议而不进行清理。

根据您的创建会议的方式，可能需要修改代码。例如，将 `CreateMeeting` 和 `CreateAttendee` 替换为：

- [CreateMeetingWithAttendees](#) — 每次会议最多可创建 10 名与会者。
- [BatchCreateAttendee](#) — 每次会议最多可创建 100 名与会者。

您可以将创建的与会者存储在数据库中，在受邀者加入会议时获取与会者信息，然后将他们与预先创建的与会者关联。

打开支持案例

如果您还有其他问题或需要业务支持，可以联系 [AWS 客户支持](#)。有关我们的支持计划的更多信息，请参阅[比较支持计划](#)页面。创建支持案例时，始终在存在问题的账户下打开。包括控制台浏览器日志、会议和与会者 ID 以及任何相关的支持案例或 GitHub 问题。

使用 Amazon Chime SDK 消息传递

您可以使用《Amazon Chime SDK 开发人员指南》中的这一节来帮助创建在 Amazon Chime SDK 服务上运行的消息传递应用程序。此 SDK 提供了创建基本消息传递应用程序所需的概念和实用信息。

主题

- [迁移到 Amazon Chime SDK 身份命名空间](#)
- [迁移到 Amazon Chime SDK 消息传递命名空间](#)
- [消息传递的先决条件](#)
- [消息传递概念](#)
- [消息传递架构](#)
- [消息类型](#)
- [开始使用](#)
- [了解系统消息](#)
- [IAM 角色示例](#)
- [了解按角色划分的授权](#)
- [流式传递消息数据](#)
- [使用弹性频道举办实时事件](#)
- [使用移动推送通知接收消息](#)
- [使用服务相关角色](#)
- [使用频道流处理消息](#)
- [使用 AppInstanceBot 作为智能频道座席](#)
- [管理消息保留日期](#)
- [用于消息传递的 UI 组件](#)
- [与客户端库集成](#)
- [将 Amazon Chime 软件开发工具包消息传递与 JavaScript](#)

迁移到 Amazon Chime SDK 身份命名空间

[Amazon Chime SDK 身份命名空间](#)是用于创建和管理 Amazon Chime SDK 身份资源 (包括 AppInstance 和 AppInstanceUser) 的 API 的专用位置。您可以使用命名空间来寻址 Amazon Chime

SDK 身份 API 终端节点所在的任何 AWS 区域。如果您刚开始使用 Amazon Chime SDK，则使用此命名空间。有关“区域”的更多信息，请参阅本指南中的 [可用区](#)。

使用 [Amazon Chime](#) 命名空间的现有应用程序应计划迁移到专用命名空间。

主题

- [迁移原因](#)
- [迁移之前](#)
- [命名空间之间的差异](#)

迁移原因

出于以下原因，我们鼓励您迁移到 [Amazon Chime SDK 身份](#) 命名空间：

选择 API 终端节点

Amazon Chime SDK 身份命名空间是唯一可以在任何[提供 API 终端节点的区域](#)中使用 API 终端节点的 API 命名空间。如果您想使用 us-east-1 以外的 API 终端节点，则必须使用 Amazon Chime SDK 身份命名空间。有关当前终端节点的更多信息，请参阅本指南中的 [API 映射](#)。

更新和新的消息传递 API

我们仅在 Amazon Chime SDK 身份命名空间中添加或更新身份 API。

迁移之前

在迁移之前，记下命名空间之间的差异。下表列出并描述了以上差异。

	Amazon Chime SDK 身份命名空间	Amazon Chime 命名空间
AWS SDK 命名空间	ChimeSDKIdentity	Chime
区域	多个	仅限 us-east-1
服务主体	https://identity.chime.amazonaws.com	https://chime.amazonaws.com

	Amazon Chime SDK 身份命名空间	Amazon Chime 命名空间
API	仅用于身份的 API	用于身份和 Amazon Chime 其他部分的 API
用户过期	可用	不可用
自动程序	可用	不可用

命名空间之间的差异

以下各节解释了 Chime 和 ChimeSDKIdentity 命名空间之间的差异。

AWS SDK 命名空间

Amazon Chime SDK 命名空间使用 Chime 正式名称。Amazon Chime SDK 身份命名空间使用 ChimeSDKIdentity 正式名称。名称的确切格式因平台而异。

例如，如果您在 Node.js 中使用 AWS SDK 来创建身份，则使用一行代码来寻址命名空间。

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

若要迁移到 ChimeSDKIdentity 命名空间，请使用新的命名空间和终端节点区域更新这行代码。

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

区域

[Amazon Chime](#) 命名空间只能寻址该 us-east-1 地区的 API 终端节点。[Amazon Chime SDK 身份](#) 命名空间可以在任何可用区域中寻址 Amazon Chime SDK 身份 API 终端节点。有关终端节点区域的最新列表，请参阅本指南中的 [可用区](#)。

终端节点

[Amazon Chime SDK 身份](#) 命名空间使用不同于 [Amazon Chime](#) 命名空间的 API 终端节点。

只有用于创建身份资源的终端节点才能用于更新这些资源。这意味着通过 eu-central-1 中的终端节点创建的 ApplInstance 只能通过 eu-central-1 进行修改。这也意味着您无法使用 us-east-1 中的

ChimeSDKIdentity 命名空间寻址通过 Chime 命名空间创建的 ApplInstance，或在创建 ApplInstance 和 ApplInstanceUser 成员的区域以外的区域中创建频道。有关当前终端节点的更多信息，请参阅本指南中的 [API 映射](#)。

服务主体

[Amazon Chime SDK 身份](#)命名空间使用了新的服务主体：Identity.chime.amazonaws.com。如果您有授予服务访问权限的 SQS、SNS 或其他 IAM 访问策略，则需要更新这些策略以授予新服务主体访问权限。

API

[Amazon Chime SDK 身份](#)命名空间仅包含用于创建和管理消息资源以及发送和接收消息的 API。[Amazon Chime](#) 命名空间包括用于 Amazon Chime 服务其他部分的 API 以及消息传递。

用户过期

创建 ApplInstanceUser 时的过期设置允许您创建临时用户。例如，您可以创建仅在大型广播期间存在的聊天用户。只有身份命名空间支持 ApplInstanceUser 的过期设置。

自动程序

您可以使用 [ApplInstanceBot](#) API 将由 Amazon Lex V2 提供支持的聊天机器人添加到您的应用程序中。您只能在身份命名空间中使用 ApplInstanceBot。有关自动程序的更多信息，请参阅本指南中的 [使用 ApplInstanceBot 作为智能频道座席](#)。

其它 API

身份命名空间的其他 API 列表越来越多，而 Chime 命名空间没有这些 API。如果您刚开始使用 Amazon Chime SDK，请使用身份命名空间来访问所有最新功能。有关当前 API 的更多信息，请参阅《Amazon Chime SDK API 参考》中的 [Amazon Chime SDK 身份](#)。

迁移到 Amazon Chime SDK 消息传递命名空间

[Amazon Chime SDK 消息传递](#)命名空间是 API 的专用位置，用于创建和管理 Amazon Chime SDK 消息传递资源。您可以使用命名空间在 Amazon Chime SDK 消息传递 API 终端节点所在的任何 AWS 区域对其进行寻址。如果您刚开始使用 Amazon Chime SDK，则使用此命名空间。有关“区域”的更多信息，请参阅本指南中的 [可用区](#)。

使用 [Amazon Chime](#) 命名空间的现有应用程序应计划迁移到专用命名空间。

主题

- [迁移原因](#)
- [迁移之前](#)
- [命名空间之间的差异](#)

迁移原因

出于以下原因，我们鼓励您迁移到 [Amazon Chime SDK 消息传递](#) 命名空间：

选择 API 终端节点

Amazon Chime SDK 消息传递命名空间是唯一可以在任何[提供 API 终端节点的区域](#)中使用 API 终端节点的 API 命名空间。如果您想使用 us-east-1 以外的 API 终端节点，则必须使用 Amazon Chime SDK 消息传递命名空间。

有关 Amazon Chime SDK 消息传递如何使用 AWS 区域的更多信息，请参阅本指南中的[可用区域](#)。

更新和新的消息传递 API

我们仅在 Amazon Chime SDK 消息传递命名空间中添加或更新消息传递 API。

迁移之前

在迁移之前，记下命名空间之间的差异。下表列出并描述了以上差异。

	Amazon Chime SDK 消息传递命名空间	Amazon Chime 命名空间
AWS SDK 命名空间	ChimeSDKMessaging	Chime
区域	多个	仅限 us-east-1
API	仅用于消息传递的 API	用于消息传递和 Amazon Chime 其他部分的 API
流	可用	不可用

	Amazon Chime SDK 消息传递命名空间	Amazon Chime 命名空间
弹性频道	可用	不可用

命名空间之间的差异

以下各节解释了 Amazon Chime 和 Amazon Chime SDK Messaging 命名空间之间的差异。

AWS SDK 命名空间

Amazon Chime SDK 命名空间使用 Chime 正式名称。Amazon Chime SDK 消息传递命名空间使用 ChimeSDKMessaging 正式名称。名称的确切格式因平台而异。

例如，如果您在 Node.js 中使用 AWS SDK 来创建消息传递，则使用一行代码来寻址命名空间。

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

要迁移到 Amazon Chime Messaging SDK，请使用新的命名空间和终端节点区域更新这行代码。

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

区域

[Amazon Chime](#) 命名空间只能寻址该 us-east-1 地区的 API 终端节点。[Amazon Chime SDK 消息传递](#)命名空间可以在任何可用区域中寻址 Amazon Chime SDK 消息传递 API 终端节点。有关最新的消息传递区域列表，请参阅本指南中的 [可用区](#)。

终端节点

[Amazon Chime SDK 消息传递](#)命名空间使用与 [Amazon Chime](#) 命名空间不同的 API 终端节点。

只能使用用于创建消息传递资源的终端节点对其进行修改。这意味着通过 eu-central-1 中的终端节点创建的消息传递资源只能通过 eu-central-1 进行修改。这意味着通过 eu-central-1 中的终端节点创建的频道只能通过 eu-central-1 进行修改。这也意味着您无法寻址通过 Chime 命名空间创建且 ChimeSDKMessaging 命名空间位于 us-east-1 中的频道。有关当前终端节点的更多信息，请参阅本指南中的 [API 映射](#)。

服务主体

[Amazon Chime SDK 消息传递](#)命名空间使用了新的服务主体

体：`messaging.chime.amazonaws.com`。如果您有授予服务访问权限的 SQS、SNS 或其他 IAM 访问策略，则需要更新这些策略以授予新服务主体访问权限。

API

[Amazon Chime SDK 消息传递](#)命名空间仅包含用于创建和管理消息传递资源以及发送和接收消息的 API。[Amazon Chime](#)命名空间包括用于 Amazon Chime 服务其他部分的 API 以及消息传递。

频道流

频道流允许开发人员在将动态消息传递给消息传递频道成员之前，对这些消息运行业务逻辑。例如，您可以创建流程，在发送消息之前从消息中删除敏感数据，例如身份证号、电话号码或污言秽语。这可以帮助实施企业传播策略或其他沟通准则。

您还可以使用频道流来执行一些功能，例如在将结果发送回给参与者之前汇总对民意调查的回复，或者通过短信发送消息。

频道流仅在 `ChimeSDKMessaging` 命名空间中可用。有关更多信息，请参阅本指南中的 [使用频道流处理消息](#)。

弹性频道

弹性频道支持大规模的聊天体验，多达 100 万聊天用户可在指定数量的子频道中自动平衡。弹性频道仅在 `ChimeSDKMessaging` 终端节点中可用。有关弹性频道的更多信息，请参阅本指南中的 [使用弹性频道举办实时事件](#)。

其它 API

消息传递命名空间中 Chime 命名空间所没有的 API 在不断增加。如果您刚开始使用 Amazon Chime SDK，请使用消息传递命名空间来访问所有最新功能。有关当前 API 的更多信息，请参阅《Amazon Chime SDK API 参考》中的 [Amazon Chime SDK 消息传递](#)。

消息传递的先决条件

您需要以下内容才能使用 Amazon Chime SDK 消息传递。

- 编程能力。
- 一个 AWS 账户。
- 使用 Amazon Chime SDK 消息传递为应用程序配置 IAM 角色的权限。

在大多数情况下，您还需要：

- 客户端应用程序：显示消息传递 UI，使用 Amazon Chime SDK 连接 Web 套接字，管理状态。
- 服务器应用程序：管理身份和用户。

消息传递概念

要有效地使用 Amazon Chime SDK 消息传递，您必须了解以下术语和概念。

AppInstance

要使用 Amazon Chime 软件开发工具包消息，您必须先创建一个。AppInstance AppInstance 包含 AppInstanceUsers 和频道。通常，您可以 AppInstance 为应用程序创建单曲。一个 AWS 账户可以有多个 AppInstances。您可以在该级别进行应用程序级别的设置，例如消息保留和直播配置。AppInstance AppInstances 由以下格式的唯一 ARN 标识：`arn:aws:chime:region:aws_account_id:app-instance/app_instance_id`

AppInstanceUser

AppInstanceUsers 是发送消息、创建频道、加入频道等的实体。通常，您可以创建 AppInstanceUser 与应用程序用户的 one-to-one 映射。您还可以创建 AppInstanceUser 连接到后端服务，这样用户就可以将消息识别为来自后端服务。AppInstanceUsers 由 ARN 识别，例如。`arn:aws:chime:region:aws_account_id:app-instance/app_instance_id/user/app_instance_user_id`您可以控制 `app_instance_user_id`，并且作为最佳实践，可重复使用应用程序已有的 ID。

频道

当您向频道添加 AppInstanceUser 时，该用户将成为成员并可以发送和接收消息。频道可以是公开的，允许任何用户将自己添加为成员，也可以是私人频道，后者只允许频道监管人添加成员。您也可以隐藏频道成员。隐藏成员可以观察对话但不能发送消息，而且他们不会被添加到频道成员资格中。

SubChannel

弹性通道的成员被划分为一个名为的逻辑容器 SubChannels。当您 AppInstanceUser 向弹性通道中添加时，用户将成为其成员，SubChannel 并且可以发送和接收该特定通道的消息 SubChannel。频道成员资格和消息处于一定 SubChannel 级别，这意味着其中一个成员发送的消息 SubChannel 不会被另一个 SubChannel 成员接收。成员被转移到不同的渠道 SubChannels，以支持渠道的弹性并促进参与度。

UserMessage

属于频道的 `AppInstanceUser` 可以发送和接收用户消息。`AppInstanceUser` 可以发送 STANDARD 或 CONTROL 消息。STANDARD 消息可以包含 4KB 的数据和 1KB 的元数据。CONTROL 消息只能包含 30 字节的数据。消息可以是 PERSISTENT 或 NON_PERSISTENT。您可以从频道历史记录中检索 PERSISTENT 消息。NON_PERSISTENT 只有当前连接到 Amazon Chime SDK 消息的频道成员才能看到消息。

系统消息

Amazon Chime SDK 会生成系统消息，以响应诸如成员加入或离开频道之类的事件。

消息传递架构

您可以将 Amazon Chime SDK 消息传递作为服务器端和客户端 SDK 使用。服务器端 API 会创建 `AppInstance` 和 `AppInstanceUser`。您可以使用各种挂钩和配置来添加特定于应用程序的业务逻辑和验证。有关如何执行该操作的更多信息，请参阅 [流式传递消息数据](#)。此外，服务器端进程可以代表 `AppInstanceUser` 调用 API，或者控制代表后端进程的专用 `AppInstanceUser`。

表示为 `AppInstanceUser` 的客户端应用程序可以直接调用 Amazon Chime SDK 消息传递 API。客户端应用程序在联机时使用该 WebSocket 协议连接到消息传递 SDK。连接后，他们会收到来自他们所属的任何频道的实时消息。断开连接后，`AppInstanceUser` 仍属于其添加的频道，它可以使用 SDK 基于 HTTP 的 API 加载这些频道的消息历史记录。

客户端应用程序有权将 API 调用作为单个 `AppInstanceUser`。要将 IAM 凭证的范围限定为单个 `AppInstanceUser`，客户端应用程序通过 AWS Cognito 身份池或小型自托管式后端 API 担任参数化 IAM 角色。有关身份验证的更多信息，请参阅 [对最终用户客户端应用程序进行身份验证](#)。相比之下，服务器端应用程序的权限通常与单个应用程序实例用户绑定，例如具有管理权限的用户，或者它们有权代表所有应用程序实例用户进行 API 调用。

消息类型

您通过频道发送消息。您可以发送 STANDARD、CONTROL 或 SYSTEM 消息。

- STANDARD 消息大小最多可达 4KB，并且包含元数据。元数据是任意的，您可以通过多种方式使用它，例如包含指向附件的链接。
- CONTROL 消息不超过 30 字节，并且不包含元数据。

- STANDARD 和 CONTROL 消息可以是永久性消息或非永久性消息。永久性消息保留在频道的历史记录中，并通过 ListChannelMessages API 调用进行查看。非永久性消息将发送给每个通过 AppInstanceUser 连接的用户 WebSocket。
- Amazon Chime SDK 会针对成员加入或离开频道之类的事件自动发送 SYSTEM 消息。

开始使用

本节中的主题介绍了如何开始生成 Amazon Chime SDK 消息传递应用程序。

主题

- [创建一个 AppInstance](#)
- [从后端服务发出 SDK 调用](#)
- [对最终用户客户端应用程序进行身份验证](#)
- [创建频道](#)
- [发送消息](#)
- [使用 ExpirationSettings](#)
- [WebSockets 用于接收消息](#)
- [配置附件](#)

创建一个 AppInstance

要使用 Amazon Chime 软件开发工具包消息，您必须先先在账户中创建亚马逊 Chime AppInstance 软件开发工具包。AWS

主题

- [建造一个 AppInstance](#)
- [创建一个 AppInstanceUser](#)

建造一个 AppInstance

为消息传递创建 **AppInstance**

1. 在 CLI 中，运行 `aws chime-sdk-identity create-app-instance --name NameOfAppInstance`。

- 在“创建响应”中，记下 AppInstanceArn 和 `arn:aws:chime:region:aws_account_id:app-instance/app_instance_id`。

创建一个 AppInstanceUser

一旦您创建了一个 AppInstance，您就可以在该 AppInstance 中创建一个 AppInstanceUser。通常是在用户首次注册或登录应用程序时执行此操作。您也可以创建代表后端服务运行的 AppInstanceUser。

以下示例介绍了如何创建一个后端 AppInstanceUser。

```
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"
```

在“创建响应”时，记下 AppInstanceUserArn。其形式为 `:arn:aws:chime:region:aws_account_id:app-instance/app_instance_id/user/app_instance_user_id`。在本例中，`app_instance_user_id` 是 “back-end-worker。”

Note

作为最佳实践，在为客户端应用程序创建 AppInstanceUser 时，请将 AppInstanceId 匹配该用户的现有唯一 ID（例如身份提供商的 sub）。该名称是附加到某些 API 实体（例如消息发送者）的可选占位符。它允许您在一个地方控制用户的显示名称，而不必从 AppInstanceUser ARN 中查找，ARN 也作为消息的发件人附上。

从后端服务发出 SDK 调用

创建代表后端服务的用户后，您就可以创建一个频道，向该频道发送消息，并从该频道读取消息。

运行以下 CLI 命令来创建一个公有通道。

```
aws chime-sdk-messaging create-channel \  
  --chime-bearer "app_instance_user_arn" \  
  --app-instance-arn "app_instance_arn" \  
  --name "channel-name"
```

```
--name "firstChannel"
```

该命令生成以下格式的 ARN : `arn:aws:chime:region:aws_account_id:app-instance/app_instance_id/channel/channel_id`.

主题

- [后端服务的 IAM 授权是如何运作的](#)
- [了解隐式 API 授权](#)
- [发送和列出频道消息](#)

后端服务的 IAM 授权是如何运作的

在上一节的 CLI 命令中，记下该 `chime-bearer` 参数。它识别创建频道和消息等资源或与之交互的用户。几乎所有 Amazon Chime SDK 消息传递 API 都以 `chime-bearer` 为参数，但本来只能由开发人员调用的 API 除外，例如 `CreateAppInstance`。

Amazon Chime SDK 消息传递 API 的 IAM 权限需要与 `chime-bearer` 参数匹配的 `app-instance-user-arn`。根据 API，可能需要其他 ARN（通常是频道 ARN）。对于像上面示例这样的后端服务，这会导致 IAM 策略，如下例所示：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "chime:SendChannelMessage",
        "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/*"
      ]
    }
  ]
}
```



```
}
```

记下 Resource 章节中的 AppInstanceUser ARN 和频道 ARN。此 IAM 策略示例向后端服务授予以 ID 为 “” 的用户身份进行 API 调用的权限back-end-worker。如果您希望后端服务能够调用使用应用程序的用户，请将 app_instance_user_arn 更改为 arn:aws:chime:region:aws_account_id:app-instance/app_instance_id/user/*。

了解隐式 API 授权

除了 IAM 策略外，Amazon Chime SDK 消息传递 API 还具有隐式权限。例如，AppInstanceUser 只能发送消息或列出用户所属频道的频道成员资格。其中一个例外情况是提升为 AppInstanceAdmin 的 AppInstanceUser。默认情况下，管理员有权访问应用程序中的所有频道。对于大多数用例，只有包含重要业务逻辑的后端服务才需要此功能。

以下 CLI 命令将后端用户提升为管理员。

```
aws chime-sdk-identity create-app-instance-admin \  
  --app-instance-admin-arn "app_instance_user_arn" \  
  --app-instance-arn "app_instance_arn"
```

发送和列出频道消息

以下 CLI 命令发送频道消息。

```
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 CLI 命令按反向的时间顺序列出频道消息。

- 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"
```

对最终用户客户端应用程序进行身份验证

您还可以通过最终用户客户端应用程序运行 Amazon Chime 软件开发工具包消息。 [从后端服务发出 SDK 调用](#) 解释了如何进行 API 调用，例如 create-channel send-channel-message、和。 list-channel-messages 浏览器和移动应用程序等最终用户客户端应用程序会进行相同的 API 调用。客户端应用程序还可以通过连接 WebSocket，以接收其所属频道的消息和事件的实时更新。本节介绍如何向限于特定应用程序实例用户的客户端应用程序提供 IAM 凭证。最终用户获得这些凭证后，可以进行 [从后端服务发出 SDK 调用](#) 中所示的 API 调用。要查看客户端应用程序的完整演示，请参阅 <https://github.com/aws-samples/amazon-chime-sdk/tree/main/apps/ch> at。有关接收来自客户端应用程序所属频道的实时消息的更多信息，请参阅 [WebSockets 用于接收消息](#)。

向最终用户提供 IAM 凭证

Amazon Chime SDK 消息传递与 AWS 身份和访问管理 (IAM) 策略进行原生集成，用于对传入的请求进行身份验证。IAM 策略定义了个人用户可以做什么。可以制定 IAM 策略来为用例提供范围有限的凭证。有关为 Amazon Chime SDK 消息传递用户创建策略的更多信息，请参阅 [IAM 角色示例](#)。

如果您已有身份提供商，则可以使用以下选项将现有身份与 Amazon Chime SDK 消息传递集成。

- 您可以使用现有的身份提供商对用户进行身份验证，然后将身份验证服务与 AWS 安全令牌服务 (STS) 集成，为客户创建自己的凭证自动售货服务。STS 提供用于担任 IAM 角色的 API。
- [如果您已经有兼容 SAML 或 OpenID 的身份提供商，我们建议您使用 Amazon Cognito 身份池，它可以抽象出对 STS SAML 的调用 AWS 。 AssumeRoleWith AssumeRoleWithWebIdentity](#) Amazon Cognito 集成了 OpenID、SAML 和公共身份提供商，如 Facebook、Login with Amazon、Google 和 Sign in with Apple。

如果您没有身份提供商，则可以开始使用 Amazon Cognito 用户群体。有关如何将 Amazon Cognito 与 Amazon Chime SDK 消息传递功能配合使用的示例，请参阅 [使用 Amazon Chime SDK 消息传递在应用程序中构建聊天功能](#)。

或者，您可以使用 [AWS STS](#) 创建自己的凭证售卖服务或构建自己的身份提供商。

使用 STS 出售凭证

[如果您已经拥有 IDP \(例如 ActiveDirectory LDAP\)，并且想要实现自定义凭证自动售货服务，或者向未经身份验证的会议与会者授予聊天权限，则可以使用 STS API。 AWS AssumeRole](#) 为此，您需要

先创建一个 Amazon Chime 软件开发工具包消息传送软件开发工具包角色。有关创建该角色的更多信息，请参阅[创建将权限委派给 IAM 用户的角色](#)。

IAM 角色将有权访问您的应用程序将使用的 Amazon Chime SDK 消息传递操作，如下所示：

```
{
  "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/*"
    ]
}
]
}

```

在本示例中，将此角色称为ChimeMessagingSampleAppUserRole。

请注意用户 ARN 资源中ChimeMessagingSampleAppUserRole策略\${my_application_user_id}中的会话标签。此会话标签在 [AssumeRole](#) API 调用中进行了参数化，以将返回的凭证限制为单个用户的权限。

[AssumeRole](#)和 [TagSession](#) API 是使用已获得认证的 IAM 实体（例如 IAM 用户）调用的。这些 API 也可以由不同的 IAM 角色调用，例如[AWS Lambda 运行角色](#)。该 IAM 身份必须具有调用AssumeRole和TagSession开启的权限ChimeMessagingSampleAppUserRole。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ],
      "Resource":
        "arn:aws:iam::my_aws_account_id:role/ChimeMessagingSampleAppUserRole"
    }
  ]
}

```

在本示例中，将此角色称为ChimeSampleAppServerRole。

您需要使用信任策略ChimeMessagingSampleAppUserRole进行设置，ChimeMessagingSampleAppServerRole以允许对其调用 [STS AssumeRole API](#)。有关将 IAM 角色与信任策略结合使用的更多信息，请参阅[如何将 IAM 角色与信任策略结合使用](#)。您可以使用 AWS IAM 角色控制台将此策略添加到ChimeMessagingSampleAppUserRole。以下示例介绍了一种典型的信任关系。

```

{

```

```

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

"AWS": "arn:aws:iam::my_aws_account_id:role/ChimeMessagingSampleAppServerRole"
        }
        "Action": "sts:AssumeRole"
      }
    ]
  }
}

```

在示例部署中，A [amazon EC2](#) 实例 AWS Lambda 或使用启动 `ChimeMessagingSampleAppServerRole`。然后，服务器：

1. 对客户端接收凭证的请求执行任何特定于应用程序的授权。
2. 在 `ChimeMessagingSampleAppUserRole` 上调用 STS `AssumeRole`，并使用标签来参数化 `${aws:PrincipalTag/my_applications_user_id}`。
3. 将 `AssumeRole` 调用中返回的凭证转发给用户。

以下示例介绍了在步骤 2 中扮演角色的 CLI 命令：

```

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

```

创建频道

您和您的最终用户可以创建频道。创建后，您或您的最终用户还需要向频道添加成员。用于创建频道的示例代码可在[上的示例应用程序](#)中找到 GitHub。

关于创建频道和添加成员的更多信息，请参阅：

- [CreateChannel](#)
- [CreateChannelMembership](#)

发送消息

使用 [SendChannelMessage](#) API 发送消息。示例代码可在 [上的示例应用程序](#) 中找到 GitHub。

使用 ExpirationSettings

创建 AppInstanceUser 或 Channel 时，您可以使用 ExpirationSettings 将这些资源配置为自动删除。ExpirationSettings 有助于降低存储成本并防止出现超出资源限制的问题。例如，您可以在 7 天后删除未使用的频道，或者删除仅为测试目的调用的 AppInstanceUser。

对于 AppInstanceUser，您可以根据用户创建时间来指定过期时间。对于 Channel，您可以根据频道的创建时间或上次消息时间来指定过期时间。后者允许您使用消息事件来自定义自动删除。

Important

资源过期后不久，ExpirationSettings 启动后台进程以删除该资源。该过程通常需要 6 个小时，但时间可能会有所不同。

已过期且尚未删除的 AppInstanceUsers 和 Channels 仍显示为有效且处于活动状态。您可以更新或删除他们的过期设置，系统会接受您的更改。

主题

- [配置 ExpirationSettings](#)
- [删除过期资源的 AWS CloudTrail 事件](#)

配置 ExpirationSettings

以下各节说明如何配置 AppInstanceUser 或 Channel 的 ExpirationSettings。

在创建资源时配置 ExpirationSettings

您可在运行 CreateAppInstanceUser 或 CreateChannel API 时配置 ExpirationSettings。如果设置 ExpirationSettings 参数，您必须授予以下 IAM 权限：

- 创建 AppInstanceUser 时的 `chime:PutAppInstanceUserExpirationSettings`
- 创建 Channel 时的 `chime:PutChannelExpirationSettings`。

以下示例使用 AWS CLI 创建在一天后过期的 AppInstanceUser。

```
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"  
}'
```

以下示例使用 AWS CLI 创建在上次收到消息一天后过期的 Channel。

```
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"  
}'
```

使用 Put API 配置 ExpirationSettings

您还可以使用 PutAppInstanceUserExpirationSettings 和 PutChannelExpirationSettings 创建、更新和删除 ExpirationSettings。

以下示例介绍了如何使用 AWS CLI 更新 AppInstanceUser 的 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"  
}'
```

以下示例介绍了如何使用 AWS CLI 删除频道的 ExpirationSettings。

```
aws chime-sdk-messaging put-channel-expiration-settings \  
--chime-bearer "app_instance_user_arn" \  
--channel-arn "channel_arn"
```

删除过期资源的 AWS CloudTrail 事件

系统删除过期的资源后，会向 AWS CloudTrail 发送 `ExpireAppInstanceUser` 或 `ExpireChannel` 事件。事件的类型取决于已删除资产的类型。

以下示例介绍了 `AppInstanceUser` 事件。

```
{
  "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 用于接收消息

您可以使用 [Amazon Chime JS 软件开发工具包](#) 通过接收消息 WebSockets，也可以使用您选择的 WebSocket 客户端库。

按照列出的顺序遵循以下主题开始使用 WebSockets：

主题

- [定义 IAM 策略](#)
- [检索终端节点](#)
- [建立连接](#)
- [使用预提取来传送频道详情](#)
- [处理事件](#)

定义 IAM 策略

首先，定义一个授予您建立 WebSocket 连接权限的 IAM 策略。以下示例策略 AppInstanceUser 授予建立 WebSocket 连接的权限。

```
"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": [
      "*"
    ]
  }
]
```

```
}  
]  
}
```

检索终端节点

以下步骤说明如何检索 WebSocket 连接中使用的端点。

1. 使用 [GetMessagingSessionEndpoint](#) API 检索 WebSocket 终端节点。
2. 使用 [GetMessagingSessionEndpoint](#) API 返回的网址来构造签名版本 4 的签名 WebSocket 网址。如果您需要帮助，可以按照 [建立连接](#) 中的说明进行操作。

Note

WebSocket URL 的格式如下：*id.region*.ws-messaging.chime.aws

建立连接

检索终端节点后，您可以使用 Connect API 建立与 Amazon Chime SDK 后端服务器的 WebSocket 连接并接收 AppInstanceUser 消息。您可以使用 AWS 签名版本 4 对请求进行签名。有关对请求进行签名的更多信息，请参阅[利用签名版本 4 对 AWS 请求进行签名](#)。

Note

要检索终端节点，您可以调用 [GetMessagingSessionEndpoint](#) API。您可以使用自己选择的 WebSocket 客户端库来连接终端节点。

请求语法

```
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 请求参数

所有 URI 请求查询参数都必须经过 URL 编码。

X-Amz-Algorithm

标识 AWS 签名版本以及用于计算签名的算法。Amazon Chime SDK 仅支持 AWS 签名版本 4 身份验证，因此其值是 AWS4-HMAC-SHA256。

X-Amz-Credential

除了访问密钥 ID 外，该参数还提供签名有效的范围（AWS 区域和服务）。该值必须与您在签名计算中使用的范围匹配。该参数值的一般形式为：

```
<yourAccessKeyId>/<date>/<awsRegion>/<awsService >/aws4_request
```

例如：

```
AKIAIOSFODNN7EXAMPLE/20201214/us-east-1/chime/aws4_request
```

X-Amz-Date

日期和时间格式必须遵循 ISO 8601 标准，并且您必须按照 yyyyMMdTTHmmssZ 将其格式化。例如，您必须将 08/01/2020 15:32:41.982-700 转换为协调世界时 (UTC) 并将其提交为 20200801T083241Z。

X-Amz-Signed-Headers

列出用于计算签名的标头。签名计算中需要以下标头：

- HTTP 主机标头。
- 您计划添加到请求的任何 x-amz-* 标头。

Note

为了提高安全性，签署计划在请求中包含的所有请求标头。

X-Amz-Signatures

提供签名以验证您的请求。此签名必须与 Amazon Chime SDK 计算的签名相匹配。如果没有，Amazon Chime SDK 会拒绝该请求。例如，733255ef022bec3f2a8701cd61d4b371f3f28c9f19EXAMPLEd48d5193d7。

X-Amz-Security-Token

如果使用来自 Security Token Service 的凭证，则为可选凭证参数。有关该服务的更多信息，请参阅 <https://docs.aws.amazon.com/STS/latest/APIReference/>。

SessionId

表示正在建立的 WebSocket 连接的唯一 ID。

UserArn

表示尝试建立连接的 AppInstanceUser 的身份。该值应为 AppInstanceUser 的 ARN。例如 `arn:aws:chime:us%2Deast%2D1:123456789012:app%2Dinstance/694d2099%2Dcb1e%2D463e%2D9d64%2D697ff5b8950e/user/johndoe`

使用预提取来传送频道详情

建立 WebSocket 连接时，可以在查询参数中指定 `prefetch-on=connect` 送达 CHANNEL_DETAILS 事件。Connect API 附带了预提取功能，该功能使用户无需额外的 API 调用即可查看丰富的聊天视图。用户可以：

- 查看上一条频道消息的预览及其时间戳。
- 查看频道的成员。
- 查看频道的未读标记。

用户使用指定的预提取参数进行连接后，该用户会收到会话已建立事件，该事件表示连接已建立。然后，用户最多可接收 50 个 CHANNEL_DETAILS 事件。如果用户的频道少于 50 个，则 Connect API 会通过 CHANNEL_DETAILS 事件预提取所有频道。如果用户有超过 50 个频道，则 API 会预提取包含未读消息的前 50 个频道和最新 LastMessageTimestamp 值。CHANNEL_DETAILS 事件按随机顺序到达，您会收到所有 50 个频道的事件。

此外，预提取还会返回 ChannelMessages 和 ChannelMemberships 的以下内容：

- ChannelMessages：[ChannelMessageSummary](#) 对象列表，由 CreatedTimestamp 按降序排序。仅包括用户可见的最新 20 条消息。如果频道中存在当前用户看不到的定向消息，则返回的消息可能

少于 20 条。ChannelMessagesHasMore 布尔值将设置为 true 以表示还有更多消息。软限额，可在 AWS 账户层面进行调整。

- ChannelMemberships : [ChannelMembershipSummary](#) 对象列表。最多包含 30 个频道成员。软限额，可在 AWS 账户层面进行调整。

此示例介绍了如何使用 prefetch-on=connect :

```
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
```

此示例介绍了一个频道的响应。您将收到所有 50 个频道的回复。

```
{
  "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
  })
}
```

处理事件

对于在建立连接后才可接收消息的 AppInstanceUser，您必须将其添加到频道中。为此，请使用 [CreateChannelMembershipAPI](#)。

Note

AppInstanceUser 始终接收其所属各频道的消息。当 AppInstance 用户断开连接时，消息传递将停止。

除非您使用 [CreateChannelMembership](#) API 明确添加消息，否则 AppInstanceAdmin 和 a ChannelModerator 不会在频道上接收消息。

下面的主题说明了如何处理事件。

主题

- [了解消息结构](#)
- [处理断开连接](#)

了解消息结构

您收到的每 WebSocket 封邮件都遵循以下格式：

```
{
  "Headers": {"key": "value"},
  "Payload": "{\"key\": \"value\"}"
}
```

标头

Amazon Chime SDK 消息传递使用以下标头键：

- x-amz-chime-event-type
- x-amz-chime-message-type
- x-amz-chime-event-reason

下一节列出并描述了标头的可能值和有效负载。

有效负载

Websocket 消息返回 JSON 字符串。JSON 字符串的结构取决于 x-amz-event-type 标头。下表列出了可能的 x-amz-chime-event-type 值和有效负载：

EventType	有效负载格式	
SESSION_ESTABLISHED	不适用。此消息在用户连接到后发送一次 WebSocket。它表示，只要频道 WebSocket 保持打开状态，就保证在用户收到SESSION_ESTABLISHED 消息后到达的频道上的任何消息或事件都将传递给用户。	
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	有效负载格式	
CHANNEL_DETAILS	<p>频道</p> <p>ChannelSummary 对象。</p> <p>ChannelMessages</p> <p>ChannelMessageSummary 对象列表，按降CreatedTimestamp 排序。包括最新的 20 条消息，但您可以在 AWS 账户级别调整该限制。</p> <p>ChannelMemberships</p> <p>ChannelMembershipSummary 对象的列表。最多返回 30 个频道成员，但您可以在 AWS 账户级别调整该限制。</p> <p>ReadMarkerTimestamp</p> <p>AppInstanceUser 上次将频道标记为已读的时间。</p>	
UPDATE_CHANNEL	Channel	
DELETE_CHANNEL		
BATCH_CREATE_CHANNEL_MEMBERSHIP	BatchChannelMembership	
CREATE_CHANNEL_MEMBERSHIP	ChannelMembership	
DELETE_CHANNEL_MEMBERSHIP		

EventType	有效负载格式	
UPDATE_CHANNEL_MEMBERSHIP		

x-amz-chime-message-类型

下表列出了 x-amz-chime-message-type 消息类型。

消息类型	描述
STANDARD	在 WebSocket 收到标准频道消息时发送。
CONTROL	在 WebSocket 收到控制频道消息时发送。
SYSTEM	Amazon Chime SDK 消息传递发送的所有其他 WebSocket 消息。

x-amz-chime-event-原因

这是特定用例支持的可选标头。标头提供有关接收特定事件的原因的信息。

事件原因	描述
subchannel_DELETED	弹性频道监管人收到的 DELETE_CHANNEL_MEMBERSHIP 事件。只有在成员资格平衡删除他们所属的子频道后，监管人才能看见。

处理断开连接

WebSocket 可能会因网络连接变化或凭证过期而断开连接。在您打开 WebSocket 后，Amazon Chime SDK 会定期向消息客户端发送 ping，以确保其仍处于连接状态。如果连接关闭，客户端会收到 WebSocket 关闭代码。客户端可以尝试也可以不尝试重新连接，具体取决于关闭代码。下表显示了客户端可用于重新连接的关闭代码。

对于关闭代码 1000 至 4000，仅针对以下消息重新连接：

关闭代码	是否可以重新连接	Reason
1001	是	正常关闭
1006	是	异常关闭
1011	是	内部服务器错误
1012	是	服务重新启动
1013	是	请稍后再试
1014	是	服务器充当网关或代理，收到来自上游服务器的无效响应。这与 HTTP 状态代码 502 类似。

对于 4XXX 代码，请务必重新连接，但以下消息除外：

关闭代码	是否可以重新连接	Reason
4002	否	客户端已启动
4003	否	禁止
4401	否	未授权

当应用程序使用关闭代码来重新连接时，应用程序应执行以下操作：

1. 再次调用 [GetMessagingSessionEndpoint](#)，以获取新的基本 URL。
2. 如果 IAM 凭证已过期，请刷新它们。
3. 通过 WebSocket 进行连接。

如果您使用 `amazon-chime-sdk-js` 库，那么在您实现 [needsRefresh\(\)](#) 属性和 [refresh\(\)](#) 方法后，就会为您处理这个问题。有关工作示例，请参阅 <https://github.com/aws-samples/amazon-chime-sdk/blob/dc11c4c76c78d28f618577706bba2087919a5635/apps/chat/src/providers/AuthProvider.jsx#L93-L101>。

配置附件

Amazon Chime SDK 允许您将消息附件作为消息元数据存储在自己的存储空间。Amazon Simple Storage Service (S3) 是最简单的附件入门方式。

使用 S3 处理附件

1. 创建 S3 存储桶以存储附件。
2. 为存储桶创建 IAM 策略，允许 Amazon Chime SDK 用户从您的 S3 存储桶上传、下载和删除附件。
3. 创建 IAM 角色供您的身份提供商使用，向用户售卖凭证以获取附件。

[示例应用程序](#) 提供了一个示例，说明如何使用 Amazon S3、Amazon Cognito 和 Amazon Chime SDK 执行此操作。

了解系统消息

Amazon Chime SDK 会向所有连接的客户端发送系统消息，告知频道中发生的事件。事件包括：

- UPDATE_CHANNEL：此事件表示对频道详细信息所做的任何更新，例如名称或元数据。
- DELETE_CHANNEL：此事件表示该频道及其所有数据（包括消息、成员资格、监管人和禁令）将被删除。
- CREATE_CHANNEL_MEMBERSHIP：此事件表示已将特定 AppInstanceUser 添加为该频道的成员。该事件还包含新 AppInstanceUser 的详细信息。
- DELETE_CHANNEL_MEMBERSHIP：此事件表示 AppInstanceUser 已从频道中移除。该事件还包含已删除的 AppInstanceUser 详细信息。
- UPDATE_CHANNEL_MEMBERSHIP：此事件仅适用于弹性频道。该事件表示成员资格平衡已将 AppInstanceUser 从一个子频道转移到另一个子频道。该事件还包含 AppInstanceUser 详细信息，以及有关 AppInstanceUser 转移到的子频道的信息。

IAM 角色示例

要让用户访问 Amazon Chime SDK 消息传递功能，您必须定义一个 IAM 角色和策略，以便在用户登录时向他们提供凭证。IAM 策略定义了用户可以访问的资源。

本节中的示例提供了您可以根据需要进行调整的基本策略。有关策略工作原理的更多信息，请参阅 [从后端服务发出 SDK 调用](#)。

此示例介绍了针对使用 Amazon Chime SDK 消息构建应用程序的开发人员的策略。

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

```
}
```

此示例介绍了一项允许用户访问 Amazon Chime SDK 用户操作的策略。

```
{
  "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/*"
    ]
}
]
}
}

```

此示例介绍了一项策略，该策略允许用户尽量减少对 Amazon Chime SDK 用户操作的访问权限。

```

{
  "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/*"
    ]
}

```

此示例显示了为建立 WebSocket 连接的策略 `AppInstanceUser`。有关 WebSocket 连接的更多信息，请参阅 [WebSockets 用于接收消息](#)。

```

{
  "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}"
      ]
    }
  ]
}

```

了解按角色划分的授权

本主题中的表格列出了应用程序实例用户可以运行的操作，具体取决于他们的角色。

图例

- 已允许：如果在 IAM policy 中指定了正确的操作/资源上下文，则可以成功运行该上下文。
- 已允许，但有限制：如果在 IAM policy 中指定了正确的操作/资源上下文，则必须满足某些条件才能成功运行操作。
- 已拒绝：即使在 IAM policy 中指定了正确的操作/资源上下文，它仍会被后端阻止。

主题

- [AppInstanceAdmin](#)

- [ChannelModerator](#)
- [成员](#)
- [非成员](#)

AppInstanceAdmin

应用程序实例管理员可以对他们作为管理员的应用程序实例中的频道执行操作。

API 名称	已允许或已拒绝	注意
UpdateChannel	已允许，但有限制	设置后无法更新 ElasticChannelConfiguration
DeleteChannel	已允许	
DescribeChannel	已允许	
ListChannel	已允许	
ListChannelMembershipsForAppInstanceUser	已允许	您也可以使用另一个 AppInstanceUser 填充 AppInstanceUserArn 。
DescribeChannelMembershipForAppInstanceUser	已允许	您也可以使用另一个 AppInstanceUser 填充 AppInstanceUserArn 。
ListChannelsModeratedByAppInstanceUser	已允许	您也可以使用另一个 AppInstanceUser 填充 AppInstanceUserArn 。
DescribeChannelModeratedByAppInstanceUser	已允许	您也可以使用另一个 AppInstanceUser 填充 AppInstanceUserArn 。不允许使用弹性频道。
CreateChannelMembership	已允许	

API 名称	已允许或已拒绝	注意
DescribeChannelMembership	已允许	
ListChannelMemberships	已允许	
DeleteChannelMembership	已允许	
SendChannelMessage	已允许，但有限制	您首先需要使用 CreateChannelMembership 为自己创建成员资格，然后调用 API。
GetChannelMessage	已允许	
ListChannelMessage	已允许	
DeleteChannelMessage	已允许	
RedactChannelMessage	已允许	
UpdateChannelMessage	已允许，但有限制	您只能编辑自己的消息。
CreateChannelModerator	已允许	
DeleteChannelModerator	已允许	
DescribeChannelModerator	已允许	
ListChannelModerator	已允许	
CreateChannelBan	已允许，但有限制	您封禁的 AppInstanceUser 不能是该频道的 AppInstanceAdmin 或监管人。

API 名称	已允许或已拒绝	注意
DeleteChannelBan	已允许，但有限制	
DescribeChannelBan	已允许	
ListChannelBan	已允许	
UpdateChannelReadMarker	已允许，但有限制	对于非弹性频道，您需要先使用 CreateChannelMembership 为自己创建成员资格，然后调用 API。 不允许用于弹性频道。
GetChannelMessage	已允许，但有限制	仅允许已发送的消息。除非您是消息发送者，否则不允许按频道流处理消息。
ListChannelMessages	已允许	
DeleteChannelMessage	已允许，但有限制	仅允许已发送的消息。
RedactChannelMessage	已允许，但有限制	仅允许已发送的消息。
UpdateChannelMessage	已允许，但有限制	您只能编辑自己发送的消息。
AssociateChannelFlow	已允许	
DisassociateChannelFlow	已允许	
GetChannelMessageStatus	已允许，但有限制	您只能获取自己的消息状态。
ListSubChannels	已允许	

ChannelModerator

频道监管人只能在他们拥有监管人角色的频道上执行操作。

Note

作为 AppInstanceAdmin 的监管人可以在该角色允许的频道上执行操作。

API 名称	已允许或已拒绝	注意
UpdateChannel	已允许	设置后无法更新 ElasticChannelConfiguration
DeleteChannel	已允许	
DescribeChannel	已允许，但有限制	您只能获取公共频道的详细信息。
ListChannel	已允许，但有限制	您只能获取公共频道的详细信息。
ListChannelMembershipsForAppInstanceUser	已允许，但有限制	您只能使用您的 ARN 作为 AppInstanceUserArn 值。
DescribeChannelMembershipForAppInstanceUser	已允许，但有限制	您只能使用您的 ARN 作为 AppInstanceUserArn 值。
ListChannelsModeratedByAppInstanceUser	已允许，但有限制	您只能使用您的 ARN 作为 AppInstanceUserArn 值。
DescribeChannelModeratedByAppInstanceUser	已允许，但有限制	您也可以使用另一个 AppInstanceUser 填充 AppInstanceUserArn 。
CreateChannelMembership	已允许	
DescribeChannelMembership	已允许	


API 名称	已允许或已拒绝	注意
ListChannelMembers hip	已允许	
DeleteChannelMembe rship	已允许	
SendChannelMessage	已允许，但有限制	您需要先使用 CreateChannelMembership 为自己创建成员资格，然后调用 API。
GetChannelMessage	已允许	
ListChannelMessage	已允许	
DeleteChannelMessage	已拒绝	
RedactChannelMessage	已允许	
UpdateChannelMessage	已允许，但有限制	您只能更新自己的消息。
CreateChannelModer ator	已允许	您需要先使用 CreateChannelMembership 为自己创建成员资格，然后调用 API。
DeleteChannelModer ator	已允许	
DescribeChannelMod erator	已允许	
ListChannelModerator	已允许	
CreateChannelBan	已允许，但有限制	您封禁的 AppInstanceUser 不能是该频道的 AppInstanceAdmin 或监管人。
DeleteChannelBan	已允许，但有限制	

API 名称	已允许或已拒绝	注意
DescribeChannelBan	已允许	
ListChannelBan	已允许	
UpdateChannelReadMarker	已允许，但有限制	对于非弹性频道，您需要先使用 CreateChannelMembership 为自己创建成员资格，然后调用 API。 不允许用于弹性频道。
GetChannelMessage	已允许，但有限制	仅允许已发送的消息。除非您是消息发送者，否则不允许按频道流处理消息。
ListChannelMessages	已允许	
DeleteChannelMessage	已拒绝	
RedactChannelMessage	已允许，但有限制	仅允许已发送的消息。
UpdateChannelMessage	已允许，但有限制	您只能编辑自己发送的消息。
AssociateChannelFlow	已允许	
DisassociateChannelFlow	已允许	
GetChannelMessageStatus	已允许，但有限制	您只能获取自己的消息状态。
ListSubChannels	已允许	

成员

如果通过 [CreateChannelMembership](#) API 将其添加到频道中，则 AppInstanceUser 成为该频道的成员。

成员只能在他们所属的频道上执行操作。

 Note

作为 AppInstanceAdmin 或 ChannelModerator 的成员可以在这两个角色允许的频道上执行操作。

API 名称	已允许或已拒绝	注意
UpdateChannel	已拒绝	
DeleteChannel	已拒绝	
DescribeChannel	已允许，但有限制	您只能获取公共频道的详细信息。
ListChannel	已允许，但有限制	您只能获取公共频道的详细信息。
ListChannelMembershipsForAppInstanceUser	已允许，但有限制	您只能使用您的 ARN 作为 AppInstanceUserArn 值。
DescribeChannelMembershipForAppInstanceUser	已允许，但有限制	您只能使用您的 ARN 作为 AppInstanceUserArn 值。
ListChannelsModeratedByAppInstanceUser	已允许，但有限制	您只能使用您的 ARN 作为 AppInstanceUserArn 值。
DescribeChannelModeratedByAppInstanceUser	已允许，但有限制	您也可以用另一个 AppInstanceUser 填充 AppInstanceUserArn 。 不允许用于弹性频道。

API 名称	已允许或已拒绝	注意
CreateChannelMembership	已允许，但有限制	您只能为 不受限 的频道添加其他成员。
DescribeChannelMembership	已允许	
ListChannelMemberships	已允许	
DeleteChannelMembership	已允许	
SendChannelMessage	已允许	
GetChannelMessage	已允许	
ListChannelMessage	已允许	
DeleteChannelMessage	已拒绝	
RedactChannelMessage	已允许，但有限制	您只能编辑自己的消息。
UpdateChannelMessage	已允许，但有限制	您只能更新自己的消息。
CreateChannelModerator	已拒绝	
DeleteChannelModerator	已拒绝	
DescribeChannelModerator	已拒绝	
ListChannelModerator	已拒绝	
CreateChannelBan	已拒绝	
DeleteChannelBan	已拒绝	

API 名称	已允许或已拒绝	注意
DescribeChannelBan	已拒绝	
ListChannelBan	已拒绝	
UpdateChannelReadMarker	已允许，但有限制	不允许用于弹性频道。
GetChannelMessage	已允许，但有限制	仅允许已发送的消息。除非您是消息发送者，否则不允许按频道流处理消息。
ListChannelMessages	已允许	
DeleteChannelMessage	已允许，但有限制	仅允许已发送的消息。
RedactChannelMessage	已允许，但有限制	仅允许已发送的消息。
UpdateChannelMessage	已允许，但有限制	您只能编辑自己发送的消息。
AssociateChannelFlow	已拒绝	
DisassociateChannelFlow	已拒绝	
GetChannelMessageStatus	已允许，但有限制	您只能获取自己的消息状态。
Listsubchannels	已拒绝	

非成员

非成员是普通 `AppInstanceUser`，除非您使用 [CreateChannelMembership](#) API 添加他们，否则他们无法执行任何与频道相关的操作。

Note

作为 AppInstanceAdmin 或 ChannelModerator 的非成员可以执行这两个角色允许的频道相关操作。

API 名称	已允许或已拒绝	注意
UpdateChannel	已拒绝	
DeleteChannel	已拒绝	
DescribeChannel	已允许，但有限制	您只能获取公共频道的详细信息。
ListChannel	已允许，但有限制	您只能获取公共频道的详细信息。
ListChannelMembershipsForAppInstanceUser	已允许，但有限制	您只能使用您的 ARN 作为 AppInstanceUserArn 值。
DescribeChannelMembershipForAppInstanceUser	已允许，但有限制	您也可以用另一个 AppInstanceUser 填充 AppInstanceArn 。 不允许用于弹性频道。
ListChannelsModeratedByAppInstanceUser	已允许，但有限制	您只能使用您的 ARN 作为 AppInstanceUserArn 值。
DescribeChannelModeratedByAppInstanceUser	已允许，但有限制	您只能使用您的 ARN 作为 AppInstanceUserArn 值。
CreateChannelMembership	已拒绝	

API 名称	已允许或已拒绝	注意
DescribeChannelMembership	已允许，但有限制	您只能获取公共频道的详细信息。
ListChannelMemberships	已允许，但有限制	您只能获取公共频道的详细信息。
DeleteChannelMembership	已拒绝	
SendChannelMessage	已拒绝	
GetChannelMessage	已允许，但有限制	您只能获取公共频道的详细信息。
ListChannelMessages	已允许，但有限制	您只能获取公共频道的详细信息。
DeleteChannelMessage	已拒绝	
RedactChannelMessage	已拒绝	
UpdateChannelMessage	已拒绝	
CreateChannelModerator	已拒绝	
DeleteChannelModerator	已拒绝	
DescribeChannelModerator	已拒绝	
ListChannelModerators	已拒绝	
CreateChannelBan	已拒绝	
DeleteChannelBan	已拒绝	
DescribeChannelBan	已拒绝	

API 名称	已允许或已拒绝	注意
ListChannelBan	已拒绝	
UpdateChannelReadMarker	已拒绝	
GetChannelMessage	已允许，但有限制	仅允许已发送的消息。除非您是消息发送者，否则不允许按频道流处理消息。
ListChannelMessages	已允许，但有限制	
DeleteChannelMessage	已拒绝	已拒绝
RedactChannelMessage	已拒绝	
UpdateChannelMessage	已拒绝	
AssociateChannelFlow	已拒绝	
DisassociateChannelFlow	已拒绝	
GetChannelMessageStatus	已允许，但有限制	您只能获取自己的消息状态。

流式传递消息数据

您可以配置 AppInstance，从而以流的形式接收消息和频道事件等数据。然后，您可以对这些数据做出实时反应。目前，Amazon Chime SDK 消息传递仅接受 Kinesis 直播作为直播目的地。要使用具有此功能的 Kinesis 流，您必须具备以下先决条件：

- Kinesis 直播必须与使用同一个 AWS 账户。AppInstance
- 流必须与 AppInstance 在同一个区域。
- 流名称的前缀以 chime-messaging- 开头。
- 您必须配置至少两个分片。每个分片每秒最多可接收 1MB 的数据，因此请相应地扩展您的数据流。
- 必须启用服务器端加密 (SSE)

配置 Kinesis 流

1. 使用上一节中的先决条件创建一个或多个 Kinesis 流，然后获取 ARN。除了 Amazon Chime 权限外，还要确保来电者还拥有 Kinesis 权限。

以下示例说明如何使用 AWS CLI 创建包含两个分片的 Kinesis 流，以及如何启用 SSE。

```
aws kinesys create-stream --stream-name chime-messaging-unique-name --
shard-count 2
```

```
aws kinesys start-stream-encryption --stream-name chime-messaging-
unique-name --encryption-type KMS --key-id "alias/aws/kinesis"
```

2. 通过调用 [PutMessagingStreamingConfigurations](#) API 配置直播。

您可以配置两种数据类型中的一种或两种，也可以为它们选择相同的流或单独的流。

以下示例说明如何使用 AWS CLI 配置 `appinstance` 为流式传输 `ChannelMessage` 和 `Channel` 数据类型。

```
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
```

数据类型的作用域如下：

DataType	生成的事件类型
ChannelMessage	CREATE_CHANNEL_MES SAGE
	REDACT_CHANNEL_MES SAGE

DataType	生成的事件类型	
	UPDATE_CHANNEL_MESSAGE	
	DELETE_CHANNEL_MESSAGE	
Channel	CREATE_CHANNEL	
	CREATE_SUB_CHANNEL	
	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. 开始从您配置的 Kinesis 流中读取数据。

Note

在配置流之前发送的任何事件都不会发送到您的 Kinesis 流中。

Data format (数据格式)

Kinesis 以 JSON 格式输出包含以下字段的记录 : EventType 和 Payload。有效负载格式取决于 EventType。下表列出了事件类型及其对应的负载格式。

EventType	有效负载格式
CREATE_CHANNEL_MESSAGE	频道消息
REDACT_CHANNEL_MESSAGE	
UPDATE_CHANNEL_MESSAGE	
DELETE_CHANNEL_MESSAGE	
CREATE_CHANNEL	Channel
UPDATE_CHANNEL	
DELETE_CHANNEL	
UPDATE_CHANNEL_EXPIRATION_SETTINGS	
CREATE_CHANNEL_MEMBERSHIP	ChannelMembership

EventType	有效负载格式	
DELETE_CHANNEL_MEMBERSHIP		
CREATE_CHANNEL_BAN	ChannelBan	
DELETE_CHANNEL_BAN		
CREATE_CHANNEL_MODERATOR	ChannelModerator	
DELETE_CHANNEL_MODERATOR		
CREATE_SUB_CHANNEL	channelARN	
DELETE_SUB_CHANNEL	SubChannelId	

使用弹性频道举办实时事件

弹性频道支持拥有多达 100 万成员的大规模聊天体验。典型用途包括观看体育赛事的各方或政治事件。您只能在美国东部（弗吉尼亚州北部）区域中使用弹性频道。

弹性频道由具有通用配置的单个或多个子频道以及不同数量（或弹性数量）的子频道组成。该配置还包括子频道中成员的最小和最大阈值。

例如，假设您创建了一个包含 100 个子频道的弹性频道，为子频道设置了 500 个成员的低门槛和 10,000 个成员的高门槛。当用户加入此示例频道时，系统会自动将他们分配到单个子频道，直到成员人数超过 10,000。此时，系统会创建一个新的子频道，并在其中添加所有新成员。当用户离开时，系统会删除子频道，并将成员分配到其余的子频道。

将受众分成子频道可以让参与者更容易关注对话。监管人的工作量也减少了，因为他们只需要观看一些子频道。此外，监管人可以使用弹性频道提供的内置工具。例如，监管人可以[禁止用户](#)进入频道，[创建监管人](#)，并使用[频道流](#)自动审核频道中的所有消息。

有关 Amazon Chime SDK 消息传递限额的更多信息，请参阅《Amazon Chime SDK 一般参考》中的[消息传递限额](#)。

主题

- [先决条件](#)
- [弹性频道概念](#)
- [其他支持的功能](#)
- [创建弹性频道](#)
- [管理弹性频道成员](#)
- [发送弹性频道消息](#)
- [了解弹性频道中的 WebSocket 系统消息](#)
- [使用 Kinesis 流接收系统消息](#)
- [在我们的演示应用程序中测试弹性频道](#)

先决条件

您必须具备以下各项条件，才能使用弹性频道。

- 了解 Amazon Chime SDK 消息传递功能，例如管理频道以及发送和接收消息。
- 能够调用 Amazon Chime SDK 消息传递 API。

弹性频道概念

要有效地使用弹性频道，您必须了解这些概念。

子频道

弹性频道将其成员划分为称为子频道的逻辑容器。当您向弹性频道添加 `AppInstanceUser` 时，用户就会成为子频道的成员。该用户可以发送和接收消息，但只能与该子频道的其他成员发送和接收消息。系统不允许来自一个子频道的消息出现在其他子频道中。

扩缩

为了支持用户互动，每个子频道都必须满足最低成员资格要求。您在创建弹性频道时提供该值。当用户加入或离开事件时，系统会将成员转移到不同的子频道，这使得整个频道具有“弹性”。子频道运行以下扩缩操作。

- **SCALE_OUT**：当新的弹性频道成员资格申请到来并且所有子频道都已满时，系统会通过创建一个新的子频道，然后将成员资格从现有子频道转移到新的子频道来进行扩展。
- **SCALE_IN**：当子频道的成员人数低于最低要求，而另一个子频道有能力容纳第一个子频道的所有成员时，**SCALE_IN** 事件会转移这些成员资格，然后删除该子频道和所有消息。

Note

如果您需要访问已删除的频道消息，则必须先打开消息流。有关更多信息，请参阅 [流式传递消息数据](#)。

成员传输

当成员资格平衡将 `AppInstanceUser` 从一个子频道转移到另一个子频道时，就会发生这种情况。传输后 `AppInstanceUser` 仍属于弹性频道。但是，新的子频道包含不同的成员资格和消息，因此转移后由 `AppInstanceUser` 发送的消息会送达上述不同成员。成员资格平衡不会影响监管人成员资格。

Note

弹性频道不支持隐藏的成员资格、成员偏好和阅读消息的时间戳。

其他支持的功能

弹性频道还支持这些消息传递功能。

- [预提取](#)
- [频道流](#)

创建弹性频道

您可以使用 [CreateChannel](#) API 中的 `ElasticChannelConfiguration` 字段来创建弹性频道。创建弹性频道后，即可创建频道成员资格。

Note

- 对于非弹性频道，创建频道的 `AppInstanceUser` 将作为成员和监管人自动添加到该频道。对于弹性频道，频道创建者仅作为监管人添加。
- `ElasticChannelConfiguration` 设置后不可更新。
- 您不能将频道从弹性频道更新为非弹性频道，反之亦然。
- 您不能在 [CreateChannel](#) API 请求中纳入成员 ARN 列表。但是，您可以包括监管人 ARN 列表。
- 您不能创建 UNRESTRICTED 类弹性频道。

管理弹性频道成员

[要管理弹性频道中的成员，请使用 `CreateChannelMembership`、`CreateChannelModerator` 和 `CreateChannelBan` API。](#) 以下信息介绍了其使用方法。

频道成员资格数

`CreateChannelMembership` API 创建子频道级别的成员资格。子频道包括监管人和普通成员。

- **监管人**：您可以将监管人添加到多个子频道。这允许监管人在他们所属的每个子频道上发送消息。向子频道添加监管人时，您必须提供 `SubChannelId`。

如果您想自动为新的子频道分配监管人，则可以[启用消息流](#)，监听子频道创建事件，然后创建监管人成员资格以响应上述事件。

最后，您可以从特定子频道或所有子频道中删除监管人。在这两种情况下，您都使用 [DeleteChannelMembership](#) API。要从特定子频道中删除监管人，您需要提供 `SubChannelId`。如果您未提供子频道的 ID，系统会将监管人从所有子频道中移除。最后，您可以使用 [ListSubChannels](#) API 来列出子频道以及每个子频道中的成员数量。

- **普通成员**：占频道成员的大多数。您只能向一个子频道添加普通成员。此外，在创建或删除频道成员资格时，您无法通过 `SubChannelId`，因为是由系统来控制创建成员资格所在的子频道。

频道的监管人数

`CreateChannelModerator` API 在弹性频道级别创建监管人。监管人可以查看所有子频道中的所有消息。当您将普通成员提升为频道监管人时，系统会删除该成员的所有现有频道成员资格。当您降级监管人时，也会发生同样的情况。

频道禁令

CreateChannelBan API 在弹性频道级别创建封禁。被封禁的 AppInstanceUser 不属于任何子频道。当您封禁某个成员时，系统会删除该成员的所有频道成员资格。

发送弹性频道消息

SendChannelMessage API 在子频道级别创建消息。要发送消息，您必须要有一个 subChannelId。您也可以使用 UpdateChannelMessage 和 RedactChannelMessage API 来编辑和删除消息，但在所有情况下，您都必须有一个 subChannelId。

Note

只有当消息发件人属于他们向其发送消息的子频道时，他们才能编辑或编写消息。如果成员资格平衡将成员转移到另一个子频道，则该成员只能编辑或编写他们新的子频道中发送的消息。

了解弹性频道中的 WebSocket 系统消息

Amazon Chime SDK 会向所有连接的客户端发送系统消息，告知频道中发生的事件。下面的列表描述了弹性频道的系统消息。

消息事件

弹性频道的事件有效负载包含该 subChannelId 字段。非弹性频道的负载保持不变。

成员事件

现在，CREATE_CHANNEL_MEMBERSHIP 和 DELETE_CHANNEL_MEMBERSHIP 事件的有效负载中包含了 subChannelId 字段。

弹性频道不支持该 BATCH_CREATE_CHANNEL_MEMBERSHIP 事件。当您调用 [BatchCreateChannelMembership](#) API 时，系统会发送单个 CREATE_CHANNEL_MEMBERSHIP 事件。

现在，您可以使用 UPDATE_CHANNEL_MEMBERSHIP 事件类型来表示成员资格信息的变化。例如，在成员从一个子频道转移到另一个子频道期间，系统会发送一个有效负载中包含新 SubChannelId 的 UPDATE_CHANNEL_MEMBERSHIP 事件，以表明该成员已被转移。

Note

系统仅将 UPDATE_CHANNEL_MEMBERSHIP 事件发送给被转移的成员，而不会发送给子频道的其他成员。为此，我们建议您使用 [ListChannelMemberships](#) API 而不是 WebSocket 来填充频道成员资格名册。有关更多信息，请参阅 [WebSockets 用于接收消息](#)。

使用 Kinesis 流接收系统消息

您可以配置 AppInstance，从而以流的形式接收数据。例如，流可以包含消息、子频道事件和频道事件。

作为其中的一部分，我们支持 CREATE_SUB_CHANNEL 和 DELETE_SUB_CHANNEL 事件。它们表示作为成员资格平衡的一部分何时创建或删除子频道。有关接收数据流的更多信息，请参阅 [流式传递消息数据](#)。

在我们的演示应用程序中测试弹性频道

您可以在 GitHub 上测试所有 Amazon Chime SDK 消息传递功能，网址为 <https://github.com/aws-samples/amazon-chime-sdk/tree/main/apps/chat>。

使用移动推送通知接收消息

您可以将 Amazon Chime SDK 消息传送配置为向移动推送通知渠道发送频道消息。Amazon Chime SDK 需要为推送通知配置 Amazon Pinpoint 应用程序。您的 Amazon Pinpoint 应用程序必须满足以下先决条件：

- 您的 Amazon Pinpoint 应用程序必须至少配置并启用 FCM 或 APNS 频道。
- 您的 Amazon Pinpoint 应用程序必须与您的 Amazon Chime 软件开发工具包应用程序实例位于相同的 AWS 账户和区域。

Note

默认情况下，推送通知渠道的所有成员都会收到推送通知，包括消息发送者。但是，您可以设置过滤规则，防止邮件发送给发件人。有关更多信息，请参阅此部分后面的 [使用筛选规则来筛选消息](#)。

主题

- [创建 Amazon Pinpoint 应用程序](#)
- [创建服务角色](#)
- [将移动设备终端节点注册为应用程序实例用户](#)
- [在启用通知的情况下发送频道消息](#)
- [接收推送通知](#)
- [调试推送通知失败](#)
- [使用筛选规则来筛选消息](#)

创建 Amazon Pinpoint 应用程序

要发送推送通知，Amazon Chime SDK 需要 Amazon Pinpoint 应用程序配置为向您的移动应用程序发送推送。以下步骤说明了如何使用 AWS 控制台创建 Pinpoint 应用程序。

创建 Amazon Pinpoint 应用程序

1. [登录 AWS 管理控制台并打开 Amazon Pinpoint 控制台](https://console.aws.amazon.com/pinpoint/)，网址为 <https://console.aws.amazon.com/pinpoint/>。

如果您是第一次使用 Amazon Pinpoint，您会看到一个介绍服务功能的页面。

2. 在开始使用部分，为您的项目输入一个名称，然后选择创建项目。
3. 在配置功能页面的推送通知旁边，选择配置。
4. 在设置推送通知页面上，切换 Apple Push Notification service (APNs)、Firebase Cloud Messaging (FCM) 或两者，然后填写必填字段。

Important

Amazon Chime SDK 目前仅支持向 APNs 和 FCM 发送推送通知。

5. 完成后，选择保存。
6. 返回 Amazon Pinpoint 控制台 <https://console.aws.amazon.com/pinpoint/> 并记下项目 ID 值。您可以将其用作 Amazon Pinpoint 应用程序的 ARN。

创建服务角色

AWS 使用服务角色向 AWS 服务授予权限，使其可以访问 AWS 资源。附加到服务角色的策略将确定服务可访问的资源以及可使用这些资源执行的操作。您为 Amazon Chime SDK 创建的服务角色允许该服务对您的 Amazon Pinpoint 应用程序进行 SendMessages 调用。

创建服务角色

1. 登录 AWS 管理控制台并打开 IAM 控制台，[网址为 https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/)。
2. 在导航窗格中选择 Policies，然后选择 Create Policy。
3. 选择 JSON 选项卡，然后将以下策略复制到文本框中。请务必将 `project_id` 替换为在上一步中创建的 Amazon Pinpoint 应用程序的编号，并将 `aws_account_id` 使用您的 AWS 账户编号替换。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Action": "mobiletargeting:SendMessages",
    "Resource": "arn:aws:mobiletargeting:region:aws_account_id:apps/project_id/
messages",
    "Effect": "Allow"
  }
}
```

4. 选择下一步：标签。
5. 选择下一步：审核，在名称字段中输入 **AmazonChimePushNotificationPolicy**，然后选择创建策略。
6. 在导航窗格中，选择 Roles（角色），然后选择 Create role（创建角色）。
7. 在创建角色页面上，选择 AWS 服务，打开 Choose a user case 列表并选择 EC2。
8. 选择 Next: Permissions，然后在搜索框中输入 **AmazonChimePushNotificationPolicy** 并选中策略旁边的复选框。
9. 选择下一步：标签。
10. 选择下一步：审核，然后在名称字段中输入 **ServiceRoleForAmazonChimePushNotification**。

Important

您必须使用上面列出的名称。Amazon Chime SDK 只接受该特定名称。

11. 选择创建角色，然后在角色页面上的搜索框中输入 **ServiceRoleForAmazonChimePushNotification**，然后选择匹配的角色。
12. 选择信任关系选项卡，选择编辑信任关系，然后将现有策略替换为以下策略。

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "messaging.chime.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

13. 选择更新信任策略。

Important

通过更改名称、权限策略或信任策略来修改角色可能会破坏推送通知功能。

将移动设备终端节点注册为应用程序实例用户

要接收推送通知，应用程序实例用户必须先使用 [RegisterAppInstanceUserEndpoint](#) API 注册移动设备。他们必须通过能够访问设备操作系统的设备令牌的移动应用程序进行注册。

为确保应用程序实例用户有权访问 ARN 中列出的 Amazon Pinpoint 应用程序，用户必须有权调用 Amazon Pinpoint ARN 上的 `mobiletargeting:GetApp`。否则，Amazon Chime SDK 在调用时会抛出 403 Forbidden 错误。 [RegisterAppInstanceUserEndpoint](#)

此示例介绍了注册终端节点所需的策略。

```
{
  "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"
    }
]
}

```

要注册终端节点

- 使用亚马逊 Pinpoint ARN 和您的设备令牌调用 API。 [RegisterAppInstanceUserEndpoint](#)

在启用通知的情况下发送频道消息

该 [SendChannelMessage](#) API 有一个可选 `PushNotification` 属性，Amazon Chime 软件开发工具包使用该属性来构建要发送到亚马逊 Pinpoint 的推送通知。目前，Amazon Chime SDK 仅支持通知标头和正文字段。

Amazon Chime SDK 还支持 APNs VoIP 推送。要将推送通知作为 APNs VoIP 推送发送，请将 `PushNotification` 属性中的类型设置为 `VOIP`。

接收推送通知

除了频道消息推送通知的标头和正文外，Amazon Chime SDK 还在数据负载中包含频道消息 ID 和频道 ARN。您可以使用该信息来加载完整的频道消息。

以下示例介绍了典型的推送通知负载。

```

{
  "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"
  }
}

```

禁用或筛选推送通知回执

Amazon Chime SDK 提供了多个选项，允许应用程序实例用户控制他们是否希望接收推送通知。

禁用所有推送通知

应用程序实例用户可以通过调用 [UpdateAppInstanceUserEndpoint](#) 并将 AllowMessages 属性设置为，完全禁用推送通知 NONE。

禁用频道的推送通知

应用程序实例用户可以通过在“PushNotification 首选项”字段 NONE 中调 [PutChannelMembershipPreferences](#) 用来禁用特定频道的推送通知。

筛选频道的推送通知

App Instance 用户可以设置过滤规则，这样他们只能使用 [PutChannelMembershipPreferences](#) API 接收特定的推送通知。有关更多信息，请参阅 [使用筛选规则来筛选消息](#)。

调试推送通知失败

Amazon Chime SDK 与亚马逊集成，以便 EventBridge 在推送消息传送失败时通知您。要进一步调试故障，您还可以查看 Amazon Pinpoint 发送的故障 [CloudWatch 指标](#)。

下表列出并描述了传送错误消息。

消息	描述
由于未知错误、异常或故障，请求处理失败。	我们遇到了内部错误。请重试。
未找到指定的资源。AppInstanceUserEndpoint 将被停用。	Amazon Pinpoint 应用程序不存在。
向 Amazon Pinpoint 发送的请求过多。	Amazon Pinpoint 已限制您的外发消息。
无法发送消息。请验证 IAM 权限策略已启用 ServiceRoleForAmazonChimePushNotification。	为 Amazon Chime SDK 创建的角色无权调用 mobiletargeting:SendMessage。请验证角色的 IAM policy。

消息	描述
无法发送消息。请验证 IAM 信任关系 ServiceRoleForAmazonChimePushNotification。	Amazon Chime SDK 没有访问推送通知角色的权限。 请验证 IAM 角色的信任策略是否包含服务主体， <code>messaging.chime.amazonaws.com</code> 。

使用筛选规则来筛选消息

Amazon Chime SDK 支持对应用程序实例用户的通道成员资格设置筛选规则，以限制他们将收到的消息。筛选规则是在通道成员资格上设置的，并将根据消息属性映射运行。消息属性映射必须是字符串键到字符串值的映射。筛选规则支持包含和专属精确字符串匹配。

Important

- Amazon Chime SDK 仅支持将转义的 JSON 字符串作为筛选规则。
- 通知渠道的所有成员都会收到推送通知，包括消息发送者。为了防止这种情况发生，请参阅下面的第一个示例规则。

要对频道成员资格设置筛选规则，请使用 [PutChannelMembershipPreferences](#) API。作为 [SendChannelMessage](#) API 调用的一部分，您可以在频道消息中包含消息属性。

主题

- [筛选规则类型](#)
- [筛选规则限制](#)
- [筛选规则示例](#)

筛选规则类型

Amazon Chime SDK 支持以下筛选规则类型：

- 包含精确字符串匹配
- 专属精确字符串匹配

- 使用 AND 或 OR 的多重筛选规则

筛选规则限制

Amazon Chime SDK 对筛选规则施加了以下限制：

- 我们只支持精确字符串匹配。
- 筛选规则的总大小为 2KB。
- 消息属性的总大小为 1KB。
- OR 筛选规则中最多有五 (5) 个单独的约束。
- 整个筛选规则的最大复杂度为 20。复杂度是根据筛选规则中键值数量的总和计算得出的：

例如，此筛选规则的复杂度为 4。

```
"FilterRule": "{\\"type\\": [\\"anything-but\\": [\\"Room\\"]],\\"mention\\": [\\"Bob\\"]}"
```

我们按如下方式计算该值：

```
Keys = "type" and "mention" - Complexity 2
Values = "Room" and "Bob" - Complexity 2

Total complexity = 4
```

筛选规则示例

以下示例显示了使用通道成员资格首选项和筛选规则的几种方法。

防止邮件发送给发件人

此过滤规则向除消息发送者之外的所有频道成员发送消息。

```
{
  "Preferences": {
    "PushNotifications": {
      "FilterRule": "{\\"type\\": [\\"anything-but\\": [\\"USER_ARM\\"]]}",
      "AllowNotifications": "FILTERED"
    }
  }
}
```

```
}
```

具有上述首选项的应用程序实例用户会收到一条包含以下属性的频道消息：

```
"MessageAttributes": {
  "senderId": {
    "StringValues": ["USER_ARN"]
  }
}
```

包含字符串匹配

此筛选规则允许包含如下消息属性的任何消息：键为“mention”且值为“Bob”。

```
{
  "Preferences": {
    "PushNotifications": {
      "FilterRule": "{\"mention\": [\"Bob\"]}",
      "AllowNotifications": "FILTERED"
    }
  }
}
```

具有上述首选项的应用实例用户会收到一条包含以下消息属性的频道消息：

```
"MessageAttributes": {
  "mention": {
    "StringValues": ["Bob", "Alice"]
  }
}
```

但是，应用程序实例用户不会收到包含以下属性的通道消息：

```
"MessageAttributes": {
  "mention": {
    "StringValues": ["Tom"]
  }
}
```

专属字符串匹配

此筛选规则允许除包含如下属性的消息以外的任何消息：键为“type”且值为“Room”。

```
{
  "Preferences": {
    "PushNotifications": {
      "FilterRule": "{\\"type\\": [{\\"anything-but\\": [\\"Room\\"]}]",
      "AllowNotifications": "FILTERED"
    }
  }
}
```

具有这些首选项的应用实例用户会收到包含以下消息属性的通道消息：

```
"MessageAttributes": {
  "type": {
    "StringValues": ["Conversation"]
  }
}
```

但是，应用程序实例用户看不到包含以下属性的通道消息：

```
"MessageAttributes": {
  "type": {
    "StringValues": ["Room"]
  }
}
```

采用 AND 逻辑的多重筛选规则

当您将筛选规则与 AND 逻辑组合使用时，消息必须满足该筛选的所有筛选条件才能应用。

```
{
  "Preferences": {
    "PushNotifications": {
      "FilterRule": "{\\"type\\": [{\\"anything-but\\": [\\"Room\\"]}],\\"mention\\": [\\"Bob\\"]}",
      "AllowNotifications": "FILTERED"
    }
  }
}
```

具有上述首选项的应用实例用户会收到包含以下消息属性的通道消息：

```
"MessageAttributes": {
  "mention": {
    "StringValues": ["Bob"]
  },
  "type": {
    "StringValues": ["Conversation"]
  }
}
```

采用 OR 逻辑的多重筛选规则

您可以使用 `$or` 来组合筛选规则与 OR 逻辑。当您使用 OR 逻辑时，消息必须满足筛选条件之一才能应用。

```
{
  "Preferences": {
    "PushNotifications": {
      "FilterRule": "{\"$or\": [{\"mention\": [\"Bob\"]}, {\"type\": [\"anything-but\": [\"Room\"]}]}\"",
      "AllowNotifications": "FILTERED"
    }
  }
}
```

具有上述首选项的应用实例用户会收到包含以下消息属性的通道消息：

```
"MessageAttributes": {
  "mention": {
    "StringValues": ["Bob"]
  }
}
```

具有上述首选项的应用实例用户会收到包含以下消息属性的通道消息：

```
"MessageAttributes": {
  "type": {
    "StringValues": ["Conversation"]
  }
}
```

使用服务相关角色

Amazon Chime SDK 使用 AWS Identity and Access Management (IAM) [服务相关角色](#)。服务相关角色是一种独特类型的 IAM 角色，它与 Amazon Chime SDK 直接关联。Amazon Chime SDK 预定义服务相关角色，并且包含该服务代表您调用其他 AWS 服务所需的所有权限。

服务相关角色可更轻松地设置 Amazon Chime SDK，因为您不必手动添加必要权限。Amazon Chime SDK 定义其服务相关角色的权限，除非另外定义，否则只有 Amazon Chime SDK 可以代入该角色。定义的权限包括信任策略和权限策略。不能将该权限策略附加到任何其他 IAM 实体。

只有在首先删除服务相关角色的相关资源后，才能删除该角色。这将保护您的 Amazon Chime SDK 资源，因为您不会无意中删除对资源的访问权限。

有关支持服务相关角色的其他服务的信息，请参阅[与 IAM 配合使用的 AWS 服务](#)。查找在 Service-Linked Role (服务相关角色) 列中具有 Yes (是) 值的服务。选择是与查看该服务文档的链接。

主题

- [为数据流使用服务相关角色](#)

为数据流使用服务相关角色

以下部分说明如何管理数据流的服务相关角色。

本节中的主题

- [服务相关角色权限](#)
- [创建服务相关角色](#)
- [编辑服务相关角色](#)
- [删除服务相关角色使用的资源](#)
- [删除服务相关角色](#)

服务相关角色权限

Amazon Chime SDK 使用名为 `AWSServiceRoleForChimeSDKMessaging` 的服务相关角色。该角色授予访问由 Amazon Chime SDK 使用或托管的 AWS 服务和资源的权限，例如用于数据流的 Kinesis 流。

`AWSServiceRoleForChimeSDKMessaging` 服务相关角色信任以下服务来代入角色：

- messaging.chime.amazonaws.com

角色权限策略允许 Amazon Chime SDK 对指定资源完成以下操作：

- 仅当使用 `kinesis.*.amazonaws.com` 发出请求时 `kms:GenerateDataKey`。
- `kinesis:PutRecord`、`kinesis:PutRecords` 或者 `kinesis:DescribeStream` 仅在以下格式的流上显示：`arn:aws:kinesis:*:*:stream/chime-messaging-*`。

以下示例显示了该策略。

```
{
  "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-*"
      ]
    }
  ]
}
```


您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅 IAM 用户指南中的[服务相关角色权限](#)。

创建服务相关角色

无需手动创建服务相关角色。当您使用 [PutMessagingStreamingConfigurations](#) API 创建数据流配置时，Amazon Chime SDK 会为您创建服务相关角色。

您也可以使用 IAM 控制台为 Amazon Chime SDK 使用案例创建服务相关角色。在 AWS CLI 或 AWS API 中，用 `messaging.chime.amazonaws.com` 服务名称创建一个服务相关角色。有关更多信息，请参阅 IAM 用户指南中的[创建服务相关角色](#)。如果您已删除此角色，则可以重复此过程再次创建。

编辑服务相关角色

创建服务相关角色后，您只能编辑其描述，且使用 IAM 进行编辑。有关更多信息，请参阅 IAM 用户指南中的[编辑服务相关角色](#)。

删除服务相关角色使用的资源

必须先删除服务相关角色使用的所有资源，然后才能使用 IAM 删除该角色。

Note

如果您试图删除 Amazon Chime SDK 正在使用的资源，则删除可能会失败。如果发生这种情况，请等待几分钟后重试。

删除 AmazonChimeServiceChatStreamingAccess 角色使用资源

使用 CLI 调用以下 API，关闭应用实例的数据流功能。

- `aws chime-sdk-messaging delete-messaging-streaming-configurations --app-instance-arn app_instance_arn`

此操作会删除您的应用程序实例的所有流配置。

删除服务相关角色

如果不再使用某个需要服务相关角色的特征或服务，我们建议您删除该角色。否则，您将拥有一个未使用实体，而该实体未得到主动监控或维护。但是，您必须先删除服务相关角色使用的资源后，才能手动删除该角色。

您可以使用 IAM 控制台、AWS CLI 或 AWS API 删除

AmazonChimeServiceRoleForChimeSDKMessaging 服务相关角色。有关更多信息，请参阅 IAM 用户指南中的[删除服务相关角色](#)。

使用频道流处理消息

您可在发送中的消息通过消息频道送达收件人之前，使用频道流在这些消息上运行业务逻辑。频道流可以执行诸如从消息中删除政府身份证号、电话号码或亵渎内容之类的操作。您还可以使用频道流来执行诸如在将结果发回给参与者之前汇总对民意调查的回复之类的功能。

先决条件

- 了解 Amazon Chime SDK 的基本功能，例如管理频道以及发送和接收消息。
- 能够调用 Amazon Chime SDK 消息传递 API。

频道流概念

要有效地使用频道流，您必须了解以下概念：

频道处理器数

对频道消息运行预处理逻辑的 AWS Lambda 函数。当您将频道与频道流关联时，系统会为频道中的每条消息调用流程中的处理器。为了减少延迟，单个处理器最适合大多数用例。最后，处理完成后，每个处理器都必须回调 Amazon Chime SDK 服务。

Note

目前，每个频道流仅支持一个处理器。如果您需要多个处理器，请提交支持工单以增加处理器。

频道流

频道流是最多可容纳三个频道处理器的容器，外加一个运行序列。您将流量与频道相关联，处理器会对发送到该频道的所有消息采取行动。

调用频道流

以下项目调用频道流：

- 新的永久性标准消息
- 新的非永久性标准消息
- 更新了永久性标准消息

Note

频道流不处理控制或系统消息。有关 Amazon Chime SDK 消息传递提供的消息类型的更多信息，请参阅 [消息类型](#)。

主题

- [设置频道处理器](#)
- [创建频道流](#)
- [关联和取消关联频道流](#)
- [发送消息](#)
- [使用 EventBridge 自动执行创建故障警报](#)

设置频道处理器

要开始使用频道流，您需要先创建一个处理器 Lambda 函数来处理您的用例的预处理。例如，您可以更新消息内容或元数据、拒绝消息并阻止发送消息，或者允许原始消息通过。

先决条件

- Lambda 函数必须与 ApplInstance 一样位于同一 AWS 账户和同一 AWS 区域中。

授予调用权限

您必须向 Amazon Chime SDK 消息收发服务授予权限才能调用 Lambda 资源。有关权限的更多信息，请参阅[使用基于资源的 AWS Lambda 策略](#)。例如：

主体：“messaging.chime.amazonaws.com”

操作：lambda:InvokeFunction

效果：允许

AWS:SourceAccount : ## *AWSAccountId*。

AWS:SourceArn : "arn:aws:chime:*region*:*AWSAccountId*: *appInstance*/"

Note

您可以提供特定的应用程序实例 ID 来调用您的处理器，也可以使用通配符允许账户中的所有 Amazon Chime SDK 应用程序实例调用您的处理器。

授予回调权限

您还需要允许您的处理器 Lambda 函数调用 ChannelFlowCallback API。有关执行此操作的信息，请参阅 AWS Lambda 开发人员指南中的 [AWS Lambda run role](#)。

您可以向 Lambda 函数的运行角色添加内联策略。此示例允许处理器调用 ChannelFlowCallback API。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "chime:ChannelFlowCallback"
      ],
      "Resource": [
        "arn:aws:chime:Region:AwsAccountId:appInstance/*"
      ]
    }
  ]
}
```

Note

遵循 Lambda 函数的最佳实践。有关更多信息，请参阅以下主题：

- [性能效率最佳实践](#)
- [使用 AWS Lambda 的最佳实践](#)
- [配置预留并发](#)

- [异步调用](#)

调用处理器 Lambda 函数

当用户发送消息时，以下输入请求会调用处理器 Lambda 函数。

```
{
  "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

正在发送到处理器的事件。该值是一个 CHANNEL_MESSAGE_EVENT 常数。

callbackID

从处理器调用 ChannelFlowCallback API 时使用的令牌。

ChannelMessage

ChannelArn : 频道的 ARN

Content : 待处理的消息内容

CreatedTimestamp : 创建消息的时间

LastEditedTimestamp : 编辑消息的时间

MessageId : 消息标识符

Metadata : 待处理的消息元数据

Persistence : 布尔值, 用于控制消息是否保留在后端。有效值 : PERSISTENT | NON_PERSISTENT

Sender : 消息发件人。类型 : [身份对象](#)。

类型 : 消息类型。ChannelFlow 仅支持 STANDARD 消息类型。有效值 : STANDARD

处理器函数决定每条消息的以下内容。

- 是更新消息内容、元数据还是两者兼而有之
- 是否拒绝消息
- 是否保留消息不变

处理完成后, 处理器 Lambda 函数会将结果发送回 Amazon Chime SDK 消息传递服务, 这样就可以将消息发送给所有收件人。消息状态一直处于 PENDING 标记状态, 直到处理器 Lambda 函数发回结果。处理器 Lambda 函数有 48 小时的时间来发送结果。在此之后, 我们无法保证消息传递, ChannelFlowCallback API 会引发“Forbidden Exception”错误消息。要发回结果, 请调用 [ChannelFlowCallback](#) API。

创建频道流

设置好处理器后, 即可使用 Amazon Chime SDK 消息传递 API 创建频道流。您可以使用 Fallback 操作来定义在频道流无法连接到处理器 Lambda 函数时是停止还是继续处理。如果处理器的回退操作为 ABORT, 则处理器会将消息状态设置为 FAILED, 并且不会发送消息。请注意, 如果频道流序列中的最后一个处理器的回退操作为 CONTINUE, 则消息被视为已处理并已发送给频道中的收件人。创建频道流后, 您可以将其与各个频道相关联。有关更多信息, 请参阅 [CreateChannelFlow](#) API 文档。

关联和取消关联频道流

当您为频道与频道流关联时, 频道流中的处理器会对发送到该频道的所有消息进行预处理。您必须是频道监管人或管理员才能调用频道流关联和取消关联 API。边走边记住这些事实。

- 在任何给定时间, 您最多可以将 1 个频道流与一个频道相关联。要关联频道流, 请调用 [AssociateChannelFlow](#) API。

- 要取消关联频道流并停止对频道消息的预处理，请调用 [DisassociateChannelFlow](#) API。

发送消息

您可以使用 `SendChannelMessage` API 向频道发送消息。对于与频道流关联的频道，处理器会分配以下状态值之一。

消息状态	描述
已发送	消息已成功处理。
PENDING	正在处理中。
FAILED	处理失败，因为无法访问处理器 Lambda 函数。
DENIED	将不会发送该消息。

接收中间状态事件

WebSocket 事件

WebSocket 事件将在成功建立连接后发送到频道。有关更多信息，请参阅 [WebSockets 用于接收消息](#)。

事件类型	状态	收件人	注意
创建频道消息	已发送	所有频道成员	预处理成功的 <code>SendChannelMessage</code> API
UPDATE_CHANNEL_MESSAGE	已发送	所有频道成员	预处理成功的 <code>UpdateChannelMessage</code> API
待定创建频道消息	PENDING	仅限消息发件人	正在进行预处理的 <code>SendChannelMessage</code> API

事件类型	状态	收件人	注意
PENDING_UPDATE_CHANNEL_MESSAGE	PENDING	仅限消息发件人	正在进行预处理的 UpdateChannelMessage API
FAILED_CREATE_CHANNEL_MESSAGE	FAILED	仅限消息发件人	预处理失败的 SendChannelMessage API
FAILED_UPDATE_CHANNEL_MESSAGE	FAILED	仅限消息发件人	预处理失败的 UpdateChannelMessage API
DENIED_CREATE_CHANNEL_MESSAGE	DENIED	仅限消息发件人	处理器拒绝该消息的 SendChannelMessage API
DENIED_UPDATE_CHANNEL_MESSAGE	DENIED	仅限消息发件人	处理器拒绝该消息的 UpdateChannelMessage API

GetChannelMessageStatus API

如果由于 WebSocket 连接不良而未收到事件，此 API 提供了另一种检索消息状态的方法。有关更多信息，请参阅 [GetChannelMessageStatus](#) API 文档。

Note

此 API 不会返回被拒消息的状态，因为我们不存储此类消息。

使用 EventBridge 自动执行创建故障警报

当调用处理器 Lambda 函数出现错误时，Amazon Chime SDK 会发送事件。创建频道流时，无论为处理器指定了什么 Fallback 操作，都会发送事件。您可以编写指定此类事件的简单规则，以及当其中任何事件与规则匹配时要执行的自动操作。有关更多信息，请参阅 [Amazon EventBridge 用户指南](#)。当出现此类错误时，根据您配置的 Fallback 操作，频道中的成员无法发送消息，或者消息将在不经过

处理的情况下流经频道。有关该 Fallback 操作的更多信息，请参阅 Amazon Chime SDK API 参考中的[处理器](#)。

此示例显示了一个典型的故障事件。

```
{
  "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"
  }
}
```

使用 AppInstanceBot 作为智能频道座席

您可以将 AppInstanceBots 用作智能频道座席。座席可以识别频道成员通过 ChannelMessages 发送的关键短语。机器人的自然语言理解模型可以解析消息。反过来，这允许一个或多个频道成员参与由机器人模型定义的自然语言对话。您提供机器人，这样您就可以控制对话的深度以及与企业系统的集成。

先决条件

- 了解 Amazon Chime SDK 的基本功能，例如创建 AppInstanceUsers、管理频道以及发送和接收消息。

- 能够调用 Amazon Chime SDK 消息传递 API。
- 了解 Amazon Lex V2 的基本功能，例如创建 Amazon Lex V2 机器人、建模意图和插槽、创建机器人版本、别名、使用会话状态以及 Lambda 钩子集成。

Important

使用 Amazon Lex V2 需遵守 [AWS 服务条款](#)，包括 AWS Machine Learning 和 Artificial Intelligence Services 的特定条款。

主题

- [创建 Amazon Lex V2 机器人](#)
- [设置 AppInstance 机器人](#)
- [为 AppInstanceBot 创建频道成员资格](#)
- [向 AppInstanceBot 发送消息](#)
- [处理来自 Amazon Lex 的消息](#)
- [处理来自 AppInstanceBot 的响应](#)
- [使用规则向 Amazon EventBridge 发送事件](#)
- [排除配置了 Amazon Lex V2 机器人的 AppInstanceBots 故障](#)

创建 Amazon Lex V2 机器人

要使用 AppInstance 机器人作为座席，您首先需要创建一个 Amazon Lex V2 机器人来管理智能座席场景中的对话交互。要开始构建 Amazon Lex V2 机器人，请参阅《Amazon Lex V2 开发人员指南》中的 [Amazon Lex V2 入门](#)。有关将 Amazon Lex V1 机器人迁移到 Amazon Lex V2 的信息，请参阅 [Amazon Lex V1 至 V2 迁移指南](#)。

主题

- [先决条件](#)
- [授予调用权限](#)
- [创建欢迎意图](#)
- [创建 Amazon Lex V2 机器人版本](#)
- [创建 Amazon Lex V2 机器人别名](#)

先决条件

您的 Amazon Lex V2 机器人必须具有以下先决条件。

- 您必须在支持 Amazon Lex V2 运行时系统终端节点的 AWS 区域中创建该机器人。
- 您必须在与 AppInstance 和 AppInstanceBot 相同的 AWS 账户和区域中创建机器人。
- 机器人必须通过基于资源的策略向 `messaging.chime.amazonaws.com` 服务主体授予调用权限。
- 机器人可以建模欢迎意图。这允许 AppInstanceBot 在加入频道后显示自己及其功能。
- 机器人应具有生产版本与别名功能才能配置 AppInstanceBot。
- 机器人必须使用支持的语言和区域设置。有关语言和区域设置的更多信息，请参阅《Amazon Lex V2 开发人员指南》中的 [Amazon Lex V2 支持的语言和区域设置](#)。

授予调用权限

对于要调用 Amazon Lex V2 机器人的 AppInstanceBot，Amazon Chime SDK 消息传递服务主体必须有权调用 Amazon Lex 机器人资源。有关 Amazon Lex V2 基于资源的策略权限的更多信息，请参阅《Amazon Lex V2 开发人员指南》中的 [Amazon Lex V2 基于资源的策略示例](#)。

以下示例介绍了基于资源的策略。

```
{
  "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

要允许一个 AppInstanceBot 调用 Amazon Lex V2 机器人，请使用 AppInstanceBot 的 ID。要允许 AppInstance 中的所有 AppInstanceBots 调用 Amazon Lex V2 机器人，请使用通配符。例如：

```
arn:aws:chime:region:aws-account-id:app-instance/app-instance-id/bot/  
*
```

创建欢迎意图

如果您在 Amazon Lex V2 机器人模型中添加了可选的欢迎意图，则 AppInstanceBot 可以在它加入频道时显示自己及其功能。欢迎意图可以显示消息，也可以发起与频道成员的对话。欢迎意图的名称可能会有所不同，您可以在 AppInstanceBot 的配置中对其进行定义。

有关意图的更多信息，请参阅《Amazon Lex V2 开发人员指南》中的[添加意图](#)。

创建 Amazon Lex V2 机器人版本

创建 Amazon Lex V2 机器人时，您只能创建草稿版本。草稿是您可以更新的机器人的工作副本。默认情况下，草稿版本与名为 TestBotAlias 的别名相关联，您只能使用草稿版机器人进行手动测试。

完成对话建模并构建草稿版机器人后，您可创建一个或多个版本，即草稿版 Lex 机器人的带编号快照。版本允许您控制客户端应用程序使用的实现。例如，您可以发布版本以用于您的工作流的不同阶段，如开发、测试部署和生产。

有关 Lex 机器人版本控制的更多信息，请参阅《Amazon Lex V2 开发人员指南》中的[创建版本](#)。

创建 Amazon Lex V2 机器人别名

一旦您创建了一个或多个版本的 Amazon Lex V2 机器人，就可以创建别名。别名充当指向 Amazon Lex V2 机器人版本的命名指针。例如，您一次只能将别名与一个版本关联。

有关 Lex 机器人别名的更多信息，请参阅《Lex V2 开发人员指南》中的[创建别名](#)。

设置 AppInstance 机器人

拥有带有型号、版本和别名的 Amazon Lex V2 机器人后，您可以使用 Amazon Chime SDK 消息传递 API 或 CLI 来创建 AppInstanceBot。有关使用 API 的更多信息，请参阅 [CreateAppInstanceBot](#) API 文档。

Note

您可以使用 `InvokedBy` 属性来配置 AppInstanceBot 的对话框交互行为。您可以配置触发机器人的消息类型，例如标准消息或定向消息。

以下示例说明如何使用 AWS CLI 创建一个 AppInstanceBot，所有带有 MENTIONS 的标准消息和定向消息都可以调用该机器人。

```
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"  
}'
```

为 AppInstanceBot 创建频道成员资格

创建 AppInstanceBot 后，即可将其作为成员添加到新的或现有频道中。了解更多信息，请参阅《Amazon Chime SDK 消息传递 API》文档中的 [CreateChannel](#) 和 [CreateChannelMembership](#)。

以下示例说明如何使用 AWS CLI 创建频道并将 AppInstanceBot 添加为成员。

```
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"  
]'
```

以下示例说明如何使用 AWS CLI 向现有频道添加 AppInstanceBot。

```
aws chime-sdk-messaging create-channel-membership \  
--chime-bearer caller_app_instance_user_arn \  
--channel-arn channel_arn \  
--member-arn app_instance_bot_arn
```

向 AppInstanceBot 发送消息

您可以使用 [SendChannelMessage](#) API 向 AppInstanceBot 发送消息。您将消息发送到 AppInstanceBot 所属的频道。如果 [自然语言理解模型](#) 识别出消息内容并引发 Amazon Lex 意图，则 AppInstanceBot 将使用频道消息进行响应并启动对话。

您还可以向频道成员发送目标消息，该成员可能是 AppInstanceUser 或 AppInstanceBot。只有目标和发件人可以查看定向消息。只有能够看到定向消息的用户才能对其采取行动。但是，管理员可以删除他们看不到的定向消息。

以下示例介绍了如何使用 AWS CLI 发送频道消息。

```
aws chime-sdk-messaging send-channel-message \  
--chime-bearer caller_app_instance_user_arn \  
--channel-arn channel_arn \  
--content content \  
--type STANDARD \  
--persistence PERSISTENT
```

处理来自 Amazon Lex 的消息

向 Amazon Lex 发送消息时，Amazon Chime SDK 消息传递会将频道和发件人的 ARN 信息作为请求属性填充 CHIME.channel.arn 和 CHIME.sender.arn。您可以使用这些属性来确定谁发送了消息以及发件人所属的频道。有关更多信息，请参阅《Amazon Lex 开发人员指南》中的 [使用 AWS Lambda 函数启用自定义逻辑](#)。

处理来自 ApplInstanceBot 的响应

当用户发送消息时，ApplInstanceBot 会以频道消息进行响应。您可以列出频道消息以获取机器人的回复。

以下示例介绍了如何使用 CLI 列出频道消息。

```
aws chime-sdk-messaging list-channel-messages \
--chime-bearer caller_app_instance_user_arn \
--channel-arn channel_arn
```

来自 ApplInstanceBot 的成功响应采用以下格式。

```
{
  "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",  
}
```

内容

该 Content 字段包含源自 Amazon Lex V2 机器人的消息列表。有关这些消息的更多信息，请参阅 Amazon Lex V2 RecognizeText API 中的[消息](#)。

以下示例说明如何在欢迎消息中使用 Content 字段。

```
{  
  "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"  
          }  
        ]  
      }  
    }  
  ]  
}
```

对于失败响应，“内容”字段包含以下格式的错误消息和代码：

```
{  
  "Code": error_code
```



```
}
```

ContentType

ContentType 是指该 Content 字段包含的负载类型，必须选中才能解析该 Content 字段。

Note

Lex V2 机器人使用的是不同的 ContentType。

ContentType 设置为 application/amz-chime-lex-msgs（对于成功响应）或者 application/amz-chime-lex-error（对于失败响应）。

MessageAttributes

MessageAttribute 是字符串键到字符串值的映射。来自 ApplInstanceBot 的响应包含以下消息属性，这些属性映射到来自 Amazon Lex 机器人的响应。

- CHIME.LEX.sessionState.intent.name：请求尝试实现的 Lex 机器人意图名称。
- CHIME.LEX.sessionState.intent.state：意图的当前状态。可能的值包括：Fulfilled、InProgress 和 Failed。
- CHIME.LEX.sessionState.originatingRequestId：向 Amazon Lex 机器人发出的特定请求的唯一标识符。该值设置为触发 ApplInstanceBot 的原始用户消息的 MessageId。
- CHIME.LEX.sessionState.sessionId：用户与机器人之间对话的唯一标识符。用户启动与机器人的聊天时，Amazon Lex 将创建一个会话。

有关 Amazon Lex 会话和会话状态的更多信息，请参阅《Amazon Lex API 参考》中的 [SessionState](#) 和《Amazon Lex V2 开发人员指南》中的 [管理会话](#)

有关 Amazon Lex V2 返回属性的更多信息，请参阅 [Amazon Lex Runtime V2 API](#)。

使用规则向 Amazon EventBridge 发送事件

当错误阻止其调用 Amazon Lex V2 机器人时，Amazon Chime SDK 会传送 EventBridge 事件。您可以创建 EventBridge 规则，以识别这些事件，并在匹配规则时自动采取行动。有关更多信息，请参阅《Amazon EventBridge 用户指南》中的 [Amazon EventBridge 规则](#)。

以下示例显示了典型故障事件。

```
{
  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)"
  }
}
```

排除配置了 Amazon Lex V2 机器人的 AppInstanceBots 故障

以下主题说明如何排查 AppInstanceBots 的常见问题。

调查发现 Amazon Lex V2 故障

当错误阻止其调用 Amazon Lex V2 机器人时，Amazon Chime SDK 消息传递会送达 [Amazon EventBridge 事件](#)。有关设置规则和配置通知目标的更多信息，请参阅 Amazon EventBridge User Guide 中的 [Amazon EventBridge 入门](#)。

如果您在 AWS CloudWatch Logs 中收到 EventBridge 事件，则可以使用 AWS CloudWatch Logs Insights，根据 Amazon Chime SDK 消息传递 detail-type 查询 EventBridge 事件。failureReason 列出了失败的原因。

以下示例介绍了典型查询。

```
fields @timestamp, @message
| filter `detail-type` = "Chime Messaging AppInstanceBot Lex Failure"
| sort @timestamp desc
```

如果 Amazon Chime SDK 消息传递可以调用您的 Amazon Lex V2 机器人，则 SDK 会发送带有错误消息的 CONTROL 消息。

排除 Amazon Lex V2 机器人权限错误故障

要让 AppInstanceBot 调用 Amazon Lex V2 机器人，Amazon Chime SDK 消息传递服务主体必须有权调用 Amazon Lex V2 Bot 资源。此外，请确保资源策略条件的 `AWS:SourceArn` 与 AppInstanceBot 的 ARN 相匹配。

有关配置 AppInstanceBot 以调用 Amazon Lex V2 机器人的更多信息，请参阅本节前面 [创建 Amazon Lex V2 机器人](#) 的内容。

排除 Amazon Lex V2 机器人节流故障

每个机器人别名的最大并发文本模式对话数 Amazon Lex 都有服务限额。如需增加限额，您可以联系 Amazon Lex 服务团队。有关更多信息，请参阅 Amazon Lex Developer Guide 中的 [Amazon Lex 准则和限额](#)。

管理消息保留日期

账户所有者可以使用 Amazon Chime SDK API 开启消息保留功能。根据管理员设置的时间段自动删除消息。保留期可以持续一天到十五年。您也可以随时使用 API 更新消息保留期或关闭消息保留。

本节中的主题

- [CLI 保留命令示例](#)
- [启用消息保留日期](#)
- [恢复和删除消息](#)

CLI 保留命令示例

以下示例介绍了用于保留的典型 CLI 命令：

正在启用

```
aws chime-sdk-identity put-app-instance-retention-settings --app-instance-arn {appInstanceArn} --app-instance-retention-settings ChannelRetentionSettings={RetentionDays=60}
```

正在更新

```
aws chime-sdk-identity put-app-instance-retention-settings --app-instance-arn {appInstanceArn} --app-instance-retention-settings ChannelRetentionSettings={RetentionDays=30}
```

正在禁用

```
aws chime-sdk-identity put-app-instance-retention-settings --app-  
instance-arn {appInstanceArn} --app-instance-retention-settings  
ChannelRetentionSettings={}
```

启用消息保留日期

您可以使用 Amazon Chime SDK API 开启消息保留功能。您也可以随时使用 API 更新消息保留期或关闭消息保留。如需了解有关配置消息保留的更多信息，请参阅 [Amazon Chime SDK API 参考](#)。

恢复和删除消息

您可以在设置或更新消息保留期后 30 天内将消息还原给用户。但是，在该 30 天宽限期之后，保留期内的所有消息都将永久删除，新消息在保留期过后也会永久删除。

Note

在 30 天宽限期内，如果您延长保留策略期限或将其关闭，那么尚未超过新保留期的消息将再次对账户中的用户可见。

AppInstanceUser 删除频道或消息时，消息也会被永久删除。

用于消息传递的 UI 组件

您可以使用组件库来减少为聊天消息传递构建 UI 所需的工作量。有关更多信息，请参阅 [Amazon Chime React 组件库](#) GitHub。

与客户端库集成

要使用 Amazon Chime SDK 的消息传递功能，您必须将您的客户端应用程序与以下客户端库集成：

- AWS SDK — 包含用于发送消息和管理资源的 API。
- 适用于 JavaScript (NPM) 的 Amazon Chime SDK 客户端库 — 一个包含 TypeScript 类型定义的 JavaScript 库，可帮助您将客户端与 Amazon Chime SDK 消息传送网络套接字集成以接收消息。

要将您的客户端应用程序与 Amazon Chime SDK 集成，请参阅客户端库 README.md 中的说明，并使用演示来学习如何构建消息传递功能。

将 Amazon Chime 软件开发工具包消息传递与 JavaScript

您可以使用 JavaScript 管理 Amazon Chime 软件开发工具包资源和发送消息。有关更多信息，请参阅 [S AWS JavaScript DK](#)。

您还可以在客户端应用程序中创建消息传递会话，以接收来自 Amazon Chime SDK 消息传递的消息。有关更多信息，请参阅 [使用上的 Amazon Chime 软件开发工具包客户端库。JavaScript GitHub](#)

使用 Amazon Chime SDK PSTN 音频服务

Note

本部分介绍 Chime SDK PSTN 音频服务，在本文档的早期版本和一些博客文章中被称为“SIP 媒体应用程序 (SMA)”。展望未来，当我们提及“SIP 媒体应用程序”时，我们指的是与 PSTN 音频服务关联的 Amazon Chime SD AWS K 控制台和软件开发工具包中的配置项目。

本部分介绍如何使用 Amazon Chime SDK 公共交换电话网 (PSTN) 音频服务。借助 PSTN Audio 服务，开发人员可以利用无服务器功能的敏捷性和操作简便性来构建自定义电话应用程序。AWS Lambda

您的 AWS Lambda 功能使用 Amazon Chime SDK 语音连接器控制电话的行为，例如播放语音提示、收集数字、录制通话、将呼叫路由到 PSTN 和会话初始协议 (SIP) 设备。以下主题提供了有关 PSTN 音频服务的概述和架构信息，包括如何构建控制呼叫的 AWS Lambda 函数。

Note

本节中的主题假设您了解该 AWS Lambda 服务。有关的更多信息 AWS Lambda，请参阅[入门 AWS Lambda](#)。此外，要成功使用 Amazon Chime SDK 的这一功能，Amazon Chime SDK 管理员必须创建至少一个 SIP 规则和一个 SIP 媒体应用程序。有关完成这些任务的更多信息，请参阅《Amazon Chime SDK 管理员指南》中的[管理 SIP 媒体应用程序](#)。

主题

- [迁移到 Amazon Chime SDK 语音命名空间](#)
- [了解电话号码、SIP 规则、SIP 媒体应用程序和 AWS Lambda 功能](#)
- [了解 PSTN 音频服务编程模型](#)
- [将呼叫和事件路由到 AWS Lambda 函数](#)
- [关于使用 PSTN 音频服务呼叫分支](#)
- [调用流程示例](#)
- [构建 PSTN 音频服务的 AWS Lambda 函数](#)

迁移到 Amazon Chime SDK 语音命名空间

[Amazon Chime SDK 语音](#)命名空间是 API 的专用位置，用于创建和管理 Amazon Chime SDK 语音资源。您可以使用命名空间在 Amazon Chime SDK 语音 API 终端节点所在的任何 AWS 区域进行寻址。如果您刚开始使用 Amazon Chime SDK，则使用此命名空间。有关“区域”的更多信息，请参阅本指南中的 [可用区](#)。

使用 [Amazon Chime](#) 命名空间的现有应用程序应计划迁移到专用命名空间，以使用更新后的 API 和新功能。

主题

- [迁移原因](#)
- [迁移之前](#)
- [命名空间之间的差异](#)

迁移原因

我们建议迁移到 [Amazon Chime SDK 语音](#)命名空间，原因如下：

选择 API 终端节点

Amazon Chime SDK 语音命名空间允许您在任何提供 [API 终端节点的区域](#)使用该终端节点。如果您想使用 us-east-1 以外的 API 终端节点，则必须使用 Amazon Chime SDK 语音命名空间。有关当前终端节点的更多信息，请参阅本指南中的 [API 映射](#)。

更新和新的语音 API

我们仅在 Amazon Chime SDK 语音命名空间中添加或更新语音 API。

迁移之前

在迁移之前，记下命名空间之间的差异。下表列出并描述了以上差异。

	Amazon Chime SDK 语音命名空间	Amazon Chime 命名空间
AWS 命名空间	ChimeSDKVoice	Chime

	Amazon Chime SDK 语音命名空间	Amazon Chime 命名空间
区域	多个	仅限 us-east-1
终端节点	https://voice-chime. <i>region</i> .amazonaws.com	service.chime.aws.amazon.com
服务主体	chime.amazonaws.com	chime.amazonaws.com
API	仅用于 PSTN 音频服务的 API	用于 PSTN 音频和 Amazon Chime 其他部分的 API
语音连接器管理	多个区域	us-east-1
语音连接器组管理	多个区域	us-east-1
SIP 媒体应用程序和 SIP 规则管理	多个区域	us-east-1
电话号码管理	多个区域	us-east-1
呼叫分析	可用	不可用
语音配置文件域	可用	不可用
紧急调用管理	多个区域	us-east-1
代理电话会话管理	多个区域	us-east-1
流管理	多个区域	us-east-1
日志和指标管理	多个区域	us-east-1

有关可用“区域”的更多信息，请参阅本指南中前面的 [PSTN 区域](#)。

命名空间之间的差异

以下各节解释了 Amazon Chime SDK Voice 和 Amazon Chime 命名空间之间的差异。

AWS 命名空间

Amazon Chime SDK 命名空间使用 Chime 正式名称。Amazon Chime SDK 语音命名空间使用 ChimeSDKVoice 正式名称。名称的确切格式因平台而异。

例如，如果您在 Node.js 中使用 AWS SDK 创建会议，则使用以下代码行寻址命名空间。

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

要迁移到 Amazon Chime SDK 语音命名空间，使用新的命名空间和终端节点区域更新该代码行。

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

区域

[Amazon Chime](#) 命名空间只能寻址该 us-east-1 地区的 API 终端节点。[Amazon Chime SDK 语音](#)命名空间可以在任何可用区域寻址 Amazon Chime SDK 语音 API 终端节点。有关语音区域的最新列表，请参阅本指南中的 [可用区](#)。

终端节点

[Amazon Chime SDK 语音](#)命名空间使用不同于 [Amazon Chime](#) 命名空间的 API 终端节点。

只能使用用于创建语音操作的终端节点对其进行修改。这意味着通过 eu-central-1 中的终端节点创建的语音操作只能通过 eu-central-1 进行修改。这也意味着您无法使用 us-east-1 中的 ChimeSDKVoice 命名空间寻址通过 Chime 命名空间创建的语音操作。有关当前终端节点的更多信息，请参阅本指南中的 [API 映射](#)。

服务主体

两个命名空间都使用 chime.amazonaws.com 服务主体。如果您有授予服务访问权限的访问策略，则无需更新这些策略。

API

[Amazon Chime SDK 语音](#)命名空间仅包含用于创建和管理语音操作的 API。[Amazon Chime](#) 命名空间包括用于语音和 Amazon Chime 服务其他部分（例如会议）的 API。

Tagging

只有 [Amazon Chime SDK 语音](#)命名空间支持标签。[有关标签的更多信息，请参阅 TagResource 和 UntagResource。](#)

媒体区域

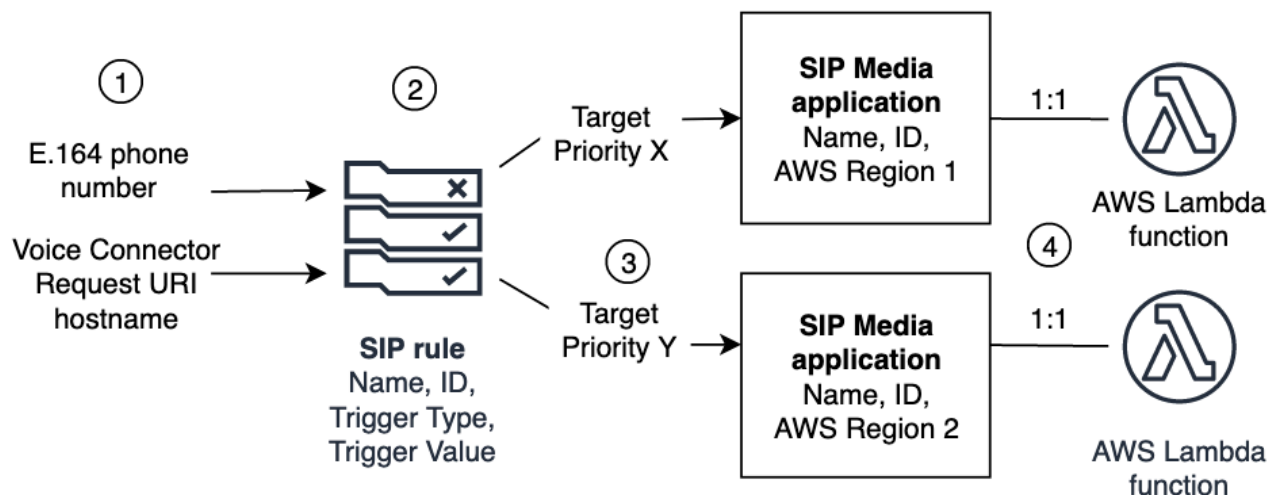
如果您使用 Chime 命名空间创建诸如语音连接器和 SIP 媒体应用程序等资源，则只能在 us-east-1 和 us-west-2 中执行此操作，并且必须使用 us-east-1 终端节点。

相反，ChimeSDKVoice 命名空间允许在所有支持的区域中创建资源，不仅仅是 us-east-1 和 us-west-2。有关区域覆盖的更多信息，请参阅 [可用区](#)。

了解电话号码、SIP 规则、SIP 媒体应用程序和 AWS Lambda 功能

在使用 PSTN 音频服务之前，Amazon Chime SDK 管理员必须配置您的电话号码并创建名为 SIP 规则和 SIP 媒体应用程序的托管对象。您可以使用 Amazon Chime 软件开发工具包控制台或 AWS 软件开发工具包来配置电话号码，以及配置 SIP 规则和 SIP 媒体应用程序托管对象。

此图片显示构成 PSTN 音频服务的托管对象之间的关系。图片中的数字对应于其下面文本的编号：



您只能将电话号码和 Amazon Chime SDK 语音连接器 (1) 分配给 SIP 规则 (2)。此外，您还必须在 PSTN 音频服务中配置电话号码或语音连接器。在接到电话号码的入站呼叫或语音连接器的出站呼叫请求时，SIP 规则将调用 SIP 媒体应用程序和相关 AWS Lambda 函数 (4)。该 AWS Lambda 功能运行一组预定义的操作，例如播放暂停的音乐或加入会议。为了提供多区域弹性，SIP 规则可以按故障转移的优先顺序指定不同 AWS 区域 (3) 中的备用目标 SIP 媒体应用程序。如果一个目标失败，PSTN 音频服务会尝试下一个目标，依此类推。请注意，每个备用目标必须位于不同的 AWS 区域。

此外，多个 SIP 媒体应用程序可以调用给定 AWS Lambda 功能。换句话说，当您创建 AWS Lambda 功能时，任何 SIP 媒体应用程序都可以使用该功能。

有关配置 SIP 媒体应用程序和规则的更多信息，请参阅《Amazon Chime SDK 管理员指南》中的[管理 SIP 媒体应用程序和规则](#)。

了解 PSTN 音频服务编程模型

PSTN 音频服务使用请求/响应编程模型，该模型反过来又使用函数。AWS Lambda 您的 AWS Lambda 函数会自动为来电和去电调用。例如，当新的来电到达时，PSTN 音频服务会使用 `NEW_INCOMING_CALL` 事件调用您的 AWS Lambda 函数，然后等待名为 Actions 的命令。例如，您的应用程序可以选择诸如播放音频提示、收集数字、录制音频或转接呼叫等的操作。这些 JSON 格式的操作将使用函数的回调发送回 PSTN 音频服务。AWS Lambda

此示例显示 PlayAudio 操作。

```
{
  "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"
    }
  }
}
```

此示例显示 RecordAudio 操作。

```
{
  "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 服务运行操作后，它会再次调用您的 AWS Lambda 函数，并显示成功或失败指示。

您的应用程序还可以拨打外线电话，并使用您的 AWS Lambda 功能来控制通话流程、来电者体验和通话环境。在这种情况下，您调用 [CreateSipMediaApplicationCallAPI](#)，然后通过 NEW_OUTBOUND_CALL 事件调用您 AWS Lambda 的 API。接听来电后，您可以返回操作，例如播放语音提示和收集用户输入的数字。您还可以使用 [UpdateSipMediaApplicationCallAPI](#) 触发您的 AWS Lambda 函数，以实现计时器、参与者静音和等候室。

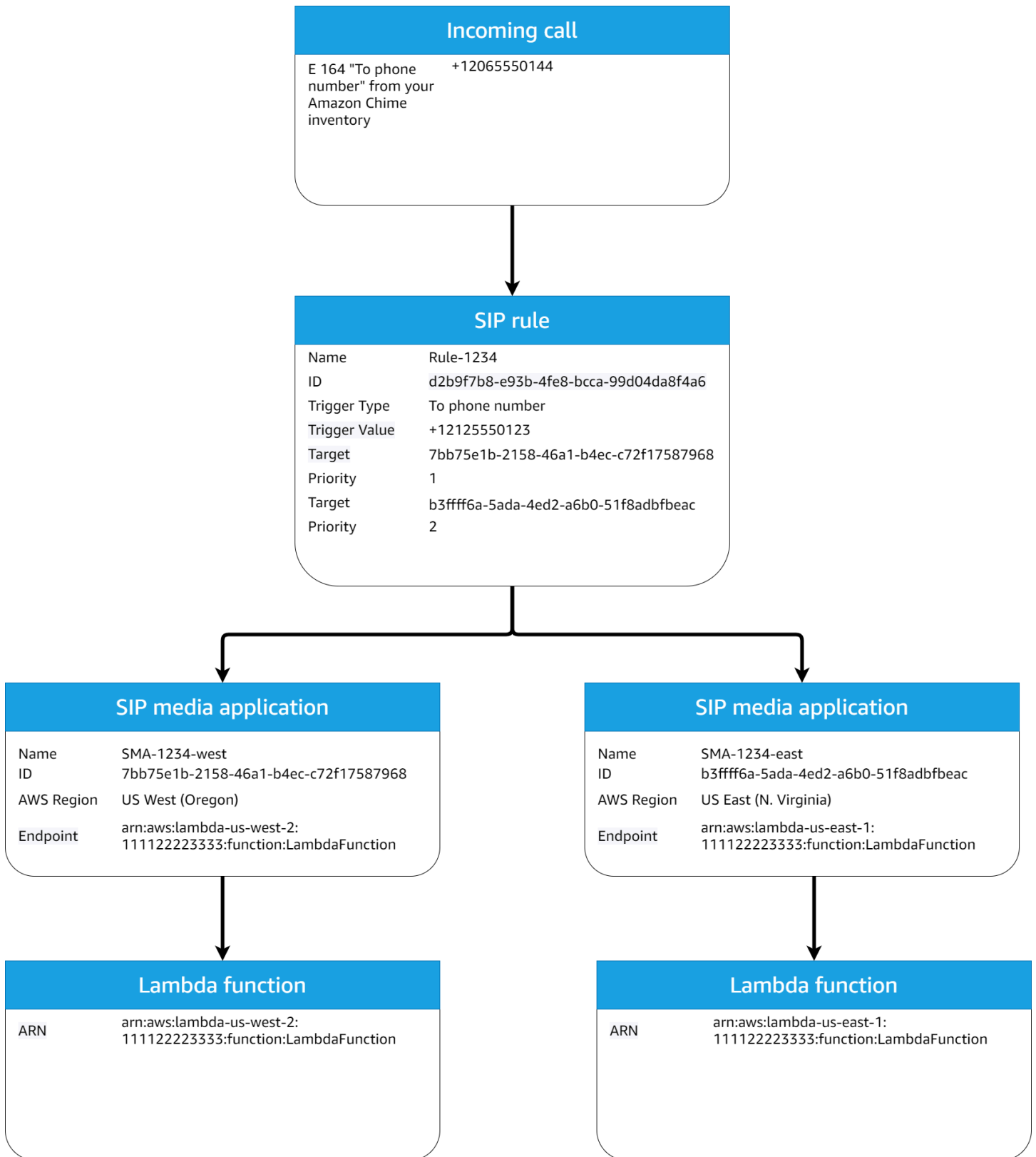
将呼叫和事件路由到 AWS Lambda 函数

PSTN Audio 服务提供以下方法来将来电转接到您的 AWS Lambda 职能部门进行治疗。

- 您可以根据被叫号码路由呼叫。为此，Amazon Chime SDK 管理员创建了一条 SIP 规则，并将触发类型设置为拨打电话号码。此电话号码必须存在于 Amazon Chime SDK 的电话号码清单中，且 AWS 账户必须与 SIP 规则相同。
- 您可以根据传入的语音连接器 SIP 呼叫的请求 URI 将呼叫路由到该 AWS Lambda 功能。为此，Amazon Chime SDK 管理员创建了一条 SIP 规则，并将触发器类型设置为请求 URI 主机名。此字段必须包含在与 SIP 规则相同的 AWS 帐户中配置的语音连接器的“出站主机名”字段中指定的完全限定域名。

然后，管理员至少配置一个目标 SIP 媒体应用程序。或者，您可以按优先顺序配置多个 SIP 媒体应用程序，以支持冗余和失效转移。例如，您可以在两个不同的 AWS 区域配置两个 SIP 媒体应用程序并指定它们的优先级顺序。如果 SIP 规则有多个目标 SIP 媒体应用程序，则按优先级顺序调用 SIP 媒体应用程序的 Lambda 函数。SIP 媒体应用程序中优先级最高的 AWS Lambda 功能（最小的数字，例如 1）首先运行。如果 PSTN 音频服务无法调用该 AWS Lambda 函数，则会调用 SIP 媒体应用程序中优先级排在第二高的优先级（次低数字，例如 2）的 AWS Lambda 函数。如果所有运行 SIP 规则中指定的 SIP 媒体应用程序的尝试都失败，则 PSTN 音频服务会挂断。

配置必要的 SIP 规则和 SIP 媒体应用程序后，PSTN 音频服务便会将来电路由到您的职能部门。AWS Lambda 下图显示了使用拨打电话号码触发类型的典型顺序。



在图中：

1. PSTN 音频服务接收拨打在同一账户的 SIP 规则中配置的电话号码的来电。AWS
2. 然后，PSTN 音频服务会评估 SIP 规则，并以最高优先级（在本例中为优先级 1）获取 SIP 媒体应用程序。
3. 然后，该服务调用与 SIP 媒体应用程序关联的 AWS Lambda 功能。
4. 可选。如果服务无法调用 AWS Lambda 与最高优先级关联的，它将尝试以第二高的优先级（在本例中为优先级 2）运行 SIP 媒体应用程序（如果存在）。
5. 可选。如果所有目标 SIP 媒体应用程序都失败，PSTN 音频服务将挂断。

下图显示了使用请求 URI 主机名触发器类型的典型规则。

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

在图中：

1. PSTN 音频服务在 Amazon Chime SDK 语音连接器上接收来电，其请求 URI 主机名与同一账户中预配置的 SIP 规则相匹配。AWS
2. 然后，该服务会评估 SIP 规则并获取优先级最低的 SIP 媒体应用程序（在本例中为优先级为 1 的唯一目标 SIP 媒体应用程序）。
3. 然后，该服务调用与 SIP 媒体应用程序关联的 AWS Lambda 功能。
4. 可选。如果服务无法调用优先级最低的关联 AWS Lambda 的，则会尝试运行优先级次低的 SIP 媒体应用程序（如果存在）。在这种情况下，只有一个目标 SIP 媒体应用程序。
5. 可选。如果所有目标 SIP 媒体应用程序都失败，PSTN 音频服务将挂断。

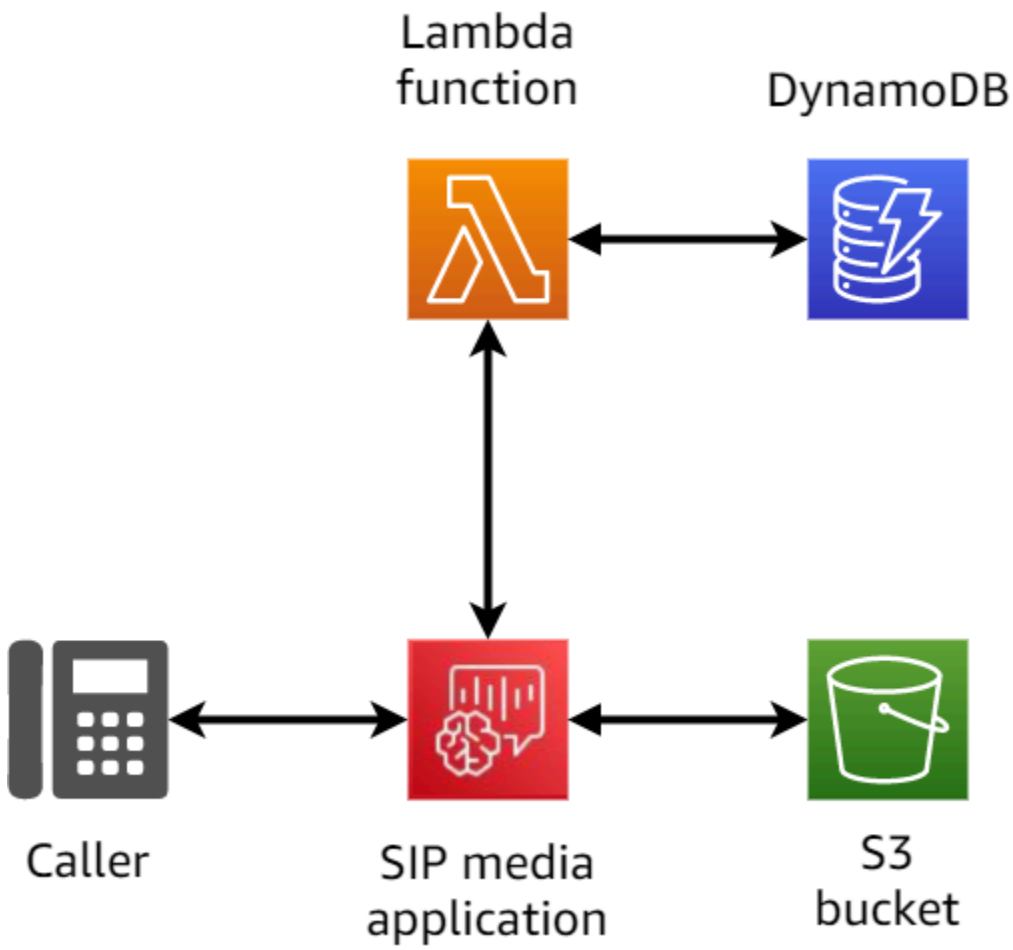
此外，您可以使用 [createSip MediaApplicationCall](#) API 创建出站呼叫，然后调用您的 AWS Lambda 函数进行额外处理。要使用此 API，您需要将已配置的 SIP 媒体应用程序 ID 指定为参数。

最后，您可以在呼叫处于活动状态时随时使用 [updateSIP API MediaApplicationCall](#) 触发您的 AWS Lambda 函数。要使用此 API，您需要将已配置的 SIP 媒体应用程序 ID 指定为参数。

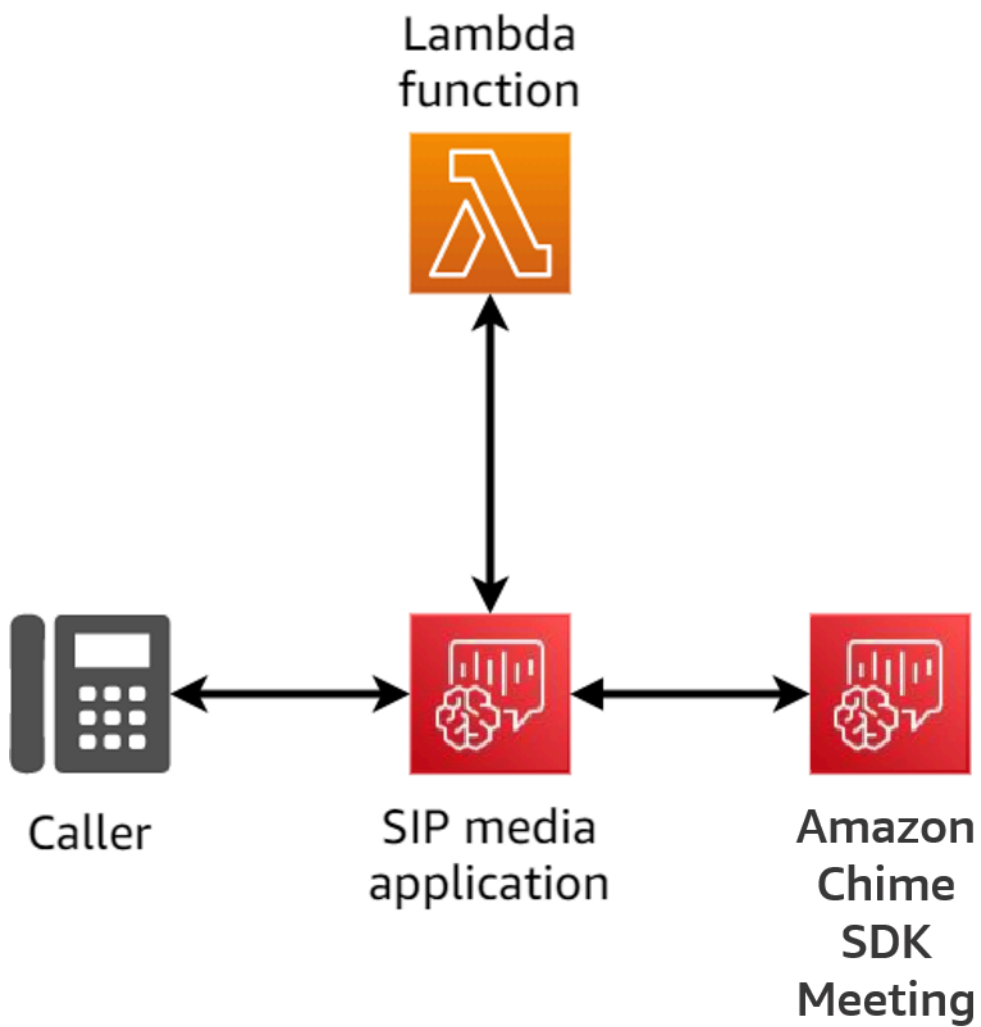
关于使用 PSTN 音频服务呼叫分支

PSTN 音频服务可以在一个或多个呼叫分支上运行。例如，当您录制或传递语音邮件时，只有一个呼叫分支，而当您加入 Amazon Chime SDK 会议时，则有多个呼叫分支。

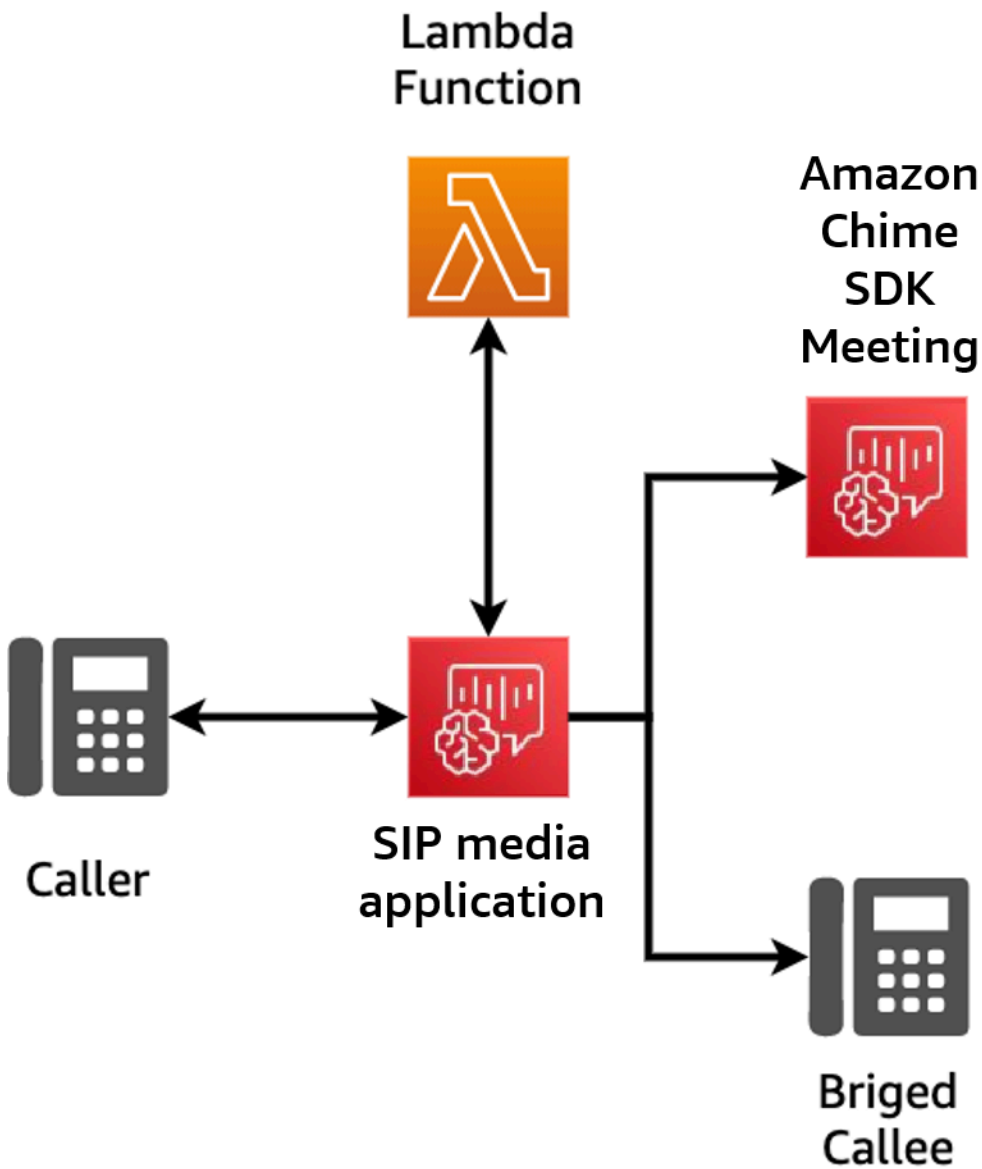
下图显示单个分支呼叫的流程。



下图显示多个分支呼叫的架构。



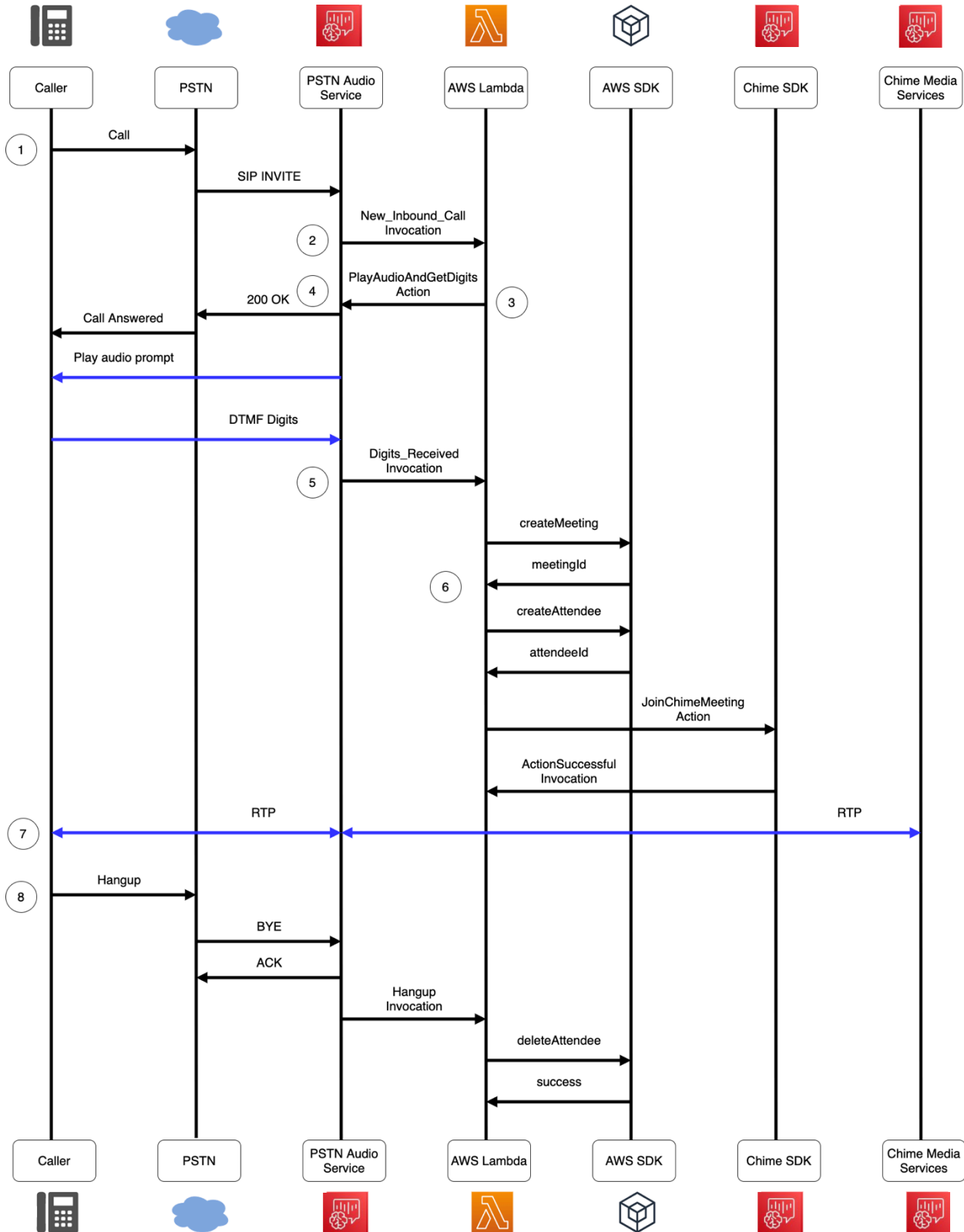
下图显示多个分支桥接呼叫的流程。



调用流程示例

此图表显示了通过 Amazon Chime SDK PSTN 音频服务的呼叫流程和客户端的功能。AWS Lambda 在此示例中，应用程序向呼叫者播放提示，收集双音多频 (DTMF) 数字，然后将它们连接到 Amazon Chime SDK 会议。

图表中的数字对应于其下方带编号的文本。



在下图中：

1. Amazon Chime SDK PSTN 音频服务接收在 SIP 规则中配置的电话号码的呼叫。
2. PSTN 音频服务获取关联的 SIP 媒体应用程序，并通过事件 (LEG-ANEW_INBOUND_CALL) 调用关联的 AWS Lambda 函数。
3. 该 AWS Lambda 函数返回操作列表 PlayAudioAndGetDigits，包括指示 PSTN 音频服务应答呼叫、向呼叫者播放音频文件以及收集呼叫者输入的 DTMF 数字。
4. PSTN 音频服务应答呼叫、播放音频提示并收集呼叫者输入的 DTMF 数字。
5. PSTN 音频服务使用 DTMF 数字输入调用该 AWS Lambda 函数。该 AWS Lambda 函数使用 AWS 软件开发工具包创建 Amazon Chime 软件开发工具包会议和会议参与者。
6. AWS 软件开发工具包返回 MeetingId 和后 AttendeeId，该 AWS Lambda 函数将返回一个操作，以加入对 Amazon Chime SDK 会议 (LEG-B) 的调用。
7. 公共交换电话网 (PSTN) 的来电者与 Amazon Chime SDK 媒体服务之间建立实时传输协议 (RTP) 会话。
8. 当 PSTN 来电者挂断电话时，PSTN 音频服务会通过 HANGUP 事件调用该 AWS Lambda 函数，该函数将删除与会者。AWS Lambda

构建 PSTN 音频服务的 AWS Lambda 函数

本节中的主题说明如何构建 PSTN 音频服务所使用的 AWS Lambda 函数。

内容

- [了解电话事件](#)
- [了解操作](#)
- [调用 AWS Lambda 函数的电话事件](#)
- [使用操作列表响应调用](#)
- [PSTN 音频服务支持的操作](#)
- [使用 SIP 标头](#)
- [使用呼叫详细信息记录](#)
- [超时和重试](#)
- [调试和故障排除](#)
- [VoiceFocus](#)
- [PSTN 音频服务词汇表](#)

了解电话事件

当调用期间发生某些事件时，音频服务会调用您的 AWS Lambda 函数。以下示例显示了事件，并且示例后的文本对每个事件进行了解释。

```
{
  "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

用于创建此事件对象的架构版本。

Sequence (序列)

调用 AWS Lambda 函数的事件的序列。每次调用期间调用函数时，序列会递增。

InvocationEventType

触发 AWS Lambda 调用的事件的类型。有关更多信息，请参阅本主题后面的 [事件类型](#)。

CallDetails

与 AWS Lambda 调用关联的呼叫的相关信息。

TransactionId

与 AWS Lambda 调用关联的呼叫的 ID。

AwsAccountId

与导致调用路由的 SIP 媒体应用程序关联的 AWS 账户 ID。

SipMediaApplicationId

与调用关联的 SIP 媒体应用程序的 ID。

参与者

调用 AWS AWS Lambda 函数的调用中有关参与者的信息。

CallId

分配给每个参与者的唯一 ID。

ParticipantTag

每个调用参与者会获得一个标签，LEG-A 或 LEG-B。

收件人

参与者的“收件人”电话号码，格式为 E.164。

发件人

参与者的“发件人”电话号码，格式为 E.164。

方向

调用分支的来源方向。Inbound 表示对音频服务的调用。Outbound 表示从音频服务发起的调用。

StartTimeInMilliseconds

以毫秒为单位的纪元时间，从参与者加入调用时开始。

状态

参与者是 Connected 或 Disconnected

事件类型

音频服务使用以下这些事件类型调用 Lambda 函数：

NEW_INBOUND_CALL

新调用由与您的 SIP 媒体应用程序关联的电话号码发起。

NEW_OUTBOUND_CALL

通过 [CreateSipMediaApplicationCall](#) API 进行新的出站调用。

ACTION_SUCCESSFUL

从 AWS Lambda 函数返回的操作已成功。成功的操作包括与成功操作相匹配的 `ActionData`。

```
"ActionData": {  
  // The previous successful action  
},
```

ACTION_FAILED

从 AWS Lambda 函数返回的操作未成功。不成功的操作包括与失败操作相匹配的 `ActionData`、错误类型以及描述失败的错误消息：

```
"ActionData": {  
  // The previous unsuccessful action  
  "ErrorType": "error-type",  
  "ErrorMessage": "error message"  
},
```

ACTION_INTERRUPTED

正在运行的操作被 [UpdateSipMediaApplicationCall](#) API 调用中断。 `ActionData` 包括中断的操作：

```
"ActionData": {  
  // The action that was interrupted  
},
```

挂断

用户或应用程序挂断调用分支。 `ActionData` 包括有关该事件的以下详细信息：


```
"ActionData": {
  "Type": "Hangup",
  "Parameters": {
    "SipResponseCode": 486,
    "CallId": "c70f341a-adde-4406-9dea-1e01d34d033d",
    "ParticipantTag": "LEG-A"
  }
},
```

类型

挂断

参数

有关 HANGUP 事件的信息。

- SipResponseCode — 与事件相关联的响应代码。最常见的代码是：
 - 0 — 正常清除
 - 480 — 无应答
 - 486 — 用户占线
- CallId 挂断的参与者的 ID。
- ParticipantTag 挂断的参与者的标签。

CALL_ANSWERED

音频服务已传入调用。除非呼叫已桥接，否则此事件会在拨出呼叫时返回。

INVALID_LAMBDA_RESPONSE

对上次 AWS Lambda 调用的响应出现问题。ActionData 包括以下附加字段：

```
"ErrorType": "error-type-1",
"ErrorMessage": "error-msg-1"
```

DIGITS_RECEIVED

应用程序在 ReceiveDigits 操作完成后收到 DTMF 位数。ActionData 包括收到的位数。

```
"ActionData": {
```

```
    "ReceivedDigits": ###  
    // The ReceiveDigits action data  
  },
```

CALL_UPDATE_REQUESTED

[UpdateSipMediaApplicationCall](#) API 已调用。ActionData 包括有关更新请求的信息：

```
  "ActionData": {  
    "Type": "CallUpdateRequest",  
    "Parameters": {  
      "Arguments": {  
        "leg": "LEG-A"  
      }  
    }  
  },  
}
```

RINGING

调用分支正在响铃

了解操作

在 PSTN 音频服务中，SIP 媒体应用程序会触发 AWS Lambda 功能。反过来，这些 AWS Lambda 函数可以返回一个称为操作的指令列表。操作是您要在电话呼叫分支上运行的项目，例如发送或接收数字、加入会议等。操作还可以返回数据，因此您可以将操作视为带有数据字段的对象。有关 PSTN 音频服务调用的操作的更多信息，请参阅 [了解电话事件](#)。

调用 AWS Lambda 函数的电话事件

音频服务调用 AWS Lambda 函数以响应不同的事件。每次调用指定调用事件类型并提供调用详细信息，包括其参与者（如适用）。以下主题描述调用 AWS Lambda 函数的音频服务事件。

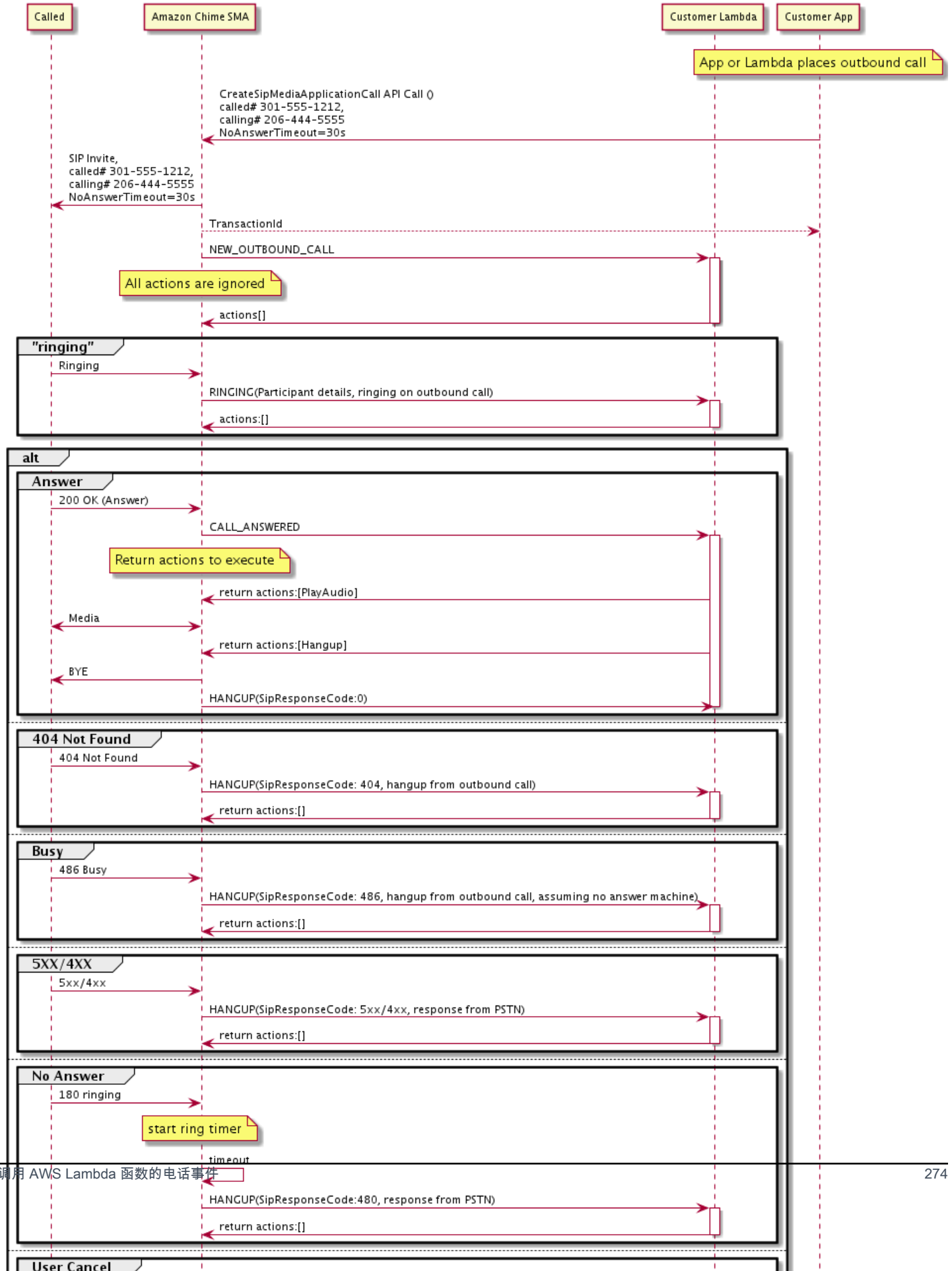
进行出站呼叫

要创建出站呼叫，您可以使用 [CreateSipMediaApplicationCall](#) API。API 调用指定 SIP media application ID 的终端节点。客户可以通过从终端节点提供不同的信令和 [SipMediaApplication](#) 操作来控制呼叫流程。

如果成功响应，API 会返回 202 http 状态码以及 transactionId，您可以将其与 [UpdateSipMediaApplicationCall](#) API 一起使用来更新正在进行的呼叫。

下图显示出站呼叫对 AWS Lambda 函数终端节点进行的呼叫。

CreateSipMediaApplicationCall() Behavior



针对出站呼叫的不同状态调用为 SIP 媒体应用程序配置的终端节点。当客户发起呼叫时，Amazon Chime SDK 会使用 NEW_OUTBOUND_CALL 调用事件类型调用终端节点。

此示例显示 NEW_OUTBOUND_CALL 的典型调用事件。

```
{
  "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": "+1xxxxxxxx",
        "Direction": "Outbound",
        "StartTimeInMilliseconds": "159700958834234"
      }
    ]
  }
}
```

任何与事件相关的 AWS Lambda 调用的响应都将被忽略。

收到接收器的 RINGING 通知时，Amazon Chime SDK 会再次调用已配置的终端节点。

此示例显示 RINGING 的典型调用事件。

```
{
  "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"
    }
  ]
}

```

任何与事件相关的 AWS Lambda 调用的响应都将被忽略。

如果接收器没有应答呼叫，或者由于错误导致呼叫失败，Chime 会断开呼叫并使用 Hangup 事件类型调用终端节点。有关 Hangup 事件类型的更多信息，请参阅 [结束呼叫](#)。

如果呼叫得到应答，Chime 会通过 CALL_ANSWERED 操作调用终端节点。此示例显示典型调用事件。

```

{
  "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"
      }
    ]
  }
}

```

此时，您可以通过使用操作列表响应调用来返回操作。如果不希望执行任何操作，则使用空列表来响应。对于每次 AWS Lambda 调用，您最多可以响应 10 次操作，并且每次呼叫可以调用 1,000 次 Lambda 函数。有关使用操作集进行响应的更多信息，请参阅 [使用操作列表响应调用](#)。

接听入站呼叫

当 NEW_INCOMING_CALL 事件发生时，音频服务会创建一个唯一 TransactionID 的唯一的 CallID，一直持续直到 HANGUP 事件发生。

您可以通过多种方式对 NEW_INCOMING_CALL 事件做出响应。例如：

- 发送 PlayAudio 或 RecordAudio 操作并自动接听电话。
- 发送 Pause 操作。
- 发送 Hangup 操作，在这种情况下，呼叫不被接听，也不会向客户收费。
- 发送 CallAndBridge 操作并将其他用户添加至呼叫中。
- 什么都不做，呼叫会在 30 秒后超时。

当接到新的入站呼叫时，SIP 媒体应用程序会使用此有效负载调用 AWS Lambda 函数。

```
{
  "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"
      }
    ]
  }
}
```

```
}  
}
```

指定响应电话事件的操作

在音频服务中，SIP 媒体应用程序调用 AWS Lambda 函数。反过来，Lambda 函数可以返回称为操作的指令列表。操作是您要在电话呼叫分支上运行的项目，例如发送或接收数字、加入会议等。有关 PSTN 音频服务调用的操作的更多信息，请参阅 [了解电话事件](#)。

当 SIP 媒体应用程序成功运行操作列表时，该应用程序会调用调用事件类型为 ACTION_SUCCESSFUL 的 AWS Lambda 函数。如果任何操作未能完成，SIP 媒体应用程序会使用 ACTION_FAILED 事件调用 AWS Lambda 函数。

仅当列表中的所有操作都成功时，SIP 媒体应用程序才会返回 ACTION_SUCCESSFUL。如果列表中的任何操作失败，SIP 媒体应用程序将使用 ACTION_FAILED 事件调用 AWS Lambda 函数，并会清除列表中失败操作之后的剩余操作。然后，SIP 媒体应用程序会运行 AWS Lambda 函数返回的下一个操作。您可以使用 ActionData 密钥识别呼叫调用的函数。

以下事件显示了 PlayAudioAndGetDigits 操作后 ACTION_SUCCESSFUL 调用事件类型的示例有效负载。

```
{  
  "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"
        }
    ]
}
}
}
}
}

```

当列表中的任何操作未能成功完成时，SIP 媒体应用程序会调用 AWS Lambda 函数以通知您操作失败，并获取一组新的操作以在该呼叫上运行。以下事件显示了 PlayAudio 操作后 ACTION_FAILED 调用事件类型的示例有效负载。

```

{
    "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"
    }
  ]
}
}
}

```

接收来电者输入

您可以使用 `ReceiveDigits` 操作收集入站 DTMF 数字并将其与正则表达式进行匹配。当 SIP 媒体应用程序收到与正则表达式匹配的数字时，它会通过 `ACTION_SUCCESSFUL` 事件调用 AWS Lambda 函数。收集的数字显示在 `ActionData` 对象的 `ReceivedDigits` 值中。

例如：

```

{
  "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"
        }
    ]
}
}
}

```

一旦呼叫者输入与您的正则表达式模式相匹配的数字，SIP 媒体应用程序就会调用 AWS Lambda 函数，该函数返回以下类型的有效负载：

```

{
    "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"
  }
]
}
```

在 GitHub 上查看工作示例：<https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>

更新正在进行的调用

作为 PSTN 音频服务的一部分，SIP 媒体应用程序允许您根据调用事件（例如传入调用或 DTMF 数字）调用用户定义的 Lambda 函数，从而设置在调用过程中运行的操作。[UpdateSipMediaApplicationCall](#) API 允许您在调用处于活动状态时随时触发 Lambda 函数，将当前操作替换为调用返回的新操作。

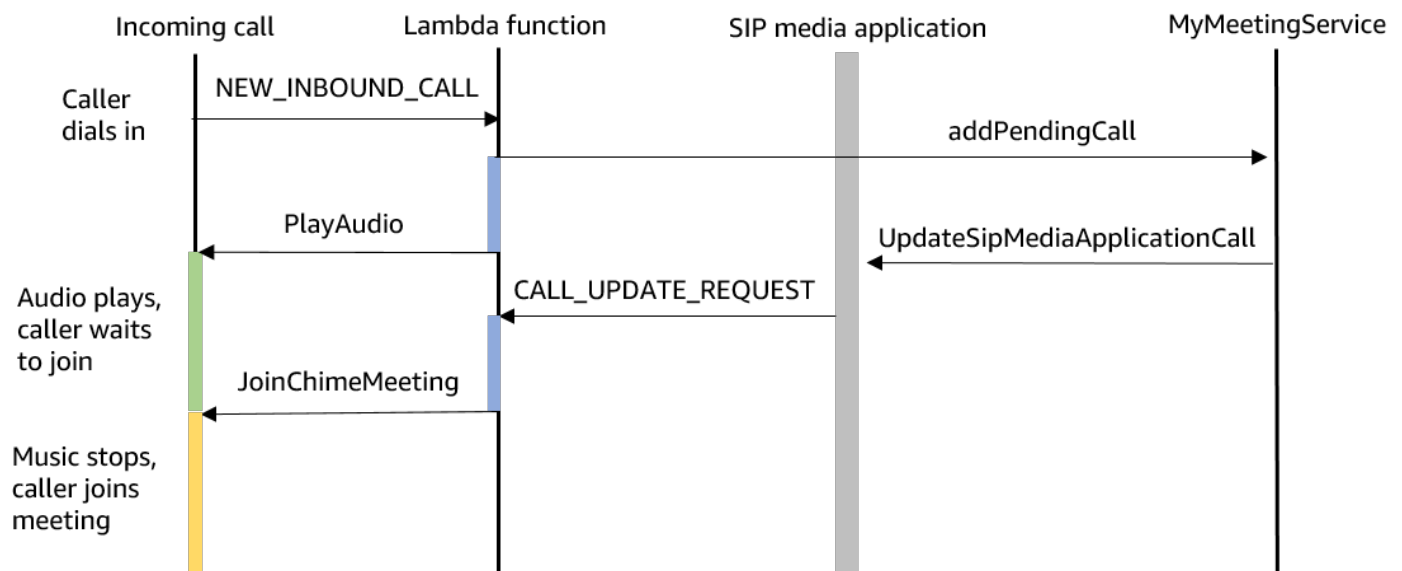
工作流程

您可以在各种情况下使用 [UpdateSipMediaApplicationCall](#) API，例如向会议添加参与者、将用户静音和取消静音、断开连接等。以下使用案例描述了典型的工作流程。

当 Amazon Chime SDK 设置会议时，用户调用并听音乐。设置完成后，Amazon Chime SDK 会停止播放音频并允许调用者加入会议。然后，假设使用单独的系统 MyMeetingService 管理会议。每个传入调用都应置于保持状态。Chime 会通知 MyMeetingService 有关传入调用的信息，然后 MyMeetingService 为每个调用创建与会者；当 MyMeetingService 准备好开始会议时，它会通知 SIP 媒体应用程序并提供加入会议的令牌。

要处理这种情况，Lambda 函数必须实现以下逻辑。

- 当新的传入调用到达时，Lambda 将使用 `NEW_INBOUND_CALL` 事件调用。Lambda 调用 `MyMeetingService` 并传递标识当前调用的 `transactionId`，然后返回 `PlayAudio` 操作。
- 当 `MyMeetingService` 准备好添加调用者到会议中时，该服务会调用 [UpdateSipMediaApplicationCall](#) API，并将调用的 `transactionId` 和 `JoinToken` 为其参数的一部分进行传递。现在，此 API 调用使用 `CALL_UPDATE_REQUESTED` 事件再次触发 Lambda 函数。MyMeetingService 将 `JoinToken` 作为事件的一部分传递给 Lambda 函数，且令牌用于将 `JoinChimeMeeting` 操作返回给 SIP 媒体应用程序，该应用程序会中断 `PlayAudio` 操作并将调用者连接到会议。



Note

[UpdateSipMediaApplicationCall](#) API 返回 HTTP 202 (已接受) SIP 媒体应用程序确认调用正在进行并且可以更新，因此它会尝试调用 Lambda 函数。调用异步执行，因此 API 的成功响应并不能保证 Lambda 函数已启动或已完成。

以下示例显示请求语法。

```

{
  "SipMediaApplicationId": "string",
  "TransactionId": "string",
  "Arguments": {
    "string": "string"
  }
}
  
```

```
}

```

请求参数

- SipMediaApplicationId — 处理调用的 SIP 媒体应用程序的 ID。
- TransactionId — 调用事务的 ID。对于进站调用，TransactionId 可以从首次调用时传递给 Lambda 函数的 NEW_INCOMING_CALL 事件中获取。对于出站调用，TransactionId 会在 [CreateSipMediaApplicationCall](#) 的响应中返回。
- 参数 — 作为 CallUpdateRequest 操作数据一部分提供给 Lambda 函数的自定义参数。可包含 0 到 20 个密钥值对。

以下示例显示典型请求。

```
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"}'
```

响应语法

```
{
  "SipMediaApplicationCall": {
    "TransactionId": "string"
  }
}
```

响应元素

- TransactionId — 调用事务的 ID，与请求 ID 相同。

以下示例代码显示 CALL_UPDATE_REQUESTED 调用事件。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 2,
  "InvocationEventType": "CALL_UPDATE_REQUESTED",
  "ActionData": {
    "Type": "CallUpdateRequest",
    "Parameters": {
      "Arguments": {
```

```

    "string": "string"
  }
}
},
"CallDetails": {
  ...
}
}

```

事件元素

- SchemaVersion — JSON 架构 (1.0) 的版本
- 序列 — 调用中事件的序列号
- InvocationEventType — Lambda 调用事件的类型，在本例中为 CALL_UPDATE_REQUESTED
- ActionData — 与 CallUpdateRequest 操作有关的数据。
 - 类型 — 操作的类型，在本例中为 CallUpdateRequest
 - 参数 — 操作的参数
 - 参数 — 作为 UpdateSipMediaApplicationCall API 请求的一部分传递的参数
- CallDetails — 有关当前调用状态的信息

了解可中断和不间断操作

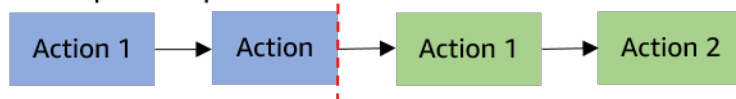
当 Lambda 函数返回新的操作列表是现有操作仍在运行时，所有正在进行的操作之后的操作都将被替换为新操作。在某些情况下，Lambda 函数会中断正在进行的操作，以便立即运行新操作。

以下图表显示典型示例。图表下方的文字解释逻辑。

Initial actions



Interrupt and replace



Replace after completion



如果操作 2 可中断，我们将其停止并运行新的操作 1。

如果操作 2 不可中断，它将在完成后运行新的操作 1。

在这两种情况下，都不会运行操作 3。

如果某个操作被中断，则使用 ACTION_INTERRUPTED 事件调用 Lambda 函数。此事件仅用于提供信息。SIP 媒体应用程序会忽略此调用返回的所有操作。

可中断操作的类型：

- PlayAudio
- RecordAudio
- Pause

示例 Lambda 函数

此示例显示典型的 Lambda 函数，该函数用于播放音频文件、传递加入令牌并更新调用。

```
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();
}
}
```

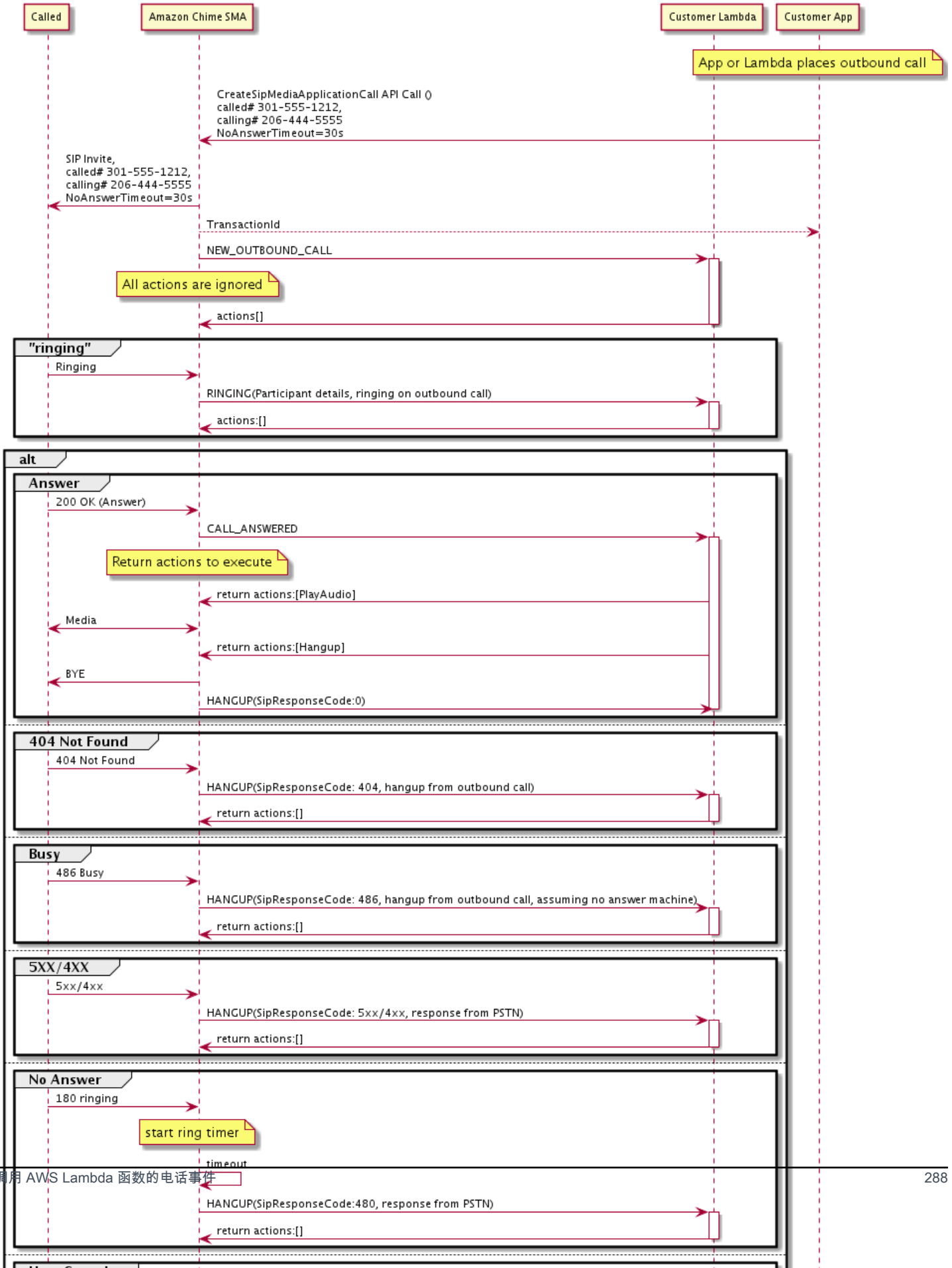
结束呼叫

您可以使用 [CreateSipMediaApplicationCall](#) API 结束出站呼叫。API 调用指定 SIP 媒体应用程序 ID 的终端节点。客户可以通过向 SIP 媒体应用程序返回操作来控制呼叫流程。

如果成功响应，API 会返回 202 http 状态码以及 `transactionId`，您可以将其与 [UpdateSipMediaApplicationCall](#) API 一起使用以更新正在进行的呼叫。

下图显示出站呼叫对 AWS Lambda 函数终端节点进行的呼叫。

CreateSipMediaApplicationCall() Behavior



调用 AWS Lambda 函数的电话事件

针对出站呼叫的不同状态调用为 SIP 媒体应用程序配置的终端节点。当客户结束呼叫时，Amazon Chime SDK 会使用 HANGUP 调用事件类型调用终端节点。

此示例显示 HANGUP 的典型调用事件。

```
{
  "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"
        }
    ]
}
}
```

端到端呼叫示例

此使用案例提供示例代码，用于接听 PSTN 呼叫者的电话、用音频消息向呼叫者打招呼、从呼叫者那里获取会议 PIN 码、播放音频以及让呼叫者加入会议。

调用事件和操作

音频服务将调用事件作为 JSON 对象传递给 AWS Lambda 函数。这些对象包括调用事件类型和任何相关的元数据。AWS Lambda 函数还以 JSON 对象的形式返回 SIP 媒体应用程序操作，这些对象包括操作类型和任何相关的元数据。

下表列出了您收到调用事件时的调用事件以及可能的 `ActionData.Type`。

调用事件	ActionData.Type
ACTION_SUCCESSFUL	CallAndBridge ReceiveDigits PlayAudio PlayAudioAndGetDigits JoinChimeMeeting ModifyChimeMeetingAttendees RecordMeeting
ACTION_FAILED	CallAndBridge PlayAudio PlayAudioAndGetDigits ModifyChimeMeetingAttendees RecordMeeting
挂断	HangUp
DIGITS_RECEIVED	ReceiveDigits

Note

要实现以下使用案例，您需要在您的 Amazon Chime SDK 库存中至少有一个电话号码、一个使用带 Amazon Resource Name (ARN) 的 AWS Lambda 函数的 SIP 媒体应用程序托管对象，以及一个使用电话号码作为触发器的 SIP 规则。

当 Amazon Chime SDK 接到规则中指定的电话号码的呼叫时，PSTN 音频服务会通过 NEW_INBOUND_CALL 调用事件类型调用 AWS Lambda 函数。

```
{
  "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"
      }
    ]
  }
}
```

您可以对 AWS Lambda 函数进行编程以验证呼叫详细信息并将其存储以备将来使用。对于 NEW_INBOUND_CALL 事件，AWS Lambda 函数会以一组操作进行响应，这些操作会播放欢迎提示并要求提供会议 PIN 码。

音频文件具有以下要求：

- 您必须播放 Amazon Simple Storage Service (S3) 存储桶中的音频文件。S3 存储桶必须与 SIP 媒体应用程序处于同一 AWS 账户。此外，您必须向 Amazon Chime SDK 语音连接器服务主体 `voiceconnector.chime.amazonaws.com` 授予 `s3:GetObject` 权限。您可以使用 S3 控制台或命令行界面 (CLI) 完成此操作。
- 您必须使用大小不超过 50 MB 的 PCM WAV 文件。Amazon Chime SDK 推荐 8kHz 单声道。
- 每个 WAV 文件的 S3 元数据必须包含 `{'ContentType': 'audio/wav'}`。

```

{
  "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 媒体应用程序在调用分支 A 上运行这些操作。假设 PlayAudioAndGetDigits 操作收到数字，则 SIP 媒体应用程序将使用 ACTION_SUCCESSFUL 事件类型调用 AWS Lambda 函数。

```

{
  "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
  }
}
}

```

您可以对 AWS Lambda 函数进行编程，以根据 CallDetails 数据识别呼叫者。您也可以验证之前收到的会议 PIN。假设 PIN 正确，您可以使用 [CreateMeeting](#) 和 [CreateAttendee](#) API 创建 Amazon Chime SDK 会议并生成会议与会者使用的加入令牌。AWS Lambda 函数以加入 Amazon Chime SDK 会议的操作作为响应。

```

{
  "SchemaVersion": "1.0",
  "Actions": [
    {

```



```

        "Type": "JoinChimeMeeting",
        "Parameters": {
            "JoinToken": "meeting-attendee-join-token"
        }
    }
]
}

```

假设 JoinToken 有效，则 SIP 媒体应用程序将加入 Amazon Chime SDK 会议并使用 ACTION_SUCCESSFUL 事件调用 AWS Lambda 函数，其中 CallDetails 包含来自 SIP 媒体应用程序和 Chime 媒体服务 (LEG-B) 的数据

```

{
  "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"
    }
  ]
}
}

```

如果您想在此时停止在调用或调用分支上运行操作，则可以使用一组空操作进行响应。

```

{
  "SchemaVersion": "1.0"
  "Actions": []
}

```

呼叫者挂断后，SIP 媒体应用程序使用 HANGUP 事件调用 AWS Lambda 函数。

```

{
  "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"
    }
  ]
}
```

如果您使用操作来响应 Hangup 事件，若没有其他 Participants 显示 Connected 的 Status，则 SIP 媒体应用程序将忽略该操作。

使用操作列表响应调用

您可以使用要对 AWS Lambda 呼叫中的各个参与者执行的操作列表来响应调用事件。每次调用最多可以使用 10 个操作进行响应，并且每次 AWS Lambda 调用可以调用 1,000 次 AWS Lambda 函数。

默认情况下，如果 Lambda 函数在 20 秒内没有响应，SIP 媒体应用程序就会超时。

下面显示一般响应结构。

```
{
  "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"
      }
    }
  ]
}
```

当该 AWS Lambda 函数向 SIP 媒体应用程序返回操作列表时，会发生以下操作：

1. 应用程序完成对呼叫的当前操作的运行。
2. 然后，应用程序将旧的操作集替换为从最新调用事件接收的一组新操作。

如果 SIP 媒体应用程序收到的操作集为 NULL，则它会保留现有操作。

PSTN 音频服务支持的操作

您可以在 AWS Lambda 函数的响应中指定不同类型的信令和媒体操作。每个操作都有不同的属性。以下主题提供示例代码并介绍了如何使用操作。

内容

- [使用 TransactionAttributes](#)
- [使用通话录音](#)
- [CallAndBridge](#)
- [挂断](#)
- [JoinChimeMeeting](#)
- [ModifyChimeMeetingAttendee \(音频静音和取消静音 \)](#)
- [Pause](#)
- [PlayAudio](#)
- [PlayAudioAndGetDigits](#)
- [ReceiveDigits](#)
- [RecordAudio](#)

- [SendDigits](#)
- [Speak](#)
- [SpeakAndGetDigits](#)
- [StartBotConversation](#)

使用 TransactionAttributes

您可以使用 TransactionAttributes 数据结构存储应用程序特定的信息，例如调用状态或会议 ID，然后将这些数据传递至 AWS Lambda 调用。这种结构无需将数据存储至外部数据库（例如 Amazon DynamoDB）中。

TransactionAttributes 是包含密钥/值对的 [JSON 对象](#)。这些对象最多可以包含 100 个密钥键/值对，并且这些对象的最大有效负载大小为 20KB。TransactionAttributes 结构中的数据会在交易的生命周期内保持存在。

当 AWS Lambda 函数传递 TransactionAttributes 至 SIP 媒体应用程序时，该应用程序会更新所有存储的属性。如果您传递具有现有密钥集的 TransactionAttributes 对象，则会更新存储的值。如果您传递不同的密钥集，则将现有值替换为来自该密钥集的值。传递一个空的 map ({}) 可擦除所有存储的值。

主题

- [设置 TransactionAttributes](#)
- [更新 TransactionAttributes](#)
- [清除 TransactionAttributes](#)
- [处理 ACTION_SUCCESSFUL 事件](#)
- [无效输入](#)

设置 TransactionAttributes

以下示例说明如何在 [PlayAudio](#) 操作时设置 TransactionAttributes 并将属性从 AWS Lambda 函数传递给 SIP 媒体应用程序。

```
{
  "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"
  }
}
```

更新 TransactionAttributes

要修改存储的 TransactionAttributes，使用新值更新 JSON 对象的内容。在以下示例中，密钥 NewKey1 和 NewKey2 已添加至 TransactionAttributes。这些密钥分别与值 NewValue1 和 NewValue2 配对。

```
{
  "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"
  }
}
```

在前面的示例中，如果您将 `NewValue1` 传递给 `key1`，则 `key1` 的现有值将被替换为 `NewValue1`。但是，将值传递给 `NewKey1` 会创建新的密钥/值对。

清除 TransactionAttributes

要清除 `TransactionAttributes` 对象的内容，使用空的 JSON 对象传递 `TransactionAttributes` 字段：

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "PlayAudio",
      "Parameters": {
        "ParticipantTag": "LEG-A",
        "AudioSource": {
          "Type": "S3",
          "BucketName": "mtg1-sipmedia-app-iad",
          "Key": "Welcome3.wav"
        }
      }
    }
  ],
  "TransactionAttributes": {}
}
```

Note

无法通过将值设置为 `null` 来清除 `TransactionAttributes` 结构中的数据。此外，忽略 `TransactionAttribute` 结构不能清除其数据。始终使用 `TransactionAttributes` 传递空的 JSON 对象以将对象从数据中清除。

处理 ACTION_SUCCESSFUL 事件

以下示例显示了成功的 [PlayAudio](#) 如何将已存储的 `TransactionAttributes` 作为 `CallDetails` 的一部分发送。

```
{
  "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"
    }
  ]
}
}

```

无效输入

以下示例显示了无效输入。在这种情况下，JSON 对象向 SIP 媒体应用程序传递的项目过多。

```

{
  "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"
}
}

```

以下示例显示了对先前给定输入的响应。此输出从 SIP 媒体应用程序传回至调用该应用程序的 AWS Lambda 函数。

```

{
  "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"
}
```

使用通话录音

SIP 媒体应用程序的通话录音操作使您能够为各种用途构建通话录音和通话后转录解决方案。例如，您可以录制客户服务电话并将其用于培训。

您可以将呼叫录音操作与 SIP 媒体应用程序配合使用。您也可以按需使用这些操作或响应 SIP 事件。

- 要在 SIP 媒体应用程序中开始按需录制呼叫，可使用 [UpdateSipMediaApplication](#) API 调用您的应用程序并返回 [StartCallRecording](#) 操作。
- 要开始通话录音以响应 SIP 事件，可在应用程序中返回 [StartCallRecording](#) 操作。

您可以暂停和恢复正在进行的录音。要暂停，则使用 [PauseCallRecording](#) 操作。要恢复，则使用 [ResumeCallRecording](#) 操作。每次暂停或恢复录音时，相应操作都会捕获一个提示暂停或恢复的音调。暂停时，操作记录静音，即 Amazon Chime SDK 跟踪暂停时长并将暂停时间计入账单中。您可以根据需要暂停和恢复录音。

要停止通话录音，则返回 [StopCallRecording](#) 操作。而当通话停止时，通话录音会自动停止，在这种情况下，您无需明确返回 [StopCallRecording](#) 操作。对于单个呼叫分支，您只能开始和停止录音一次。

Amazon Chime SDK 将通话录音发送至您选择的 Amazon S3 存储桶中。Amazon S3 存储桶必须属于您的 AWS 账户。呼叫停止后，SIP 媒体应用程序会将录音传送到 [StartCallRecording](#) 操作的 `Destination` 参数中指定的文件夹。Amazon Chime SDK 以开放的 WAV 格式记录通话。录制传入和传出轨道的呼叫使用立体声模式，传入轨道位于左声道，传出轨道位于右声道。如果您只录制传入或传出轨道，则系统将使用单声道模式。

Note

使用此功能录制的录音可能会受有关电子通信录制的法律或法规的约束。您和您的最终用户有责任遵守与录音有关的所有适用法律，包括适当通知录制会话或通信中的所有参与者正在录制会话或通信，并获得他们的许可。

通话录音计费

Amazon Chime SDK 按分钟时长向您收取呼叫分支启用通话录音的费用，该时间包括所有暂停时间。将通话录音传送到您的 Amazon S3 存储桶后，您需要为通话录音使用量付费。

录制音频轨道

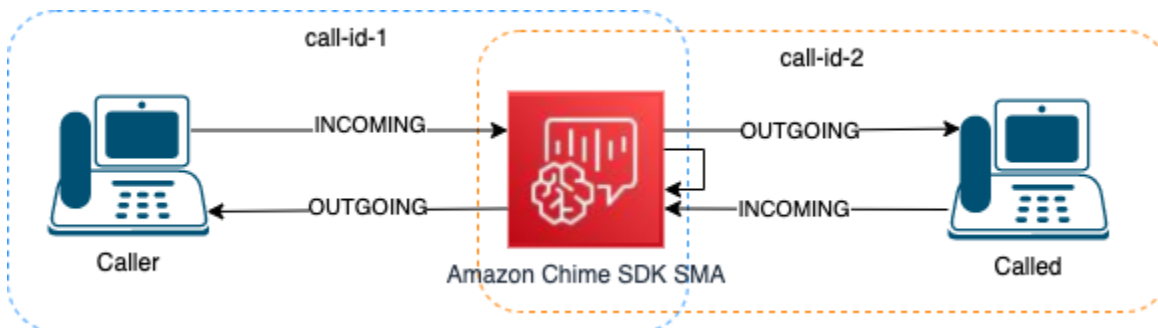
您可以只录制传入或传出呼叫轨道，也可以同时录制两个呼叫轨道。

此图显示了典型的单个分支或非桥接来电。



该呼叫只有一条分支，callID 为 call-id-1。INCOMING 音频轨道是从呼叫者到 SIP 媒体应用程序的音频。OUTGOING 音频轨道是从 SIP 媒体应用程序到呼叫者的音频。您的 SIP 媒体应用程序指定您要录制的呼叫的 CallId。要录制拨打电话的参与者，则指定 INCOMING。要录制接听电话的参与者，则指定 OUTGOING。要录制两个参与者，则指定 BOTH。

此图显示带有两个参与者的典型桥接呼叫。



在此示例中，呼叫有两个呼叫分支，call-id-1 和 call-id-2，并且 call-id-1 桥接至 call-id-2。这会创建四个音频轨道，即两个呼叫 ID 的传入和传出音频流。您可以指定要录制呼叫 ID 和音频轨道。例如，如

如果要录制被叫参与者的音频轨道，则可以通过将 `call-id-2` 指定为 `CallId` 并将 `INCOMING` 指定为轨道录制 `INCOMING` 音频轨道。

如果要录制呼叫者听到的所有内容，则可以通过将 `call-id-1` 指定为 `CallId` 并将 `OUTGOING` 指定为轨道录制 `OUTGOING` 音频轨道。如果要录制 `Caller` 所说和听到的所有音频，则可以通过将 `call-id-1` 指定为 `CallId` 并将 `BOTH` 指定为轨道来录制 `BOTH` 音频轨道。

使用案例示例

SIP 媒体应用程序提供通话录音操作作为构建模块。它们使您可以灵活地为您的业务使用案例构建通话录音解决方案。以下案例说明一些常见的使用场景。

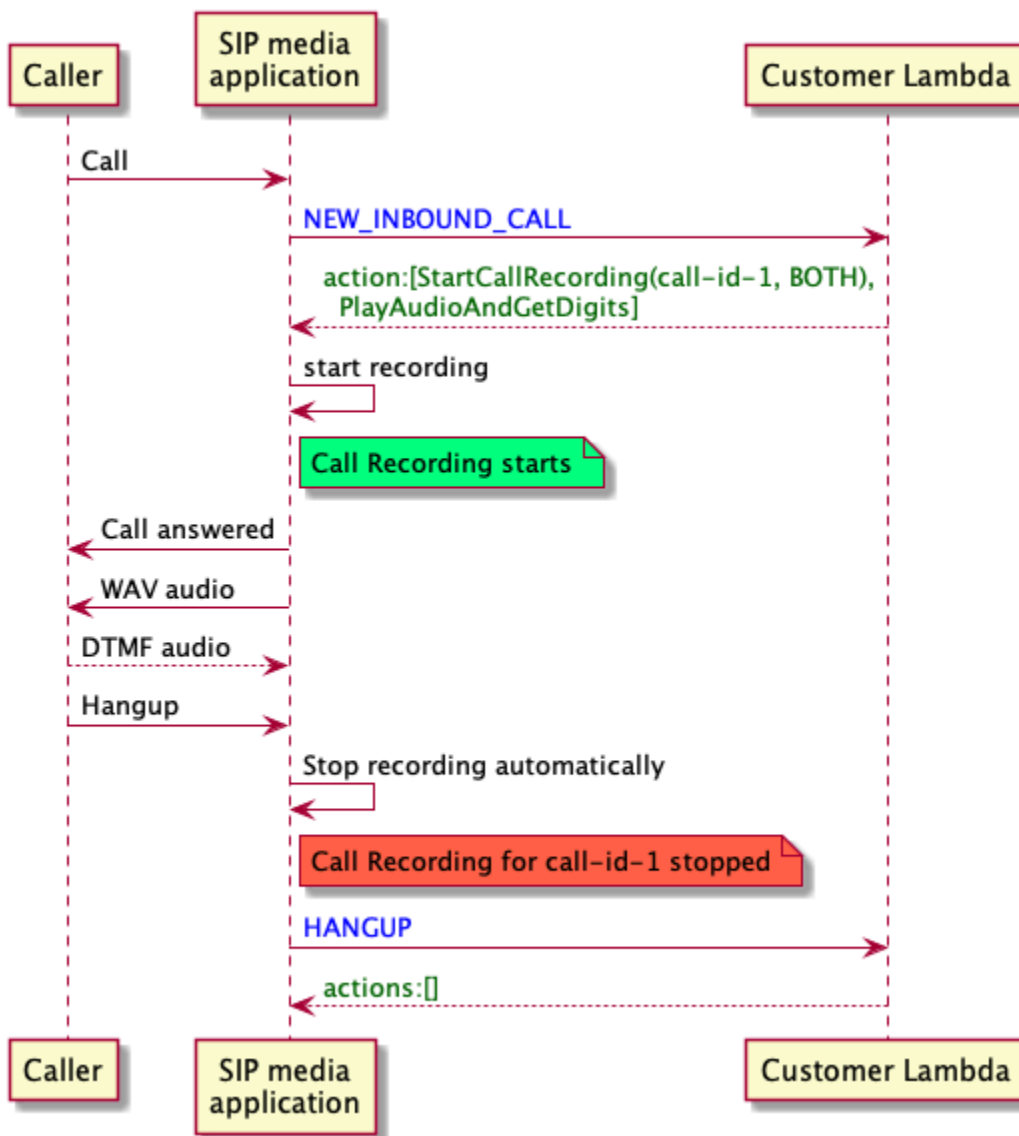
主题

- [案例 1：录制涉及 SIP 操作的单个分支呼叫](#)
- [案例 2：在桥接通话中有选择地录制音频](#)
- [案例 3：录制多个通话分支](#)
- [案例 4：按需录制（包括暂停和恢复）](#)

案例 1：录制涉及 SIP 操作的单个分支呼叫

您可以录制呼叫者和由 SIP 媒体应用程序操作生成的任何音频，例如 [PlayAudio](#) 和 [PlayAudioAndGetDigits](#) 操作。在录制过程中，如果呼叫者按下某个数字，则录音会捕获该数字的音调。此示例使用 `PlayAudioAndGetDigits` 操作，但交互式语音应答 (IVR) 可能是一系列复杂的 SIP 媒体应用程序操作。

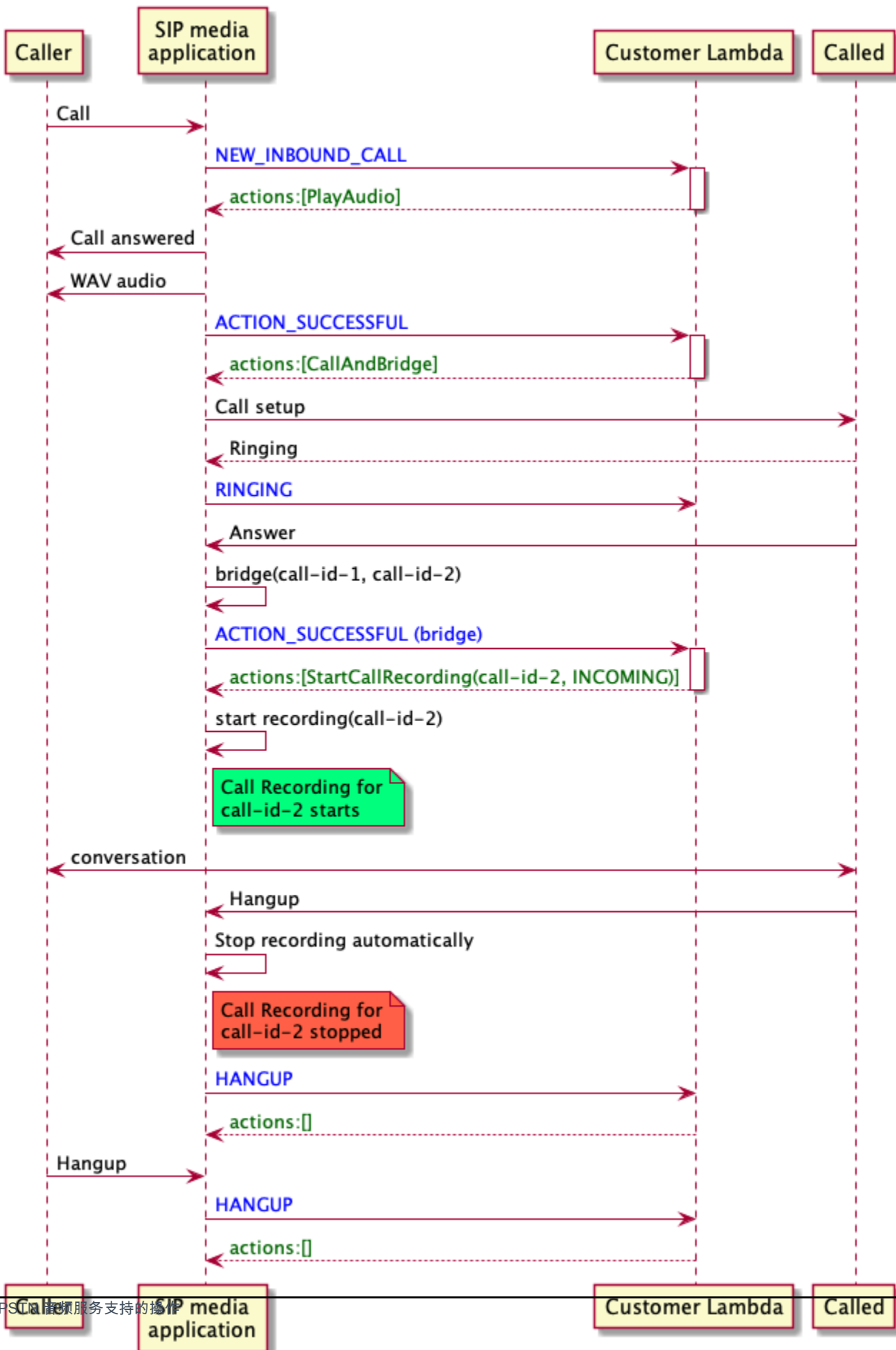
在此示例中，SIP 媒体应用程序会记录呼叫者与 SIP 媒体应用程序本身之间的音频轨道。建立呼叫后，录音开始，并在呼叫者挂断后停止录音。建立呼叫后，计费开始，并在呼叫者挂断后停止计费。



案例 2：在桥接通话中有选择地录制音频

您可以有选择地录制单个呼叫参与者的音频轨道。您可以使用此功能有选择地启用仅针对特定参与者的通话录音。

在此示例中，SIP 媒体应用程序通过将 `call-id-2` 指定为 `CallId` 并将 `INCOMING` 指定为轨道来记录被叫方与 SIP 媒体应用程序本身之间的传入音频轨道。通话录音从呼叫者桥接到被叫方时开始，同时也开始计费。当被叫方挂断后，录音就会停止，同时计费也相应停止。此录音仅包含被叫方的音频轨道。

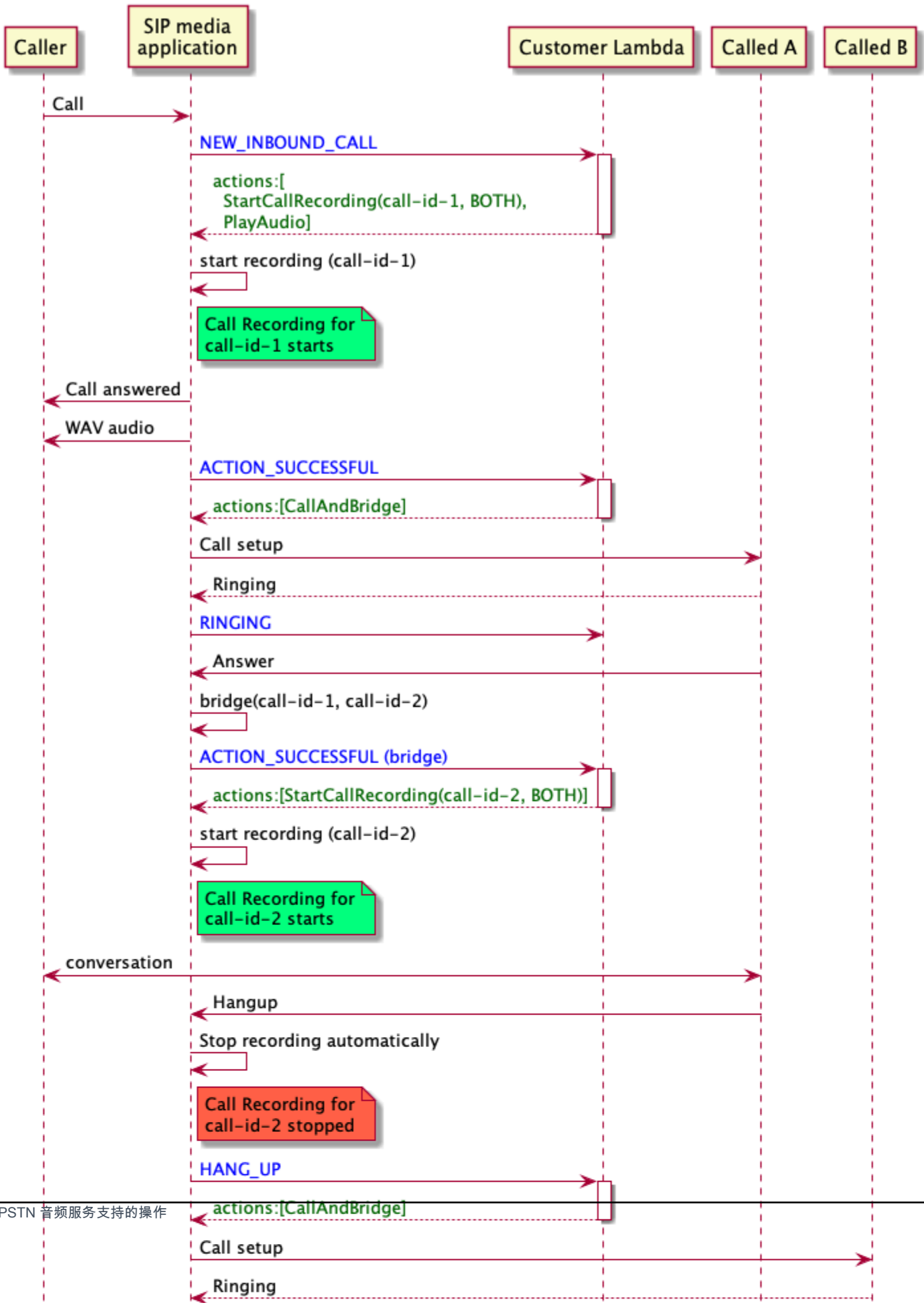


案例 3：录制多个通话分支

您可以录制多个呼叫分支。例如，假设您将呼叫桥接到参与者。当该参与者挂断电话时，电话将桥接到另一位参与者。您可以为所有三个呼叫分支启用通话录音。

此示例显示了三个单独的录音文件。第一个呼叫分支的录音捕获呼叫者、您的应用程序以及桥接到通话中的两个参与者之间的对话。第二个呼叫分支的录音捕获呼叫者和第一个参与者之间的对话。第三个呼叫分支的录音捕获呼叫者和第二个参与者之间的对话。

此案例创建了三个呼叫分支，计费适用于每个呼叫分支的开始与结束之间。也就是说，系统会向您的 S3 存储桶传送三段录音，每段都需要付费。

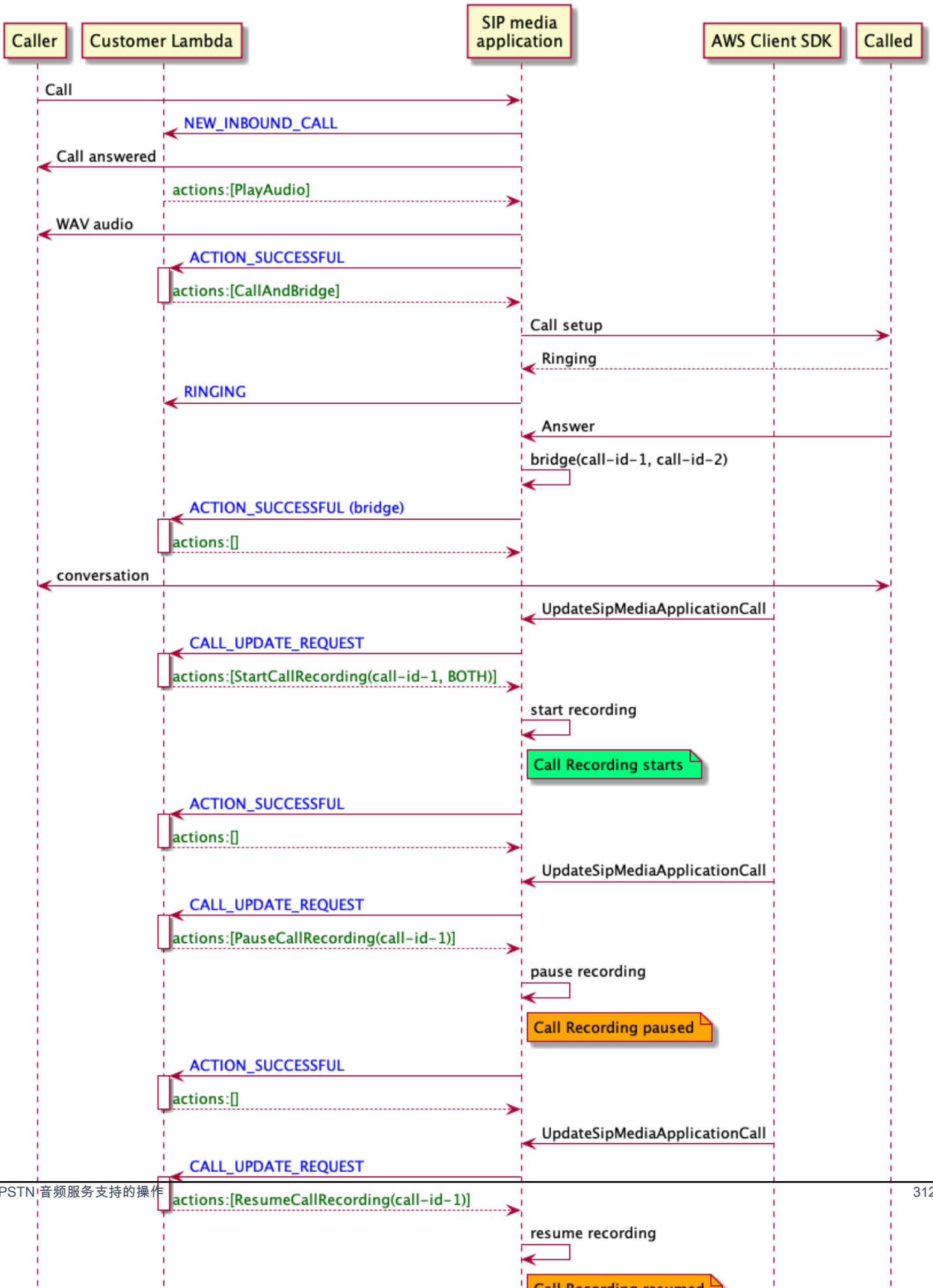


案例 4：按需录制（包括暂停和恢复）

您可以使用 [UpdateSipMediaApplicationCall](#) API 按需开始、停止、暂停和恢复通话录音。您可以构建一个调用 `UpdateSipMediaApplicationCall` API 并调用 SIP 媒体应用程序以返回呼叫录音操作的客户端应用程序。

您的最终用户使用客户端应用程序控制通话录音。例如，在呼叫中心中，座席将使用桌面客户端应用程序按需触发呼叫录音操作。在呼叫中心示例中，座席可能会征求呼叫者的同意以允许录制通话，并且在同意后在客户端应用程序中单击以开始录音。在另一个示例中，呼叫者可能需要提供诸如社会保险号 (SSN) 之类的信息。但是，呼叫中心政策要求座席不得记录诸如客户的 SSN 之类的信息。当客户提供信息时，座席可以单击应用程序暂停录音，然后再次单击恢复录制。座席处理呼叫者的请求后，单击应用程序以停止录音并挂断呼叫。

在此使用案例中，SIP 媒体应用程序会记录呼叫者和 SIP 媒体应用程序之间的音频轨道。由于 `call-id-1` 分支桥接到 `call-id-2` 分支，因此系统会录制两条分支上的音频，包括呼叫者和被叫者。录制和计费从 `UpdateSipMediaApplicationCall` API 调用 `StartCallRecording` 操作时开始。录制和计费从 `UpdateSipMediaApplicationCall` API 调用 `StopCallRecording` 操作时停止。作为提醒，暂停录制不会更改其持续时间，并且您需要为所有暂停时间付费。



SIP 媒体应用程序的通话录音操作

您可以在 SIP 媒体应用程序 AWS Lambda 函数的响应中指定不同的呼叫录音操作。以下主题提供示例代码并介绍了如何使用操作。

主题

- [StartCallRecording](#)
- [StopCallRecording](#)
- [PauseCallRecording](#)
- [ResumeCallRecording](#)

StartCallRecording

StartCallRecording 操作开始呼叫分支的录音。您可以在 SIP 媒体应用程序中按需或作为 SIP 事件的响应开始通话录音。

- 要开始按需录制通话，您可以使用 UpdateSipMediaApplication API 调用您的应用程序并返回 StartCallRecording 操作。
- 要开始通话录音以响应 SIP 事件，可在应用程序中返回 StartCallRecording 操作。

您可以指定录制传入分支的音频轨道和/或传出分支的音频轨道。下面几个部分介绍如何使用 StartCallRecording 操作。

Note

使用此功能录制的录音可能会受有关电子通信录制的法律或法规的约束。您和您的最终用户有责任遵守与录音有关的所有适用法律，包括适当通知录制会话或通信中的所有参与者正在录制会话或通信，并获得他们的许可。

主题

- [请求 StartCallRecording 操作](#)
- [指定录制目标](#)
- [授予 Amazon S3 存储桶权限](#)
- [操作成功响应](#)

- [操作错误响应](#)

请求 StartCallRecording 操作

以下示例显示如何为 BOTH 轨道请求 StartCallRecording 操作。

```
{
  "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

描述 — AWS Lambda 函数调用的 CallDetails 中参与者的 CallId

允许的值 — 有效的调用 ID

必填 — 是

默认值 — 无

Track

描述 — 通话录音的音频 Track。

允许的值 — BOTH、INCOMING 或 OUTGOING

必填 — 是

默认值 — 无

Destination.Type

描述 — 目的地类型。只允许使用 Amazon S3。

允许的值 — Amazon S3

必填 — 是

默认值 — 无

Destination.Location

描述 — 有效的 Amazon S3 存储桶和可选 Amazon S3 键前缀。存储桶必须有权访问 Amazon Chime SDK 语音连接器服务主体，即 `voiceconnector.chime.amazonaws.com`。

允许的值 — Amazon Chime SDK 有权访问 `s3:PutObject` 和 `s3:PutObjectAcl` 操作的有效 Amazon S3 路径。

必填 — 是

默认值 — 无

指定录制目标

Amazon Chime SDK 将通话录音发送至您的 Amazon S3 存储桶。该存储桶必须属于您的 AWS 账户。您需要在 `StartCallRecording` 操作的 `Destination` 参数中指定存储桶的位置。`Destination` 参数中的 `Type` 字段必须是 `S3`。`Location` 字段由您的 Amazon S3 存储桶以及用于传送通话录音的可选对象键前缀组成。

SIP 媒体应用程序使用指定的 `Location`、呼叫分支的日期和时间、事务 ID 和呼叫 ID 格式化 Amazon S3 对象键。`StartCallRecording` 操作响应会返回完整的 Amazon S3 对象键。

当您仅在 `Location` 字段中提供 Amazon S3 存储桶时，SIP 媒体应用程序会在 Amazon S3 路径上附加一个默认前缀 `Amazon-Chime-SMA-Call-Recordings`。SIP 媒体应用程序还会附加通话开始时间的年、月和日，以帮助梳理录音。以下示例显示带有默认前缀的一般格式的 Amazon S3 路径。此示例使用 `myRecordingBucket` 作为 `Location` 值。

```
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
```

以下示例显示通话录音 Amazon S3 路径中表示的数据。

```
s3Bucket/Amazon-Chime-SMA-Call-Recordings/year/month/date/year-month-date-hour-minute-second-millisecond_transactionId_callId.wav
```

当您在 Location 字段中提供 Amazon S3 存储桶和对象键前缀时，SIP 媒体应用程序将在目标 Amazon S3 路径中使用您的对象键前缀，而非默认前缀。以下示例显示带有您的前缀的一般格式的通话录音 Amazon S3 路径。例如，您可以将 myRecordingBucket/technicalSupport/english 指定为 Location。

```
myRecordingBucket/technicalSupport/english/2019/03/01/2019-03-01-17-10-00-010_c4640e3b1478-40fb-8e38-6f6213adf70b_7ab7748e-b47d-4620-ae2c-152617d3333c.wav
```

以下示例显示 Amazon S3 路径中的数据。

```
s3Bucket/yourObjectKeyPrefix/year/month/date/year-month-date-hour-minute-second-millisecond_transactionId_callId.wav
```

发送到您的 Amazon S3 存储桶的录音包含有关呼叫分支的其他 [Amazon S3 对象元数据](#)。下表列出了受支持的 Amazon S3 对象元数据。

名称	描述
transaction-id	呼叫的事务 ID。
call-id	AWS Lambda 函数调用的 CallDetails 中参与者的 CallId
recording-duration	通话录音时长 (秒)
recording-audio-file-format	以互联网媒体类型表示的通话录音音频文件格式

授予 Amazon S3 存储桶权限

您的目标 Amazon S3 存储桶必须与您的应用程序处于同一 AWS 账户。此外，您必须向 Amazon Chime SDK 语音连接器服务主体，即 voiceconnector.chime.amazonaws.com，授予 s3:PutObject 和 s3:PutObjectAcl 的访问权限。以下示例授予适当的权限。

```
{
  "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 音频服务代表您的 SIP 媒体应用程序对 S3 存储桶进行读写操作。为避免[混淆代理问题](#)，您可以将 S3 存储桶的访问限制为单个 SIP 媒体应用程序。

```
{
  "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"
    }
}
]
}

```

操作成功响应

当通话录音在呼叫分支上成功启动后，SIP 媒体应用程序会调用 ACTION_SUCCESSFUL 事件类型的 AWS Lambda 函数。响应中会返回通话记录的位置。

```

{
  "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": {
    ...
  }
}

```

操作错误响应

对于验证错误，SIP 媒体应用程序会在调用 AWS Lambda 函数时显示相应的错误消息。下表列出错误消息。

错误	消息	Reason
InvalidActionParameter	操作的 CallId 参数无效。	任何参数无效。
SystemException	运行操作时出现系统错误。	运行操作时出现另一种类型的系统错误。

当操作未能在呼叫分支上录制媒体时，SIP 媒体应用程序将调用 ActionFailed 事件类型的 AWS Lambda 函数。

以下示例显示典型错误响应。

```
{
  "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": {
  ...
}
}
```

在 GitHub 上查看工作示例：<https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>

StopCallRecording

StopCallRecording 操作停止呼叫分支的录音。通话结束后，录音会自动停止，您的应用程序无需明确返回 StopCallRecording 操作。呼叫分支的录音一旦停止，无法重新开始，录音将传送到 StartCallRecording 操作中指定的目的地。

以下示例停止 call-id-1 呼叫分支的录音。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "StopCallRecording",
      "Parameters": {
        "CallId": "call-id-1"
      }
    }
  ]
}
```

CallId

描述 — AWS Lambda 函数调用的 CallDetails 中参与者的 CallId

允许的值 — 有效的调用 ID

必填 — 是

默认值 — 无

在 GitHub 上查看工作示例：<https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>

PauseCallRecording

PauseCallRecording 操作暂停呼叫分支的录音。每次暂停录音时，录音会捕获提示暂停的音调。当您暂停时，录制会继续，但只会捕捉静音。暂停录制不会影响录制的总时长。您可以根据需要暂停和恢复录音。

以下示例暂停录制。

```
{
```

```
"SchemaVersion": "1.0",
"Actions": [
  {
    "Type": "PauseCallRecording",
    "Parameters": {
      "CallId": "call-id-1"
    }
  }
]
```

CallId

描述 — AWS Lambda 函数调用的 CallDetails 中参与者的 CallId

允许的值 — 有效的调用 ID

必填 — 是

默认值 — 无

在 GitHub 上查看工作示例：<https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>

ResumeCallRecording

ResumeCallRecording 操作会恢复呼叫分支的录制。重新开始录制之前，会播放一段简短的音调。在呼叫分支期间，您可以多次暂停和恢复录音。

以下示例恢复录制。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "ResumeCallRecording",
      "Parameters": {
        "CallId": "call-id-1"
      }
    }
  ]
}
```

CallId

描述 — AWS Lambda 函数调用的 CallDetails 中参与者的 CallId

允许的值 — 有效的调用 ID

必填 — 是

默认值 — 无

在 GitHub 上查看工作示例：<https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>

CallAndBridge

向 PSTN 电话号码或配置为 Amazon Chime SDK 语音连接器或 Amazon Chime SDK 语音连接器群组的 SIP 中继创建出站调用，然后将其与现有调用分支桥接。拨打电话号码时使用 PSTN 和调用 SIP 中继时使用 AWS。

现有的呼叫段可以是使用 [createSip MediaApplicationCall API 创建](#) 的出站呼叫段，也可以是由 SIP 规则创建的通过事件调用该 AWS Lambda 函数的入站路段。NewInboundCall 当您语音连接器或语音连接器群组终端节点实施 CallAndBridge 操作时，必须指定语音连接器或语音连接器群组的 Amazon Resource Number (ARN)。

您还可以在出站呼叫段和 AWS Lambda 功能中添加自定义 SIP 标头。自定义标头允许您传递楼层号和邮政编码等值。有关自定义标头的更多信息，请参阅 [使用 SIP 标头](#)。

以下示例代码显示桥接到 PSTN 终端节点的典型操作。

```
{
  "SchemaVersion": "1.0",
  "Actions": [{
    "Type": "CallAndBridge",
    "Parameters": {
      "CallTimeoutSeconds": 30,
      "CallerIdNumber": "e164PhoneNumber", // required
      "Endpoints": [{
        "BridgeEndpointType": "PSTN", // required
        "Uri": "e164PhoneNumber", // required
      }],
    }
  ]
}
```

```
]
}
```

以下示例显示使用语音连接器或语音连接器群组以及自定义 SIP 标头的典型操作。

```
{
  "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

描述 — 调用超时之前的时间间隔。计时器从调用设置时开始

允许的值 — 介于 1 到 120 之间 (含)

必填 — 否

默认值 — 30

CallerIdNumber

描述 — 属于客户的号码，或 A 分支的发件人编号

允许的值 — E.164 格式的有效电话号码

必填 — 是

默认值 — 无

端点

描述 — 调用的终端节点

允许的值：

- BridgeEndpointType — 适用于语音连接器和语音连接器群组的 AWS，否则为 PSTN。
- Arn — 语音连接器或语音连接器群组的 ARN。仅在 AWS 用作 BridgeEndpointType 时需要。
- Uri — URI 值取决于终端节点的类型。

对于 PSTN 终端节点，URI 必须是有效的 E.164 电话号码。

对于 AWS 终端节点，URI 值设置 Request-URI 的 user 部分。您必须使用[增强的 Backus-Naur 格式](#)。所需长度：介于 1 到 30 之间（含）。使用以下值：a-z, A-Z, 0-9, &, =, +, \$, /, %, -, _, !, ~, *, (,), (.)

Request-URI 的主机值来自目标语音连接器的入站路由。下面示例显示带有 AWS 终端节点的 CallAndBridge 操作。

```
{
  "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"
    }
  }
]
}

```

有关入站路由和语音连接器的更多信息，请参阅[编辑 Amazon Chime SDK 语音连接器设置](#)。

必填 — 是

默认值 — 无

SipHeaders

描述 — 允许您传递其他值。仅与 AWS 终端节点类型一起使用。

允许的值 — 有效的 SIP 标头

必填 — 否

默认值 — 无

以下示例显示使用 PSTN 终端节点的成功 CallAndBridge 操作：

```

{
  "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"
    }
  ]
}
}
}

```

以下示例显示失败的 CallAndBridge 操作。

```

{
  "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",
        .....
      }
    ]
  }
}
```

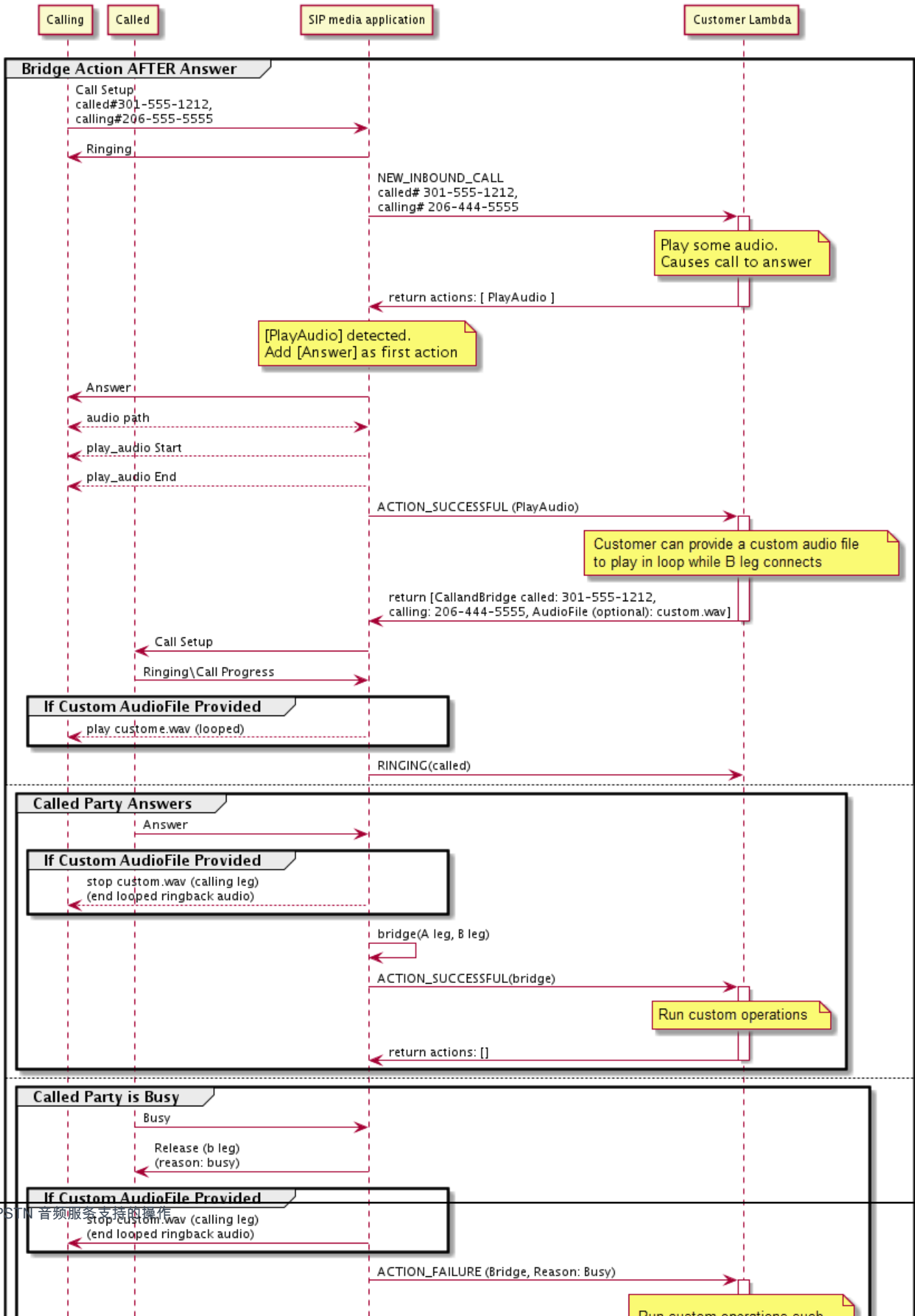
调用流

CallAndBridge 操作为现有调用分支提供不同的调用信令和音频体验，具体取决于参数和分支是否已连接。

下图显示入站调用分支 A 已连接时具有不同参数的调用流。

CallandBridge

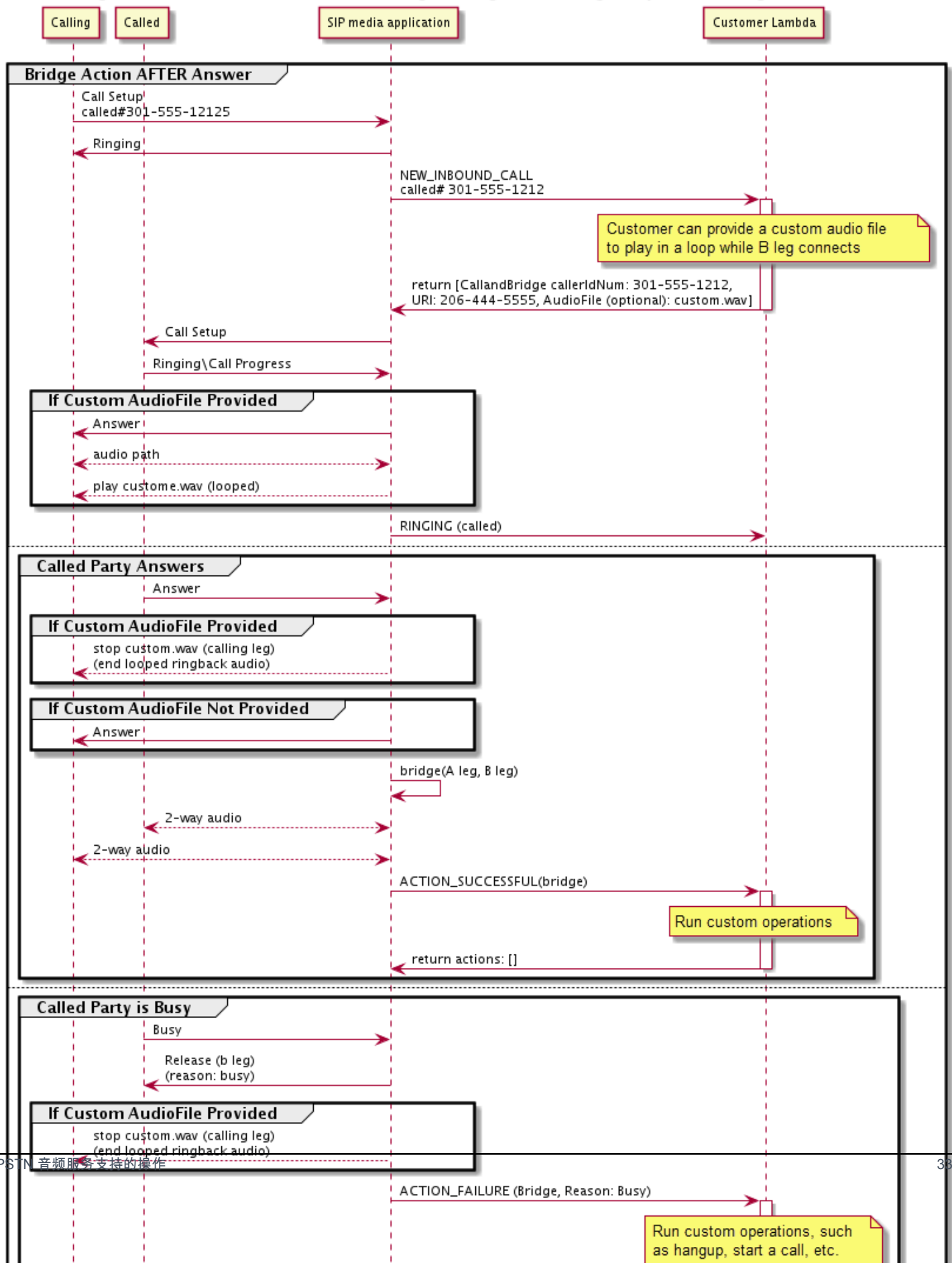
A leg has already been answered. Bridge A leg with B leg only once B leg is answered



下图显示了未应答呼叫的调用流。

CallandBridge

A leg has NOT been answered. Bridge A leg with B leg only once B leg is answered



其他详细信息

记住有关 CallAndBridge 操作的这些事实。

- CallTimeoutSeconds— 此计时器在 B-Leg 上发送 SIP 邀请时启动。您可以设置所需的目标值，但上游运营商可以忽略该值。
- CallerIdNumber— 此电话号码必须属于客户，或者是 A-Leg 的发件人号码。
- 挂断行为和边缘情况 — 如果一个调用分支挂断，则另一个调用分支不会自动挂断调用。当向 AWS Lambda 函数发送 Hangup 事件时，必须独立断开剩余段的连接。如果调用分支仍未挂断，则将对调用进行计费，直到该调用挂断为止。例如，以下情况可能会导致意外收费：
 - 您尝试桥接到目标电话号码。目的地占线，直接将呼叫发送到语音信箱。从音频服务的角度来看，进入语音信箱就是应答呼叫。A-Leg 挂断了，但是 B-Leg 继续收听语音信箱消息。当 B-Leg 收听时，您会被收费。
 - 作为最佳实践，使用 AWS Lambda 函数或呼叫另一端的一方独立挂断每个呼叫段。
- 账单 — 使用 CallAndBridge 时，您需要为以下内容付费：
 - 为 PSTN 创建的每个调用分支（A-Leg、B-Leg 等）的活动呼叫分钟数。
 - 音频服务使用分钟数。

参见以下方面的工作示例 GitHub：

- <https://github.com/aws-samples/amazon-chime-sma-bridging>
- <https://github.com/aws-samples/amazon-chime-sma-call-转发>
- <https://github.com/aws-samples/amazon-chime-sma-on-需求录音>

挂断

向调用的一个分支发送带有 SipStatusCode 的 Hangup 值。

当音频服务运行调用分支上的 Hangup 操作时：

- 对于只有一个调用分支的调用，SIP 媒体应用程序使用 HANGUP 事件调用 AWS Lambda 函数并忽略响应。然后，调用将断开。
- 对于桥接到另一个调用分支（分支 B）的调用分支（分支 A），如果 Hangup 操作与桥接调用分支（分支 B）相关联，则 PSTN 音频服务会断开桥接调用分支，然后使用分支 B 的 HANGUP 的事件调用 Lambda 函数。然后，PSTN 音频服务会运行从该 Lambda 调用返回的所有操作。

- 对于桥接到另一个调用分支 (分支 B) 的调用分支 (分支 A) ，如果 Hangup 操作与初始调用分支 (分支 A) 相关联，则 PSTN 音频服务会断开初始调用分支，然后使用分支 A 的 HANGUP 事件调用 Lambda 函数。然后，PSTN 音频服务会运行从该 Lambda 调用返回的所有操作。
- 对于使用 JoinMeeting 操作加入会议的调用分支，如果 Hangup 操作与会议分支 (通常是分支 B) 相关联，则调用方将断开与会议的连接并收到该 Hangup 操作的 ACTION_SUCCESSFUL 事件。

以下示例显示典型的 Hangup 操作。

```
{
  "Type": "Hangup",
  "Parameters": {
    "CallId": "call-id-1",
    "ParticipantTag": "LEG-A",
    "SipResponseCode": "0"
  }
}
```

CallId

描述 — AWS Lambda 函数调用的 CallDetails 中参与者的 CallId

允许的值 — 有效的调用 ID

必填 — 否

默认值 — 无

ParticipantTag

描述 — CallDetails 中一个已连接参与者的 ParticipantTag

允许的值 — LEG-A 或 LEG-B

必填 — 否

默认值 — 已调用 callLeg 的 ParticipantTag，如指定 CallId，则忽略该值

SipResponseCode

描述 — 任何受支持的 SIP 响应代码

允许值 — 480 — 不可用；486 — 占线；0 — 正常终止

必填 — 否

默认值 — 0

用户结束调用后，SIP 媒体应用程序使用 AWS Lambda 中列出的代码调用 [结束呼叫](#) 函数。

在 GitHub 上查看工作示例：

- <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-outbound-call-notifications>
- <https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>

JoinChimeMeeting

提供与会者加入令牌即可加入 Amazon Chime SDK 会议。为此，您可以对 [CreateMeeting](#) 和 [CreateAttendee](#) API 进行 AWS SDK 调用，以获得令牌并在操作中进行传递。请参阅以下示例。

Note

您无法在桥接调用上运行此操作。

```
{
  "Type": "JoinChimeMeeting",
  "Parameters": {
    "JoinToken": "meeting-attendee-join-token",
    "CallId": "call-id-1",
    "ParticipantTag": "LEG-A",
    "MeetingId": "meeting-id"
  }
}
```

JoinToken

描述 — Amazon Chime SDK 会议与会者的有效加入令牌

允许的值 — 有效加入令牌

必填 — 是

默认值 — 无

CallId

描述 — AWS Lambda 函数调用的 CallDetails 中参与者的 CallId

允许的值 — 有效的调用 ID

必填 — 否

默认值 — 无

ParticipantTag

描述 — CallDetails 中一个已连接参与者的 ParticipantTag

允许的值 — LEG-A

必填 — 否

默认值 — 已调用 callLeg 的 ParticipantTag，如指定 CallId，则忽略该值

MeetingId

描述 — 与 JoinToken 关联的有效 Amazon Chime SDK 会议 ID。如果使用 [Amazon Chime](#) 命名空间中的 API 创建会议，则无需提供会议 ID。如果使用 [Amazon Chime SDK 会议](#) 命名空间中的 API 创建会议，则需要提供会议 ID。使用用于创建会议的 API 终端节点加入会议。

允许的值 — 有效的会议 ID

必填 — 否。

默认值 — 无。

运行此操作后，SIP 媒体应用程序始终会调用 AWS Lambda 函数。它会返回 ACTION_SUCCESSFUL 或 ACTION_FAILED 调用事件类型。以下示例显示成功调用事件结构。

```
{
  "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": {
    ...
  }
}

```

错误处理

当桥接会议时出现验证错误时，SIP 应用程序会使用下表所示的错误消息之一调用其 AWS Lambda 函数。

错误	消息	Reason
InvalidActionParameter	JoinToken 参数值无效。	该操作的任何其他参数无效或缺失。
SystemException	运行操作时出现系统错误。	运行操作时出现另一种类型的系统错误。

以下示例显示了典型故障事件。

```

{
  "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": {
    ...
  }
}

```

在 GitHub 上查看工作示例：<https://github.com/aws-samples/amazon-chime-sma-update-call>

ModifyChimeMeetingAttendee (音频静音和取消静音)

允许 SIP 媒体应用程序通过提供 Amazon Chime SDK 会议 ID 和与会者列表修改电话与会者的状态。

Note

此操作目前支持对电话与会者进行静音和取消静音操作。此外，用户必须使用 `JoinChimeMeeting` 操作加入会议。可以在 `participantTag="LEG-B"` 上或对应的 `CallId` 上执行此操作。

此操作仅适用于从 SIP 媒体应用程序加入 "+ 13605550122"、LEG-B 的 `callLeg` 或从 SIP 媒体应用程序加入会议的分支。

```
{
  "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"]
      }
    }
  ]
}
```

操作

描述 — 要对与会者列表执行的操作

允许的值 — 静音、取消静音

必填 — 是

默认值 — 无

MeetingId

描述 — 与会者所属会议的 ID

允许的值 — 有效的会议 ID 设置静音或取消静音的人员也必须属于会议。

必填 — 是

默认值 — 无

CallId

描述 — 与会者所属会议的 ID

允许的值 — 有效的调用 ID。

必填 — 否

默认值 — 无

ParticipantTag

描述 — 分配给与会者的标签。

允许的值 — 有效的标签。

必填 — 否

默认值 — 无

AttendeeList

描述 — 要静音或取消静音的与会者 ID 列表

允许的值 — 有效的与会者 ID 列表

必填 — 是

默认值 — 无，最大值 100

执行此操作后，音频服务始终调用 ACTION_SUCCESSFUL 或 ACTION_FAILED 调用事件类型的 AWS Lambda 函数。以下示例代码显示了典型的 ACTION_SUCCESSFUL 调用事件。

```
{
  "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": {
  ...
}
}

```

错误处理

如果指令参数无效或 API 故障，SIP 媒体应用程序会调用带有特定于失败指令或 API 的错误消息的 AWS Lambda 函数。

错误	消息	Reason
InvalidActionParameter	ModifyChimeMeetingAttendees Operation 参数值无效	Operation 值必须为“静音”或“取消静音”。
	会议 ID 参数值无效。	会议 ID 为空。
	与会者列表参数值无效。	与会者 ID 列表为空，或者超过最大值 100。
	对调用的操作无效。	调用没有桥接。
	调用未连接到 Chime Meeting。	与会者未连接到 Chime Meeting。
	一位或多位与会者并不是此会议的一部分。所有与会者必须是会议的一部分。	与会者无权修改会议中的与会者。
SystemException	运行操作时出现系统错误。	运行操作时出现系统错误。

以下示例代码显示了典型失败事件。

```
{
  "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": {
    ...
  }
}
```

在 GitHub 上查看工作示例：

- <https://github.com/aws-samples/amazon-chime-sma-bridging>.
- <https://github.com/aws-samples/amazon-chime-sma-update-call>

Pause

在指定时间内暂停调用。

```
{
  "Type": "Pause",
  "Parameters": {
    "CallId": "call-id-1",
    "ParticipantTag": "LEG-A",
    "DurationInMilliseconds": "3000"
  }
}
```

CallId

描述 — AWS Lambda 函数调用的 CallDetails 中参与者的 CallId

允许的值 — 有效的调用 ID

必填 — 否

默认值 — 无

ParticipantTag

描述 — CallDetails 中一个已连接参与者的 ParticipantTag

允许的值 — LEG-A 或 LEG-B

必填 — 否

默认值 — 已调用 callLeg 的 ParticipantTag，如指定 CallId，则忽略该值

DurationInMilliseconds

描述 — 暂停的持续时间（以毫秒为单位）

允许的值 — 大于 0 的整数

必填 — 是

默认值 — 无

在 GitHub 上查看工作示例：

- <https://github.com/aws-samples/amazon-chime-sma-outbound-call-notifications>
- <https://github.com/aws-samples/amazon-chime-sma-on-demand-recording>

PlayAudio

在调用的任一分支上播放音频文件。音频可以重复播放任意次数。可以使用 PlaybackTerminators 中设置的 DTMF 数字终止正在播放的音频。

目前，Amazon Chime SDK 仅支持从 Amazon Simple Storage Service (Amazon S3) 存储桶播放音频文件。S3 存储桶必须与 SIP 媒体应用程序处于同一 AWS 账户。此外，您必须向 Amazon Chime SDK 语音连接器服务主体授予 s3:GetObject 权限。您可以使用 S3 控制台或命令行界面 (CLI) 完成此操作。

以下代码示例显示典型的存储桶策略。

```
{
  "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"
        }
      }
    }
  ]
}
```

音频服务代表您的 Sip 媒体应用程序对 S3 存储桶进行读写操作。为避免[混淆代理问题](#)，您可以将 S3 存储桶的访问限制为单个 SIP 媒体应用程序。

```
{
  "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"
    }
}
]
}

```

以下代码示例显示典型的操作。

```

{
  "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

描述 — CallDetails 中的参与者的 CallId。

允许的值 — 有效的调用 ID。

必填 — 否，如 ParticipantTag 已存在。

默认值 — 无。

ParticipantTag

描述 — CallDetails 中一个已连接参与者的 ParticipantTag。

允许的值 — LEG-A 或 LEG-B。

必填 — 否，如 CallId 已存在。

默认值 — 被调用的 callLeg 的 ParticipantTag。如果您指定 CallId，则忽略该值。

PlaybackTerminator

描述 — 使用用户的 DTMF 输入终止正在播放的音频

允许的值 — 由以下值组成的数组：“0”“1”、“2”、“3”、“4”、“5”、“6”、“7”、“8”、“9”、“#”、“*”

必填 — 否

默认值 — 无

重复

描述 — 将音频重复播放指定次数

允许的值 — 大于零的整数

必填 — 否

默认值 — 1

AudioSource.Type

描述 — 音频文件来源的类型。

允许的值 — S3。

必填 — 是。

默认值 — 无。

AudioSource.BucketName

描述 — 对于 S3 来源类型，S3 存储桶必须属于 SIP 应用程序所在的 AWS 账户。存储桶必须有权访问 Amazon Chime SDK 语音连接器服务主体，即 `voiceconnector.chime.amazonaws.com`。

允许的值 — Amazon Chime SDK 有权访问 `s3:GetObject` 操作的有效 S3 存储桶。

必填 — 是。

默认值 — 无。

AudioSource.key

描述 — 对于 S3 源类型，`AudioSource.BucketName` 属性中指定的 S3 存储桶中的文件名。

允许的值 — 有效的音频文件。

必填 — 是。

默认值 — 无。

SIP 媒体应用程序尝试从来源 URL 播放音频。您可以使用大小不超过 50MB 未压缩的原始 PCM .wav 文件。Amazon Chime SDK 推荐 8kHz 单声道。

当拨号方案中的最后一条指令为 PlayAudio 并且文件已完成播放，或者如果用户通过按键停止播放时，应用程序将使用以下代码示例所示的事件调用 AWS Lambda 函数。

```
{
  "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",
      }
    }
  }
}
```

终止数字停止音频后，音频将不会重复播放。

错误处理

当验证文件包含错误或运行操作时出现错误，SIP 媒体应用程序会通过适当的错误代码调用 AWS Lambda 函数。

错误	消息	Reason
InvalidAudioSource	音频来源参数无效。	出现此错误可能基于多种原因。例如，由于权限问题或 URL 问题，SIP 媒体应用程序无法访问文件。或者，由于格式、持续时间、大小等原因，音频文件可能无法通过验证。

错误	消息	Reason
SystemException	运行操作时出现系统错误。	运行操作时出现另一个系统错误。
InvalidActionParameter	操作的 CallId 或 ParticipantTag 参数无效。	该操作包含无效参数。

以下代码示例显示典型的调用失败。

```
{
  "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": {
  ...
}
}
```

在 GitHub 上查看工作示例：

- <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-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

播放音频并收集 DTMF 数字。如果发生失败，例如用户未输入正确的 DTMF 数字，该操作将播放“失败”音频，然后重播主音频，直到 SIP 媒体应用程序耗尽 Repeat 参数中定义的尝试次数。

您必须播放 S3 存储桶中的音频文件。S3 存储桶必须与 SIP 媒体应用程序处于同一 AWS 账户。此外，您必须向 [Amazon Chime SDK 语音连接器服务主体](#) 授予 s3:GetObject 的访问权限 (voiceconnector.chime.amazonaws.com)。您可以使用 S3 控制台或 CLI 完成该操作。

以下代码示例显示典型的 S3 存储桶策略。

```
{
  "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"
        }
      }
    }
  ]
}
```

音频服务代表您的 Sip 媒体应用程序对 S3 存储桶进行读写操作。为避免[混淆代理问题](#)，您可以将 S3 存储桶的访问限制为单个 SIP 媒体应用程序。

```
{
  "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"
        }
    }
}
]
}

```

以下示例显示典型的 PlayAudioAndGetDigits 操作。

```

{
  "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

描述 — AWS Lambda 函数调用的 CallDetails 中参与者的 CallId

允许的值 — 有效的调用 ID

必填 — 否

默认值 — 无

ParticipantTag

描述 — CallDetails 中一个已连接参与者的 ParticipantTag

允许的值 — LEG-A 或 LEG-B

必填 — 否

默认值 — 已调用 callLeg 的 ParticipantTag，如指定 CallId，则忽略该值

InputDigitsRegex

描述 — 正则表达式模式

允许的值 — 有效的正则表达式模式

必填 — 否

默认值 — 无

AudioSource.Type

描述 — 音频文件来源的类型

允许的值 — 一个 S3 存储桶

必填 — 是

默认值 — "S3"

AudioSource.BucketName

描述 — 对于 S3 AudioSource.Type 值，S3 存储桶必须属于 SIP 应用程序所在的 AWS 账户。S3 存储桶必须有权访问 [Amazon Chime SDK 语音连接器服务主体](https://docs.aws.amazon.com/chime/latest/APIReference/welcome.html) (voiceconnector.chime.amazonaws.com)。

允许的值 — Amazon Chime SDK 具有 `s3:GetObject` 操作权限的有效 S3 存储桶。

必填 — 是

默认值 — 无

AudioSource.Key

描述 — `AudioSource.BucketName` S3 存储桶中音频对象的密钥名称。

允许的值 — 有效的音频文件

必填 — 是

默认值 — 无

FailureAudioSource.Type

描述 — `FailureAudioSource.BucketName` S3 存储桶中音频对象的密钥名称。

允许的值 — S3

必填 — 是

默认值 — 无

FailureAudioSource.BucketName

描述 — 对于 S3 来源类型，S3 存储桶必须属于 SIP 应用程序所在的 AWS 账户。[Amazon Chime SDK 语音连接器服务主体](#) (`voiceconnector.chime.amazonaws.com`) 必须有权访问 S3 存储桶。

允许的值 — Amazon Chime SDK 具有 `s3:GetObject` 操作权限的有效 S3 存储桶。

必填 — 是

默认值 — 无

FailureAudioSource.Key

描述 — `FailureAudioSource.BucketName` S3 存储桶中音频对象的密钥名称。

允许的值 — 有效的音频文件

必填 — 是

默认值 — 无

MinNumberOfDigits

描述 — 在超时或播放“调用失败”音频之前捕获的最小数字位数。

允许的值 — ≥ 0

必填 — 否

默认值 — 0

MaxNumberOfDigits

描述 — 在没有终止数字的情况下停止前要捕获的最大数字位数。

允许的值 — $> \text{MinNumberOfDigits}$

必填 — 否

默认值 — 128

TerminatorDigits

描述 — 当用户输入的数字小于 `MaxNumberOfDigits` 时，用于结束输入的数字

允许的值 — 以下数字中的任意一个：0123456789#*

必填 — 否

默认值 — #

InBetweenDigitsDurationInMilliseconds

描述 — 播放 `FailureAudio` 之前数字输入之间的等待时间（以毫秒为单位）。

允许的值 — > 0

必填 — 否

默认值 — 如未指定，默认为 `RepeatDurationInMilliseconds` 值

重复

描述 — 尝试获取数字的总次数。

允许的值 — > 0

必填 — 否

默认值 — 1

RepeatDurationInMilliseconds

描述 — Repeat 尝试之间的等待时间 (以毫秒为单位)

允许的值 — >0

必填 — 是

默认值 — 无

SIP 媒体应用程序始终在运行 PlayAudioAndGetDigits 操作后调用其 AWS Lambda 功能，事件类型为 ACTION_SUCCESSFUL 或 ACTION_FAILED 调用。当应用程序成功收集数字时，它会在 ActionData 对象中设置 ReceivedDigits 值。以下示例显示 AWS Lambda 函数的调用事件结构。

```
{
  "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": {
    ...
  }
}

```

错误处理

发生验证错误时，SIP 媒体应用程序会调用 AWS Lambda 函数并显示相应的错误消息。下表列出可能的错误消息。

错误	消息	Reason
InvalidAudioSource	音频来源参数值无效。	出现此错误可能基于多种原因。例如，由于权限问题或 S3 存储桶问题，SIP 媒体应用程序无法访问文件。或者，由于持续时间、大小或不受支持的格式等原因，音频文件可能无法通过验证。
InvalidActionParameter	操作的 CallId 或 ParticipantTag 参数无效。	CallId、ParticipantTag 或其他参数无效。
SystemException	运行操作时出现系统错误。	运行操作时出现一个系统错误。

当操作由于超时或重试次数过多而无法收集指定数字位数时，SIP 媒体应用程序将使用 ACTION_FAILED 调用事件类型调用 AWS Lambda 函数。

```

{
  "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": {
    ...
  }
}

```

在 GitHub 上查看工作示例：

- <https://github.com/aws-samples/amazon-chime-sma-bridging>.
- <https://github.com/aws-samples/amazon-chime-sma-update-call>

ReceiveDigits

当用户输入与此操作中指定的正则表达式模式匹配的数字时，SIP 媒体应用程序将调用 AWS Lambda 函数。

```

{
  "Type": "ReceiveDigits",
  "Parameters": {

```

```
    "CallId": "call-id-1",
    "ParticipantTag": "LEG-A",
    "InputDigitsRegex": "\\d{2}#$",
    "InBetweenDigitsDurationInMilliseconds": 1000,
    "FlushDigitsDurationInMilliseconds": 10000
  }
}
```

CallId

描述 — AWS Lambda 函数调用的 CallDetails 中参与者的 CallId

允许的值 — 有效的调用 ID

必填 — 否

默认值 — 无

ParticipantTag

描述 — CallDetails 中一个已连接参与者的 ParticipantTag

允许的值 — LEG-A 或 LEG-B

必填 — 否

默认值 — 已调用 callLeg 的 ParticipantTag，如指定 CallId，则忽略该值

InputDigitsRegex

描述 — 正则表达式模式

允许的值 — 有效的正则表达式模式

必填 — 是

默认值 — 无

InBetweenDigitsDurationInMilliseconds

描述 — 检查输入是否与正则表达式模式匹配前数字之间的时间间隔

允许的值 — 持续时间（以毫秒为单位）

必填 — 是

默认值 — 无

FlushDigitsDurationInMilliseconds

描述 — 刷新收到的 DTMF 数字并发送到 AWS Lambda 函数的时间间隔。如果 SIP 媒体应用程序在时间间隔结束后收到新的数字，则计时器将重新启动。

允许的值 — InBetweenDigitsDurationInMilliseconds

必填 — 是

默认值 — 无

SIP 媒体应用程序在调用期间放弃 DTMF 数字，直至收到新的 `ReceiveDigits` 操作。`FlushDigitsDurationInMilliseconds` 间隔时间从 SIP 媒体应用程序收到第一个 DTMF 数字时开始。如果用户在间隔时间到期之前输入了正确的数字，则 SIP 媒体应用程序将调用 [接收来电者输入](#) 中描述的 AWS Lambda 功能。

如果用户输入与正则表达式模式不匹配，SIP 媒体应用程序将重复“失败”的音频文件消息，直到应用程序耗尽重复计数或用户输入有效数字。

在 GitHub 上查看工作示例：

- <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

允许 SIP 媒体应用程序录制给定调用 ID 的媒体。例如，语音邮件应用程序和会议参与者通知。应用程序会进行录制，直到达到您设置的持续时间，或用户按下其中一个 `RecordingTerminators`，或应用程序检测到沉默。在这些情况下，该操作会指示您的应用程序将生成的媒体文件放入指定的 S3 存储桶中。S3 存储桶必须与 SIP 媒体应用程序处于同一 AWS 账户。此外，该操作必须给予 `s3:PutObject` 和 `s3:PutObjectAcl` 访问 Amazon Chime SDK 语音连接器服务主体的权限。[Amazon Chime SDK 语音连接器服务主体](#) (`voiceconnector.chime.amazonaws.com`)。

Note

使用此功能录制的录音可能会受有关电子通信录制的法律或法规的约束。您和您的最终用户有责任遵守与录音有关的所有适用法律，包括适当通知录制会话或通信中的所有参与者正在录制会话或通信，并获得他们的许可。

以下示例向 `s3:PutObject` 和 `s3:PutObjectAcl` 提供了访问 Amazon Chime SDK 语音连接器服务主体的权限。

```
{
  "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/*"
    }
  ]
}
```

以下示例显示在调用者按下井号键 (#) 后，或者在 10 秒钟内没有活动，或者调用者保持沉默 3 秒钟后会停止录制，并将生成的媒体文件写入 `RecordingDestination` 参数定义的位置。

Note

本示例使用 `CallId` 参数。您可以改用 `ParticipantTag` 参数，但不能同时使用两者。

```
{
  "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

描述 — AWS Lambda 函数调用的 CallDetails 中参与者的 CallId

允许的值 — 有效的调用 ID

必填 — 否

默认值 — 无

ParticipantTag

描述 — CallDetails 中一个已连接参与者的 ParticipantTag

允许的值 — LEG-A 或 LEG-B

必填 — 否

默认值 — 已调用 callLeg 的 ParticipantTag，如指定 CallId，则忽略该值

RecordingDestination.Type

描述 — 目的地类型。仅 S3。

允许的值 — S3

必填 — 是

默认值 — 无

RecordingDestination.BucketName

描述 — 有效的 S3 存储桶名称。存储桶必须有权访问 [Amazon Chime SDK 语音连接器服务主体](#)，voiceconnector.chime.amazonaws.com。

允许的值 — Amazon Chime SDK 有权访问 s3:PutObject 和 s3:PutObjectAcl 操作的有效 S3 存储桶。

必填 — 是

默认值 — 无

RecordingDestination.Prefix

描述 — 录制文件的 S3 前缀

允许的值 — 包含最多 979 个安全字符的有效前缀名称。有关安全字符的更多信息，请参阅 Amazon Simple Storage Service 用户指南中的[安全字符](#)。

必填 — 否

默认值 — 无。如果未指定，则录制文件将保存到 S3 存储桶的根目录。

DurationInSeconds

描述 — 录制的持续时间，以秒为单位

允许的值 — >0

必填 — 否

默认值 — 无

SilenceDurationInSeconds

描述 — 录制停止后沉默的持续时间，以秒为单位 如果未指定，则禁用沉默检测。

允许的值 — [1;1000]

必填 — 否

默认值 — 200

SilenceThreshold

描述 — 被视为“沉默”的噪音水平。如果您未指定 SilenceDurationInSeconds，此参数将被忽略。

参考值（将噪音视为沉默的噪声水平和阈值）：

- 1 — 30dB 或以下，例如安静的房间
- 100 — 40-50dB，例如耳语或安静的办公室
- 200 — 60dB，例如拥挤的办公室

- 1000 — 75dB，例如吵闹的人或音乐

允许的值 — [1;1000]

必填 — 否

默认值 — 200

RecordingTerminators

描述 — 列出所有可用的录制终止符。

允许的值 — 来自 [123456789*0#] 的个位数和符号的数组

必填 — 是

默认值 — 无

处理 ACTION_SUCCESSFUL 事件

录制结束后，Amazon Chime SDK SIP 媒体应用程序会调用 AWS Lambda 函数，并将 ACTION_SUCCESSFUL 事件以及调用结果传递给该函数。

```
{
  "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` 事件包含 `ActionData`，其中包含以下字段：

Type

描述 — 操作的类型，`RecordAudio`。

参数

描述 — 操作的参数。

`RecordingDestination.Type`

描述 — 目的地类型。仅 `S3`。

`RecordingDestination.BucketName`

描述 — 包含录制文件的 `S3` 存储桶。

`RecordingDestination.Key`

描述 — 录制文件的 `S3` 密钥。

`RecordingTerminatorUsed`

描述 — 用于停止录制的终止符 — `RecordingTerminators` 中传递的终止符之一。如果录制在达到最大持续时间 (`DurationInSeconds`) 后或由于沉默 (`SilenceDurationInSeconds`) 而停止，则输出中不包含此密钥值对。

错误处理

对于验证错误，SIP 媒体应用程序会在调用 AWS Lambda 函数时显示相应的错误消息。下表列出可能的错误消息。

错误	消息	Reason
<code>InvalidActionParameter</code>	操作的 <code>CallId</code> 或 <code>ParticipantTag</code> 参数无效。	任何参数无效。
	<code>DurationInSeconds</code> 参数值无效。	
	<code>SilenceDurationInSeconds</code> 参数值无效。	

错误	消息	Reason
	<p>SilenceThreshold 参数值无效。</p> <p>RecordingDestination 参数值无效。</p> <p>将录音上传到 S3 存储桶时出错。</p>	
SystemException	运行操作时出现系统错误。	运行操作时出现另一种类型的系统错误。

处理 ACTION_FAILED 事件

当操作未能在呼叫分支上录制媒体时，SIP 媒体应用程序将调用 ACTION_FAILED 事件类型的 AWS Lambda 函数。请参阅以下示例。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 5,
  "InvocationEventType": "ACTION_FAILED",
  "ActionData": {
    "Type": "RecordAudio",
    "Parameters": {
      ...
    },
    "ErrorType": "InvalidActionParameter",
    "ErrorMessage": "RecordingDestination parameter value is invalid."
  },
  "CallDetails": {
    ...
  }
}
```

在 GitHub 上查看工作示例：<https://github.com/aws-samples/amazon-chime-sma-bridging>

SendDigits

在调用的任何分支上发送最多 50 个双音多频 (DTMF) 数字。信号可包括：

- 数字 0 到 9
- 特殊字符星号 (*) 和井号 (#)
- 网络控制信号 A、B、C、D
- 逗号字符 (,)。该信号在前一个信号和下一个信号之间增加了 0.5 秒的延迟。

主题

- [使用 sendDigits 操作](#)
- [处理 ACTION_SUCCESSFUL 事件](#)
- [处理 ACTION_FAILED 事件](#)
- [调用流](#)

使用 sendDigits 操作

以下示例显示典型的 SendDigits 操作：

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "SendDigits",
      "Parameters": {
        "CallId": "call-id-1", // required
        "Digits": ",, *1234,56,7890ABCD#", // required
        "ToneDurationInMilliseconds": 100 // optional
      }
    }
  ]
}
```

CallId

描述 — AWS Lambda 函数调用的 CallDetails 中参与者的 CallId

允许的值 — 有效的调用 ID

必填 — 是

默认值 — 无

数字

描述 — 要在调用分支上发送的数字，对应于 CallId

允许的值 — 0-9、*、#、A、B、C、D、逗号(,)

必填 — 是

默认值 — 无

ToneDurationInMilliseconds

描述 — 传输每个数字所允许的时间（以毫秒为单位）。

允许的值 — 介于 50 和 24000 之间的任何整数

必填 — 否

默认值 — 250

处理 ACTION_SUCCESSFUL 事件

以下示例显示 SendDigits 操作的典型 ACTION_SUCCESSFUL 事件。

```
{
  "SchemaVersion": "1.0",
  "Sequence": 3,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData": {
    "Type": "SendDigits",
    "Parameters": {
      "Digits": "1,2A#",
      "ToneDurationInMilliseconds": 100,
      "CallId": "call-id-1"
    },
    "CallDetails": {
      ...
    }
  }
}
```

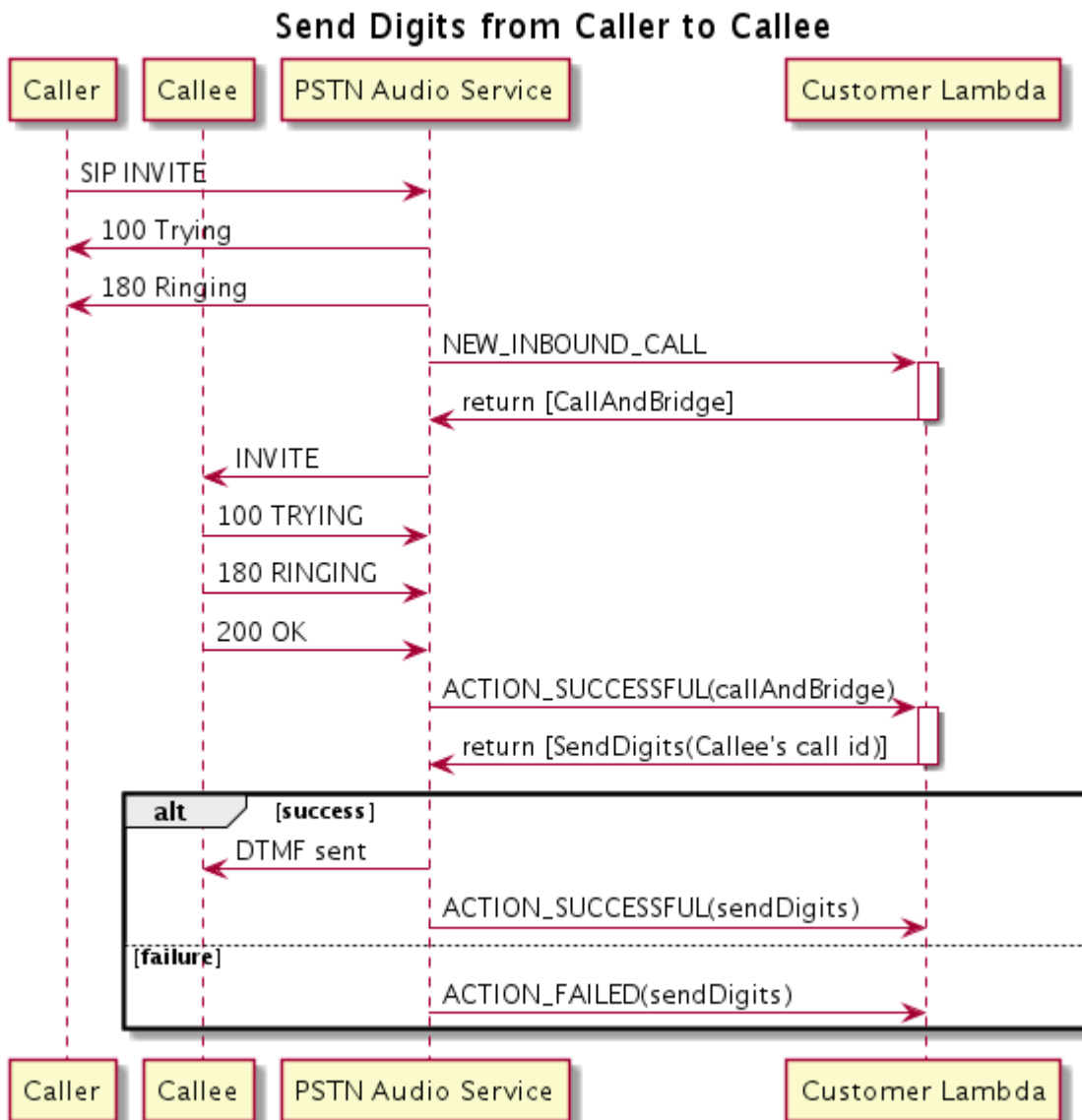
处理 ACTION_FAILED 事件

以下示例显示 SendDigits 操作的典型 ACTION_FAILED 事件。

```
{
  "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": {
    ...
  }
}
```

调用流

下图显示从呼叫方向被叫方发送数字的程序流程。



Speak

您可以通过提供文本在任何调用分支上播放语音。您可以输入纯文本或语音合成标记语言 (SSML)。SSML 可以通过添加暂停、强调某些单词或更改说话风格等，更好地控制 Amazon Chime SDK 生成语音的方式。

Amazon Chime SDK 使用 Amazon Polly 服务将文本转换为语音。Amazon Polly 允许您在标准或神经引擎之间进行选择，以提高语音质量。Amazon Polly 支持超过 20 种语言和 60 种语音，可自定义应用程序的用户体验。Amazon Chime SDK 免费提供语音功能，但使用 Amazon Polly 需要付费。有关定价信息，请参阅 Amazon Polly [定价页面](#) 或账单控制面板。

⚠ Important

使用 Amazon Polly 需遵守 [AWS 服务条款](#)，包括 AWS 机器学习和人工智能服务的特定条款。

主题

- [使用 Speak 操作](#)
- [处理 ACTION_SUCCESSFUL 事件](#)
- [处理 ACTION_FAILED 事件](#)
- [程序流程](#)

使用 Speak 操作

以下示例显示典型的 Speak 操作。

```
{
  "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

描述 — Lambda 函数调用的 CallDetails 中参与者的 CallId

允许的值 — 有效的调用 ID

必填 — 是

默认值 — 无

文本

描述 — 指定要合成语音的输入文本。如果指定 `ssml` 为 `TextType`，按照 SSML 格式输入文本。

允许的值 — 字符串

必填 — 是

默认值 — 无

Engine

描述 — 指定在处理语音合成文本时使用的引擎：标准引擎或神经引擎。

允许的值 — `standard` | `neural`

必填 — 否

默认值：标准

LanguageCode

描述 — 指定语言代码。仅在使用双语语音时需要。如果您使用没有语言代码的双语语音，则使用双语语音的默认语言。

允许的值 — [Amazon Polly 语言代码](#)

必填 — 否

默认值 — 无

TextType

描述 — 指定输入文本的类型，即纯文本或 SSML。如果未指定输入类型，则使用纯文本作为默认值。有关 SSML 的更多信息，请参阅 Amazon Polly 开发人员指南中的[由 SSML 文档生成语音](#)。

允许的值 — `ssml` | `text`

必填 — 否

默认值 — 无

Voiceld

描述 — 指定要使用的语音的 ID。

允许的值 — [Amazon Polly 语音 ID](#)

必填 — 否

默认值 — Joanna

处理 ACTION_SUCCESSFUL 事件

以下示例显示了使用 Amazon Polly 的 Joanna 声音将文本“Hello World”合成英语语音操作的典型 ACTION_SUCCESSFUL 事件。

```
{
  "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": {
    ...
  }
}
```

处理 ACTION_FAILED 事件

以下示例显示了与上一个示例中使用相同事件的典型 ACTION_FAILED 事件。

```
{
  "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":{
  ...
}
}

```

错误处理

此表列出并描述了 Speak 操作引发的错误消息。

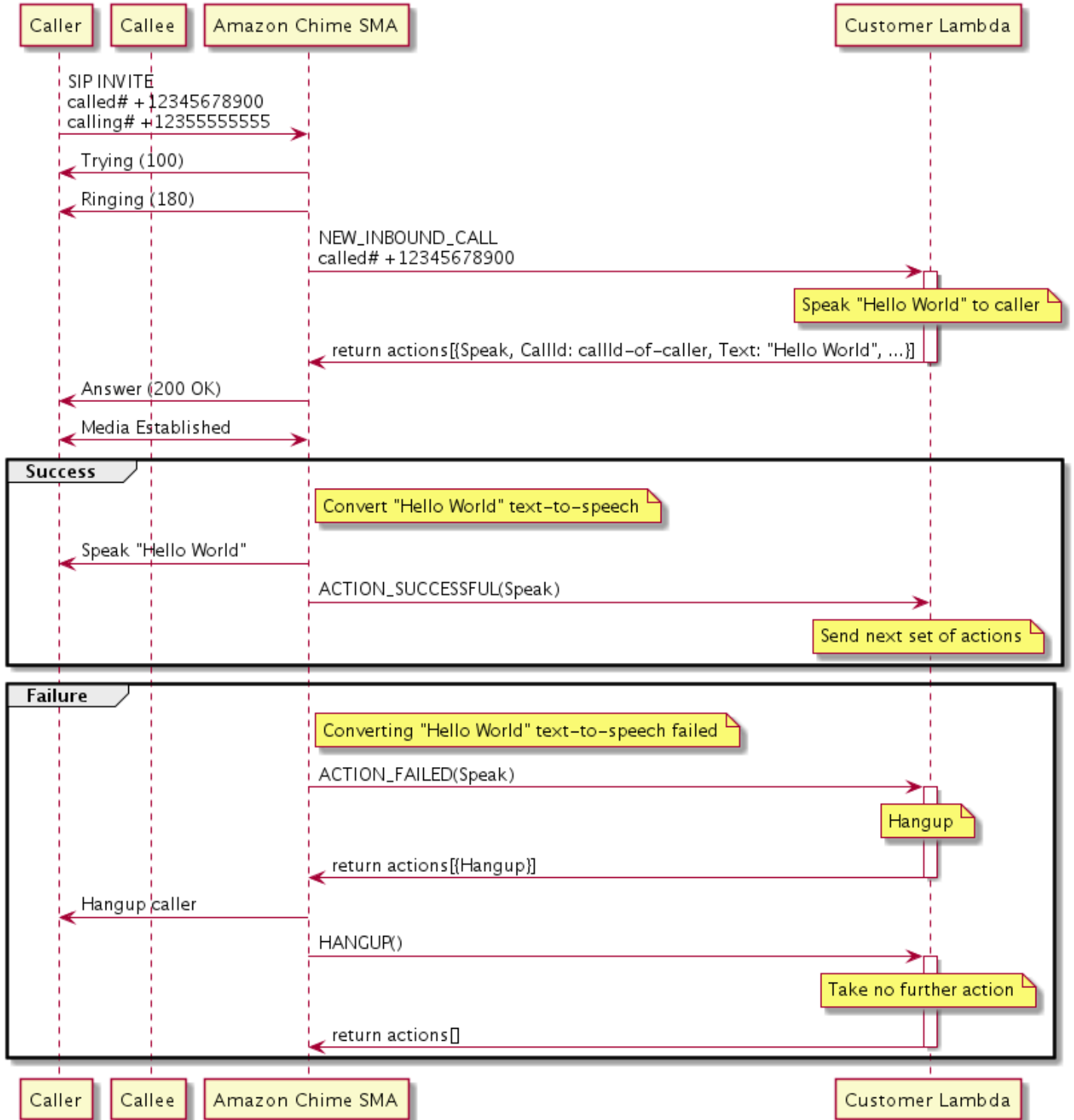
错误	消息	Reason
AccessDenied	AWSServiceRoleForAmazonChimeVoiceConnector 服务相关角色配置不正确。	用于向 Amazon Polly 发出请求的服务关联角色不存在或缺少权限。要解决这一问题，请参阅 使用 Amazon Chime SDK 语音连接器服务相关角色 部分中的相关步骤
InvalidActionParameter		验证操作参数时出错。有关参数的更多信息，请参阅 Amazon Polly 开发人员指南中的 SynthesizeSpeech API 。
ActionExecutionThrottled	Amazon Polly 正在限制合成语音的请求。	向 Amazon Polly 发出的请求返回限制异常。 有关 Amazon Polly 限制限额的更多信息 ，请参阅 https://docs.aws.amazon.com/polly/latest/dg/limits.html#limits-throttle 。
MissingRequiredActionParameter	Text 是必填参数。	操作参数必须有 Text 值

错误	消息	Reason
MissingRequiredActionParameter	Text 限制为 1,000 个字符	文本超出了字符限制。
SystemException	运行操作时出现系统错误。	运行操作时出现一个系统错误。

程序流程

下图显示了为调用者启用 Speak 操作的程序流程。在此示例中，调用者听到的文本是

Enable Speak action for Caller in SMA



在图中

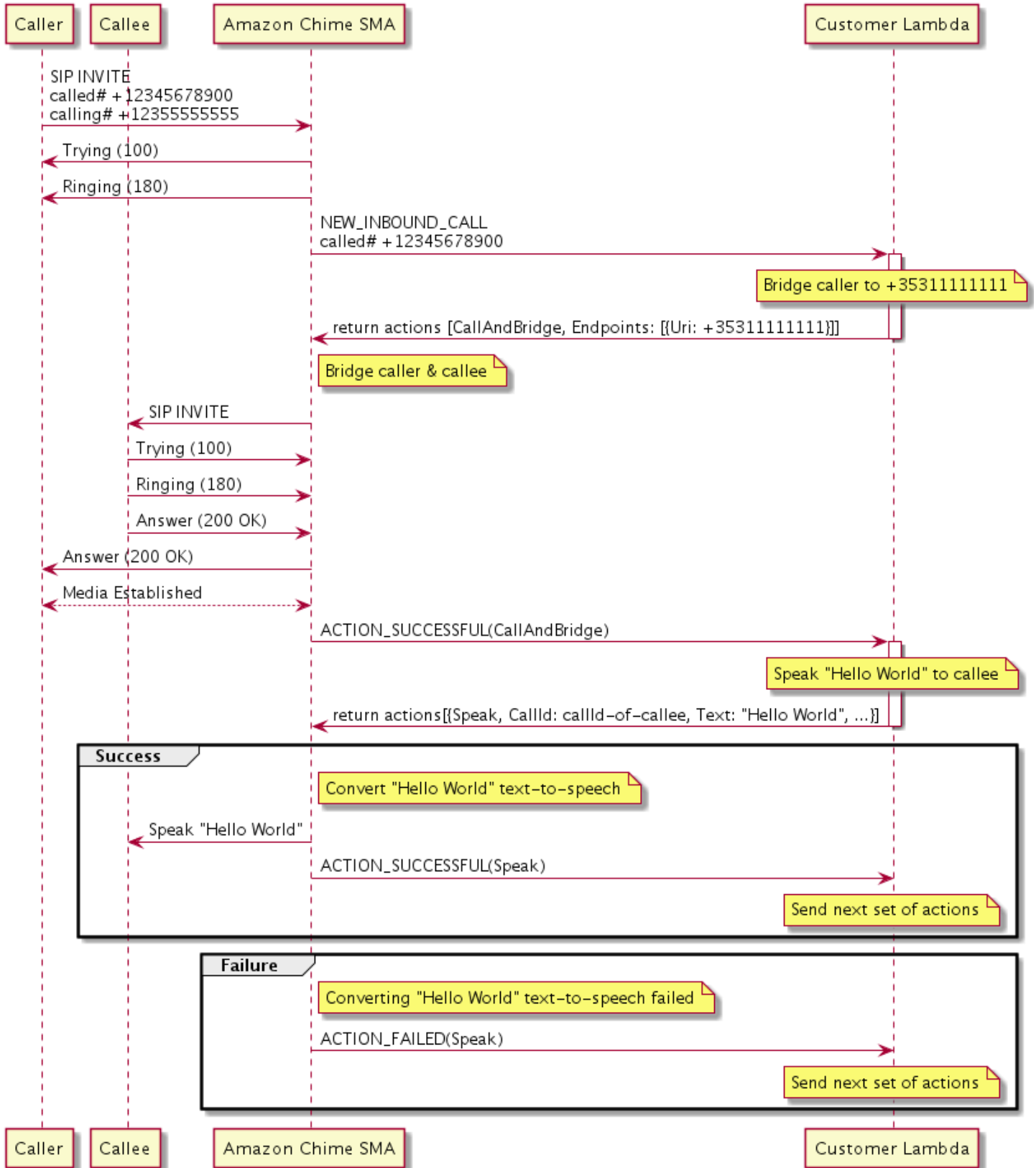
调用者使用软件电话输入注册到 SIP 媒体应用程序的号码。应用程序使用 SIP INVITE 法并向调用者发送 Trying (100) 响应。这表明下一跳服务器收到了调用请求。然后，SIP 应用程序使用 INVITE 联系终端节点。建立连接后，应用程序会向调用者发送 Ringing (180) 响应，并开始发出警报。

然后，SIP 媒体应用程序向 Lambda 函数发送 NEW_INBOUND_CALL 事件，Lambda 函数以包含调用者 ID 和要转换为语音的文本的 Speak 操作进行响应。然后，SIP 应用程序发送 200 (OK) 响应，表示调用已应答。该协议还启用媒体。

如果 Speak 操作成功并将文本转换为语音，则它会向 SIP 媒体应用程序返回一个 ACTION_SUCCESSFUL 事件，而该应用程序会返回下一组操作。如果操作失败，SIP 媒体应用程序会向 Lambda 函数发送 ACTION_FAILED 事件，而 Lambda 函数会以一组 Hangup 操作进行响应。应用程序挂断调用者并向 Lambda 函数返回 HANGUP 事件，而该函数不会执行任何进一步操作。

下图显示了为被调用者启用 Speak 操作的程序流程。

Enable Speak action for Callee in SMA



在图中

调用者输入注册到 SIP 媒体应用程序的号码，应用程序的响应如前图所述。当 Lambda 函数收到 `NEW_INBOUND_CALL` 事件时，它会将 [the section called “CallAndBridge”](#) 操作返回给 SIP 应用程序。然后，应用程序使用 SIP INVITE 法向被调用者发送 Trying (100) 和 Ringing (180) 响应。

如果被调用者应答，SIP 媒体应用程序会收到 200 (OK) 响应，并向调用者发送相同的响应。这会建立媒体，并且 SIP 应用程序会向 Lambda 函数发送 [the section called “CallAndBridge”](#) 操作的 `ACTION_SUCCESSFUL` 事件。然后，该函数将 Speak 操作和数据返回给 SIP 应用程序，后者会转换

SpeakAndGetDigits

通过提供文本播放语音，并从客户那里收集双音多频 (DTMF) 数字。文本可以是纯文本或者语音合成标记语言 (SSML) 增强型文本，以通过添加暂停、强调某些单词或更改说话风格以及其他支持的 SSML 功能更好地控制 Amazon Chime SDK 生成语音的方式。如果发生失败，例如用户未输入正确的 DTMF 数字，该操作将播放“失败”语音，然后重播主语音，直到 SIP 媒体应用程序耗尽 Repeat 参数中定义的尝试次数。

Amazon Chime SDK 使用 Amazon Polly — 一项将文本转换为逼真语音的云服务，可提供标准和神经引擎，以提高语音质量并支持超过 20 种语言和 60 种语音。Amazon Polly 免费提供语音功能，但使用 Amazon Polly 需要付费。有关定价信息，请参阅 Amazon Polly [定价页面](#) 或账单控制面板。

Important

使用 Amazon Polly 需遵守 [AWS 服务条款](#)，包括 AWS 机器学习和人工智能服务的特定条款。

主题

- [使用 SpeakAndGetDigits 操作](#)
- [处理 ACTION_SUCCESSFUL 事件](#)
- [处理 ACTION_FAILED 事件](#)
- [使用 Amazon Chime SDK 语音连接器服务相关角色](#)

使用 SpeakAndGetDigits 操作

以下示例显示典型的 SpeakAndGetDigits 操作：

```
{
```



```

"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

描述 — Lambda 函数调用的 CallDetails 中参与者的 CallId。

允许的值 — 有效的 callID

必填 — 是

默认值 — 否

InputDigitsRegex

描述 — 一种正则表达式模式，帮助确保用户输入正确的数字和字母。

允许的值 — 有效的正则表达式模式

必填 — 否

默认值 — 无

SpeechParameters.Engine

描述 — 指定在处理语音合成文本时使用的引擎：标准引擎或神经引擎。

允许的值 — standard | neural

必填 — 否

默认值 — 标准

SpeechParameters.LanguageCode

描述 — 指定语言代码。仅在使用双语语音时需要。如果使用双语语音但未指定语言代码，则使用双语语音的默认语言。

允许的值 — [Amazon Polly 语言代码](#)

必填 — 否

默认值 — 无

SpeechParameters.Text

描述 — 指定输入文本。如果指定 ssm1 为 SpeechParameters.TextType，必须按照 SSML 格式输入文本。有关 SSML 的更多信息，请参阅 Amazon Polly 开发人员指南中的[由 SSML 文档生成语音](#)。

允许的值 — 字符串

必填 — 是

默认值 — 无

SpeechParameters.TextType

描述 — 指定 `SpeechParameters.Text` 的文本格式。如未指定，默认使用 `text`。有关 SSML 的更多信息，请参阅 Amazon Polly 开发人员指南中的 [由 SSML 文档生成语音](#)。

允许的值 — `ssml | text`

必填 — 否

默认值 — `text`

SpeechParameters.VoiceId

描述 — 用于朗读 `SpeechParameters.Text` 中文本的 Amazon Polly 语音的 ID。

允许的值 — [Amazon Polly 语音 ID](#)

必填 — 否

默认值 — `Joanna`

FailureSpeechParameters.Engine

描述 — 指定在处理客户在为语音合成输入无效响应而使用故障消息时使用的引擎：标准引擎或神经引擎。

允许的值 — `standard | neural`

必填 — 否

默认值 — `SpeechParameters.Engine` 值

FailureSpeechParameters.LanguageCode

描述 — 指定客户输入无效响应时使用的语言代码。仅在使用双语语音时需要。如果您使用双语语音但未指定语言代码，则使用双语语音的默认语言。

允许的值 — [Amazon Polly 语言代码](#)

必填 — 否

默认值 — `SpeechParameters.LanguageCode` 值。

FailureSpeechParameters.Text

描述 — 指定客户输入无效响应时所说的输入文本。如果指定 `ssml` 为 `FailureSpeechParameters.TextType`，必须按照 SSML 格式输入文本。

允许的值 — 字符串

必填 — 是

默认值 — 无

FailureSpeechParameters.TextType

描述 — 指定 `FailureSpeechParameters.Text` 中指定的输入文本是纯文本还是 SSML。默认值为纯文本。有关更多信息，请参阅 Amazon Polly 开发人员指南中的 [由 SSML 文档生成语音](#)。

允许的值 — `ssml | text`

必填 — 否

默认值 — `SpeechParameters.Text` 值

FailureSpeechParameters.VoiceId

描述 — 用于朗读 `FailureSpeechParameters.Text` 中字符串的语音的 ID。

允许的值 — [Amazon Polly 语音 ID](#)

必填 — 是

默认值 — `SpeechParameters.VoiceId` 值

MinNumberOfDigits

描述 — 在超时或播放“调用失败”消息之前捕获的最小数字。

允许的值 — 大于或等于零

必填 — 否

默认值 — 0

MaxNumberOfDigits

描述 — 在没有终止数字的情况下停止前要捕获的最大数字位数。

允许的值 — 大于 `MinNumberOfDigits`

必填 — 否

默认值 — 128

TerminatorDigits

描述 — 当用户输入的数字小于 MaxNumberOfDigits 时，用于结束输入的数字

允许的值 — 以下任何一值：0、1、2、3、4、5、6、7、8、9、# 或 *

必填 — 否

默认值 — #

InBetweenDigitsDurationInMilliseconds

描述 — 播放失败语音之前数字输入之间的等待时间（以毫秒为单位）。

允许的值 — 大于零

必填 — 否

默认值 — 如未指定，默认为 RepeatDurationInMilliseconds 值

重复

描述 — 尝试获取数字的总次数。如果忽略此参数，则默认值为尝试收集数字一次。

允许的值 — 大于零

必填 — 否

默认值 — 1

RepeatDurationInMilliseconds

描述 — 每次尝试获取数字的超时时间（以毫秒为单位）。

允许的值 — 大于零

必填 — 是

默认值 — 无

处理 ACTION_SUCCESSFUL 事件

以下示例显示典型的 ACTION_SUCCESSFUL 事件。

```
{
```

```

"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":{
  ...
}
}

```

处理 ACTION_FAILED 事件

以下示例显示了典型的 ACTION_FAILED 事件。

```

{
  "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":{
  ...
}
}

```

错误处理

此表列出并描述了 Speak 操作引发的错误消息。

错误	消息	Reason
AccessDenied	AWSServiceRoleForAmazonChimeVoiceConnector 角色配置不正确。	用于向 Amazon Polly 发出请求的角色不存在或缺少权限。要解决这一问题，请参阅 使用

错误	消息	Reason
		Amazon Chime SDK 语音连接器服务相关角色 部分中的相关步骤
InvalidActionParameter		验证操作参数时出错。要查看此操作的可用参数及其选项，请参阅 Amazon Polly 开发人员指南中的 SynthesizeSpeech 。
MissingRequiredActionParameter	Text 是必填参数。	操作参数必须有 Text 值
MissingRequiredActionParameter	Text 限制为 1,000 个字符	文本超出了字符限制。
SystemException	运行操作时出现系统错误。	运行操作时出现一个系统错误。

使用 Amazon Chime SDK 语音连接器服务相关角色

无需手动为 Speak 或 SpeakAndGetDigits 操作创建该服务相关角色。当您在 Amazon Chime SDK 控制台、AWS Command Line Interface 或 AWS API 中创建或更新 SIP 媒体应用程序时，Amazon Chime SDK 会为您创建服务相关角色。

有关更多信息，请参阅 Amazon Chime SDK 管理员指南中的 [使用 Amazon Chime 服务相关角色](#)。

StartBotConversation

StartBotConversation 操作可在最终用户和您的 Amazon Lex v2 机器人之间建立语音对话。用户向机器人提供所需的信息。然后，机器人将信息返回到公共交换电话网 (PSTN) Audio Lambda 函数，然后该函数执行请求的任务。

例如，机器人可以在对话开始时播放欢迎消息（可选），以简要描述 PSTN Audio Lambda 函数可以执行的任务。对话在用户和机器人之间来回进行，直到机器人收集到所需的信息。对话结束后，Amazon Chime SDK 会通过操作成功事件调用您的 PSTN Audio Lambda 函数，该事件包含机器人收集的信息。您的 PSTN Audio Lambda 函数处理信息并执行请求的任务。

音频服务为您的用户提供逼真的对话互动。例如，用户可以在音频提示结束之前打断机器人并回答问题。此外，用户还可以使用语音和 DTMF 数字的任意组合来提供信息。机器人会等待用户提供输入后再做出响应。您可以配置机器人等待用户完成讲话后多久开始解释任何语音输入。用户还可以指示机器人在通话期间需要时间来检索其他信息（例如信用卡号）时等待。

在机器人对话期间，StartBotConversation 操作使用 Amazon Lex 和 Amazon Polly。适用 Amazon Lex 和 Amazon Polly 的标准费用。有关更多定价信息，请参阅 [Amazon Lex 流传输对话定价](#) 和 [Amazon Polly 定价](#) 页面。

Note

您无法对桥接呼叫或已加入 Amazon Chime SDK 会议的呼叫执行此操作。

Important

使用 Amazon Lex 和 Amazon Polly 须遵守 [AWS 服务条款](#)，包括 AWS 机器学习和人工智能服务的特定条款。

主题

- [StartBotConversation 语法](#)
- [使用动 StartBotConversation 作](#)
- [处理 ACTION_SUCCESSFUL 事件](#)
- [处理 ACTION_FAILED 事件](#)
- [授予使用机器人的权限](#)
- [配置语音和 DTMF 超时](#)
- [在对话期间使用 DTMF 输入](#)
- [账单和服务限额](#)

StartBotConversation 语法

以下示例显示典型 StartBotConversation 语法。

```
{
  "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

描述- AWS Lambda 函数调用中CallDetails参与者的描述。CallIDStartBotConversation 操作使用此 ID 作为机器人的 SessionId。呼叫中发生的所有机器人对话共享相同的对话会话。您可以使用 [Amazon Lex PutSession](#) API 修改您的用户和机器人之间的会话状态。有关更多信息，请参阅 Amazon Lex 开发人员指南中的[使用 Amazon Lex v2 API 管理会话](#)。

允许的值 — 有效的调用 ID。

必填 — 否，如 ParticipantTag 已存在。

默认值 — 无。

ParticipantTag

描述 — CallDetails 中一个已连接参与者的 ParticipantTag。

允许的值 — LEG-A

必填 — 否，如 CallId 已存在。

默认值 — 被调用的 callLeg 的 ParticipantTag。如果您指定 CallDetails，则忽略该值。

BotAliasArn

描述 — 您的 Lex 机器人的机器人别名 ARN。您必须在与您的 PSTN Audio 应用程序相同的 AWS 区域创建机器人。有效的 Amazon Lex 机器人别名采用以下格式：`arn:aws:lex:region:awsAccountId:bot-alias/botId/botAliasId`，其中，`region` 是您的机器人所在的 AWS 区域。`awsAccountId` 是创建您的 Amazon Lex 机器人时使用的 AWS 账户 ID。`botId` 值是您在创建机器人时为该机器人指定的标识符。您可以在 Amazon Lex 控制台的机器人详情页面上查看该机器人 ID。`botAliasId` 是您在创建机器人时为该机器人别名指定的标识符。您可以在 Amazon Lex 控制台的别名页面上查看该机器人别名 ID。

允许的值 — 有效的机器人 ARN。

必填 — 是。

默认值 — 无

LocaleId

描述 — 用于机器人的区域设置的标识符。有关区域设置和语言代码的列表，请参阅 [Amazon Lex 支持的语言和区域设置](#)。

允许的值 — [Amazon Lex 支持的语言和区域设置](#)

必填 — 否。

默认值 — en_US

配置

描述 — 对话配置，包括会话状态和欢迎消息。Configuration 对象的 JSON 字符串表示形式总大小限制为 10KB。

允许的值 — Configuration 对象。

必填 — 否。

默认值 — 无。

配置。SessionState

描述 — 用户与 Amazon Lex V2 的会话状态。

允许的值 — `SessionState` 对象。

必填 — 否。

默认值 — 无。

配置。 `SessionState`。 `SessionAttributes`

描述 — 表示会话特定上下文信息的键/值对的映射。该映射包含在 Amazon Lex v2 与客户端应用程序之间传递的应用程序信息。

允许的值 — 字符串到字符串的映射。

必填 — 否。

默认值 — 无。

配置。 `SessionState`。 `DialogAction.Type`

描述 — 机器人与用户交互时采取的下一步操作。可能的值：

- 委托 Amazon Lex v2 决定下一步操作。
- `ElicitIntent` 下一个操作会引起用户的意图。

允许的值 — `Delegate` | `ElicitIntent`。

必填 — 否。

默认值 — 无。

配置。 `WelcomeMessages`

描述 — 对话开始时要发送给用户的消息列表。如果设置了 `welcomeMessage` 字段，则必须将 `DialogAction.Type` 值设置为 `ElicitIntent`。

允许的值 — 消息对象

必填 — 否。

默认值 — 无。

配置。 `WelcomeMessages`. 内容

描述 — 欢迎消息文本。

允许的值 — 字符串

必填 — 否。

默认值 — 无。

配置。WelcomeMessages。ContentType

描述 — 表示欢迎消息类型。

允许的值 — PlainText | SSML

- PlainText— 该消息包含纯文本 UTF-8 文本。
- SSML — 消息包含语音输出设置格式的文本。

必填 — 是。

默认值 — 无。

使用 `StartBotConversation` 作

以下示例显示典型的 `StartBotConversation` 操作。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "StartBotConversation",
      "Parameters": {
        "CallId": "call-id-1",
        "BotAliasArn": "arn:aws:lex:us-east-1:123456789012:bot-alias/ABCDEFGHIJ/MNOPQRSTU",
        "LocaleId": "en_US",
        "Configuration": {
          "SessionState": {
            "SessionAttributes": {
              "mykey1": "myvalue1"
            },
            "DialogAction": {
              "Type": "ElicitIntent"
            }
          },
          "WelcomeMessages": [
            {
              "Content": "Welcome. How can I help you?",
              "ContentType": "PlainText"
            }
          ]
        }
      }
    }
  ]
}
```

```

    }
  ]
}
}
}
]
}
}

```

处理 ACTION_SUCCESSFUL 事件

以下示例显示 StartBotConversation 操作的典型 ACTION_SUCCESSFUL 事件。

```

{
  "SchemaVersion": "1.0",
  "Sequence": number,
  "InvocationEventType": "ACTION_SUCCESSFUL",
  "ActionData":
  {
    "CallId": "string",
    "Type": "StartBotConversation",
    "Parameters": {
      // parameters provided in the StartBotConversation action.
    },
    "CallDetails": {
      // Information about the call associated with the AWS Lambda invocation.
    },
    "IntentResult": {
      "SessionId": "string",
      "SessionState": {
        "SessionAttributes": {
          "string": "string"
        },
        "Intent": {
          "Name": "string",
          "Slots": {
            "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

用户与机器人之间对话的结果。

SessionId

机器人对话会话的标识符。用户启动与您的机器人的对话时，Amazon Lex 会创建一个会话。会话封装了您的用户和机器人之间交换的信息。StartBotConversation 操作使用此调用 ID 作为机器人的 SessionId。您可以使用 Lex [PutSession](#) API 修改用户和机器人之间的会话状态。有关更多信息，请参阅 Amazon Lex 开发人员指南中的 [使用 Amazon Lex v2 API 管理会话](#)。

SessionState

用户的 Amazon Lex V2 会话的状态。

SessionState.SessionAttributes

表示会话特定上下文信息的键/值对的映射。该映射包含在您的机器人所附的 Lambda 函数和 PSTN 音频 Lambda 函数之间传递的机器人对话信息。

解释

由 Amazon Lex 得出的可能满足客户言论的意图清单。NluConfidence 得分最高的意图变为意图 SessionState。

解释。 NluConfidence.Score

表示 Amazon Lex v2 对意图满足用户意图的置信度的分数。范围介于 0.00 和 1.00 之间。分数越高，置信度越高。

意图

用户要执行的操作。

Intent.Name

意图的名称。

Intent.Slots

意图中所有插槽的映射。插槽的名称映射于该插槽的值。如果某个插槽尚未填满，则该值为 null。

Intent.Slots.Value

插槽的值。

Intent.Slots.Values

用户为插槽提供的一或多个值的列表。

Intent.Slots.Value。 OriginalValue

为该插槽输入的用户回复文本。

Intent.Slots.Value。 InterpretedValue

描述 — Amazon Lex v2 为该插槽确定的值。实际值取决于机器人的值选择策略设置。您可以选择使用用户输入的值，也可以让 Amazon Lex v2 选择 resolvedValues 列表中的第一个值。

Intent.Slots.Value。 ResolvedValues

Amazon Lex v2 识别的该插槽的其他值列表。

Intent.State

描述 — 意图的实现信息。可能的值：

- 失败 — Lambda 函数未能实现意图。
- 已完成 — Lambda 函数已实现意图。
- ReadyForFulfillment— 意图信息已存在，您的 LambdaFunction 可以实现意图。

意图。ConfirmationState

描述 — 表示意图的确认。可能的值：

- 已确认 — 意图已实现。
- 已拒绝 — 用户对确认提示响应“否”。
- 否 — 不提示用户进行确认；或者系统已提示用户，但用户未确认也未拒绝提示。

处理 ACTION_FAILED 事件

以下示例显示 StartBotConversation 操作的典型 ACTION_FAILED 事件。

```
{
  "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

唯一标识错误条件的字符串。

ErrorMessage

错误条件的一般描述。

错误代码

下表列出了 Lambda 函数在 ACTION_FAILED 事件中可能返回的错误消息。

错误	描述
InvalidActionParameter	一个或多个操作参数无效。错误消息描述无效参数。
SystemException	运行操作时出现系统错误。
ResourceNotFound	未找到指定机器人。
ResourceAccessDenied	对存储桶的访问被拒绝。
ActionExecutionThrottled	已超出机器人对话服务限制。错误消息描述特定服务限制已超出。

授予使用机器人的权限

以下示例授予亚马逊 Chime 软件开发工具包调用 Amazon Lex [StartConversation](#) API 的权限。您必须明确授予音频服务使用您的机器人的权限。服务主体必须使用条件数据块。条件数据块必须使用全局上下文键 `AWS:SourceAccount` 和 `AWS:SourceArn`。`AWS:SourceAccount` 是您的 AWS 账户 ID。`AWS:SourceArn` 是调用 Lex 机器人的 PSTN 音频应用程序的资源 ARN。

```
{
  "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"
      }
    }
  }
}
```

配置语音和 DTMF 超时

在捕获用户输入时，您可以配置语音和 DTMF 超时。您还可以在开始与机器人对话时通过会话属性配置超时，并在必要时在 Lambda 函数中将其覆盖。Amazon Lex 可以为意图或机器人设置多个插槽。由于您可以指定会话属性应用于意图和时隙级别，因此可以指定仅在收集特定类型的输入时设置该属性。例如，您可以在收集账号时指定比收集日期时更长的超时。您可以在会话属性键中使用通配符。

例如，要将所有意图的所有插槽的语音超时设置为 4000 毫秒，您可以使用 `x-amz-lex:start-timeout-ms:*:*` 作为会话属性名称和 `4000` 作为会话属性值来提供会话属性。有关更多信息，请参阅 Amazon Lex 开发人员指南中的[配置捕获用户输入的超时时间](#)。

在对话期间使用 DTMF 输入

Amazon Lex 机器人支持对话期间的语音和键盘输入。机器人将键盘输入解释为 DTMF 数字。您可以使用井号键 (#) 提示联系人结束输入，并使用星号键 (*) 取消会话。如果您未提示客户使用井号键结束输入，Lex 将在其他按键操作 5 秒钟后停止等待。

账单和服务限额

AWS 向您收取以下费用：

- Amazon Chime SDK 用于呼叫的使用量。有关更多信息，请参阅 [Amazon Chime SDK 定价](#)。
- Amazon Lex 用于解释用户语音的使用量。有关更多信息，请参阅 [Amazon Lex 流传输对话定价](#)。
- Amazon Polly 用于合成机器人文本回复的使用量。有关更多信息，请参阅 [Amazon Polly 定价](#)。

此外，您还需要了解以下服务限额：

- Amazon Chime SDK 的服务配额是你可以在 PSTN Audio 操作中使用的最大数量的 Amazon Lex 机器人。[StartBotConversation](#) 有关更多信息，请参阅《AWS 一般参考》中的 [SIP 中继和语音配额](#)。

- Amazon Lex 对每个 Lex 机器人的最大并发语音对话数都有服务限额。如需增加限额，您可以联系 Amazon Lex 服务团队。有关更多信息，请参阅 Amazon Lex 开发人员指南中的 [Amazon Lex 指南和限额](#)。
- Amazon Polly 具有用于合成文本响应的服务限额。如需增加限额，您可以联系 Amazon Polly 服务团队。有关 Amazon Polly 服务限额的更多信息，请参阅 Amazon Polly 开发人员指南中的 [Amazon Polly 中的限额](#)。

使用 SIP 标头

现在，当您想要与 SIP 基础设施交换调用上下文信息时，可以在 AWS Lambda 函数中发送和接收用户对用户标头、转移标头和自定义 SIP 标头。

- 用户对用户 (UUI) 标头可用于发送调用控制数据。该数据由发起会话的应用程序插入，并由接收会话的应用程序使用。它不用于任何基本的 SIP 功能。例如，您可以使用调用中心中的 UUI 标头在座席之间传递有关调用的信息。
- 转移标头用于显示调用从何处转移以及转移原因。您可以使用此标头查看来自其他 SIP 座席的转移信息或者将其传递出去。
- 自定义 SIP 标头可用于传递所需的任何其他信息。例如，如果想要传递账户 ID，您可以创建一个名为“X-Account-Id”的 X 标头并添加此信息。

您必须在自定义 SIP 标头前面加上 x-。标头在 AWS Lambda 函数中公开，并在进站调用期间作为 NEW_INBOUND_CALL 事件的一部分接收。您还可在触发 [CallAndBridge](#) 操作或 [CreateSipMediaApplicationCall](#) API 时，在出站调用分支中包含这些标头。

Lambda 函数的 Participants 部分包含 SipHeaders 字段。该字段在您收到自定义标头或填充 User-to-User 或 Diversion 标头时可用。

该示例显示了 AWS Lambda 调用包含 SIP 标头时的预期响应。

```
{
  "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"
    }
  ]
}
}

```

以下示例显示了由于 SipHeaders 参数的输入无效而导致的成功 [CallAndBridge](#) 操作。

```

{
  "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"
    }
  }
}

```

以下示例显示了由无效 SipHeaders 参数导致的 [CallAndBridge](#) 操作失败。

```

{
  "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"
    }
]
}
```

使用 sip-headers 字段

当您触发 [CreateSipMediaApplicationCall](#) API 时，可选 SipHeaders 字段允许您将自定义 SIP 标头传递至出站调用分支。有效标头密钥必须包括以下内容之一：

- x- 前缀
- User-to-User 标头
- Diversion 标头

X-AMZN 为保留的标头。如果您在 API 调用中使用此标头，它将失败。标头的最大长度为 2048 个字符。

以下示例显示了命令行界面中带有可选 SipHeaders 参数的典型 [CreateSipMediaApplicationCall](#) API。

```
create-sip-media-application-call
  --from-phone-number value // (string)
  --to-phone-number value // (string)
  --sip-media-application-id value // (string)
  --sip-headers // (map)
```

有关更多信息，请参阅 [SIP 中传输用户到用户调用控制信息的机制](#) 以及 [SIP 中的转移指示](#)。

使用呼叫详细信息记录

Amazon Chime SDK 管理员可以配置 Amazon Chime SDK 语音连接器以存储呼叫详细记录 (CDR)。有关配置 Amazon Chime SDK 语音连接器以存储 CDR 的更多信息，请参阅《Amazon Chime SDK 管理员指南》中的[管理 Amazon Chime SDK 中的全局配置](#)。

启用 CDR 后，每次调用时，SIP 媒体应用程序都会将记录发送到 S3 存储桶中名为 Amazon-Chime-SMADRs 的文件夹。

下表列出了 CDR 的属性并显示它们的正确格式。记录包含此处列出的所有呼叫的所有字段。

值	描述
"AwsAccountId": " <i>AWS-account-ID</i> "	与启动 PSTN 使用的 SIP 媒体应用程序关联的 AWS 账户 ID
"TransactionId": " <i>transaction-ID</i> "	呼叫的事务 ID。
"CallId": " <i>SIP-media-application-call-ID</i> "	与使用情况相关的参与者的呼叫 ID。
"VoiceConnectorId": " <i>voice-connector-ID</i> "	Amazon Chime SDK 语音连接器 ID UUID
"Status": " <i>status</i> "	呼叫状态 (已完成、失败)
"BillableDurationSeconds": " <i>billable-duration-in-seconds</i> "	呼叫的可计费时长 (以秒为单位)
"SchemaVersion": " <i>schema-version</i> "	CDR 架构版本
"SourcePhoneNumber": " <i>12075550155</i> "	E.164 发起电话号码
"SourcePhoneNumberName": " <i>North Campus Reception</i> "	分配给源电话号码的名称
"DestinationPhoneNumber": " <i>13605551214</i> "	E.164 目标电话号码

值	描述
"DestinationPhoneNumberName": " <i>South Campus Reception</i> "	分配给目标电话号码的名称
"UsageType": " <i>usage-type</i> "	价目表 API 中行项目的使用详情
"ServiceCode": " <i>service-code</i> "	价目表 API 中的服务代码
"Direction": " <i>direction</i> "	呼叫方向，Outbound 或 Inbound。
"TimeStampEpochSeconds": " <i>start-time-epochseconds</i> "	记录的时间戳，格式为纪元/Unix 时间戳
"Region": " <i>AWS-region</i> "	AWS Amazon Chime SDK 语音连接器所在区域
"SipRuleId": " <i>sip-rule-id</i> "	呼叫到达 PSTN 音频服务时触发的 sip 规则的 ID
"SipApplicationId": " <i>sip-application-id</i> "	处理呼叫的 SIP 应用程序的 ID。
"CallLegTriggerType": " <i>trigger-type</i> "	触发呼叫的事件类型
"BillableVoiceFocusSeconds": " <i>billable-voice-focus-in-seconds</i> "	Voice Focus 的计费使用量，以秒为单位

超时和重试

PSTN 音频服务与功能同步交互。AWS Lambda 在重试调用之前，应用程序会等待 5 秒钟让 AWS Lambda 函数响应。当函数返回包含 4 个 XX 状态码之一的错误时，默认情况下，SIP 媒体应用程序仅重试调用一次。如果重试次数用完，则调用将以 480 Unavailable 错误代码终止。有关 AWS Lambda 错误的更多信息，请参阅中的 [调用问题疑难解答](#)。 [AWS Lambda](#)

调试和故障排除

您可以使用以下信息诊断和修复在使用 Amazon Chime SDK PSTN Audio 服务时可能遇到的常见问题。

主题

- [查看日志](#)
- [调试意外挂断](#)
- [调试意外 ACTION_FAILED 事件](#)

查看日志

如果您正在调试 SIP 媒体应用程序，请查看 CloudWatch 日志，了解与该应用程序关联的 AWS Lambda 功能。

接下来，检查与 SIP 媒体应用程序关联的日志。根据需要，您可以配置 SIP 媒体应用程序进行日志记录。有关更多信息，请参阅 Amazon Chime SDK 管理员指南中的[使用 SIP 媒体应用程序](#)。如果您启用日志记录，则可以在 `/aws/ChimeSipMediaApplicationSipMessages/` SIP 媒体应用程序 Id 日志组中的 CloudWatch 上中找到日志。

调试意外挂断

- 验证您的 AWS Lambda 策略是否授予 voiceconnector.chime.amazonaws.com 服务主体 `lambda:InvokeFunction` 权限。
- 检查您的 AWS Lambda 函数的日志，确保其已成功调用。
- 如果日志显示传入事件和返回操作，请验证在调用 AWS Lambda 函数时没有返回挂断操作。
- 查看您的 SIP 媒体应用程序的 CloudWatch 日志。下表列出可能显示的一些消息。

消息	解析
AWS Lambda 客户端操作已超时。	该功能需要至少 20 秒才能完成。将响应时间缩短到 20 秒以内。
调用 AWS Lambda 函数时访问被拒绝。	AWS Lambda 函数未提供允许该服务访问 Amazon Chime SDK 语音连接器服务主体的策略。向 <code>voiceconnector.chime.amazon</code>

消息	解析
AWS Lambda 函数被节流。	aws.com 服务主体提供 AWS Lambda 策略中的 <code>lambda:InvokeFunction</code> 权限。 音频服务无法调用您的 AWS Lambda 函数，因为该函数已被节流。有关更多信息，请参阅 https://aws.amazon.com/premiumsupport/knowledge-center/lambda-troubleshoot-throttling/ 。
读取操作列表时出错。	PSTN 音频服务无法解析 AWS Lambda 函数返回的操作。检查日志中是否有 <code>ACTION_FAILED</code> 事件，并查阅失败操作文档，以确保已正确对其进行编码。
调用请求中的架构版本与响应中的架构版本不匹配。	检查您的日志，并确保您的请求和响应使用相同的架构版本。
已指定不受支持的操作名称	AWS Lambda 函数返回 PSTN 音频服务无法识别的操作。确保操作拼写正确，并查看操作文档。
操作列表为空。	对 <code>NEW_INCOMING_CALL</code> 事件的响应未返回任何操作。返回响应该事件的操作。
响应中指定的操作过多。	您为响应 AWS Lambda 调用返回了 10 次以上的操作。返回 10 次或更少操作。
响应为空白或为空	您返回了 <code>null</code> 或空字符串。确保响应对象至少包含 <code>SchemaVersion</code> 字段。

调试意外 ACTION_FAILED 事件

如果收到意外 `ACTION_FAILED` 事件，请检查以下几点：

操作	错误类型	错误消息	解析
CallAndBridge 、 PlayAudio 和 PlayAudioAndGetDigits	InvalidAudioSource	无法访问 S3 存储桶或音频文件。	<ul style="list-style-type: none"> • 确保 S3 存储桶与 SIP 媒体应用程序处于同一 AWS 账户。 • 确保 S3 存储桶已向 voiceconnector.chime.amazonaws.com 服务主体授予 s3:GetObject 权限。
PlayAudio 和 PlayAudioAndGetDigits	InvalidAudioSource	音频来源参数值无效。	<ul style="list-style-type: none"> • 确保使用有效的 Type，例如 S3。 • 确保 S3 存储桶已向 voiceconnector.chime.amazonaws.com 服务主体授予 s3:GetObject 权限。 • 确保 BucketName 字段不为 null 或空。 • 确保 Key 字段不为 null 或空。
CallAndBridge	InvalidAudioSource	铃声参数值无效。	<ul style="list-style-type: none"> • 确保使用有效的 Type，例如 S3。 • 确保 BucketName 字段不为 null 或空。 • 确保 Key 字段不为 null 或空。

操作	错误类型	错误消息	解析
	InvalidActionParameter	提供的终端节点数量无效。	确保终端节点不为 null 或零且不大于 1。
	InvalidActionParameter	终端节点参数无效。	<ul style="list-style-type: none"> • 确保提供了终端节点 URI 值。 • 如果终端节点类型为 PSTN，确保 Uri 字段中提供的电话号码是有效的 E.164 电话号码。 • 如果终端节点类型为 PSTN，确保 ARN 字段未设置或设置为 null。
	InvalidActionParameter	来电者 ID 无效。	在 CallerId 字段中提供有效的 E.164 格式的电话号码。
	InvalidActionParameter	未定义来电者 ID。	在 CallerId 字段中提供有效的 E.164 格式的电话号码。
	InvalidActionParameter	MaxCallTimeout 参数无效。超时必须介于 0 到 120 秒之间。	将 MaxCallTimeout 间隔设置为 0 到 120 秒之间的值。
	InvalidActionParameter	提供的来电者 ID 号码无效。号码必须归该 AWS 账户所有，或者是 LEG-A 的 From 号码。	<ul style="list-style-type: none"> • 确保 CallerId 号码已配置并与 SIP 媒体应用程序相同的 AWS 账户关联。 • 如果该号码没有与账户关联，则它必须与 LEG-A From 字段中的号码相匹配。

操作	错误类型	错误消息	解析
	InvalidActionParameter	提供的 SIP 标头无效：{标头}。	<ul style="list-style-type: none"> 移除所有内部自定义标头：x-vine、x-amzn、x-vc、x-canary、x-voice。 确保您的自定义标头以 x- 开头。您也可以将它们设置为 user-to-user 或 diversion。
JoinChimeMeeting	InvalidActionParameter	JoinToken 参数值无效。	<ul style="list-style-type: none"> 验证会议加入令牌是否正确。 验证与该令牌关联的参与者是否仍是会议的有效与会者。 验证会议是否仍然存在。
ModifyChimeMeetingAttendee (音频静音和取消静音)	InvalidActionParameter	ModifyChimeMeetingAttendees 操作的 Operation 字段中的值无效。	确保服务支持静音和取消静音操作。
	InvalidActionParameter	会议 ID 参数无效。	确保会议 ID 正确无误。
	InvalidActionParameter	与会者列表参数无效。	您没有提供任何与会者，或者您提供的与会者超过 100 人。提供 1 到 100 名与会者。

操作	错误类型	错误消息	解析
	InvalidActionParameter	一位或多位与会者并不是此会议的一部分。所有与会者必须是会议的一部分。	操作中提供的其中一位与会者不是指定会议的有效参与者。移除所有不在会议中的与会者。
Pause	InvalidActionParameter	Duration 参数无效。	将暂停持续时间设置为 100 到 30000 之间。
PlayAudioAndGetDigits	InvalidActionParameter	MaxNumberOfDigits 参数无效。	确保 MaxNumberOfDigits 介于 0 和 128 之间，并且大于 MinNumberOfDigits /
	InvalidActionParameter	RepeatDurationInMilliseconds 参数无效。	确保该 RepeatDurationInMilliseconds 值为正。
	InvalidActionParameter	InputDigitsRegex 参数无效。	确保 InputDigitsRegex 是有效的正则表达式模式。
ReceiveDigits	InvalidActionParameter	InBetweenDigitsDurationInMilliseconds 参数无效。	确保值大于 0。

操作	错误类型	错误消息	解析
	InvalidActionParameter	FlushDigitsDurationInMilliseconds 参数无效。	FlushDigitsDurationInMilliseconds 间隔小于或等于 InBetweenDigitsDurationInMilliseconds 间隔。使 InBetweenDigitsDurationInMilliseconds 间隔大于 FlushDigitsDurationInMilliseconds 间隔。
	InvalidActionParameter	InputDigitsRegex 参数无效。	确保值不为空或 null。
RecordAudio	InvalidActionParameter	RecordingDestination 参数无效。	<ul style="list-style-type: none"> 验证 Type 字段是否有效，例如 S3。 验证 BucketName 字段是否为空或 null。 验证前缀是否包含有效字符。 验证前缀是否小于或等于 979 个字节。
	InvalidActionParameter	DurationInSeconds 参数无效。	DurationInSeconds 不得为空且必须大于 0。

操作	错误类型	错误消息	解析
	InvalidActionParameter	SilenceThreshold 参数无效。	SilenceThreshold 不能为空且必须介于 1 和 1000 之间。
	InvalidActionParameter	SilenceDurationInSeconds 参数无效。	SilenceDurationInSeconds 不得为空且必须大于 0。
	InvalidActionParameter	将录音上传到 S3 存储桶时出错。	<ul style="list-style-type: none"> • 确保 S3 存储桶与 SIP 媒体应用程序处于同一 AWS 账户。 • 确保 S3 存储桶已向 voiceconnector.chime.amazonaws.com 服务主体授予 s:PutObject 和 :PutObjectAcl 权限。
StartCallRecording	InvalidActionParameter	录制轨道无效。选择 INCOMING、OUTGOING 或 BOTH。	将 Track 参数设置为 INCOMING、OUTGOING 或 BOTH。
	InvalidActionParameter	Destination 参数无效。	<ul style="list-style-type: none"> • 确保 Type 字段包含有效值，例如 S3。 • 确保 Location 字段不为 null 或空。
VoiceFocus	MissingRequiredActionParameter	缺少必需操作参数。	为 Enable 参数提供有效的布尔值。

VoiceFocus

能够将 Amazon Voice Focus 噪音抑制应用于公共交换电话网络 (PSTN) 调用的入站和出站调用分支。当您应用 Amazon Voice Focus 时，它可以在不影响人类语音的情况下减少背景噪音。这可以使当前发言人更容易被听见。

要创建入站调用分支，使用 [SIP 规则](#)，该规则可调用带有 `NewInboundCall` 事件的 AWS Lambda 函数。您可以使用 [CallAndBridge](#) 操作或使用 [CreateSIPMediaApplicationCall API](#) 操作创建出站调用分支。有关 Amazon Voice Focus 的更多信息，请参阅 [Amazon Chime SDK 的噪音消除工作原理](#)。

Amazon Voice Focus 可减少不必要的非语音噪音，包括：

- 环境噪音 — 风、风扇、自来水
- 背景噪音 — 割草机、狗吠声
- 前景噪音 — 打字、纸张哗啦声

Note

当您使用 Amazon Voice Focus 时，AWS 会根据每个调用分支的活跃调用分钟数和 SIP 媒体应用程序的每分钟使用量向您收费。

此示例显示了一个典型的 VoiceFocus 操作。

```
{
  "SchemaVersion": "1.0",
  "Actions": [
    {
      "Type": "VoiceFocus",
      "Parameters": {
        "Enable": True/False,           // required
        "CallId": "call-id-1",       // required
      }
    }
  ]
}
```

Enable

描述 — 启用或禁用 Amazon Voice Focus

允许的值 — True | False

必填 — 是

默认值 — 无

CallId

描述 — AWS Lambda 函数调用的 CallDetails 中参与者的 CallId

允许的值 — 有效的调用 ID

必填 — 是

默认值 — 无

此示例显示了 VoiceFocus 操作的成功 ACTION_SUCCESSFUL 事件。

```
{
  "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"
      }
    ]
  }
}
```

```

    ]
  }
}

```

此示例显示了 VoiceFocus 操作的典型 ACTION_FAILED 事件。

```

{
  "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",
      .....
    }
  ]
}
}

```

错误处理

出于安全考虑，PSTN 音频操作的每个客户账户 (CPS) 的限制为每秒 5 个调用请求。当调用请求超过 5 个 CPS 限制时，操作会返回错误消息。此表列出了 VoiceFocus 操作返回的错误消息。

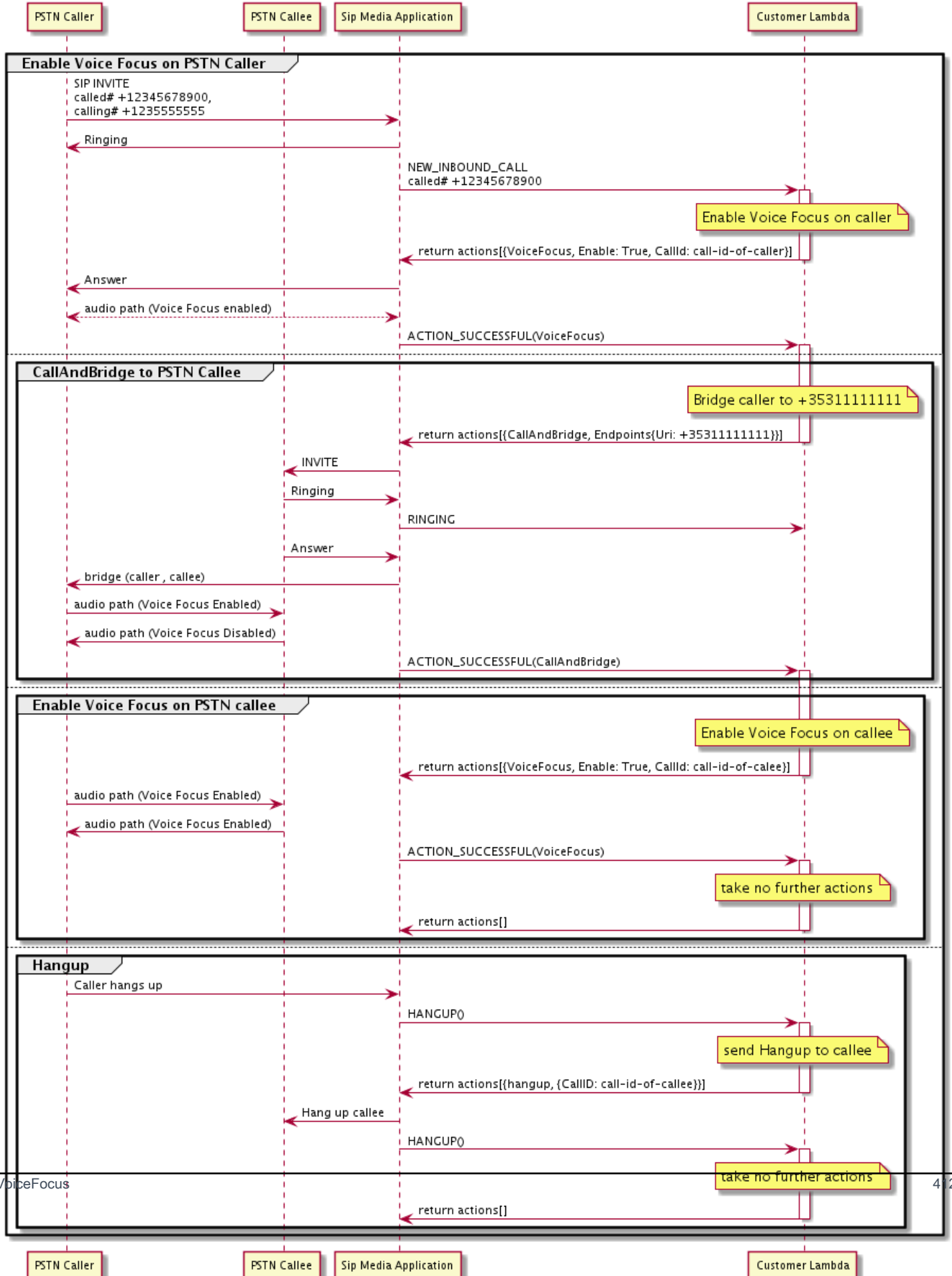
错误	消息	Reason
ActionExecutionThrottled	运行操作失败。已达到每秒最大操作数。	每秒的 Voice Focus 操作请求数超过系统限制。

错误	消息	Reason
MissingRequiredActionParameter	缺少必需操作参数。	运行操作时缺少一个或多个必需参数。
SystemException	运行操作时出现系统错误。	运行操作时出现一个系统错误。

调用流

此图表显示了针对两个 PSTN 调用之间的 CallAndBridge 操作启用和禁用 Amazon Voice Focus 的调用流。

Voice Focus between 2 PSTN parties



PSTN 音频服务词汇表

| [A](#) | [C](#) | [E](#) | [I](#) | [L](#) | [M](#) | [N](#) | [O](#) | [P](#) | [S](#) | [T](#) | [V](#) |

A

操作

在 AWS Lambda 函数中，操作是您要在电话呼叫分支中运行的项目，例如发送或接收数字、加入会议等。有关 PSTN 音频服务支持的操作的更多信息，请参阅 [PSTN 音频服务支持的操作](#)。

AWS Lambda

一种计算服务，可使您无需预置或管理服务器即可运行几乎所有类型的应用程序或后端服务的代码。

AWS Lambda 函数

在 PSTN 音频服务环境中，为响应由 SIP 媒体应用程序传递的数据（例如拨打出站呼叫）而运行的函数。

C

呼叫详细信息记录

来自 Amazon Chime SDK 语音连接器呼叫的数据，例如账户 ID、来源电话号码和目的地国家/地区。这些记录作为对象存放在您账户的 Amazon Simple Storage Service (S3) 存储桶中。有关更多信息，请参阅 Amazon Chime SDK 管理员指南中的 [在 Amazon Chime SDK 中管理全局设置](#)。有关记录架构的更多信息，请参阅本指南中的 [使用呼叫详细信息记录](#)。

呼叫 ID

分配给所有传入呼叫分支的 ID。

呼叫分支

呼叫的一部分。在 Amazon Chime SDK 应用程序中，呼叫可以来自有效的电话号码、PSTN 或 Amazon Chime SDK 语音连接器。有关更多信息，请参阅本指南中的 [关于使用 PSTN 音频服务呼叫分支](#)。

Carrier

提供移动服务的公司。无线运营商的缩写。

Amazon Chime

由 AWS 提供的统一通信和协作服务。

Amazon Chime SDK

开发人员使用的软件开发套件，用于添加实时媒体和通信至自定义通信应用程序。

E

E.164

PSTN 音频服务中唯一可接受的电话号码格式。一项 ITU-T 建议，号码使用 1 - 3 位数的国家/地区代码，后面是最多 12 位数的订阅用户号码。例如：美国：+14155552671，英国：+442071838750 44，澳大利亚：+61285993444。

终端节点

一种硬件设备或软件服务，例如电话或统一通信应用程序。

EventBridge

一种无服务器事件总线服务，使您能够将您的应用程序与来自多种来源的数据连接起来。

Note

SIP 媒体应用程序不向 EventBridge 发送数据。有关更多信息，请参阅 Amazon Chime SDK 管理员指南中的[使用 EventBridge 自动运行 Amazon Chime SDK](#)。

I

IVR

交互式语音应答。一种允许人们通过语音识别或触音键盘与计算机操作的电话系统进行交互的系统。

L

分支

请参阅[Call leg](#)。

M

媒体

可在 Amazon Chime SDK 会议期间使用的音频、视频或聊天消息。自定义通信应用程序可以包含每种媒体类型中的一种或多种。

媒体管道

一种在 Amazon Chime SDK 会议期间流式传输和捕获音频、视频、消息和事件的机制。有关更多信息，请参阅本指南中的 [创建 Amazon Chime SDK 媒体管道](#)。

N

携号转网

在电话运营商或统一通信系统之间移动电话号码的能力。

O

起源

接收来自 PSTN 的呼叫并将该呼叫移交给 VoIP 终端节点的过程。

P

参与者标签

分配给每个呼叫参与者的标识符，LEG-A 或 LEG-B。

策略

Amazon Chime SDK 需要以下策略类型：

- IAM 用户策略 — 定义身份和访问管理用户权限的策略。
- 会议策略 — 一种策略，允许一个用户在会议期间共享屏幕时控制另一个用户的计算机，并允许会议参与者选择通过接听 Amazon Chime SDK 的电话呼叫加入会议。

PSTN

公用交换电话网路。提供电话呼叫功能的基础设施和服务。

PSTN 音频服务

一种 Amazon Chime SDK 服务，使开发人员能够添加音频功能至自己的通信解决方案。

R

路由

使用 Amazon Chime SDK 来创建的应用程序使用一种或多种路由类型：

- 网络路由 — 为网络中或多个网络之间或跨多个网络的流量选择路径的过程。
- 交互路由 — 确保呼叫转到正确的接收者或终端节点的过程。
- 呼叫路由 — 一种呼叫管理功能，可将入站呼叫排队并分配给预定义的接收者或终端节点。

S

SBC

会话边界控制器。为保护基于 SIP 的互联网协议语音 (VoIP) 网络而部署的网络元素。

Sequence (序列)

调用 AWS Lambda 函数的事件序列。每次在呼叫期间调用函数时，序列都会递增。

服务限制/服务限额

Amazon Chime SDK 允许的最大资源数量，例如会议、音频流或内容共享。有关更多信息，请参阅本指南中的 [Audio](#)。

SIP

会话启动协议，一种用于启动、维护和终止实时会话的信令协议，包括语音、视频和消息收发应用程序的任意组合。有关更多信息，请参阅 [SIP：会话启动协议](#)。

SIP 标头

AWS Lambda 函数中的参数，包含呼叫控制数据以及其他数据，例如用户账户 ID。

SIP 媒体应用程序

将值从 SIP 规则传递到目标 AWS Lambda 函数的托管型对象。开发人员可以调用 [CreateSipMediaApplication](#) API 来创建 SIP 媒体应用程序，但他们必须具有管理权限才可以这样做。

SIP 规则

一种托管型对象，将 Amazon Chime SDK 语音连接器 URI 的电话号码传递到目标 SIP 媒体应用程序。

SIP 中继

请参阅 [Amazon Chime SDK Voice Connector](#)。

SMA

请参阅 SIP 媒体应用程序。

SMA ID

请参阅 SIP 媒体应用程序。

T

电信

电信服务提供商。

终止

结束呼叫的过程。

事务

包含一个或多个呼叫分支的呼叫。有关更多信息，请参阅本指南中的 [关于使用 PSTN 音频服务呼叫分支](#)。

事务 ID

包含多个呼叫分支的事务的 ID。有关更多信息，请参阅本指南中的 [关于使用 PSTN 音频服务呼叫分支](#)。

V

Amazon Chime SDK 语音连接器

为电话系统提供会话启动协议 (SIP) 中继服务的对象。管理员使用 Amazon Chime SDK 管理控制台来创建和管理语音连接器。有关更多信息，请参阅 Amazon Chime SDK 管理员指南中的 [管理 Amazon Chime SDK 语音连接器](#)。

Amazon Chime SDK 语音连接器组

包含来自不同 AWS 区域的多个语音连接器的包装器。组允许入站呼叫跨区域故障转移，从而创建容错机制。有关更多信息，请参阅 Amazon Chime SDK 管理员指南中的[管理 Amazon Chime SDK 语音连接器组](#)。

使用 Amazon Chime SDK 呼叫分析

本节中的主题介绍如何使用 Amazon Chime SDK 呼叫分析从呼叫数据中生成见解数据。

Amazon Chime SDK 呼叫分析为开发者提供了低代码解决方案，用于从实时音频中生成具有成本效益的见解，包括音频摄取、分析、警报和数据湖集成。呼叫分析使您能够通过 Amazon Transcribe 和 Transcribe 呼叫分析 (TCA) 集成以及在本地通过 Amazon Chime SDK 语音分析生成见解。呼叫分析还可以记录对您的 Amazon S3 存储桶的呼叫。

您可以使用以下方法配置和运行呼叫分析。

- 使用 Amazon Chime SDK 控制台创建呼叫分析配置并将其与 Amazon Chime SDK Voice Connector 相关联。在此过程中，您可以启用通话录音和分析。您无需编写代码即可完成此过程。
- 使用一组 Amazon Chime SDK API [Amazon Chime SDK](#) API 以编程方式创建和运行配置。

有关更多信息，请参阅本节后文中的 [创建呼叫分析配置](#) 和 [使用呼叫分析配置](#)。

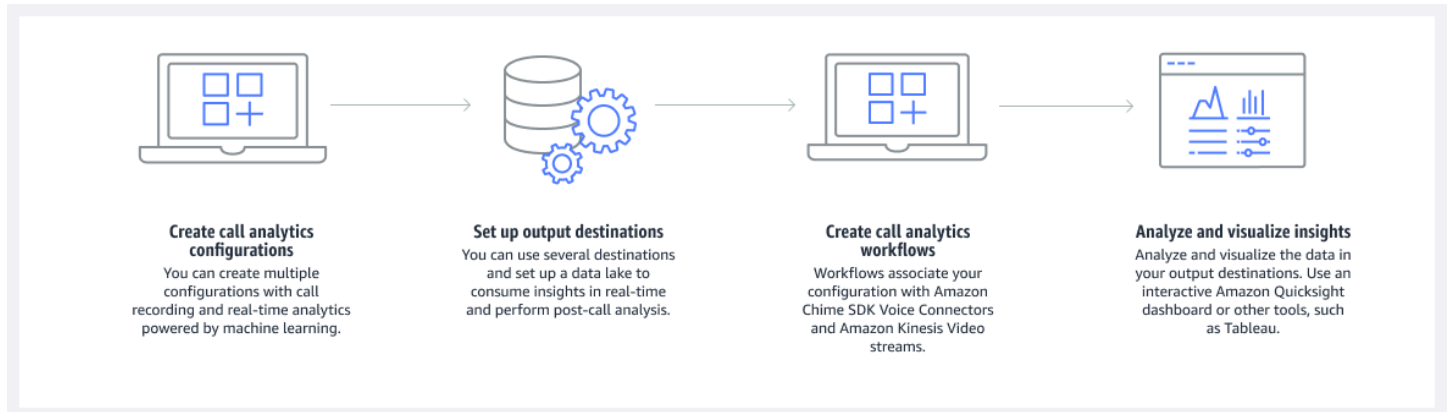
主题

- [什么是 Amazon Chime SDK 呼叫分析](#)
- [呼叫分析术语](#)
- [创建呼叫分析配置](#)
- [使用呼叫分析配置](#)
- [管理呼叫分析管道](#)
- [暂停和恢复呼叫分析管道](#)
- [使用通话分析资源访问角色](#)
- [了解通话分析状态](#)
- [使用 Amazon 监控呼叫分析渠道 CloudWatch](#)
- [呼叫分析处理器和输出目的地](#)
- [呼叫分析数据模型](#)
- [使用 Amazon Chime SDK 语音分析](#)
- [呼叫分析服务限额](#)

什么是 Amazon Chime SDK 呼叫分析

Amazon Chime SDK 呼叫分析是一种低代码解决方案，用于从实时音频中生成具有成本效益的见解，包括音频摄取、录音、语音分析、警报和数据湖等功能。您可以创建可重复使用的呼叫分析配置，确定要为工作流程启用哪些 AWS 机器学习集成和音频处理功能，从而使用呼叫分析生成基于机器学习的见解。然后，您可以将呼叫分析配置用于各种媒体来源，例如 Voice Connector 或 Amazon Kinesis Video Streams。呼叫分析通过与 Amazon Transcribe 和 Transcribe 呼叫分析 (TCA) 的集成以及在本地通过 [Amazon Chime SDK 语音分析](#)（一项在呼叫分析下运行的服务）生成见解。

要使用呼叫分析，请按照以下步骤操作：



在下图中：

1. 您首先要创建呼叫分析配置。
2. 您可以设置输出目的地和可选的数据湖。
3. 您可以创建将您的配置与 Voice Connector 和 Amazon Kinesis Video Streams 关联的工作流程。
4. 您可以分析并可视化您的见解。

您可以使用 Amazon Chime SDK 控制台创建呼叫分析配置，并启用自动启动呼叫分析。如果您需要控制适用于给定类型呼叫的配置，则可以使用 API 来创建配置。无论哪种方式，配置都包含以下相关详细信息：为呼叫音频调用的 AWS 机器学习服务、启用通话录音以及见解、元数据和录音的目的地。呼叫分析提供以下目的地：

- Amazon Kinesis Data Stream (KDS)。您可以使用 KDS 接收实时通话见解，然后将其集成到您的应用程序中。例如，您可以整合实时见解以在客户电话中为销售或客户支持代理提供帮助，或者使用这些见解来增强生成式 AI 提示和摘要。
- 配置作为数据仓库的 Amazon S3 存储桶。存储桶以 Parquet 格式存储数据。Parquet 是一种开源文件格式，旨在压缩和存储大量数据。然后，您可以使用 Amazon Athena 查询使用简单查询语言

(SQL) 的数据，或者将数据移至现有的数据仓库以与您的业务数据配对。例如，您可以进行通话后汇总分析，以了解客户来电的有效性、产品的问题领域或培训员工以实现更好客户成果的机会。

除了这些目的地，呼叫分析还支持实时警报，您可以根据见解预先配置这些警报。提醒将发送到亚马逊 EventBridge。

Note

创建呼叫分析配置时，无需选择特定的音频源。这使您可以跨多个音频源重复使用配置。例如，配置可以启用通话录音并提供呼叫转录。然后，您可以将该配置与 Chime SDK Voice Connector 一起使用，也可以通过 Kinesis 视频流使用音频流。您还可以在多个 Voice Connector 之间共享配置。每个呼叫分析配置都是唯一的，并由 ARN 标识。

呼叫分析术语

以下是便于您了解如何使用 Amazon Chime SDK 呼叫分析的核心术语和概念。

Amazon Athena

一项交互式查询服务，您可以使用该服务通过标准 SQL 在 Amazon S3 中分析数据。Athena 无服务器，因此您无需管理基础设施，只需为运行的查询付费。要使用 Athena，请指向 Amazon S3 中的数据，定义架构，然后使用标准 SQL 查询。您还可以使用工作组对用户进行分组，并控制他们在运行查询时可以访问的资源。工作组使您能够管理查询并行性，并在不同的用户组和工作负载之间确定查询执行的优先级。有关更多信息，请参阅[什么是 Amazon Athena](#)。

Amazon Kinesis Data Firehose

提取、转换、加载 (ETL) 服务，可靠地捕获、转换、传送流数据到数据湖、数据存储和分析服务。有关更多信息，请参阅[什么是 Amazon Kinesis Data Firehose](#)。

呼叫分析数据仓库

可选存储呼叫分析数据。仓库将数据以基于 Parquet 的格式存储在 Amazon S3 存储桶中。您可以使用标准 SQL 来查询数据。您可以在呼叫分析配置中启用仓库。

Glue 数据目录

用于存储各种数据来源的数据资产的集中式元数据存储库。该目录由数据库和表组成。对于呼叫分析，表中的元数据会将 Amazon S3 存储桶的位置告知 Athena。元数据还指定数据结构，例如列

名、数据类型和表名。数据库仅保存数据集的元数据和架构信息。有关更多信息，请参阅本节后文中的 [Glue 数据目录表结构](#)。

媒体见解管道

由唯一 `MediaPipelineId` 标识的临时资源。通过使用呼叫分析管道配置和运行时系统参数创建。运行时系统参数指定管道的数据来源。

媒体见解管道配置

用于创建媒体见解管道的静态配置。您可以使用配置来实例化一个或多个管道。

媒体见解管道配置元素

媒体见解管道配置元素包括使用处理器元素处理媒体或使用接收器元素提供发送生成见解的指令。

媒体见解管道任务

媒体见解管道的临时子资源。任务包含有关特定流 ARN 和频道 ID 的进程状态的元数据。由唯一 ID 识别。通过在媒体见解管道上启动语音分析而创建。

发言人搜索

一种语音分析功能，可帮助您识别呼叫参与者。

语音分析

一项 Amazon Chime SDK 功能，包括发言人搜索和语音分析。

语音嵌入

呼叫者声音的矢量表示形式，外加一个唯一的 ID。

语音增强

一种可提高电话通话音频质量的系统。

语音配置文件

语音嵌入、其 ID 和到期日期的组合。

语音配置文件域

语音配置文件集合。

语音音调分析

一种语音分析功能，可让您分析呼叫者的声音，以了解 positive、negative、或 neutral 情绪。

有关用于创建 Call Insights 配置、启动管道和运行语音分析的 API 的更多信息，请参阅《Amazon Chime SDK API 参考》中的 [Amazon Chime SDK 媒体管道](#)。

Note

我们强烈建议使用媒体见解管道 API 来运行呼叫分析，因为只有这些 API 才能提供新功能。有关媒体管道和语音命名空间之间区别的更多信息，请参阅本节后文中的 [使用语音 API 来运行语音分析](#)。

创建呼叫分析配置

要使用呼叫分析，首先要创建一个配置，这是一个静态结构，用于保存创建呼叫分析管道所需的信息。您可以使用 Amazon Chime 软件开发工具包控制台来创建配置，也可以调用 API。

[CreateMediaInsightsPipelineConfiguration](#)

呼叫分析配置包括有关音频处理器的详细信息，例如录音、语音分析或 Amazon Transcribe。它还包括见解目的地和警报事件配置。或者，您可以将呼叫数据保存到 Amazon S3 存储桶，以便进一步分析。

但是，配置不包括特定的音频源。这允许您在多个呼叫分析工作流程中重复使用配置。例如，您可以对不同的 Voice Connector 使用相同的呼叫分析配置，也可以跨不同的 Amazon Kinesis Video Streams (KVS) 来源使用相同的呼叫分析配置。

当通过 Voice Connector 进行 SIP 呼叫时，或者当新媒体被发送到 Amazon Kinesis Video Streams (KVS) 时，您可以使用这些配置来创建管道。反过来，管道根据配置中的规格处理媒体。

您可以随时以编程方式停止管道。当 Voice Connector 呼叫结束时，管道也会停止处理媒体。此外，您还可以暂停管道。这样做会禁用对底层 Amazon 机器学习服务的调用，并在需要时恢复调用。但是，当您暂停管道时，会运行通话录音。

以下各节说明了创建呼叫分析配置的先决条件以及如何创建呼叫分析配置。

主题

- [先决条件](#)
- [使用 Amazon Chime SDK 控制台创建呼叫分析配置](#)
- [使用 API 创建呼叫分析配置。](#)
- [将配置与 Voice Connector 关联](#)

先决条件

在创建呼叫分析配置之前，您必须具有以下项目。您可以使用 AWS 控制台创建这些项目：

- Amazon Chime SDK Voice Connector。如果不是，请参阅[创建 Amazon Chime SDK Voice Connector](#)。您还必须：
 - 为 Voice Connector 启用流媒体。有关更多信息，请参阅[《亚马逊 Chime 软件开发工具包管理员指南》中的“EventBridge使用自动化 Amazon Chime 软件开发工具包”](#)
 - 将 Voice Connector 配置为使用呼叫分析。有关更多信息，请参阅《Amazon Chime SDK 管理员指南》中的[配置 Voice Connector 以使用呼叫分析](#)。
- 亚马逊的 EventBridge 目标。如果没有，请参阅[《Amazon Chime 软件开发工具包管理员指南》中的“使用 EventBridge监控亚马逊 Chime 软件开发工具包”](#)。
- 一种与服务相关的角色，允许语音连接器访问对 EventBridge 目标的操作。有关更多信息，请参阅《Amazon Chime SDK 管理员指南》中的[使用 Amazon Chime SDK Voice Connector 服务相关角色策略](#)。
- Amazon Kinesis Data Streams。如果不是，请参阅《Amazon Kinesis Streams 开发者指南》中的[创建和管理流](#)。语音分析和转录需要 Kinesis Data Streams。
- 要离线分析通话，您必须创建 Amazon Chime SDK 数据湖。要执行此操作，请参阅本指南后面的[创建 Amazon Chime SDK 数据湖](#)。

使用 Amazon Chime SDK 控制台创建呼叫分析配置

在创建了上一节中列出的先决条件后，您可以使用 Amazon Chime SDK 控制台创建一个或多个呼叫分析配置。您还可以使用该控制台将一个或多个 Voice Connector 与配置关联。完成该过程后，呼叫分析将使用您在创建配置时启用的功能开始运行。

您可以按照以下步骤创建呼叫分析配置：

1. 指定配置详细信息，包括名称和可选标签。
2. 配置您的录制设置。创建呼叫分析配置，包括录音和基于机器学习的见解。
3. 配置您的分析服务。
4. 选择用于获取实时见解的输出目的地。创建可选的数据湖以执行通话后分析。
5. 创建新的服务角色或使用现有角色。
6. 设置实时提醒，在满足特定条件 EventBridge 时通过 Amazon 发送通知。

7. 检查您的设置，然后创建配置。

创建配置后，您可以通过将 Voice Connector 与配置关联来启用呼叫分析。完成此操作后，当呼叫进入该 Voice Connector 时，呼叫分析会自动启动。有关更多信息，请参阅本节后文中的 [将配置与 Voice Connector 关联](#)。

以下章节介绍如何完成流程的每个步骤。按列出的顺序展开各章节。

指定配置详细信息

若要指定配置详细信息

1. 通过以下网址打开 Amazon Chime 控制台：<https://console.aws.amazon.com/chime-sdk/home>。
2. 在导航窗格中的呼叫分析下，选择配置，然后选择创建配置。
3. 在 Basic information (基本信息) 中，执行以下操作：
 - a. 为配置输入名称。名称应反映您的用例和所有标签。
 - b. (可选) 在标签下，选择添加新标签，然后输入标签键和可选值。您可以定义键和值。标签可帮助您查询配置。
 - c. 请选择 Next (下一步) 。

配置录制

若要配置录制

- 在配置录制页面上，执行以下操作：
 - a. 选中激活通话录音复选框。这样可以为 Voice Connector 呼叫或 KVS 流启用录制并将数据发送到您的 Amazon S3 存储桶。
 - b. 在文件格式下，选择采用 PCM 编码的 WAV 以获得最佳音频质量。

—或者—

选择采用 OPUS 编码的 OGG 来压缩音频并优化存储空间。

- c. (可选) 根据需要，选择 Create an Amazon S3 bucket 链接，然后按照以下步骤创建 Amazon S3 存储桶。
- d. 输入您的 Amazon S3 存储桶的 URI，或者选择浏览以找到存储桶。
- e. (可选) 选择激活语音增强以帮助提高录音的音频质量。

- f. 请选择 Next (下一步) 。

了解语音增强

创建呼叫分析配置时，您可以启用通话录音并将录制的呼叫存储在 Amazon S3 存储桶中。作为其中的一部分，您还可以启用语音增强功能并提高存储通话的音频质量。语音增强功能仅适用于启用该功能后生成的录音。启用语音增强功能后，除了原始录音外，还会创建增强录音，并存储在相同的 Amazon S3 存储桶和格式中。语音增强功能将为长达 30 分钟的通话生成增强录音。长度超过 30 分钟的通话不会生成增强录音。

电话经过窄带过滤，采样频率为 8kHz。语音增强将采样率从 8kHz 提高到 16kHz，并使用机器学习模型将频率内容从窄带扩展到宽带，使语音听起来更自然。语音增强还使用一种名为 Amazon Voice Focus 的降噪模型来帮助减少增强音频中的背景噪音。

语音增强还使用名为 Voice Focus 的降噪模型。该模型有助于减少增强音频中的背景噪音。语音增强功能将模型应用于升级后的 16kHz 音频。

Note

仅在美国东部（弗吉尼亚州北部）地区和美国西部（俄勒冈州）地区支持语音增强功能。

语音增强录音元数据通过您配置的 KDS 发布到现有的 AWS Glue 数据目录表 `call_analytics_recording_metadata` 中。为了识别语音增强型通话录音中的原始通话录音记录，在 KDS 通知和粘合表 `call_analytics_recording_metadata` 中添加了一个名为 `deta il-sub VoiceEnhancementtype` 的新字段，该字段具有值。有关数据仓库架构的更多信息，请参阅 [呼叫分析数据模型](#)。

语音增强文件格式

请注意以下有关增强录音文件的注意事项。

- 增强录音将写入与普通录音相同的 Amazon S3 存储桶。您可以通过调用 [S 3](#) 或 `S3 RecordingSinkRuntimeConfiguration API RecordingSinkConfiguration` 或使用 [Amazon Chime](#) 软件开发工具包控制台来配置目标。
- 增强录音在基本文件名后面附加了 `_enhanced`。
- 增强录音保持与原始录音相同的文件格式。您可以通过调用 [S 3](#) 或 `S3 RecordingSinkRuntimeConfiguration API RecordingSinkConfiguration` 或使用 [Amazon Chime](#) 软件开发工具包控制台来配置文件格式。

以下示例介绍了一个典型的文件名格式。

```
s3://original_file_name_enhanced.wav
```

或者

```
s3://original_file_name_enhanced.ogg
```

配置分析服务

Amazon Transcribe 提供通话文本转录。然后，您可以使用转录来增强其他机器学习服务，例如 Amazon Comprehend 或您自己的机器学习模型。

Note

Amazon Transcribe 还提供自动语言识别功能。但是，您不能在自定义语言模型或内容编辑中使用该功能。此外，如果您将语言识别与其他功能一起使用，则只能使用这些功能支持的语言。有关更多信息，请参阅《Amazon Transcribe 开发者指南》中的[附带流媒体转录的语言识别](#)。

Amazon Transcribe 通话分析功能是一款基于机器学习的 API，可提供通话记录、情绪和实时对话见解。该服务无需记笔记，并且可以对检测到的问题立即采取行动。该服务还提供通话后分析，例如呼叫者情绪、呼叫驱动程序、非通话时间、中断、通话速度和对话特征。

Note

默认情况下，通话后分析会将通话录音流式传输到您的 Amazon S3 存储桶。为避免创建重复的录音，请勿同时启用通话录音和通话后分析。

最后，Transcribe 通话分析可以根据特定的短语自动标记对话，并帮助编辑音频和文本中的敏感信息。有关呼叫分析媒体处理器、其生成的见解以及输出目的地的更多信息，请参阅本节后文中的[呼叫分析处理器和输出目的地](#)。

若要配置分析服务

1. 在配置分析服务页面上，选中语音分析或转录服务旁边的复选框。您可以同时选择这两个项目。

选中语音分析复选框以启用发言人搜索和语调分析的任意组合。

选中转录服务复选框以启用 Amazon Transcribe 或 Transcribe 呼叫分析。

a. 若要启用发言人搜索

- 选中 Yes, I agree to the Consent Acknowledgement for Amazon Chime SDK voice analytics 复选框，然后选择接受。

b. 启用语调分析

- 选中语调分析复选框。

c. 启用 Amazon Transcribe

i. 选择 Amazon Transcribe 按钮。

ii. 在语言设置下，执行以下任一操作：

- A. 如果您的呼叫者只讲一种语言，请选择特定语言，然后打开语言列表并选择语言。
- B. 如果您的呼叫者讲多种语言，您可以自动识别这些语言。选择 Automatic language detection。
- C. 打开 Language options for automatic language identification 列表并选择至少两种语言。
- D. (可选) 打开首选语言列表并指定首选语言。当您在上一步中选择的语言具有匹配的置信度分数时，该服务会转录首选语言。
- E. (可选) 展开内容删除设置，选择一个或多个选项，然后选择显示的一个或多个其他选项。帮助文本解释了每个选项。
- F. (可选) 展开其他设置，选择一个或多个选项，然后选择显示的一个或多个其他选项。帮助文本解释了每个选项。

d. 若要启用 Amazon Transcribe 通话分析功能

i. 选择 Amazon Transcribe 通话分析功能按钮。

ii. 打开语言列表并选择一种语言。

iii. (可选) 展开内容删除设置，选择一个或多个选项，然后选择显示的一个或多个其他选项。帮助文本解释了每个选项。

iv. (可选) 展开其他设置，选择一个或多个选项，然后选择显示的一个或多个其他选项。帮助文本解释了每个选项。

v. (可选) 展开通话后分析设置，然后执行以下操作：

- A. 选中通话后分析复选框。

- B. 输入 Amazon S3 存储桶的 URI。
 - C. 选择内容密文类型。
2. 完成选择后，选择下一步。

配置输出详细信息

完成媒体处理步骤后，您可以为分析输出选择目的地。呼叫分析通过 Amazon Kinesis Data Streams 提供实时见解，也可以选择通过您选择的 Amazon S3 存储桶中的数据仓库提供实时见解。要创建数据仓库，请使用 CloudFormation 模板。此模板可帮助您创建基础设施，将呼叫元数据和见解传送到 Amazon S3 存储桶。有关创建数据仓库的更多信息，请参阅本节后文中的 [创建 Amazon Chime SDK 数据湖](#)。有关数据仓库架构的更多信息，请参阅本节后文中的 [呼叫分析数据模型](#)。

如果您在上一节中启用了语音分析，则还可以添加语音分析通知目的地，例如 AWS Lambda、Amazon Simple Queue Service 或 Amazon Simple Notification Service。以下步骤说明了操作方法。

若要配置输出详细信息

1. 打开 Kinesis Data Streams 列表并选择您的数据流。

Note

如果要对数据进行可视化，则必须选择 Amazon S3 存储桶和 Amazon Kinesis Data Firehose 使用的 Kinesis Data Streams。

2. (可选) 展开 Additional voice analytics notification destinations，然后选择 AWS Lambda、Amazon SNS 和 Amazon SQS 目的地的任意组合。
3. (可选) 在分析和可视化见解下，选中使用数据湖执行历史分析复选框。有关数据湖的更多信息，请参阅本节后文中的 [创建 Amazon Chime SDK 数据湖](#)。
4. 在完成后，选择下一步。

配置访问权限

要启用呼叫分析，机器学习服务和其他资源必须具有访问数据媒体和提供见解的权限。您可以使用现有的服务角色或使用控制台创建新的角色。有关角色的详细信息，请参阅本节后文中的 [使用通话分析资源访问角色](#)。

若要配置访问权限

1. 在配置访问权限页面上，执行以下任一操作：
 1. 选择创建和使用新的服务角色。
 2. 在服务角色名后缀框中，输入该角色的描述性后缀。

—或者—

1. 选择使用现有服务角色。
 2. 打开服务角色列表并选择一个角色。
2. 请选择 Next (下一步) 。

(可选) 配置实时警报

Important

要使用实时提醒，您必须先启用 Amazon Transcribe 或 Amazon Transcribe 分析。

您可以创建一组向 Amazon 发送实时提醒的规则 EventBridge。在分析会话期间，当 Amazon Transcribe 或 Amazon Transcribe 通话分析功能生成的见解与您的指定规则相匹配时，就会发送警报。警报具有详细类型 Media Insights Rules Matched。EventBridge 支持与 Amazon Lambda、Amazon SQS 和 Amazon SNS 等下游服务集成，以触发终端用户的通知或启动其他自定义业务逻辑。有关更多信息，请参阅本节后文中的 [使用亚马逊 EventBridge 通知](#)。

若要配置警报

1. 在实时警报下，选择 Active real-time alerts。
2. 在规则下，选择创建规则。
3. 在规则名称框中，输入规则的名称。
4. 打开规则类型列表并选择要使用的规则类型。
5. 使用显示的控件向规则添加关键字并应用逻辑，例如 mentioned 或 not mentioned。
6. 请选择 Next (下一步) 。

审核和创建

若要创建配置文件

1. 查看每个章节的设置。根据需要选择编辑以更改设置。
2. 选择创建配置。

您的配置会显示在 Amazon Chime SDK 控制台的配置页面上。

使用 API 创建呼叫分析配置。

您可以通过编程方式创建 Voice Connector 和呼叫分析配置，然后将其关联以启动呼叫分析工作流程。本指南假设您知道如何编写代码。

您使用的 API 因工作流程类型而异。例如，要录制音频，您需要先调用 [CreateMediaInsightsPipelineConfiguration](#) API 来创建呼叫分析配置。然后，您可以 [CreateVoiceConnector](#) 致电创建语音连接器。最后，您可以使用 [PutVoiceConnectorStreamingConfiguration](#) API 将配置与语音连接器关联起来。

相比之下，要使用 Kinesis 视频流制作器录制音频，则需要调用 API [CreateMediaInsightsPipelineConfiguration](#)，然后调用 API。 [CreateMediaInsightsPipeline](#)

有关使用呼叫分析配置来启用不同工作流程的更多信息，请参阅本节后文 [使用呼叫分析配置](#) 中的工作流程。

将配置与 Voice Connector 关联

使用控制台创建呼叫分析配置后，您可以通过将 Voice Connector 与其关联来使用该配置。然后，Voice Connector 会自动调用您的配置中指定的分析服务。Voice Connector 会为每个呼叫调用呼叫分析。

关联 Voice Connector

1. 通过以下网址打开 Amazon Chime 控制台：<https://console.aws.amazon.com/chime-sdk/home>。
2. 在导航窗格的 SIP 中继下，选择 Voice Connector。
3. 选择要与配置关联的 Voice Connector 的名称，然后选择流式传输选项卡。
4. 如果尚未选中，选择开始以便开始向 Kinesis 视频流进行流式传输。
5. 在呼叫分析下，选择激活，然后在出现的菜单上，选择您的呼叫分析配置 ARN。
6. 选择保存。

Note

启用、禁用或修改与 Voice Connector 关联的配置后，请等待 5 分钟，让新设置通过服务传播并生效。

有关呼叫分析配置的更多信息，请参阅《Amazon Chime SDK 管理员指南》中的[管理呼叫分析](#)。

有关使用呼叫分析配置来启用不同工作流程的更多信息，请参阅本节后文中的[使用呼叫分析配置](#)。

使用呼叫分析配置

要使用呼叫分析配置处理音频，必须创建呼叫分析管道，也称为媒体见解管道。管道是在呼叫期间创建的，用于处理音频，并在呼叫结束时终止。呼叫分析管道需要呼叫分析配置的 ARN 以及有关音频源的信息。呼叫分析配置包括有关音频处理器、见解目的地和警报事件配置的详细信息，但不包括音频源。这允许您在不同的呼叫分析工作流程中重复使用配置，例如使用不同的 Voice Connector 或 KVS 源。呼叫分析管道调用配置中指定的机器学习服务并录制音频。呼叫结束后，您可以手动或自动停止管道。

您可以在各种用例中使用呼叫分析管道。以下工作流程介绍了使用呼叫分析配置和管道的潜在方法。

主题

- [通话录音工作流程](#)
- [基于机器学习的分析工作流程](#)

通话录音工作流程

本节中的主题列出并描述了录制通话和 Kinesis 视频流的工作流程。

录制 Voice Connector 通话

在以下情况下使用此工作流程：

- 您已经使用或计划使用 Voice Connector 将 SIP 媒体引入呼叫分析。

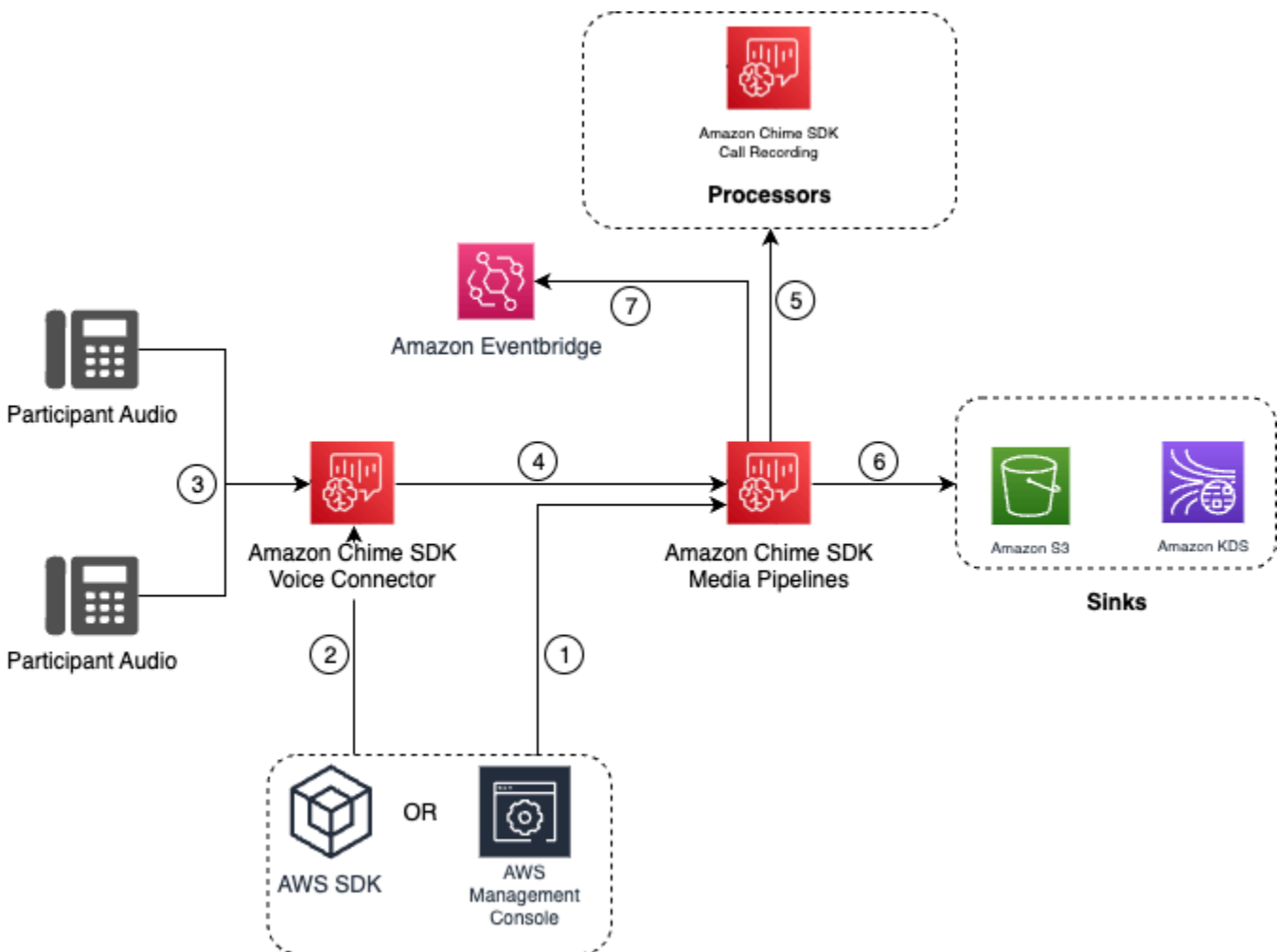
Note

Voice Connector 支持 SIP 和 SIPREC。有关更多信息，请参阅《Amazon Chime SDK 管理员指南》中的[管理 Amazon Chime SDK Voice Connector](#)。

- 您想以低延迟自动录制到您选择的 Amazon Simple Storage Service 目的地的 SIP 或 SIPREC 呼叫。
- 您想使用 Amazon Chime SDK 控制台来创建配置并将其与 Voice Connector 关联。
- 您想对每个 Voice Connector 呼叫应用相同的录音配置。如果要对一个或多个 Voice Connector 应用多种配置，请参阅下一节。

要启用以编程方式调用，请使用以下 Amazon Chime SDK API。使用 [CreateMediaInsightsPipelineConfiguration](#) API 创建呼叫分析配置，[CreateVoiceConnector](#) 创建语音连接器，然后使用 [PutVoiceConnectorStreamingConfiguration](#) API 将配置关联到语音连接器。有关更多信息，请参阅《Amazon Chime SDK 管理员指南》中的[配置 Voice Connector 以使用呼叫分析](#)。

下图显示了 Voice Connector 启动呼叫录音会话时的数据流。图表中的数字对应于下面的带编号的文本。



在图中：

1. 使用 Amazon Chime 软件开发工具包控制台或 [CreateMediaInsightsPipelineConfiguration](#) API 创建呼叫分析配置。在配置创建过程中，您只需激活通话录音，选择所需的录音文件格式，然后指定用于存储录音文件的 Amazon S3 目标即可。有关更多信息，请参阅《Amazon Chime SDK 管理员指南》中的 [创建呼叫分析配置](#)。
2. 您可以使用 Amazon Chime 软件开发工具包控制台或 [PutVoiceConnectorStreamingConfiguration](#) API 将配置与语音连接器关联起来。要使用控制台，请参阅 [配置 Voice Connector 以使用呼叫分析](#)。
3. 在拨出呼叫期间，Voice Connector 会接收每个呼叫参与者的音频。
4. 如果将呼叫分析录音配置附加到 Voice Connector，则 Voice Connector 服务将使用媒体管道服务启动呼叫分析录音会话。
5. 媒体管道服务启动监视正在进行的呼叫的呼叫录音处理器。
6. 通话结束后，媒体管道服务会将通话录音文件传送到指定的 Amazon S3 存储桶，并通过 Amazon Kinesis Data Streams 提供录音元数据。如果启用了数据仓库，则呼叫元数据也会进入 Amazon Simple Storage Service 数据仓库。如果使用 SIPREC 将 SIP 音频整合到呼叫分析中，则呼叫元数据会纳入表格格式的 SIPREC 元数据。有关录音表的更多信息，请参阅本节后文中的 [Glue 数据目录表](#)。
7. 媒体管道服务将管道状态事件发送到默认的 Amazon EventBridge。有关更多信息，请参阅本指南中的 [使用 EventBridge 通知](#)。

Note

请注意，您必须启用 Voice Connector 流媒体才能使用 Voice Connector 进行录音。此功能允许将通话数据流式传输到您账户中由 Voice Connector 管理的 Kinesis 视频流。有关更多信息，请参阅《Amazon Chime SDK 管理员指南》中的 [将 Amazon Chime SDK Voice Connector 媒体流式传输到 Kinesis 视频流](#)。

您还可以将 Voice Connector 创建的呼叫数据存储到 Kinesis 视频流中，持续时间各不相同，从几小时到几天甚至几年。选择不保留数据会限制呼叫数据的可用性，便于立即使用。Kinesis 视频流的成本是根据使用的带宽和总存储空间来确定的。您可以随时在 Voice Connector 流媒体配置中调整数据保留期。要启用呼叫分析录音，您必须确保 Kinesis 视频流将数据保留足够长的时间以进行呼叫分析。您可以通过指定合适的数据保留期来达到此目的。

您可以根据需要将呼叫见解管道配置与任意数量的 Voice Connector 相关链接。您也可以为每个 Voice Connector 创建不同的配置。语音连接器使用代

表您 `AWSServiceRoleForAmazonChimeVoiceConnector` 为每个交易 ID 调用 [CreateMediaInsightsPipeline](#) API 一次。有关该角色的信息，请参阅《Amazon Chime SDK 管理员指南》中的 [为 Amazon Chime SDK Voice Connector 使用 Amazon Chime SDK 服务相关角色](#)。

与 Amazon Kinesis 视频流制作人一起录制

在以下情况下，您可以录制 Amazon Kinesis 视频流：

- 您需要对呼叫应用不同的配置，而不是为每个 Voice Connector 呼叫使用相同的配置。
- 您想要录制未由 Voice Connector 处理的 SIP 或非 SIP 音频。

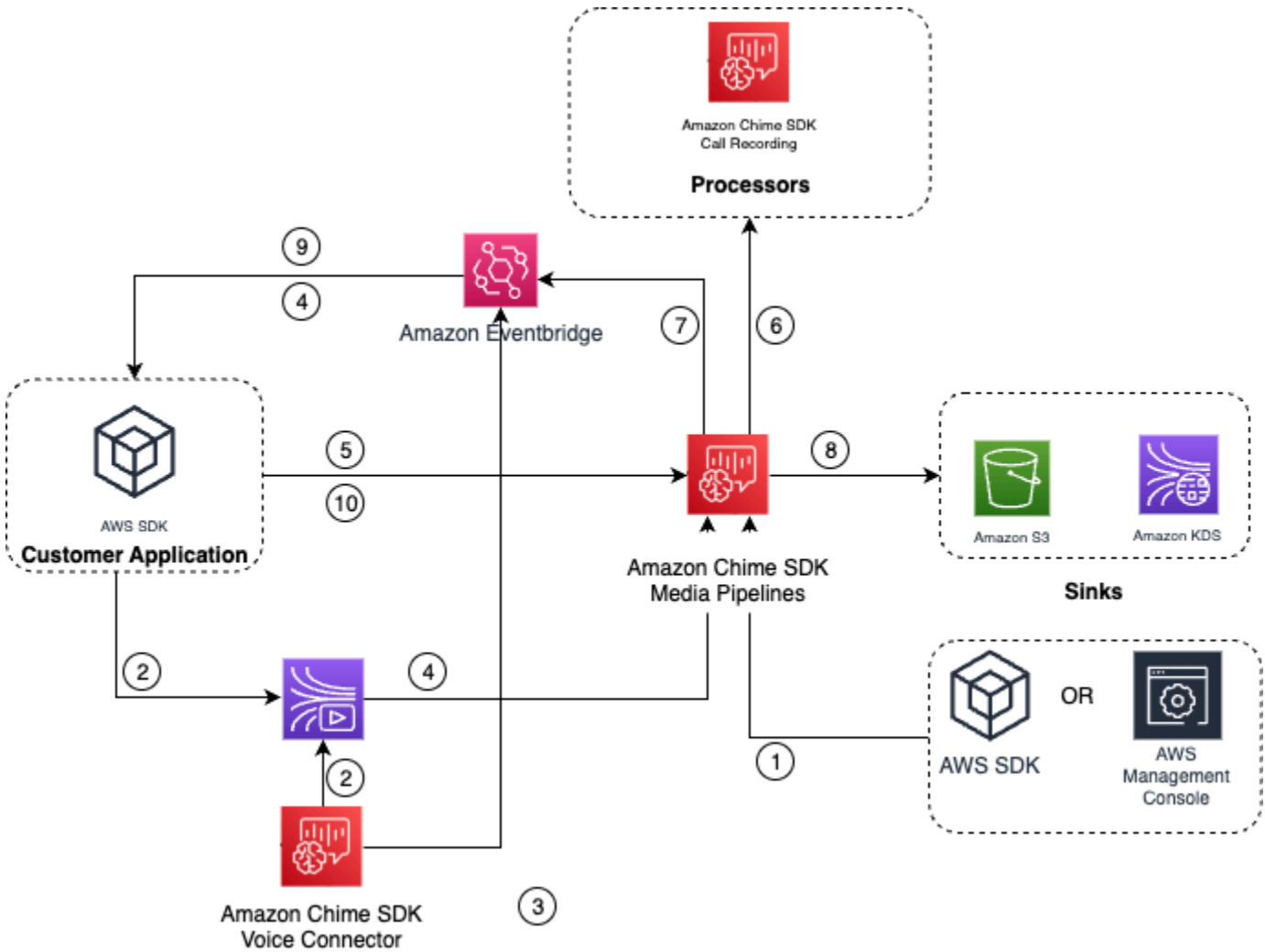
要使用此通话录音选项，您需要将音频发布到 Kinesis Video Streams (KVS)，然后使用 KVS 直播频道信息和呼叫 [CreateMediaInsightsPipeline](#) 分析配置 ARN 调用 API。

Note

呼叫分析 API 最多支持两个音频频道。您也可以启用 Voice Connector 流式传输，然后使用语音连接器 EventBridge 通知中发布的 KVS 信息启动通话录音。

在调用 [CreateMediaInsightsPipeline](#) API 时，您可以选择是否为每个 KVS 直播频道定义指定片段编号。如果您提供片段编号，则呼叫分析将开始处理该片段上的流。如果您未指定片段 ID，则呼叫分析将开始处理来自最新可用片段的流。

下图显示了 Voice Connector 启动呼叫录音会话时的数据流。图表中的数字对应于下面的带编号的文本。



在图中：

1. 您可以使用 Amazon Chime 软件开发工具包控制台或 [CreateMediaInsightsPipelineConfiguration](#) API 来创建通话录音配置。
2. 使用 AWS SDK 创建可将外部音频推送到 KVS 的应用程序，或者启用 Voice Connector 流式传输将呼叫音频自动发布到 KVS。有关更多信息，请参阅《Amazon Chime SDK 管理员指南》中的[将 Amazon Chime SDK Voice Connector 媒体流式传输到 Kinesis 视频流](#)。
3. 如果启用了语音连接器流式传输，则语音连接器服务会将通知发送到默认设置 EventBridge。
4. 如果是 Voice Connector 直播，您的应用程序可以使用来自的 Amazon Chime Voice Connector 流媒体 STARTED 事件 EventBridge 来收集有关通话时段的 KVS 直播信息。
5. 当您的应用程序获得来自 Voice Connector 流媒体事件或外部来源的音频信息后，您的应用程序就会调用 Amazon Ch [CreateMediaInsightsPipeline](#) SDK API。
6. 媒体管道服务启动监视正在进行的呼叫的呼叫录音处理器。

7. 媒体管道服务将管道状态事件发送到默认的 Amazon EventBridge。有关更多信息，请参阅[使用 EventBridge 通知](#)。
8. 通话完成后，媒体管道服务会将通话录音文件传送到指定的 Amazon S3 存储桶，并通过 Amazon Kinesis Data Streams 提供录音元数据。如果启用了数据仓库，则呼叫元数据也将发送到 Amazon S3 数据仓库。如果使用 SIPREC 将 SIP 音频整合到呼叫分析中，则呼叫元数据会纳入采用方便表格格式的 SIPREC 元数据。有关录音表的更多信息，请参阅本节后文中的[Glue 数据目录表](#)。
9. 您的应用程序可以使用发布到 Amazon 的事件监控管道，如果是语音连接器，还可以监控通话状态 EventBridge。有关更多信息，请参阅本指南中的[使用 EventBridge 通知](#)。
10. 要终止录音，请调用 [DeleteMediaPipeline](#) API 终止通话录音。

有关基于 API 的录制和示例，请参阅本指南中的[Amazon S3 录制接收器](#)。

使用 CLI 开始录制

本节中的示例向您说明如何执行以下操作：

- 使用 CLI 运行呼叫分析配置并调用 [CreateMediaInsightsPipeline](#)。
- 使用 CLI 指定录制目标、音频文件格式和音频文件名。

主题

- [运行配置并启动管道](#)
- [设置目的地、名称和格式](#)

运行配置并启动管道

使用以下命令运行配置并启动媒体见解管道。pipeline.json 文件包含配置设置。

```
aws chime-sdk-media-pipeline create-media-insights-pipeline --cli-input-json file://pipeline.json
```

以下示例介绍了典型的 pipeline.json 文件。

```
{
  "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` 是您在创建呼叫分析配置后收到的配置 ARN。

设置目的地、名称和格式

以下示例使用名为 `MyRecordingBucket` 的文件夹用作 `S3SinkConfiguration.Destination` 值，将 `Opus` 用作 `RecordingFileFormat` 值。

```
arn:aws:s3:::MyRecordingBucket/voice-connector-id/transaction-id_year-month-date-hour-minute-second-millisecond.ogg
```

以下示例将 `MyRecordingBucket` 用作 `S3SinkConfiguration.Destination` 值，将 `Wav` 用作 `RecordingFileFormat` 值。

```
arn:aws:s3:::MyRecordingBucket/voice-connector-id/transaction-id_year-month-date-hour-minute-second-millisecond.wav
```

基于机器学习的分析工作流程

下面几节介绍如何从 Amazon Chime SDK 呼叫分析工具包提供的机器学习分析功能。

Note

如果您计划在同一 Kinesis 视频流上运行多个机器学习分析，则可能需要提高视频流 `GetMedia` 和 `GetMediaForFragmentList` 的连接级别限制。有关更多信息，请参阅《Kinesis 视频流开发者指南》中的 [Kinesis 视频流限制](#)。

使用 Voice Connector 自动启动呼叫分析

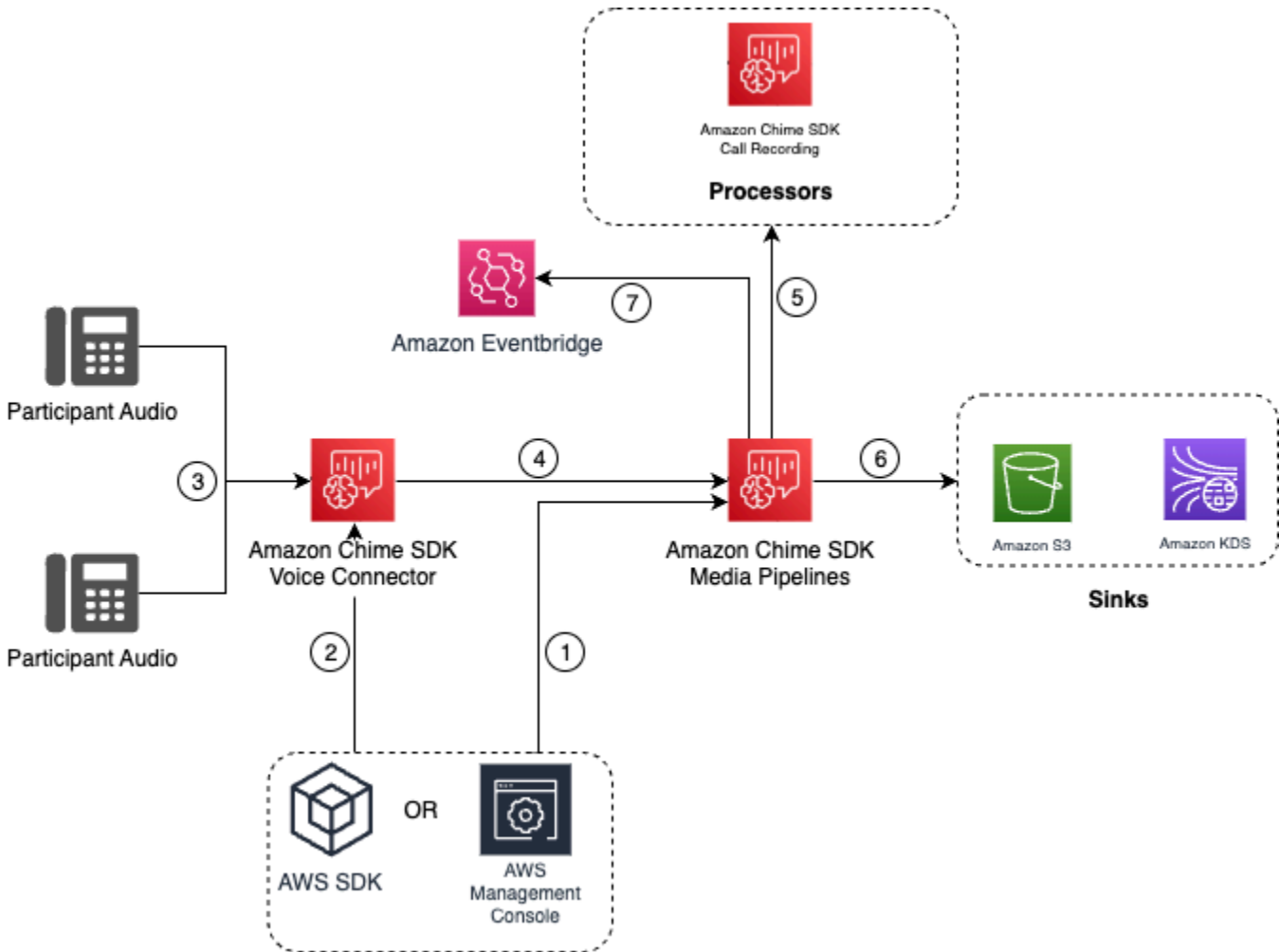
在以下情况下使用此工作流程：

- 您需要控制台驱动的设置。
- 您已经使用或计划使用 Voice Connector 将 SIP 媒体引入呼叫分析。Voice Connector 支持 SIP 和 SIPREC。有关配置 Voice Connector 的更多信息，请参阅[管理 Amazon Chime SDK Voice Connector](#)。
- 您希望将相同的媒体见解配置应用于每个 Voice Connector 呼叫。
- 您需要使用 Amazon Chime SDK 语音分析，这需要 Voice Connector 或媒体见解管道。

要在 Amazon Chime SDK 控制台中启用此工作流程，请按照[配置 Voice Connector 以使用呼叫分析](#)中的步骤创建录音配置。

要以编程方式启用此工作流程，请使用以下 [CreateMediaInsightsPipelineConfiguration](#) API：API 创建呼叫分析配置，然后使用 [PutVoiceConnectorStreamingConfiguration](#) API 将配置关联到语音连接器。有关更多信息，请参阅《Amazon Chime SDK 管理员指南》中的[配置 Voice Connector 以使用语音分析](#)。

下图显示了 Voice Connector 启动呼叫分析会话时的数据流。图表中的数字对应于下面的带编号的文本。



在图中：

1. 您可以使用 Amazon Chime 软件开发工具包控制台或 [CreateMediaInsightsPipelineConfiguration](#) API 来创建媒体见解管道配置。
2. 您可以使用 Amazon Chime 软件开发工具包控制台或 [PutVoiceConnectorStreamingConfiguration](#) API 将配置与语音连接器关联起来。要将现有配置与 Voice Connector 相关联，请参阅《Amazon Chime SDK 管理员指南》中的 [配置 Voice Connector 以使用呼叫分析](#)。
3. 在拨出呼叫期间，Voice Connector 会接收每个呼叫参与者的音频。
4. 由于与呼叫分析的内置集成，如果将呼叫分析配置附加到 Voice Connector，则 Voice Connector 服务将使用媒体管道服务启动呼叫分析会话。
5. 媒体管道服务调用配置中指定的一个或多个媒体处理器。

6. 媒体管道服务根据配置将输出数据发送到一个或多个目的地。例如，您可以通过 Amazon Kinesis Data Streams 发送实时分析，如果已配置，则可以将呼叫元数据和分析数据发送到 Amazon S3 数据仓库。
7. 媒体管道服务将管道状态事件发送到默认的 Amazon EventBridge。如果您已配置规则，则这些规则的通知也将发送到 EventBridge Amazon。有关更多信息，请参阅[使用 EventBridge 通知](#)。

Note

- 语音分析处理器只有在您调用 [StartSpeakerSearchTask](#) 或 [StartVoiceToneAnalysisTask](#) API 时才会自动启动。
- 您必须启用 Voice Connector 流媒体才能将呼叫分析与 Voice Connector 结合使用。此功能允许将通话数据流式传输到您账户中由 Voice Connector 管理的 Kinesis 视频流。有关更多信息，请参阅《Amazon Chime SDK 管理员指南》中的[将 Amazon Chime SDK Voice Connector 媒体流式传输到 Kinesis 视频流](#)。

您可以将 Voice Connector 通话数据存储到 Kinesis 视频流中，存储时间各不相同，从数小时到数年不等。选择不保留数据会限制呼叫数据的可用性，便于立即使用。Kinesis 视频流的成本是根据使用的带宽和总存储空间来确定的。通过编辑 Voice Connector 的流媒体配置，可以随时调整数据保留期。要启用呼叫分析录音，您必须确保 Kinesis 视频流在呼叫分析完成之前保留数据。您可以通过指定合适的保留期来达到此目的。

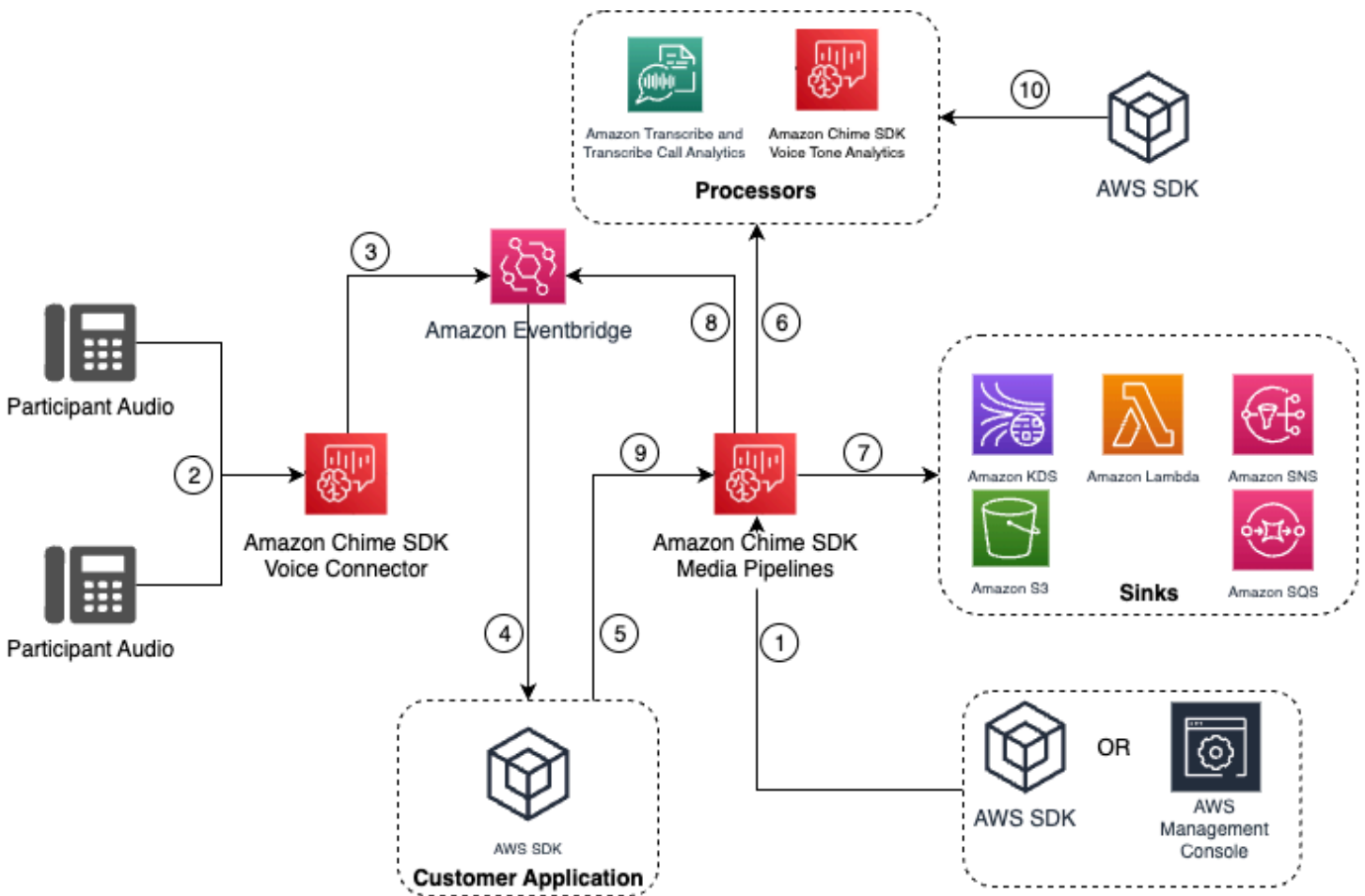
您可以根据需要将媒体见解管道配置与任意数量的 Voice Connector 相关联。您也可以为每个 Voice Connector 创建不同的配置。语音连接器使用代表您 `AWSServiceRoleForAmazonChimeVoiceConnector` 为每个交易 ID 调用 [CreateMediaInsightsPipeline](#) API 一次。有关该角色的信息，请参阅《Amazon Chime SDK 管理员指南》中的[为 Amazon Chime SDK Voice Connector 使用 Amazon Chime SDK 服务相关角色](#)。

将呼叫分析 API 与 Voice Connector 配合使用

如果您使用 Voice Connector，但需要控制何时应用呼叫分析配置以及将配置应用于哪个呼叫，请使用此工作流程。

要使用此方法，您需要为 Voice Connector 发布的事件创建 EventBridge 目标，然后使用这些事件触发呼叫分析管道 API。有关更多信息，请参阅[《亚马逊 Chime 软件开发工具包管理员指南》中的“EventBridge 使用自动化 Amazon Chime 软件开发工具包”](#)。

下图显示了在 Voice Connector 中使用呼叫分析时如何实现更精细的控制。图表中的数字对应于下方文本中的数字。



在下图中：

1. 您可以使用 Amazon Chime 软件开发工具包控制台或 [CreateMediaInsightsPipelineConfiguration](#) API 来创建媒体见解管道配置。
2. 在拨出呼叫期间，Voice Connector 将接收参与者的音频。
3. 语音连接器将通话音频发送到 Kinesis Video Stream，并将相应的事件发送到。EventBridge 这些事件具有流和通话元数据。
4. 您的应用程序是 EventBridge 通过 Target EventBridge t 订阅的。
5. 您的应用程序调用了 Amazon Ch [CreateMediaInsightsPipeline](#) 软件开发工具包 API。
6. 媒体管道服务根据媒体见解管道配置中的处理器元素调用一个或多个媒体处理器。
7. 媒体管道服务根据配置将输出数据发送到一个或多个目的地。Amazon Chime SDK 呼叫分析将通过 Amazon Kinesis Data Streams 提供实时分析，如果已配置，则可对 Amazon S3 数据仓库进行调用元数据分析。

8. 媒体管道服务将事件发送到 Amazon EventBridge。如果您已配置规则，则这些规则的通知也将发送到 EventBridge Amazon。
9. 您可以通过调用 [UpdateMediaInsightsPipelineStatus](#) API 来暂停或恢复呼叫分析会话。

Note

通话录音不支持暂停和恢复通话。此外，当您暂停会话时，为呼叫启动的语音分析任务也会停止。要重新启动它们，必须调用 [StartSpeakerSearchTask](#) 或 [StartVoiceToneAnalysisTask](#) API。

10. 如果您在配置期间选择语音分析，则可以通过调用 [StartSpeakerSearchTask](#) 或 [StartVoiceToneAnalysisTask](#) API 开始语音分析。

与 Kinesis 视频流制作人一起使用呼叫分析

要使用此选项，您需要将音频数据发布到 Kinesis Video Streams (KVS)，然后使用 KVS 直播 [CreateMediaInsightsPipeline](#) 频道信息调用 API。

Note

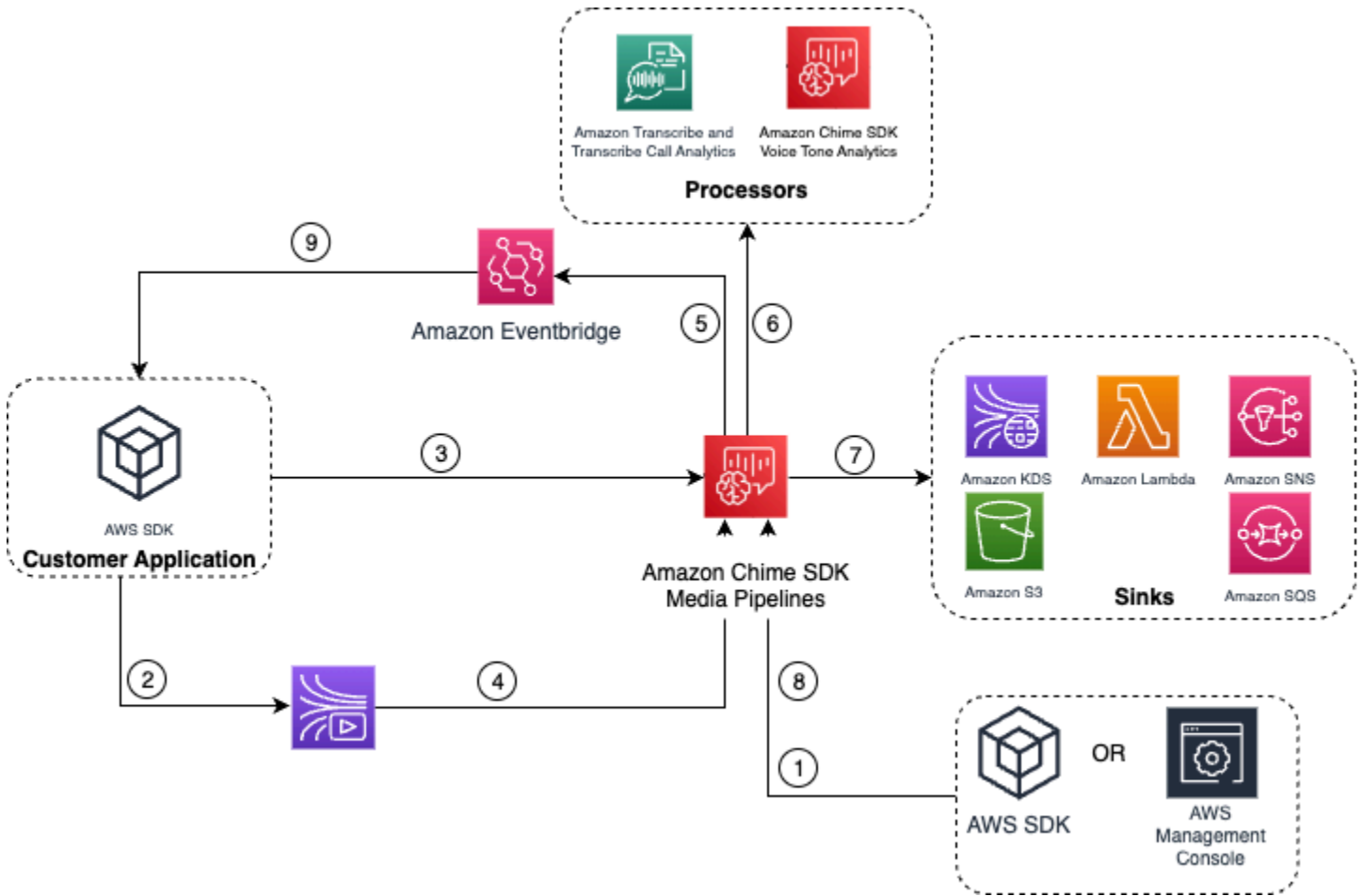
呼叫分析 API 最多支持两个音频频道。

在调用 [CreateMediaInsightsPipeline](#) API 时，您可以为每个 KVS 直播频道定义指定片段编号。如果您提供片段编号，则呼叫分析将开始处理该片段上的流。否则，呼叫分析将开始处理来自最新可用片段的流。

呼叫分析支持 PCM 音频（仅限带签名的 16 位小端序音频格式，不包括 WAV），音频采样率介于 8kHz 和 48kHz 之间。低质量的音频（例如电话音频）通常在 8,000Hz 左右。高质量音频的范围通常在 16,000Hz 到 48,000Hz 之间。您指定的采样率必须与音频的采样率相匹配。有关更多信息，请参阅 [KinesisVideoStreamSourceRuntimeConfiguration](#) 《亚马逊 Chime 软件开发工具包 API 参考》。

Kinesis Video Streams 创建者开发工具包提供了一组库，您可以用这些库将音频数据流式传输到 Kinesis 视频流。有关更多信息，请参阅《Amazon Kinesis Video Streams 开发者指南》中的 [Kinesis 视频流创建者库](#)。

下图显示了与自定义 Kinesis Video Streams 创建者一起使用呼叫分析时的数据流。图表中的数字对应于下面的带编号的文本。



1. 您可以使用AWS控制台或 [CreateMediaInsightsPipelineConfiguration](#) API 创建媒体见解管道配置。
2. 您可以使用 Kinesis Video Streams 创建者向 Kinesis 视频流写入音频。
3. 您的应用程序调用 API。 [CreateMediaInsightsPipeline](#)
4. 媒体管道服务从客户的 Kinesis 视频流读取音频。
5. 媒体管道服务将事件发送到 Amazon EventBridge。如果您已配置规则，则这些规则的通知也将发送到 EventBridge Amazon。
6. 媒体管道服务调用一个或多个处理器元素。
7. 媒体管道服务将输出数据发送到一个或多个接收器元素。
8. 您可以通过调用 [UpdateMediaInsightsPipelineStatus](#) API 来暂停或恢复呼叫分析会话。

Note

通话录音不支持暂停和恢复。

9. 您的应用程序可以处理 Amazon EventBridge 事件以触发自定义业务工作流程。

10. 如果您在创建配置时选择语音分析，则您的应用程序可以通过调用 [StartSpeakerSearchTask](#) 或 [StartVoiceToneAnalysisTask](#) API 来启动语音分析。

管理呼叫分析管道

您可以通过调用、和 [DeleteMediaPipeline](#) API 来读取、列出和删除媒体见解管道。 [GetMediaPipelineListMediaPipelines](#)

在满足以下任意条件时，媒体见解管道停止：

- 任何 Kinesis 视频流在 15 秒钟内都不会向 InProgress 管道发送任何新片段。
- [DeleteMediaPipeline](#) API 被调用。
- 媒体见解管道是在 8 个多小时前创建的。系统会自动停止管道。
- 媒体见解管道暂停超过 2 小时。系统会自动停止管道。

暂停和恢复呼叫分析管道

要暂停和恢复媒体见解管道，请使用 Pause 或 Resume 操作调用 [UpdateMediaInsightsPipelineStatus](#) API。为此，您需要在 Identifier 字段中传递管道的 ID 或 ARN。

Warning

警告：提供 Pause 状态后，UpdateMediaInsightsPipelineStatus API 会停止在媒体见解管道上启动的所有语音分析任务。提供 Resume 状态后，任务不会恢复，必须重新启动。在重新开始任务之前，您必须提供所有必要的通知并获得发言人的所有必要同意。有关更多信息，请参阅《Amazon Chime 软件开发工具包 API 参考》中的 [StartSpeakerSearchTask](#) 或 [StartVoiceToneAnalysisTask](#)。

暂停后，管道停止向处理器发送媒体以及向 Kinesis Data Streams 和数据仓库写入数据。当您 Resume 管道时，服务会发送流上最新的可用数据块。暂停超过 2 小时后，媒体见解管道会自动停止。请注意，通话录音不支持暂停和恢复。

有关更多信息，请参阅以下主题：

- [使用 EventBridge 通知](#)。

- [StartSelectorType.NOW](#) 在《亚马逊 Kinesis Video Streams 开发者指南》中。
- [Amazon Transcribe 通话分析功能处理器](#)。

Note

在管道暂停期间，您需要为通话分析的使用量付费。但是，您无需为通过资源访问角色访问的 AWS 服务（例如 Amazon Transcribe 和 Amazon Kinesis）付费。

通过在标识符字段中传递配置名称或 ARN，您可以使用 [GetMediaInsightsPipelineConfiguration](#)、[UpdateMediaInsightsPipelineConfiguration](#)、和 [DeleteMediaInsightsPipelineConfiguration](#) API 读取、更新和删除现有呼叫分析配置。

您可以通过调用 [ListMediaInsightsPipelineConfigurations](#) API 列出配置。

使用通话分析资源访问角色

调用账户必须创建媒体见解管道配置使用的资源访问角色。您无法使用跨账户角色。

根据您在创建通话分析配置时启用的功能，您必须使用其他资源策略。展开以下部分以了解更多信息。

所需的最小策略

该角色至少需要以下策略：

```
{
  "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"
      }
    }
  }
]
}

```

您还必须使用以下信任策略：

```

{
  "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 政策

如果您使用 KinesisDataStreamSink，请添加以下策略：

```

{
  "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"
        }
      }
    }
  ]
}

```

S3 RecordingSink 策略

如果您使用 S3RecordingSink，请添加以下策略：

```
{
  "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"
      }
    }
  }
]
```

通话后分析策略

如果您使用 AmazonTranscribeCallAnalyticsProcessor 的通话后分析功能，请添加以下策略：

```
{
  "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 政策

如果您使用 VoiceEnhancementSinkConfiguration 元素，添加以下策略：

```
{
  "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 策略

如果您使用 VoiceAnalyticsProcessor，则根据您定义的接收器添加 LambdaFunctionSink、SqsQueueSink 和 SnsTopicSink 的策略。

LambdaFunctionSink 策略：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "lambda:InvokeFunction",
        "lambda:GetPolicy"
      ],
      "Resource": [
        "arn:aws:lambda:us-east-1:111122223333:function:function_name"
      ],
      "Effect": "Allow"
    }
  ]
}
```

SqsQueueSink 策略

```
{
  "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 策略 :

```
{
  "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"
    }
  }
}
```

了解通话分析状态

在执行以下一项或两项操作时，媒体见解管道会跟踪一组状态：

- 使用多种机器学习处理元素，例如 Amazon Transcribe 和语音分析。
- 启用带或不带机器学习处理功能的通话录音。

要获取管道和元素状态，请使用 [GetMediaPipelineAPI](#) 和 [EventBridge 通知](#)。

要获取语音分析任务的状态，请使用 [GetSpeakerSearchTask](#) 和 [GetVoiceToneAnalysisTaskAPI](#) 以及 [语音分析通知目标](#)。

媒体见解管道跟踪以下状态。

- 管道状态：呼叫分析管道（也称为媒体见解管道）的总体状态。这由元素状态决定。
- 元素状态：各个媒体见解管道配置元素的处理状态。
- 任务状态：为语音分析而启动的媒体见解管道任务的处理状态。VoiceAnalyticsProcessor 元素状态由任务状态决定。呼叫分析管道中没有其他元素具有任务状态。

有关媒体见解管道任务的更多信息，请参阅本指南前文中的 [呼叫分析术语](#)。

并非所有媒体见解配置元素类型都有元素状态。通常，只有“处理器”类型的媒体见解配置元素才具有元素状态。此外，Amazon S3 录音和语音增强接收器具有处理器状态。具体而言，以下媒体见解配置元素类型存在元素状态：

- AmazonTranscribeProcessor

- AmazonTranscribeCallAnalyticsProcessor
- S3RecordingSink
- VoiceAnalyticsProcessor
- VoiceEnhancementSink

管道状态由元素状态决定，如下所示：

管道状态	状况
NotStarted	所有元素状态均未启动。
Initializing	至少有一个元素正在初始化，其余元素尚未启动。
InProgress	至少有一个元素正在进行中。
失败	至少有一个元素失败，其余元素已停止。
Stopping	有关停止条件的完整列表，请参阅 管理呼叫分析管道 。
Stopped	所有元素都已停止。
Paused	所有元素都已暂停。

与其他元素状态不同，该 VoiceAnalyticsProcessor 元素有一些细微差别。如前所述，与 Amazon Chime SDK 语音分析功能相对应的 VoiceAnalyticsProcessor 元素状态由和创建的任务状态决定。 [StartSpeakerSearchTask StartVoiceToneAnalysisTask](#)

- VoiceAnalyticsProcessor 的元素状态从 NotStarted 状态开始，因为 StartSpeakerSearchTask 和 StartVoiceToneAnalysisTask 必须先调用，然后元素才能将状态更改为 Initializing，之后改为 InProgress。
- 只要一个任务已启动，并且在任务运行时不满足 [停止条件](#)，VoiceAnalyticsProcessor 就会一直处于 InProgress 状态。
- 尽管 VoiceAnalyticsProcessor 可能是 InProgress，但您只需要为任务处理的持续时间付费。

- 若要清理已启动至少一项语音分析任务且不再运行任务的媒体见解管道，您必须调用 `DeleteMediaPipeline`。
- 只要任务成功运行或完成，`VoiceAnalyticsProcessor` 元素的状态就会保持在 `InProgress`。

使用 Amazon 监控呼叫分析渠道 CloudWatch

您可以使用亚马逊 CloudWatch 来监控 Amazon Chime SDK 调用分析管道。还可以设置特定阈值监视警报，在达到对应阈值时发送通知或采取行动。有关的更多信息 CloudWatch，请参阅 [Amazon CloudWatch 用户指南](#)。

主题

- [先决条件](#)
- [通话分析指标](#)
- [CloudWatch 管道指标的维度](#)

先决条件

要使用 CloudWatch 指标，您必须先创建一个媒体管道服务相关角色，该角色授予向 Amazon CloudWatch 发布服务指标的权限。有关此服务相关角色的更多信息，请参阅本指南中的 [为媒体管道创建服务相关角色](#)。

通话分析指标

Amazon Chime SDK 通话分析会将以下指标发布到您使用媒体见解配置创建的媒体见解管道的 AWS/ChimeSDK 命名空间。

指标	描述
<code>MediaInsightsPipelineCreated</code>	已成功创建媒体见解管道。 单位：计数
<code>MediaInsightsPipelineStopped</code>	已成功停止媒体见解管道。 单位：计数
<code>MediaInsightsPipelineFailed</code>	媒体见解管道失败。

指标	描述
	单位：计数
MediaInsightsPipelineDuration	从管道创建到停止/失败之间的时间。 单位：秒
MediaInsightsPipelineBillingDuration	媒体见解管道计费持续时间。 单位：计数
RecordingFileSize	录音文件大小。 单位：字节
RecordingDuration	录音持续时间。 单位：秒

CloudWatch 管道指标的维度

下表列出了可用于监控呼叫分析管道的 CloudWatch 维度。

维度	描述
MediaInsightsPipelineConfigurationId	媒体见解管道配置 ID。
MediaInsightsPipelineConfigurationName	媒体见解管道配置名称。

呼叫分析处理器和输出目的地

对于每个媒体见解管道配置，您只能指定一次唯一元素。所有处理器和接收器必须位于同一个 AWS 账户中，并且您必须在与您调用的终端节点相同的 AWS 区域中创建它们。例如，如果您使用 Amazon Chime SDK 媒体管道的 us-east-1 终端节点，则无法传递来自 us-west-2 区域的 Kinesis Data Stream。

展开每个部分，了解有关每个目的地的信息。

Amazon Transcribe 通话分析功能处理器目的地

支持的接收器：KinesisDataStreamSink。

您不能将此处理器与 Amazon Transcribe 处理器结合使用。有关 Amazon Transcribe 呼叫分析的更多信息，请参阅《Amazon Transcribe 开发者指南》中的[实时通话分析](#)。如果您通过在 AmazonTranscribeCallAnalyticsProcessorConfiguration API 调用中添加 PostCallAnalyticsSettings 来启用[通话后分析](#)，则当媒体见解管道停止并处理完成时，您将在指定的 Amazon S3 位置收到构件。

Note

如果您暂停管道超过 35 秒然后恢复管道，则会在 Amazon S3 存储桶中生成具有不同会话 ID 的单独文件中的通话后构件。

通话后构件包括分析 JSON 文件和录音 WAV 或 Opus 文件。作为元数据章节中 onetimeMetadata 的一部分，对于每个 Amazon Transcribe 通话分析功能通话后会话，已编辑（如果您启用内容编辑）和未编辑录音文件的 Amazon S3 存储桶 URL 都会发送到 Kinesis Data Streams 一次。

使用 Amazon Transcribe 通话分析功能进行呼叫分析采用 Kinesis 视频流中的音频数据输入。

- 支持的媒体编码：PCM 签名的 16 位小端序音频。
- 支持的媒体采样率：介于 8,000Hz 和 48,000Hz 之间。

Amazon Transcribe 分析流程的 StreamConfiguration 输入：

- 必须为每个流指定 KinesisVideoStreamArn。
- （可选）KVS FragmentNumber 使用指定片段之后的数据块启动呼叫分析作业。如果未提供，则它将使用 Kinesis 视频流中的最新片段。
- StreamChannelDefinition 定义了发言人。Amazon Transcribe 通话分析功能需要双声道音频。调用 [CreateMediaInsightsPipeline](#) API 时，必须指定哪个扬声器在哪个频道上。例如，如果您的座席先发言，则将 ChannelId 设置为 0 以表示第一个频道，并将 ParticipantRole 设置为 AGENT 表示座席正在发言。

Note

当您使用 Voice Connector 创建带有 Amazon Transcribe 通话分析功能处理器的 MediaInsightsPipeline 时，Voice Connector 账户分支音频是 AGENT，PSTN 分支音频是用于 ParticipantRole 的 CUSTOMER。
对于 Voice Connector SIPREC，我们依赖于 SIPREC 元数据。在大多数情况下，认为 AGENT 是词典值最低的流标签。

以下示例介绍了一个双声道音频流的 Kinesis 视频流输入。

```
"StreamChannelDefinition" : {
  "NumberOfChannels" : 2
  "ChannelDefinitions": [
    {
      "ChannelId": 0,
      "ParticipantRole": "AGENT"
    },
    {
      "ChannelId": 1,
      "ParticipantRole": "CUSTOMER"
    }
  ]
}
```

相比之下，以下示例介绍了来自两个不同的 Kinesis 视频流的两个单声道输入。

```
KVS-1:
  "StreamChannelDefinition" : {
    "NumberOfChannels" : 1
    "ChannelDefinitions": [
      {
        "ChannelId": 0,
        "ParticipantRole": "AGENT"
      }
    ]
  }
KVS-2:
  "StreamChannelDefinition" : {
    "NumberOfChannels" : 1
    "ChannelDefinitions": [
```

```

    {
      "ChannelId": 1,
      "ParticipantRole": "CUSTOMER"
    }
  ]
}

```

Amazon Transcribe 通话分析功能输出

每条 Amazon Transcribe 记录都包含 UtteranceEvent 或 CategoryEvent，但不会两者兼而有之。CategoryEvents 具有 TranscribeCallAnalyticsCategoryEvent 的 detail-type。

以下示例介绍了 Amazon Transcribe 的一次性元数据输出格式。

```

{
  "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"
}
```

以下示例介绍了 Amazon Transcribe 通话分析功能输出格式。

```
{
  "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,
      "VocabularyFilterMatch": boolean,
      "Stable": boolean,
      "ItemType": "string",
      "BeginOffsetMillis": number,
      "EndOffsetMillis": number,
    }, ]
    "Entities": [{
      "Content": "string",
      "Confidence": number,
```



```

    "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"
}

```

以下示例显示了通话分析 Amazon S3 录音的更新元数据格式。

```

{
    "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 通话录音元数据

以下示例显示了用于对 Alice 和 Bob 两个人之间的 SIP 通话进行录音的元数据。两个参与者都发送和接收音频和视频。为简单起见，此示例只有 SIP 和 SDP 的片段，SRC 将每名参与者的流记录到 SRS，而不进行混合。

```

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">
        <name 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="8zc6e0lYTlWIINA6GR+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>

```

以下示例显示了一名通话参与者与另一名参与者接通电话时已更新的元数据。在这种情况下，participant_id srfBEImCRp2QB23b7Mpk0w== 只接收媒体流而不发送任何媒体，因此省略了 send XML 元素。与之相反，participant_id zSfPoSvdSDCmU3A3TRDxAw== 向另一名参与者发送媒体，但不接收来自该参与者的媒体，因此省略了 recv XML 元素。

```

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>8zc6e01YTLWIINA6GR+3ag==</recv>
      <recv>EiXGlc+4TruqqoDaNE76ag==</recv>
    </participantstreamassoc>
    <participantstreamassoc
      participant_id="zSfPoSvdSDCmU3A3TRDxAw==">
      <send>8zc6e01YTLWIINA6GR+3ag==</send>
      <send>EiXGlc+4TruqqoDaNE76ag==</send>
    </participantstreamassoc>
  </recording>
```

以下示例显示了通话恢复时的元数据更新。有效负载现在包含 send 和 recv XML 元素。

```
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>8zc6e01Yt1WIINA6GR+3ag==</recv>
    <recv>EiXG1c+4TruqqoDaNE76ag==</recv>
  </participantstreamassoc>
  <participantstreamassoc
    participant_id="zSfPoSvdSDCmU3A3TRDxAw==">
    <send>8zc6e01Yt1WIINA6GR+3ag==</send>
    <send>EiXG1c+4TruqqoDaNE76ag==</send>
    <recv>i1Pz3to5hGk8fuXl+PbwCw==</recv>
    <recv>UAAMm5GRQKSCMVvLy14rFw==</recv>
  </participantstreamassoc>
</recording>
```

Amazon Transcribe 处理器目的地

支持的接收器：KinesisDataStreamSink。

您不能将此处理器与 Amazon Transcribe 通话分析功能结合使用。有关 Amazon Transcribe 的输入和输出的更多信息，请参阅《Amazon Transcribe 开发者指南》中的[转录流式传输音频](#)。

Amazon Transcribe 的呼叫分析会话采用 Kinesis 视频流中的音频数据输入。

- 支持 MediaEncoding：PCM 签名的 16 位小端音频。
- 支持的 MediaSampleRate 采样率：介于 8,000 Hz 和 48,000 Hz 之间。

Amazon Transcribe 处理器的 StreamConfiguration 输入：

- 必须为每个流指定 KinesisVideoStreamArn。
- (可选) KVS FragmentNumber：使用特定片段之后的数据块启动呼叫分析作业。如果未提供，它将使用 Kinesis 视频流中最新的可用区块。
- StreamChannelDefinition Amazon Transcribe 目前支持两个频道的音频。您必须在运行时系统 StreamChannelDefinition 中指定 NumberOfChannels。此外，如果您在两个不同的频道中发送单声道音频，则必须通过 ChannelId。在您的转录中，频道被分配了标签 ch_0 和 ch_1。以下示例显示了一个单声道音频频道流的 KVS 输入。


```
"StreamChannelDefinition" : {"
  NumberOfChannels" : 1
}
```

以下示例显示了两个不同流中两个单声道音频输入的 KVS 输入。

```
KVS-1:
  "StreamChannelDefinition" : {
    "NumberOfChannels" : 1
    "ChannelDefinitions": [
      {
        "ChannelId": 0
      }
    ]
  }
KVS-2:
  "StreamChannelDefinition" : {
    "NumberOfChannels" : 1
    "ChannelDefinitions": [
      {
        "ChannelId": 1
      }
    ]
  }
```

Note

对于 Voice Connector 使用 Amazon Transcribe 处理器创建的 MediaInsightsPipeline，Voice Connector 账户分支音频分配给 channel-0，PSTN 分支音频分配给 channel-1。

对于 Voice Connector SIPREC，我们依赖于 SIPREC 元数据。在大多数情况下，会将词典值最低的流标签分配给 channel-0。

对于 Amazon Transcribe 和 Amazon Transcribe 通话分析功能处理器，如果您传递两个 Kinesis 视频流，并且每个流都包含一个单声道音频频道，则在处理 Transcribe 或 Transcribe 呼叫分析数据之前，我们会将两个频道交错成一个音频流。

Amazon Transcribe 输出

以下示例介绍了适用于 Amazon Transcribe 的一次性元数据输出格式。

```

{
  "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"
}

```

以下示例介绍了 Amazon Transcribe 输出格式。

```

{
  "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
      }
    ]
  }
}
```

语音分析处理器目的地

支持的接收器：KinesisDataStreamSink、SqsQueueSink、SnsTopicSink 和 LambdaFunctionSink。

您可以将此处理器与 Amazon Transcribe 通话分析功能处理器、Amazon Transcribe 处理器或通话录音结合使用。您必须使用 [StartSpeakerSearchTask](#) 或 [StartVoiceToneAnalysisTask](#) API 来调用语音分析处理器。有关使用语音分析的更多信息，请参阅[使用 Amazon Chime SDK 语音分析](#)。

将 Kinesis Data Streams 用作接收器

呼叫分析生成的 Kinesis Data Streams (KDS) 记录包括媒体管道 ID、详细信息类型、元数据和处理器特定部分。有关使用来自 Kinesis Data Streams 的数据的更多信息，请参阅《Amazon Kinesis Streams 开发者指南》中的[从 Amazon Kinesis Data Streams 读取数据](#)。要使用此接收器创建配置，您必须拥有对指定流的 `kinesis:DescribeStream` 权限。

元数据

生成的 KDS 记录 `metadata` 部分包含在 API 调用 `CallAnalyticsRuntimeMetadata` 期间指定的任何键值对。[CreateMediaInsightsPipeline](#) 如果呼叫分析会话是由 Voice Connector 发起的，则元数据部分将自动填充以下参数：

- `transactionId`
- `fromNumber`
- `toNumber`
- `callId`
- `voiceConnectorId`
- `direction`

除了上面显示的参数外，Voice Connector 启动的呼叫分析会话的元数据部分还将填入一个包含以下内容的 `oneTimeMetadata` 字段：

- `inviteHeaders`
- `siprecMetadata`

这在会话开始时只发布到 Kinesis Data Streams `detail-type a CallAnalyticsMetadata Streams` 一次，并且有。

您可以在 `MediaInsightsRuntimeMetadata` 为每个 [CreateMediaInsightsPipeline](#) API 调用传递唯一标识符，这样您就可以唯一标识传送到 Kinesis 数据流的每条记录的来源。

亚马逊 S3 通话录音

通话分析录音从 KVS 流中读取音频，将其录制为音频文件，然后将该文件上传到指定的 Amazon S3 存储桶。录音后，通话分析还会将通话元数据以及文件位置发送到 KDS。如果您启用数据仓库，则呼叫元数据（如果使用了 SIPREC，则包括 SIPREC 元数据）将以一组 Parquet 表的形式传送到数据仓库，供您查询。

与任何其他呼叫分析处理器一样，您需要先为管道创建配置。您可以使用 Amazon Chime SDK 控制台或者 CLI 来创建配置。然后，您使用 CLI 控制台创建管道。有关使用控制台创建录制配置的更多信息，请参阅本节前文中的 [创建呼叫分析配置](#)。有关使用录制工作流程的更多信息，请参阅本节前文中的 [通话录音工作流程](#)。

若要使用 CLI 创建配置

运行以下命令：

```
aws chime-sdk-media-pipeline create-media-insights-pipeline-configuration --cli-input-json file://configuration.json
```

以下示例介绍了一个仅启用录制的配置 JSON 文件：

```
{
  "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"
  }
]
}
```

请记住以下事项：

- 要通过 Kinesis Video Streams Video Streams 启用通话录音，音频应采用 PCM 签名 16 位小端字节序。采样速率必须为 8kHz。
- 构建者必须为 Kinesis Video Stream 设置足够长的数据保留期，以确保呼叫分析保留和使用这些片段。
- 如果您单独或与其他处理器组合启用通话录音，则必须提供两个 Kinesis Video Stream ARN 进行录音。通话录音不支持单个立体声音频输入。

Amazon S3 通话录音元数据输出

以下示例显示了呼叫分析 Amazon S3 录音的元数据输出格式。

```
{
  "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"
}
```

启用语音增强

要启用语音增强，请在 [CreateMediaInsightsPipelineConfiguration](#) API 调用中加入一个 `VoiceEnhancementSinkConfiguration` 元素。

本示例将展示典型元素。

```
{
  "Type": "VoiceEnhancementSink",
  "VoiceEnhancementSinkConfiguration": {
    "Disabled": Boolean (string) // FALSE ==> Voice Enhancement will be performed
  }
}
```

要更新配置，请将该 `VoiceEnhancementSinkConfiguration` 元素添加到 [UpdateMediaInsightsPipelineConfiguration](#) API 调用中。当你这样做时，[GetMediaInsightsPipelineConfiguration](#) API 会将该 `VoiceEnhancementSinkConfiguration` 元素包含在结果中。

此示例请求说明如何启用语音增强和 Amazon S3 录音。

```
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 元素始终需要呼叫分析配置中的 S3RecordingSink 元素。

将转录与录音接收器相结合

您可以将 Amazon Transcribe 和 Amazon Transcribe 通话分析功能处理器与 Amazon S3 录音接收器结合使用。除了 Amazon RecordingSinkConfiguration Transcribe 处理器之外，生成器还可以通过 [CreateMediaInsightsPipelineConfiguration](#) API 调用或使用控制台传递 S3。

与 Amazon S3 录音接收器配合使用，您可以使用 Amazon Transcribe 或 Amazon Transcribe 通话分析功能处理器，但不能两者兼而有之。除了带或不带转录处理器的录音接收器外，您还可以在相同的配置中添加语音分析。

Note

您可以使用上面列出的任何处理器启用录制。但是，如果您同时启用 Amazon Transcribe 通话分析功能和 Amazon S3 通话录音，您将收到重复的录音文件，一个来自 Amazon Transcribe 通话分析功能，另一个来自 Amazon S3 通话录音。

请记住以下事项：

- 您必须使用唯一的 `MediaInsightsPipelineConfigurationName`。
- 有关 `ResourceAccessRoleArn` 的更多信息，请参阅本指南中的 [使用通话分析资源访问角色](#)。
- `Destination` 值必须是 S3 路径 ARN。Amazon S3 存储桶必须由同一账户拥有。
- 如果您使用带有 Transcribe 和录制功能的配置来创建管道，则暂停和恢复仅显示在 Kinesis Data Streams 生成的见解中。KVS 流中的所有数据都被记录下来并上传到 Amazon S3。
- 如果配置除了录音之外还使用 Amazon 转录或转录呼叫分析 (TCA)，则媒体见解管道会实时提供转录或转录呼叫分析见解，然后在通话结束时提供 Amazon S3 录音。如果转录服务在呼叫分析期间失败，S3 录音作业仍会尝试运行。相反，Amazon S3 录制失败不会影响转录见解，因为它是在转录完成后运行的。

此示例显示了使用 Amazon Transcribe 处理器和 Amazon S3 录音接收器的配置。该示例还启用了部分结果稳定功能，这可以减少输出延迟，但可能会影响准确性。有关更多信息，请参阅《Amazon Transcribe 开发者指南》中的 [部分结果稳定功能](#)。

```
{
  "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",
      "PiiEntityTypes": "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"
    }
  ]
}
```

使用亚马逊 EventBridge 通知

Amazon Chime SDK 呼叫分析支持在媒体见解渠道的状态发生变化或呼叫分析实时警报条件满足时，向默认 EventBridge 总线发送事件。对于媒体见解管道错误状态更新，我们建议您配置一个 EventBridge 目标，以便在资源异步失败时通知您。来电分析通知有 `aws.chime` 来源和各种详细信息类型，将在以下各节中共享。有关更多信息，请参阅 [Amazon EventBridge 用户指南](#)。

主题

- [状态更新](#)
- [实时提醒](#)

状态更新

媒体洞察管道会在呼叫分析会话进行时发送 EventBridge 通知，要么成功结束，要么遇到错误。在以下情况下，您会收到一条包含“媒体见解状态变更”详细信息的 EventBridge 通知：

- 媒体见解管道的状态发生了变化。
- 媒体见解管道元素的状态发生了变化。
- 所有管道元素都已停止。
- 任何管道元素都出现故障。

详细信息部分始终包含以下字段：

- `version`
- `mediaInsightsPipelineArn`
- `eventType`

如果媒体见解管道包含多个元素，例如分析处理器和数据接收器，则详细信息部分还包括一个 `mediaInsightsPipelineElementStatuses` 字段。此字段表示管道中每个元素的状态。每个管道元素可能的状态可能为：

- `NotStarted`
- `InProgress`
- `Stopped`
- `Failed`

详细信息部分还包括在 [CreateMediaInsightsPipeline](#) API 调用 `MediaInsightsRuntimeMetadata` 期间在中指定的任何键值对。如果呼叫分析会话是由 Voice Connector 发起的，则元数据部分将自动填充以下参数：

- `transactionId`
- `fromNumber`
- `toNumber`
- `callId`
- `voiceConnectorId`
- `direction`

每当媒体见解管道包含单个元素时，就会出现以下事件类型。展开每个部分以了解更多信息。

Amazon Chime SDK 媒体见解正在进行中

此示例介绍了典型的事件结构。

```
{
  "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 媒体见解已暂停

此示例介绍了典型的事件结构。

```
{
  "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 媒体见解已停止

此示例介绍了典型的事件结构。

```
{
```

```
"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 媒体见解暂时失败

表示服务遇到了临时故障，将尝试重试。您无需进行操作。

此示例介绍了典型的事件结构。

```
{
  "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 媒体见解永久失败

表示需要您采取措施的故障。使用 `failureReason` 对问题进行故障排除。典型故障可能包括：

- 资源访问角色权限不足
- 资源丢失或已删除
- 呼叫分析代表您调用 AWS 服务（例如 Amazon Transcribe 或 Amazon Kinesis）从而产生的限制。
- KVS 流上的媒体格式不兼容

此示例介绍了典型的事件结构。

```
{
  "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 字段为可选项。例如，典型的原因可能是 Access denied when assuming resource access role。

每当为由 Amazon Chime SDK Voice Connector 启动的呼叫分析会话创建媒体见解管道或创建尝试失败时，都会出现以下事件类型。展开每个部分以了解更多信息。

Amazon Chime SDK 媒体见解已创建

此示例显示了一个典型的成功事件。

```
{
  "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",
  }
}
```

创建 Amazon Chime SDK 媒体见解失败

此示例显示了一个典型的故障事件。

```
{
```

```

"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"
}
}

```

当媒体见解管道包含多个元素时，可能会出现以下事件类型。示例通知针对与 S3RecordingSink 组合使用的 AmazonTranscribeProcessor。展开每个部分以了解更多信息。

AmazonTranscribeProcessor 正在进行且 S3 RecordingSink 尚未启动

此示例介绍了典型的事件结构。

```

{
  "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 已成功且 S3 RecordingSink 正在进行中

此示例介绍了典型的事件结构。

```

{
  "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 已失败且 S3 RecordingSink 正在进行中

此示例介绍了典型的事件结构。

```

{
    "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 已失败且 S3 RecordingSink 已成功

此示例介绍了典型的事件结构。

```

{
  "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 已成功且 S3 RecordingSink 失败

此示例介绍了典型的事件结构。

```

{
  "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 已暂停且 S3 RecordingSink 尚未启动

此示例介绍了典型的事件结构。

```
{
  "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 已暂时失败且 S3 RecordingSink 尚未启动

此示例介绍了典型的事件结构。

```
{
  "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 然后 S3 RecordingSink 成功了

此示例介绍了典型的事件结构。

```
{
  "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"
}
}

```

S3 RecordingSink 成功且正在 VoiceEnhancement 进行中

此示例介绍了典型的事件结构。

```

{
  "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
  }
]
}
}

```

由于呼叫时间超过 30 分钟，S3 RecordingSink 成功 VoiceEnhancement 但失败了

此示例介绍了典型的事件结构。

```

{
  "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
      }
    ]
  }
}

```



```
    ]  
  }  
}
```

由于呼叫时间少于 30 分钟，S3 RecordingSink 成功 VoiceEnhancement 但失败了

此示例介绍了典型的事件结构。

```
{  
  "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  
      }  
    ]  
  }  
}
```

实时提醒

Note

只有 Amazon Transcribe 和 Amazon Transcribe 通话分析功能处理器支持实时提醒。

Amazon Chime SDK 呼叫分析允许开发者设置规则，以便在分析会话期间通过处理器发送实时警报。提醒将以详细类型发送到亚马逊 EventBridge Media Insights Rules Matched。EventBridge 支持与 Lambda、Amazon SQS 和 Amazon SNS 等下游服务集成，以触发终端用户的通知或启动其他自定义业务逻辑。

针对 `MediaInsightsPipelineConfiguration` 将实时警报设置为 `RealTimeAlertConfiguration` 字段的一部分。您可以使用 Amazon Chime 软件开发工具包控制台来配置该字段，也可以调用 [CreateMediaInsightsPipelineConfiguration](#) 或 [UpdateMediaInsightsPipelineConfiguration](#) API。

此示例说明如何通过 API 创建或更新实时警报配置。

```
{
  "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"
    }
}
]
}

```

实时警报配置中的每条规则都是独立触发的。如果同时满足多个规则条件，您可能会收到多条 EventBridge 通知。要为警报创建规则列表，可以在以下规则类型中进行选择：

关键字匹配

在言语或转录事件中匹配一组指定的关键字或短语时发出警报。在以下情况下，您可以将警报配置为发出事件：

- 朗读任何指定的关键字，并且将 Negate 设置为 false。
- 如果将 Negate 设置为 true，则在整个通话过程中，不朗读所有指定的关键字。

Amazon Transcribe 和 Amazon Transcribe 分析支持这种规则类型。

情绪分析

当特定情绪类型持续一段滚动窗口期时发出警报。只有 Transcribe 呼叫分析支持此规则。

问题检测

在语音事件中检测到问题时发出警报。只有 Transcribe 呼叫分析支持此规则类型。

以下示例介绍了一个 KeywordMatch 规则的实时警报事件。

```

{
    "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",
}
}
```

有些 EventBridge 字段特定于匹配的规则类型：

关键字匹配字段

utteranceId：如果您使用 Amazon Transcribe 通话分析功能，则为包含匹配关键字的转录 ID。仅适用于口语关键词匹配。

resultId：如果您使用 Amazon Transcribe，则为包含匹配关键字的转录 ID。仅适用于口语关键词匹配。

beginTimestamp：包含匹配关键字的转录开始时间。仅适用于口语关键词匹配。

情感分析字段

beginTimestamp：匹配情绪的滚动窗口的开始时间。

endTimestamp：匹配情绪的滚动窗口的结束时间。

创建 Amazon Chime SDK 数据湖

Amazon Chime SDK 呼叫分析数据湖允许您将基于机器学习的见解和来自 Amazon Kinesis Data Streams 的任何元数据流传输到您的 Amazon S3 存储桶。例如，使用数据湖访问录制文件的 URL。要创建数据湖，您可以从 Amazon Chime 软件开发工具包控制台部署一组 AWS CloudFormation 模板，也可以使用以编程方式部署一组模板。AWS CLI 数据湖使您能够通过引用 Amazon Athena 中的 AWS Glue 数据表来查询您的呼叫元数据和语音分析数据。

主题

- [先决条件](#)
- [数据湖术语和概念](#)
- [创建多个数据湖](#)
- [数据湖区域可用性](#)

- [数据湖架构](#)
- [数据湖设置](#)

先决条件

要创建 Amazon Chime SDK 数据湖，您必须具备以下物品：

- Amazon Kinesis Data Streams。有关更多信息，请参阅《Amazon Kinesis 流开发者指南》中的[通过 AWS 管理控制台创建流](#)。
- 一个 S3 存储桶。有关更多信息，请参阅《Amazon S3 用户指南》中的[创建第一个 Amazon S3 存储桶](#)。

数据湖术语和概念

使用以下术语和概念来理解数据湖的工作原理。

Amazon Kinesis Data Firehose

提取、转换、加载 (ETL) 服务，可靠地捕获、转换、传送流数据到数据湖、数据存储和分析服务。有关更多信息，请参阅[什么是 Amazon Kinesis Data Firehose ?](#)

Amazon Athena

Amazon Athena 是一种交互式查询服务，让您能够使用标准 SQL 分析 Amazon S3 中的数据。Athena 无服务器，因此您无需管理基础设施，只需为运行的查询付费。要使用 Athena，请指向 Amazon S3 中的数据，定义架构，然后使用标准 SQL 查询。您还可以使用工作组对用户进行分组，并控制他们在运行查询时可以访问的资源。工作组使您能够管理查询并行性，并在不同的用户组和工作负载之间确定查询执行的优先级。

Glue 数据目录

在 Amazon Athena 中，表和数据库包含详细说明底层源数据架构的元数据。每个数据集都必须在 Athena 中有一个对应的表。表中的元数据将 Amazon S3 存储桶的位置告知 Athena。元数据还指定数据结构，例如列名、数据类型和表名。数据库仅保存数据集的元数据和架构信息。

创建多个数据湖

通过提供唯一的 Glue 数据库名称来指定呼叫见解的存储位置，可以创建多个数据湖。对于给定 AWS 账户，可以有多种呼叫分析配置，每种配置都有相应的数据湖。这意味着可以将数据分离应用于某些用

例，例如自定义保留策略和数据存储方式的访问策略。可以采用不同的安全策略来访问见解、记录和元数据。

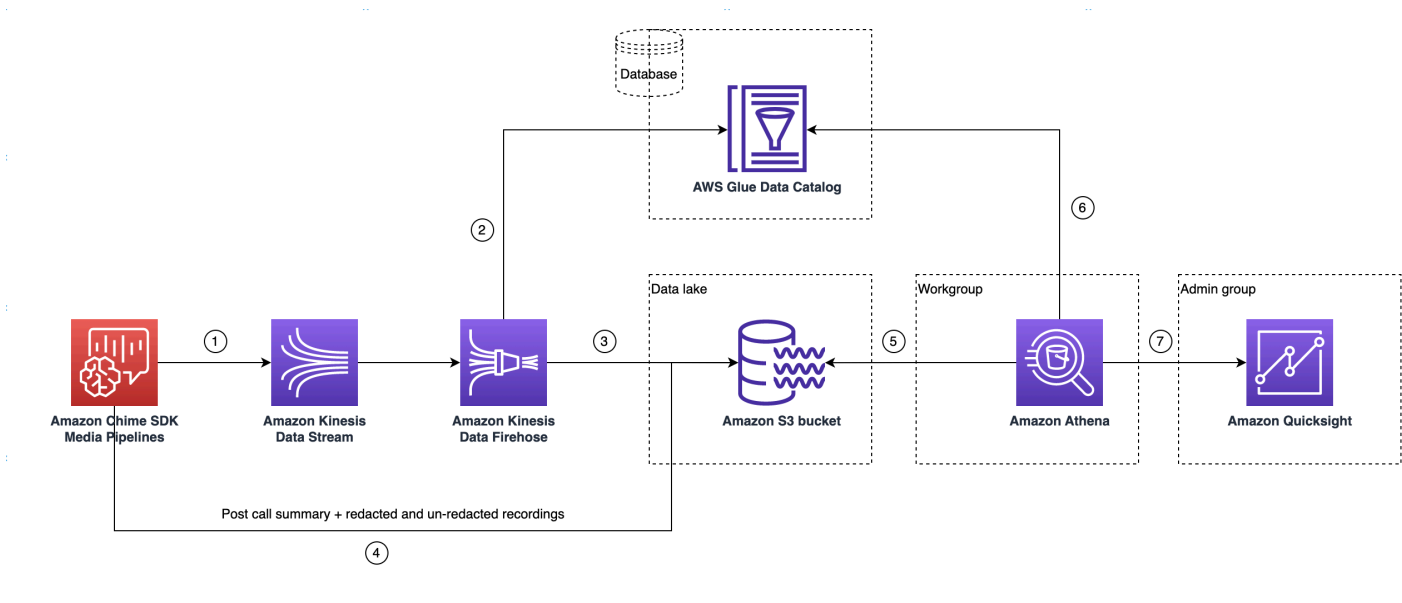
数据湖区域可用性

Amazon Chime SDK 数据湖在以下区域中推出。

区域	Glue 表	Amazon QuickSight
us-east-1	可用	可用
us-west-2	可用	可用
eu-central-1	可用	可用

数据湖架构

下图显示的是数据湖的架构。绘图中的数字对应于下面的编号文本。



在图中，使用AWS控制台从媒体见解管道配置设置工作流程部署 CloudFormation 模板后，以下数据将流向 Amazon S3 存储桶：

1. Amazon Chime SDK 呼叫分析将开始将实时数据流式传输到客户的 Kinesis Data Streams。
2. Amazon Kinesis Firehose 会缓冲这些实时数据，直到其累积 128MB，或时间达到 60 秒，以先到者为准。然后，Firehose 使用 Glue 数据目录中的

amazon_chime_sdk_call_analytics_firehose_schema 来压缩数据，并将 JSON 记录转换为 Parquet 文件。

3. Parquet 文件以分区格式存放在您的 Amazon S3 存储桶中。
4. 除了实时数据外，通话后的 Amazon Transcribe 通话分析功能摘要 .wav 文件（如果在配置中指定，则经过编辑和未编辑）和通话录音 .wav 文件也将发送到您的 Amazon S3 存储桶。
5. 您可以使用 Amazon Athena 和标准 SQL 来查询 Amazon S3 存储桶中的数据。
6. 该 CloudFormation 模板还创建了一个 Glue 数据目录，用于通过 Athena 查询此通话后摘要数据。
7. Amazon S3 存储桶上的所有数据也可以使用亚马逊 QuickSight 进行可视化。QuickSight 使用亚马逊 Athena 与亚马逊 S3 存储桶建立连接。

Amazon Athena 表使用以下功能来优化查询性能：

数据分区

分区可将您的表格分成多个部分，并根据日期、国家和地区等列值将相关数据保存在一起。分区充当虚拟列。在这种情况下，CloudFormation 模板在创建表时定义分区，这有助于减少每次查询扫描的数据量并提高性能。您还可以按分区筛选，以限制查询所扫描的数据量。有关更多信息，请参阅《Amazon Athena 用户指南》中的 [在 Athena 中对数据进行分区](#)。

此示例介绍了日期为 2023 年 1 月 1 日的分区结构：

i.

```
s3://example-bucket/amazon_chime_sdk_data_lake
                               /serviceType=CallAnalytics/detailType={DETAIL_TYPE}/
year=2023
                               /month=01/day=01/example-file.parquet
```

ii. 其中，DETAIL_TYPE 为下列项之一：

- a. CallAnalyticsMetadata
- b. TranscribeCallAnalytics
- c. TranscribeCallAnalyticsCategoryEvents
- d. Transcribe
- e. Recording
- f. VoiceAnalyticsStatus
- g. SpeakerSearchStatus
- h. VoiceToneAnalysisStatus

优化列式数据存储的生成

Apache Parquet 使用按列压缩、基于数据类型的压缩和谓词下推来存储数据。更好的压缩率或跳过数据块意味着从 Amazon S3 存储桶中读取更少的字节。这样可以实现更佳的查询性能和更低的成本。为了进行此优化，在 Amazon Kinesis Data Firehose 中启用了从 JSON 到 parquet 的数据转换。

分区投影

这项 Athena 功能会自动为每天创建分区，以提高基于日期的查询性能。

数据湖设置

使用 Amazon Chime SDK 控制台完成以下步骤。

1. 启动 Amazon Chime SDK 控制台 (<https://console.aws.amazon.com/chime-sdk/home>)，然后在导航窗格的呼叫分析下，选择配置。
2. 完成步骤 1，选择下一步，然后在步骤 2 页面上，选中语音分析复选框。
3. 在“输出详细信息”下，选中“要执行历史分析的数据仓库”复选框，然后选择“部署 CloudFormation 堆栈”链接。

系统会将您发送到 CloudFormation 控制台中的快速创建堆栈页面。

4. 输入堆栈的名称，然后输入以下参数：
 - a. DataLakeType— 选择“创建通话分析” DataLake。
 - b. KinesisDataStreamName：选择您的流。它应该是用于呼叫分析流的流。
 - c. S3BucketURI：选择您的 Amazon S3 存储桶。URI 必须具有前缀 `s3://bucket-name`
 - d. GlueDatabaseName：选择一个唯一的 AWS Glue 数据库名称。您不能重复使用 AWS 账户中的现有数据库。
5. 选中确认复选框，然后选择 Create data lake。等待 10 分钟让系统创建数据湖。

使用 AWS CLI 的数据湖设置

AWS CLI 用于创建具有调用创建堆栈权限 CloudFormation 的角色。按照以下步骤创建和设置 IAM 角色。有关更多信息，请参阅《AWS CloudFormation 用户指南》中的[创建堆栈](#)。

1. 创建一个名为 AmazonChimeSdkCallAnalytics-Datalake-Provisioning-Role 的角色，并将信任策略附加到允许担任该角色的角色。CloudFormation

1. 使用以下模板创建 IAM 信任策略，并将文件保存为 .json 格式。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

2. 运行 `aws iam create-role` 命令并将信任策略作为参数传递。

```
aws iam create-role \
--role-name AmazonChimeSdkCallAnalytics-Datalake-Provisioning-Role
--assume-role-policy-document file://role-trust-policy.json
```

3. 记下响应中返回的角色 ARN。在下一步中需要用到角色 ARN。

2. 创建具有创建 CloudFormation 堆栈权限的策略。

1. 使用以下模板创建 IAM 策略，并将文件保存为 .json 格式。调用 `create-policy` 时需要此文件。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DeployCloudFormationStack",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack"
      ],
      "Resource": "*"
    }
  ]
}
```

```
}
```

2. 运行 `aws iam create-policy` 并将创建堆栈策略作为参数传递。

```
aws iam create-policy --policy-name  
testCreateStackPolicy  
--policy-document file://create-cloudformation-stack-policy.json
```

3. 记下响应中返回的角色 ARN。在下一步中需要用到角色 ARN。

3. 将 `aws iam attach-role-policy` 策略附加到该角色。

```
aws iam attach-role-policy --role-name {Role name  
created above}  
--policy-arn {Policy ARN created above}
```

4. 创建 CloudFormation 堆栈并输入所需参数：`aws cloudformation create-stack`。

为每种 `ParameterKey` 使用提供参数值 `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}
```

由数据湖设置创建的资源

下表列出了创建数据湖时创建的资源。

资源类型	资源名称和描述	服务名称
AWS Glue 数据目录数据库	GlueDatabaseName— 对属于通话见解和语音分析的所有 AWS Glue 数据表进行逻辑分组。	呼叫分析、语音分析
AWS Glue 数据目录表	amazon_chime_sdk_call_analytics_firehose_schema : 提供给 Kinesis Firehose 的呼叫分析语音分析的组合架构。	呼叫分析、语音分析
	call_analytics_metadata : 呼叫分析元数据的架构。包含 SIP 元数据和。 OneTimeMetadata	呼叫分析
	call_analytics_recording_metadata : 录音架构和语音增强元数据	呼叫分析、语音分析
	tranc@@@ ribe_call_analytics — Payload “Utterance TranscribeCallAnalytics	呼叫分析
	tranc@@@ ribe_call_analytics_category_events — Payload 的架构 TranscribeCallAnalytics	呼叫分析
	transcribe_call_analytics_post_call : 通话后转录呼叫分析摘要有效负载的架构	呼叫分析
	transcribe : 转录有效载荷的架构	呼叫分析
	voice_analytics_status : 语音分析就绪事件的架构	语音分析
	speaker_search_status : 标识匹配的架构	语音分析
voice_tone_analysis_status : 语调分析事件架构	语音分析	
Amazon Kinesis Data Firehose	AmazonChimeSDK-Call-Analytics- UUID - ##### Kinesis Data Firehose 管道数据	呼叫分析、语音分析
Amazon Athena 工作组	GlueDatabaseName-AmazonChime SDK DataAnalytics — 逻辑用户组，用于控制他们在运行查询时可以访问的资源。	呼叫分析、语音分析

配置亚马逊 QuickSight 控制面板

设置数据湖后，您可以使用预定义的指标配置 Amazon QuickSight 控制面板，以实现数据的可视化。您可以使用以下控制面板：

- Transcribe 呼叫分析 + 语音分析。指标包括 turn-by-turn 笔录、检测到的问题、结果、实体检测和语音配置文件 ID 匹配的摘要和详细视觉效果。
- Transcribe + 语音分析。指标包括 turn-by-turn 笔录、词汇匹配、语音和语音配置文件 ID 匹配的摘要和详细视觉效果。

以下主题说明了如何设置亚马逊 QuickSight 账户（如果您还没有 Amazon 账户）以及如何配置控制面板。

主题

- [创建 QuickSight 账户](#)
- [配置您的 QuickSight 账户](#)
- [创建 QuickSight 仪表板](#)

创建 QuickSight 账户

本节中的步骤说明了如何创建 Amazon QuickSight 账户。如果您已有账户，请跳到 [创建 QuickSight 仪表板](#)。

您可以通过以下方式创建 QuickSight 账户：

- 使用亚马逊 CloudFormation 模板。
- 使用 Amazon Chime SDK 控制台。

先决条件

开始之前，请收集以下信息：

- Amazon S3 存储桶的呼叫分析的名称。
- 通知电子邮件地址。系统将 QuickSight 通知发送到该地址。

使用 CloudFormation 模板创建账户

以下步骤说明了如何通过部署亚马逊 CloudFormation 模板来创建亚马逊 QuickSight 账户。该流程仅向您订阅企业版账户。有关定价的信息，请参阅 [Amazon QuickSight 定价](#)。

部署模板

1. 启动 AWS 控制台并登录到您的 AWS 账户。
2. 将以下 URL 粘贴到浏览器的地址栏中。确保按照指示输入您的区域。

```
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. 在快速创建堆栈页面上，输入以下内容：
 - a. 在堆栈名称下输入您的账户名称。
 - b. QuickSightNotificationEmail在您之前收集的电子邮件地址下。
 - c. 在“创建新账户”下 QuickSightSubscriptionForDataVisualization，选择“创建新AWS QuickSight 账户”。
 - d. 在 S3 下 BucketName，输入您的亚马逊 S3 存储桶的名称。
 - e. 选择“我确认 AWS CloudFormation 可能会创建 IAM 资源”。复选框。
4. 选择创建堆栈。

系统创建堆栈需要约 10 分钟的时间。

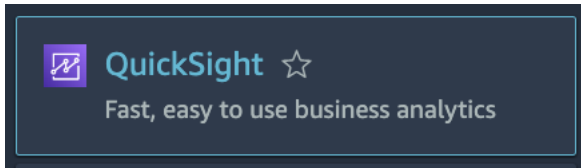
5. 构建完成后，选择 Go to Amazon QuickSight 并输入您的电子邮件地址以登录您的账户。

使用控制台创建账户

以下步骤说明了如何使用 Amazon Chime 软件开发工具包控制台创建亚马逊 QuickSight 账户。您必须使用企业版或企业版 + Q 账户。

要使用 控制台

1. [通过 https://console.aws.amazon.com/chime-sdk/home](https://console.aws.amazon.com/chime-sdk/home) 启动 Amazon Chime SDK 控制台 **QuickSight**，搜索并在搜索结果中选择。QuickSight



2. 选择“注册” QuickSight。
3. 选择企业版或企业版 + Q，然后选择继续。
4. 输入您的名字、姓氏、电话号码和您之前收集的电子邮件地址，然后选择继续。
5. 执行以下操作：
 - i. 在身份验证方法下，选择一个选项。

Note

如果您选择联合用户选项，则需要正确的 IAM 权限。有关更多信息，请参阅《[亚马逊 QuickSight 用户指南](#)》中的[注册亚马逊 QuickSight 订阅](#)。

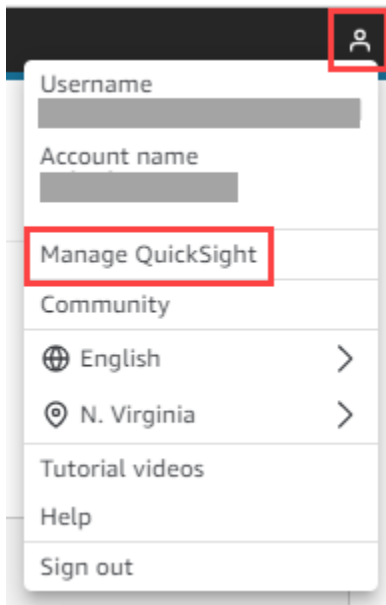
- ii. 在“QuickSight 区域”下，选择一个区域。
- iii. 在账户信息下，输入账户名称和您之前收集的电子邮件地址。
- iv. 在“QuickSight 访问 AWS 服务”下，使用默认角色，或者选择使用现有角色并从列表选择一个角色。
- v.（可选）根据需要，在 Allow access and autodiscovery for these resources 下，选择其他资源。
- vi. 完成后选择完成。
- vii. 构建完成后，选择 Go to Amazon QuickSight 并输入您的电子邮件地址以登录您的账户。

配置您的 QuickSight 账户

登录 QuickSight 账户后，您需要配置安全性并将自己添加到设置过程中创建的群组中。

若要配置安全性

1. 选择右上角的个人资料图标，然后 QuickSight 从出现的菜单中选择“管理”。



2. 在导航窗格中，选择安全性和权限。
3. 在“QuickSight 访问AWS服务”下，选择“管理”，并确保选择了以下服务。
 - Amazon Redshift
 - Amazon RDS
 - Amazon S3
 - Amazon Athena
 - IAM
4. 选择 [Select Amazon S3 buckets](#) 链接。
5. 选中您的 Amazon S3 存储桶旁边的复选框，然后在 Athena 工作组的写入权限列中选中右边的复选框。
6. 选择完成。
7. 选择保存。

将自己加入群组

1. 在导航窗格中，选择管理组，然后选择名称中包含管理员的组。例如，S3 BucketName — 管理员。
2. 选择添加用户，然后在出现的框中输入您的电子邮件别名。

您的姓名显示为Admin – 您的别名。
3. 选择添加。

创建 QuickSight 仪表板

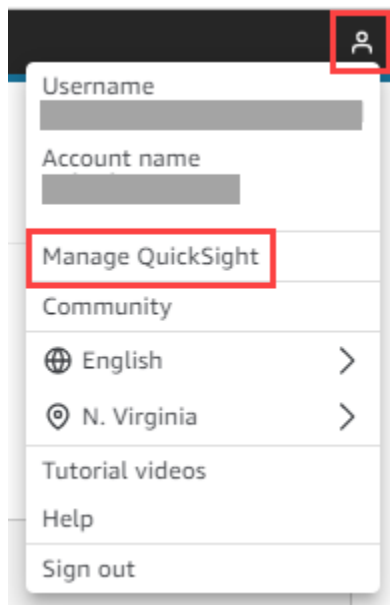
创建数据湖后，您可以创建一个可视化数据的 QuickSight 仪表板。您可以使用亚马逊 CloudFormation 模板或 Amazon Chime 软件开发工具包控制台来创建控制面板。以下步骤解释了这两种方法。

若要使用模板

1. 启动 Amazon CloudFormation 控制台。
2. 将以下链接粘贴到浏览器的地址栏中：[https://*region*.console.aws.amazon.com/cloudformation/home?region=*region*#/stacks/quickcreate?templateURL=https://chime-sdk-assets.s3.amazonaws.com/public_templates/AmazonChimeSDKQuickSightDashboards.yaml](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. 在快速创建堆栈页面的堆栈名称下，输入账户的名称。
4. 在下方 ActiveQuickSightAccount，选择 True。
5. 在 QuicksightDashboardSelection“呼叫分析”-“转录呼叫分析和语音分析”控制面板或“呼叫分析”-“转录和语音分析”控制面板。
6. 在 Amazon S3 下 BucketName，输入您的亚马逊 S3 存储桶的 URI。
7. 在下方 GlueDatabaseName，输入要在其上部署 QuickSight 仪表板的 Glue 数据库。
8. 选中我确认 AWS CloudFormation 可能会创建 IAM 资源复选框，然后选择创建堆栈。

手动配置 QuickSight 仪表板

1. 导航到您的 QuickSight 账户。
2. 在右上角选择个人资料图标，然后选择管理 QuickSight。



3. 在导航窗格中，请选择管理组，然后选择设置过程创建的组。
4. 选择添加用户，输入您的电子邮件地址，然后选择添加。

系统需要 10 分钟来部署该页面。

5. 使用 Amazon Chime 软件开发工具包控制台登录您的 QuickSight 账户并使用控制面板。

呼叫分析数据模型

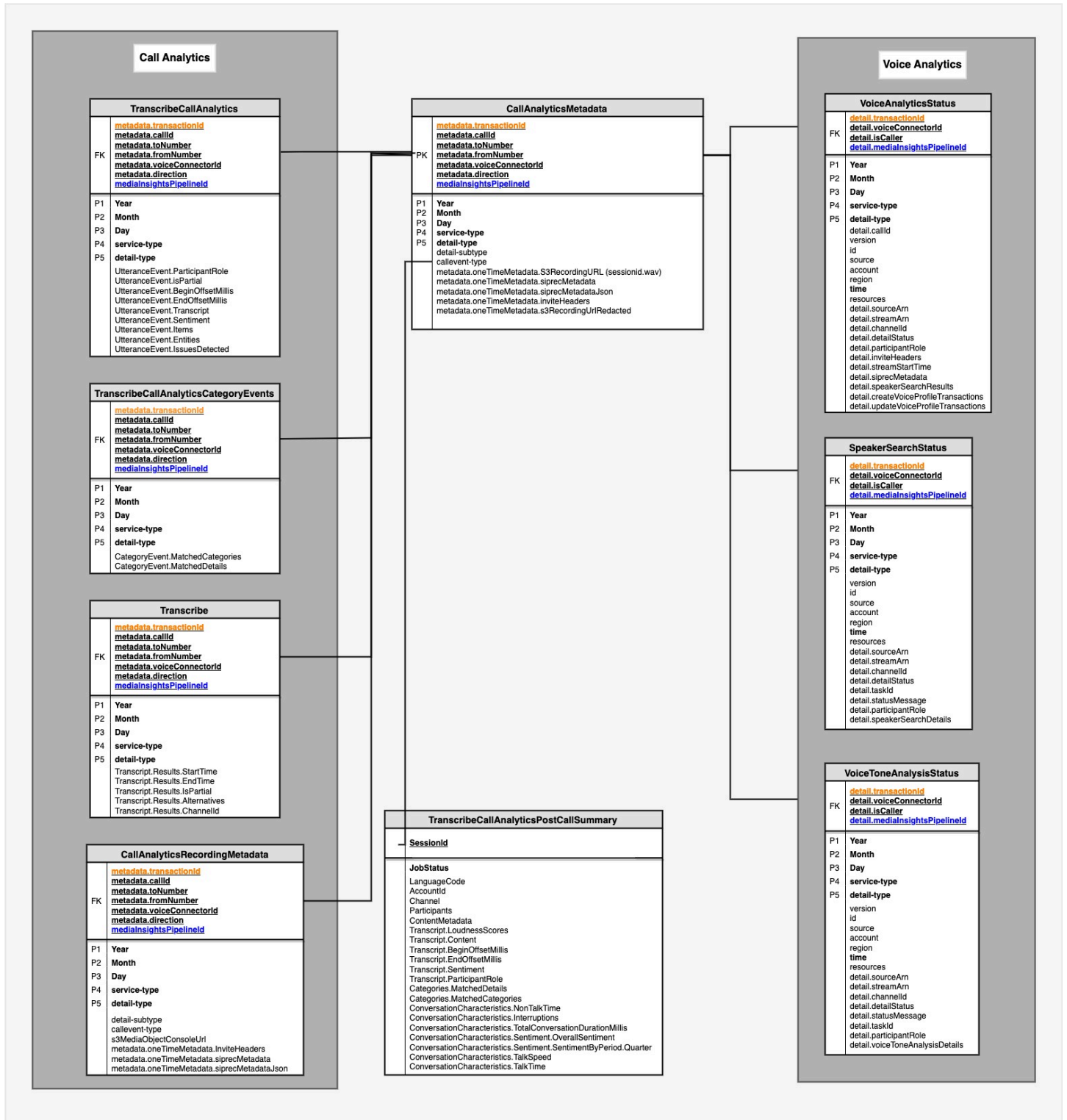
本部分中的信息列出并描述了 Amazon Chime SDK 呼叫分析数据模型，即 AWS Glue 数据目录中的一组表。

主题

- [Glue 数据目录表结构](#)
- [Glue 数据目录表](#)
- [示例查询](#)

Glue 数据目录表结构

下图显示了为 Amazon Chime SDK 呼叫分析和语音分析会话创建的 AWS Glue 数据目录的表结构。



下一部分列出并描述了目录中的表和字段。

Glue 数据目录表

下表列出并描述了 Amazon Chime SDK 呼叫分析 Glue 数据目录中的列、数据类型和元素。

主题

- [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

列名称	数据类型	元素	定义
time	字符串		事件生成时间戳 ISO 8601。
detail-type	字符串		与 service-type 相关的功能类型。
service-type	字符串		AWS 服务的名称，VoiceAnalytics 或 CallAnalytics。
detail-subtype	字符串		用于记录和 CallAnalyticsMetadata detail-type。
callevnt-type	字符串		与 SIP 关联的事件类型，例如更新、暂停、恢复

列名称	数据类型	元素	定义
mediaInsightsPipelineId	字符串		Amazon Chime SDK 媒体见解管线 ID。
metadata	字符串	voiceConnectorId	Amazon Chime SDK 语音连接器 ID。
		callId	与使用情况相关的参与者的呼叫 ID。
		transactionId	呼叫的事务 ID。
		fromNumber	E.164 发起电话号码。
		toNumber	E.164 目标电话号码。
		方向	呼叫方向，出站或进站。
		oneTimeMetadata.s3RecordingUrl	由 Transcribe 呼叫分析发出的媒体对象的 Amazon S3 存储桶 URL。
		oneTimeMetadata.s3RecordingUrlRedacted	由 Transcribe 呼叫分析发出的编辑后媒体对象的 Amazon S3 存储桶 URL。
		oneTimeMetadata.siprecMetadata	与呼叫相关的 XML 格式的 SIPREC 元数据。
oneTimeMetadata.siprecMetadataJson	与呼叫相关的 JSON 格式的 SIPREC 元数据。		

列名称	数据类型	元素	定义
		oneTimeMetadata.In viteHeaders	邀请标头。

call_analytics_recording_metadata

列名称	数据类型	元素	定义
time	字符串		事件生成时间戳 ISO 8601。
detail-type	字符串		与 service-type 相关的功能类型。
service-type	字符串		AWS 服务的名称，VoiceAnalytics 或 CallAnalytics。
detail-subtype	字符串		用于记录和 CallAnalyticsMetadata detail-type。
callevent-type	字符串		与 SIP 关联的事件类型
mediaInsightsPipelineId	字符串		Amazon Chime SDK 媒体见解管线 ID。
s3MediaObjectConsolidationUrl	字符串		媒体对象的 S3 存储桶 URL。
metadata	字符串	voiceConnectorId	Amazon Chime SDK 语音连接器 ID。
		callId	与使用情况相关的参与者的呼叫 ID。

列名称	数据类型	元素	定义
		transactionId	呼叫的事务 ID。
		fromNumber	E.164 发起电话号码。
		toNumber	E.164 目标电话号码。
		方向	呼叫方向，出站或进站。
		语音增强	与 service-type 相关的功能子类型。
		oneTimeMetadata.siprecMetadata	与呼叫相关的 XML 格式的 SIPREC 元数据。
		oneTimeMetadata.siprecMetadataJson	与呼叫相关的 JSON 格式的 SIPREC 元数据。
		oneTimeMetadata.InviteHeaders	邀请标头。

transcribe_call_analytics

列名称	数据类型	元素	定义
time	字符串		事件生成时间戳 ISO 8601。
detail-type	字符串		与 service-type 相关的功能类型。
service-type	字符串		AWS 服务的名称，VoiceAnalytics 或 CallAnalytics。

列名称	数据类型	元素	定义
mediaInsightsPipelineId	字符串		Amazon Chime SDK 媒体见解管线 ID。
metadata	字符串	voiceConnectorId	Amazon Chime 语音连接器 ID。
		callId	与使用情况相关的参与者的呼叫 ID。
		transactionId	呼叫的事务 ID。
		fromNumber	E.164 发起电话号码。
		toNumber	E.164 目标电话号码。
		方向	呼叫方向，Outbound 或 Inbound。
UtteranceEvent	struct	UtteranceId	与指定 Utterance Event 相关的唯一标识符。
		IsPartial	指示 Utterance Event 中的片段是完整的 (FALSE) 还是部分的 (TRUE)。
		ParticipantRole	为每个音频通道提供发言人角色，可以是客户或座席。
		BeginOffsetMillis	从音频流开始到 UtteranceEvent 开始之间的时间，以毫秒为单位。

列名称	数据类型	元素	定义
		EndOffsetMillis	从音频流开始到 UtteranceEvent 开始之间的时间，以毫秒为单位。
		转录	包含转录后文本。
		情绪	提供在指定片段中检测到的情绪。
		Items.beginoffsetmillis	转录后内容的开始时间，以毫秒为单位。
		Items.endoffsetmillis	转录后内容的结束时间，以毫秒为单位。
		Items.itemtype	识别内容的类型。 选项：PRONUNCIATION（口语词）和 PUNCTUATION。
		Items.content	转录的单词或标点符号。
		Items.confidence	与您的转录中的单词或短语相关的置信度分数。分数是介于 0 和 1 之间的值。值越大，指示识别内容与您的媒体中所说内容正确匹配的可能性越高。

列名称	数据类型	元素	定义
		Items.vocabularyfiltermatch	指示指定内容是否与您的请求中所含词汇筛选器中的单词匹配。如果为 true，则存在词汇筛选匹配项。
		Items.stable	已启用部分结果稳定，“稳定”指示指定内容是稳定的 (true)，还是在片段完成后可能发生变化 (false)。
		IssuesDetected.characteroffsets_begin	提供识别匹配项的第一个字符的字符计数。例如，与议题或类别相关的第一个字符与片段转录中的内容相匹配。
		IssuesDetected.characteroffsets_end	提供识别匹配项的最后一个字符的字符计数。例如，与议题或类别相关的最后一个字符与片段转录中的内容相匹配。
		Entities.beginoffsetmillis	识别为 PII 的言语的开始时间，以毫秒为单位。
		Entities.endoffsetmillis	识别为 PII 的言语的结束时间，以毫秒为单位。
		Entities.category	识别信息的类别。唯一的类别是 PII。

列名称	数据类型	元素	定义
		Entities.type	识别 PII 的类型。例如，NAME 或 CREDIT_DEBIT_NUMBER。
		Entities.content	识别为 PII 的一个或多个单词。
		Entities.confidence	与您的音频中识别 PII 实体相关的置信度分数。置信度分数介于 0 和 1 之间。值越大，指示识别实体与您的媒体中所说实体正确匹配的可能性越高。

transcribe_call_analytics_category_events

列名称	数据类型	元素	定义
time	字符串		事件生成时间戳 ISO 8601。
detail-type	字符串		与 service-type 相关的功能类型。
service-type	字符串		AWS 服务的名称，VoiceAnalytics 或 CallAnalytics。
mediaInsightsPipelineId	字符串		Amazon Chime SDK 媒体见解管线 ID。

列名称	数据类型	元素	定义
metadata	字符串	voiceConnectorId	Amazon Chime 语音连接器 ID。
		callId	与使用情况相关的参与者的呼叫 ID。
		transactionId	呼叫的事务 ID。
		fromNumber	E.164 发起电话号码。
		toNumber	E.164 目标电话号码。
		方向	呼叫方向，出站或进站。
CategoryEvent	数组	MatchedCategories	列出由用户定义的类别中的匹配项。

transcribe_call_analytics_post_call

列名称	数据类型	元素	定义
JobStatus	字符串		事件生成时间戳 ISO 8601。
LanguageCode	字符串		与 service-type 相关的功能类型。
转录	struct	LoudnessScores	<p>测量每位参与者的说话音量。使用此指标来查看呼叫者或座席是否大声说话或者大喊大叫，这通常表示愤怒。</p> <p>该指标表示为标准化值（给定片段中每秒</p>

列名称	数据类型	元素	定义
			的话音水平)，范围从 0 到 100，其中值越高表示声音越大。
		内容	包含转录后文本。
		Id	与指定 Utterance Event 相关的唯一标识符。
		BeginOffsetMillis	从音频流开始到 UtteranceEvent 开始之间的时间，以毫秒为单位。
		EndOffsetMillis	从音频流开始到 UtteranceEvent 开始之间的时间，以毫秒为单位。
		情绪	提供在指定转录片段中检测到的情绪。
		ParticipantRole	为每个音频通道提供发言人角色，可以是客户或座席。
		IssuesDetected.CharacterOffsets.Begin	提供识别匹配项的第一个字符的字符偏移量。例如，转录片段中与议题相关的第一个字符。

列名称	数据类型	元素	定义
		IssuesDetected.CharacterOffsets.End	提供识别匹配项的最后一个字符的字符偏移量。例如，转录片段中与议题相关的最后一个字符。
		OutcomesDetected.CharacterOffsets.Begin	提供在呼叫中识别的结果或决议。
		OutcomesDetected.CharacterOffsets.End	
		ActionItemsDetected.CharacterOffsets.Begin	列出在呼叫中识别的所有操作内容。
		ActionItemsDetected.CharacterOffsets.End	
AccountId	字符串		AWS 账户 ID
类别	struct	MatchedCategories	列出匹配的类别。
		MatchedDetails	列出从音频流开始到检测到该类别中的匹配项的时间，以毫秒为单位。
频道	字符串	频道	指示语音通道。
参与者	数组	ParticipantRole	为每个音频通道提供发言人角色，CUSTOMER 或 AGENT。

列名称	数据类型	元素	定义
ConversationCharacteristics	struct	NonTalkTime	测量不含语音的时间段。使用此指标来查找长时间的静默，例如客户等待的时间过长。
		中断	测量一个参与者是否以及何时打断了另一个参与者的话。频繁中断可能与粗鲁或愤怒相关，也可能与一个或两个参与者的消极情绪相关。
		TotalConversationDurationMillis	对话的总时长。
		Sentiment.OverallSentiment.AGENT	代理的 OverallSentiment 标签。
		Sentiment.OverallSentiment.CUSTOMER	Customer 的 OverallSentiment 标签。
		Sentiment.SentimentByPeriod.QUARTER.AGENT	Agent 的各季情绪标签。
		Sentiment.SentimentByPeriod.QUARTER.CUSTOMER	Customer 的各季情绪标签。

列名称	数据类型	元素	定义
		TalkSpeed	测量两个参与者的说话速度。如果一个参与者说话太快，理解力就会受到影响。该指标以每分钟单词数来衡量。
		TalkTime	测量每个参与者在呼叫期间的说话时长，以毫秒为单位。使用此指标来帮助识别是否有一名参与者在呼叫中占据主导地位，或者对话是否平衡。
SessionId	字符串		呼叫的 SessionId
ContentMetadata	字符串		根据客户指定的配置来标记原始内容与编辑后内容的字段。

transcribe

列名称	数据类型	元素	定义
time	字符串		事件生成时间戳 ISO 8601。
detail-type	字符串		与 service-type 相关的功能类型。
service-type	字符串		AWS 服务的名称，VoiceAnalytics 或 CallAnalytics。

列名称	数据类型	元素	定义
mediaInsightsPipelineId	字符串		Amazon Chime SDK 媒体见解管线 ID。
metadata	字符串	voiceConnectorId	Amazon Chime 语音连接器 ID。
		callId	与使用情况相关的参与者的呼叫 ID。
		transactionId	呼叫的事务 ID。
		fromNumber	E.164 发起电话号码。
		toNumber	E.164 目标电话号码。
		方向	呼叫方向，Outbound 或 Inbound。
TranscriptEvent	struct	ResultId	Result 的唯一标识符。
		StartTime	Result 的开始时间，以毫秒为单位。
		EndTime	Result 的结束时间，以毫秒为单位。
		IsPartial	表明片段是否是完整的。如果 IsPartial 为 true，则片段不是完整的。否则，片段是完整的。
		ChannelId	与音频流相关的通道的 ID。

列名称	数据类型	元素	定义
		Alternatives.Entities	包含您的转录输出中识别为个人身份信息 (PII) 的实体。
		Alternatives.Items .Confidence	与您的转录中的单词或短语相关的置信度分数。置信度分数是介于 0 和 1 之间的值。值越大，指示识别内容与您的媒体中所说内容正确匹配的可能性越高。
		Alternatives.Items .Content	转录的单词或标点符号。
		Alternatives.Items .EndTime	转录后内容的结束时间，以毫秒为单位。
		Alternatives.Items .Speaker	如果启用了发言人分区，Speaker 就会标记指定内容的发言人。
		Alternatives.Items .Stable	如果启用了部分结果稳定，Stable 会指示指定项目是稳定的 (true)，还是在片段完成后可能发生变化 (false)。
		Alternatives.Items .StartTime	转录后内容的开始时间，以毫秒为单位。

列名称	数据类型	元素	定义
		Alternatives.Items.Type	识别内容的类型。 选项：PRONUNCIATION（口语词）和 PUNCTUATION。
		Alternatives.Items.VocabularyFilterMatch	指示指定内容是否与 您请求中所含词汇筛选器中的单词匹配。 如果为 true，则存在 词汇筛选匹配项。
		Alternatives.Transcript	包含转录后文本。

voice_analytics_status

列名称	数据类型	元素	定义
time	字符串		事件生成时间戳 ISO 8601。
detail-type	字符串		与 service-type 相关的 功能类型。
service-type	字符串		AWS 服务的名称，VoiceAnalytics 或 CallAnalytics。
源	字符串		生成事件的 AWS 服务。
account	字符串		AWS 账户 ID。
region	字符串		AWS 账户区域。
version	字符串		事件架构的版本。

列名称	数据类型	元素	定义
id	字符串		事件的唯一 ID
detail	struct	taskId	任务的唯一 ID。
		isCaller	指示参与者是否为呼叫者。
		streamStartTime	流的开始时间。
		transactionId	呼叫的事务 ID。
		voiceConnectorId	Amazon Chime 语音连接器 ID。
		callId	与使用情况相关的参与者的呼叫 ID。
		detailStatus	与 service-type 相关的详细功能类型。
		statusMessage	任务 ID 成功或失败的状态。
		mediaInsightsPipelineId	Amazon Chime SDK 媒体见解管线 ID。此字段仅在通过媒体管道 SDK 而不是语音 SDK 启动的发言人搜索任务时才会填充。
sourceArn	在其上运行任务的资源 ARN		

列名称	数据类型	元素	定义
		streamArn	为其运行任务的 Kinesis 视频流 ARN。此字段仅在通过媒体管道 SDK 而不是语音 SDK 启动的发言人搜索任务时才会填充。
		channelId	为其运行任务的 StreamArn 的通道。此字段仅在通过媒体管道 SDK 而不是语音 SDK 启动的发言人搜索任务时才会填充。
		speakerSearchDetails.voiceProfileId	已注册语音配置文件的 ID，其语音嵌入与呼叫中的发言人非常匹配。
		speakerSearchDetails.confidenceScore	介于 [0, 1] 之间的数字，其中，数字越大，意味着机器学习模型对语音配置文件匹配越有信心。

speaker_search_status

列名称	数据类型	元素	定义
time	字符串		事件生成时间戳 ISO 8601。
detail-type	字符串		与 service-type 相关的功能类型。

列名称	数据类型	元素	定义
service-type	字符串		AWS 服务的名称，VoiceAnalytics 或 CallAnalytics。
源	字符串		生成事件的 AWS 服务。
account	字符串		AWS 账户 ID。
region	字符串		AWS 账户区域。
version	字符串		事件架构的版本。
id	字符串		事件的唯一 ID
detail	struct	taskId	任务的唯一 ID。
		isCaller	指示参与者是否为呼叫者。
		transactionId	呼叫的事务 ID。如果任务源自通过语音连接器拨打的呼叫，则会填充此字段。
		voiceConnectorId	Amazon Chime 语音连接器 ID。如果任务源自通过语音连接器拨打的呼叫，则会填充此字段。
		mediaInsightsPipelineId	媒体见解管线 ID。此字段仅在通过媒体管道 SDK 而不是语音 SDK 启动的发言人搜索任务时才会填充。

列名称	数据类型	元素	定义
		sourceArn	在其上运行任务的资源 ARN。
		streamArn	为其运行任务的 Kinesis 视频流 ARN。此字段仅在通过媒体管道 SDK 而不是语音 SDK 启动的发言人搜索任务时才会填充。
		channelId	为其运行任务的 StreamArn 的通道。此字段仅在通过媒体管道 SDK 而不是语音 SDK 启动的发言人搜索任务时才会填充。
		participantRole	与 streamArn 中 channelId 相关的参与者角色。此字段仅在通过媒体管道 SDK 而不是语音 SDK 启动的发言人搜索任务时才会填充。
		detailStatus	与 service-type 相关的详细功能类型。
		statusMessage	任务 ID 的状态，成功或失败。
		speakerSearchDetails.voiceProfileId	已注册语音配置文件的 ID，其语音嵌入与呼叫中的发言人非常匹配。

列名称	数据类型	元素	定义
		speakerSearchDetails.confidenceScore	介于 [0, 1] 之间的数字，其中，数字越大，意味着机器学习模型对语音配置文件匹配越有信心。

voice_tone_analysis_status

列名称	数据类型	元素	定义
time	字符串		事件生成时间戳 ISO 8601。
detail-type	字符串		与 service-type 相关的功能类型。
service-type	字符串		AWS 服务的名称，VoiceAnalytics 或 CallAnalytics。
源	字符串		生成事件的 AWS 服务。
account	字符串		AWS 账户 ID。
region	字符串		AWS 账户区域。
version	字符串		事件架构的版本。
id	字符串		事件的唯一 ID
detail	struct	taskId	任务的唯一 ID。
		isCaller	指示参与者是否为呼叫者。

列名称	数据类型	元素	定义
		transactionId	呼叫的事务 ID。如果任务源自通过语音连接器拨打的呼叫，则会填充此字段。
		voiceConnectorId	Amazon Chime 语音连接器 ID。如果任务源自通过语音连接器拨打的呼叫，则会填充此字段。
		mediaInsightsPipelineId	媒体见解管线 ID。此字段仅在通过媒体管道 SDK 而不是语音 SDK 启动的发言人搜索任务时才会填充。
		sourceArn	在其上运行任务的资源 ARN。
		streamArn	为其运行任务的 Kinesis 视频流 ARN。此字段仅在通过媒体管道 SDK 而不是语音 SDK 启动的发言人搜索任务时才会填充。
		channelId	为其运行任务的 StreamArn 的通道。此字段仅在通过媒体管道 SDK 而不是语音 SDK 启动的发言人搜索任务时才会填充。

列名称	数据类型	元素	定义
		participantRole	与 streamArn 中 channelId 相关的参与者角色。此字段仅在通过媒体管道 SDK 而不是语音 SDK 启动的发言人搜索任务时才会填充。
		statusMessage	任务 ID 成功或失败的状态。
		voiceToneAnalysisDetails.startFragmentNumber	与 streamArn 相关的起始片段编号。
		voiceToneAnalysisDetails.currentAverageVoiceTone.startTime	ISO8601 格式的起始时间戳，表示当前平均情绪所依据的发言人呼叫音频。
		voiceToneAnalysisDetails.currentAverageVoiceTone.endTime	ISO8601 格式的结束时间戳，表示当前平均情绪所依据的发言人呼叫音频。
		voiceToneAnalysisDetails.currentAverageVoiceTone.beginOffsetMillis	当前平均情绪所依据的发言人呼叫音频起始片段的起始偏移量，以毫秒为单位。
		voiceToneAnalysisDetails.currentAverageVoiceTone.endOffsetMillis	当前平均情绪所依据的发言人呼叫音频起始片段的结束偏移量，以毫秒为单位。

列名称	数据类型	元素	定义
		voiceToneAnalysisDetails.currentAverageVoiceTone.voiceToneScore.positive	发言人积极情绪的概率可能性介于 [0, 1] 之间。
		voiceToneAnalysisDetails.currentAverageVoiceTone.voiceToneScore.negative	发言人消极情绪的概率可能性介于 [0, 1] 之间。
		voiceToneAnalysisDetails.currentAverageVoiceTone.voiceToneScore.neutral	发言人中性情绪的概率可能性介于 [0, 1] 之间。
		voiceToneAnalysisDetails.currentAverageVoiceTone.voiceToneLabel	最高概率具有平均语音音调分数的标签。
		voiceToneAnalysisDetails.overallAverageVoiceTone.startTime	ISO8601 格式的起始时间戳，表示总体平均情绪所依据的发言人呼叫音频。
		voiceToneAnalysisDetails.overallAverageVoiceTone.endTime	ISO8601 格式的结束时间戳，表示总体平均情绪所依据的发言人呼叫音频。
		voiceToneAnalysisDetails.overallAverageVoiceTone.beginOffsetMillis	总体平均情绪所依据的发言人呼叫音频起始片段的起始偏移量，以毫秒为单位。

列名称	数据类型	元素	定义
		voiceToneAnalysisDetails.overallAverageVoiceTone.endOffsetMillis	总体平均情绪所依据的发言人呼叫音频起始片段的结束偏移量，以毫秒为单位。
		voiceToneAnalysisDetails.overallAverageVoiceTone.voiceToneScore.positive	发言人积极情绪的概率可能性介于 [0, 1] 之间。
		voiceToneAnalysisDetails.overallAverageVoiceTone.voiceToneScore.negative	发言人消极情绪的概率可能性介于 [0, 1] 之间。
		voiceToneAnalysisDetails.overallAverageVoiceTone.voiceToneScore.neutral	发言人中性情绪的概率可能性介于 [0, 1] 之间。
		voiceToneAnalysisDetails.overallAverageVoiceTone.voiceToneLabel	情绪分数最高的情绪标签（积极、消极或中性）。

示例查询

使用这些示例查询来提取和组织您的 Amazon Chime SDK 呼叫分析 Glue 数据目录中的数据。

Note

有关连接 Amazon Athena 并查询您的 Glue 数据目录的信息，请参阅[使用 ODBC 连接 Amazon Athena](#)。

根据需要展开每个部分。

从 call_analytics_metadata 表中的元数据 (字符串数据类型) 中提取值

call_analytics_metadata 具有采用 JSON 字符串格式的 metadata 字段。使用 Athena 中的 [json_extract_scalar 函数](#) 来查询此字符串中的元素。

```
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"
```

在 call_analytics_metadata 表中查询 SIPRECMetadata 更新

call_analytics_metadata 字段的元数据字段采用 JSON 字符串格式。metadata 还有另一个名为 oneTimeMetadata 的嵌套对象，该对象包含原始 XML 格式和转换后的 JSON 格式的 SIPRec 元数据。使用 Athena 中的 json_extract_scalar 函数来查询此字符串中的元素。

```
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";
```

从 `call_analytics_recording_metadata` 表中的元数据 (字符串数据类型) 中提取值

`call_analytics_recording_metadata` 具有采用 JSON 字符串格式的元数据字段。使用 Athena 中的 [json_extract_scalar 函数](#) 来查询此字符串中的元素。

```
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"
```

从 `voice_analytics_status` 表中的详细信息 (结构数据类型) 中提取值

`voice_analytics_status` 具有采用 struct 数据类型的详细信息字段。以下示例展示了如何查询 struct 数据类型字段：

```
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"
```

连接 `voice_analytics_status` 表和 `call_analytics_metadata` 表

以下示例查询连接了 `call_analytics_metadata` 和 `voice_analytics_status`：

```
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')

```

从 `transcribe_call_analytics_post_call` 表中提取转录

`transcribe_call_analytics_post_call` 具有结构格式的转录字段，带有嵌套数组。使用以下查询来取消数组嵌套：

```

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)

```

连接 `transcribe_call_analytics_post_call` 和 `call_analytics_metadata` 表

以下查询连接了 `transcribe_call_analytics_post_call` 和 `call_analytics_metadata` 表：

```

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
```

查询语音增强呼叫记录的媒体对象 URL

以下示例查询连接了 Voice enhancement call recording URL :

```
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"
```

使用 Amazon Chime SDK 语音分析

Amazon Chime SDK 语音分析功能使您能够实现发言人搜索和语音音调分析。您可以使用发言人搜索来识别和注册新呼叫者，识别重复呼叫者，并为这些识别分配置信度分数。您可以使用语音音调分析来预测呼叫者的情绪为 `negative`、`neutral` 或 `positive`。

您可以将语音分析作为 Amazon Chime SDK 呼叫分析会话的可选组件来运行。

语音分析可与媒体见解管线或 Amazon Chime SDK 语音连接器呼叫结合使用。我们建议使用 [媒体管道 SDK](#) 并在媒体见解管线上调用任务，以便对任务及有关信息进行更精细的控制。

您可以使用语音连接器来确保向后兼容，但我们仅使用新功能来更新媒体见解管线 API。

有关创建和使用语音连接器的更多信息，请参阅 Amazon Chime SDK 管理员指南中的 [管理 Amazon Chime SDK 语音连接器](#)。

语音分析还提供：

- 异步任务处理。彼此独立运行任务。
- 在您处理见解时进行控制。

您可以通过调用 [StartSpeakerSearchTask](#) 和 [StartVoiceToneAnalysisTask](#) API 来启动语音分析。

以下主题介绍如何使用语音分析。

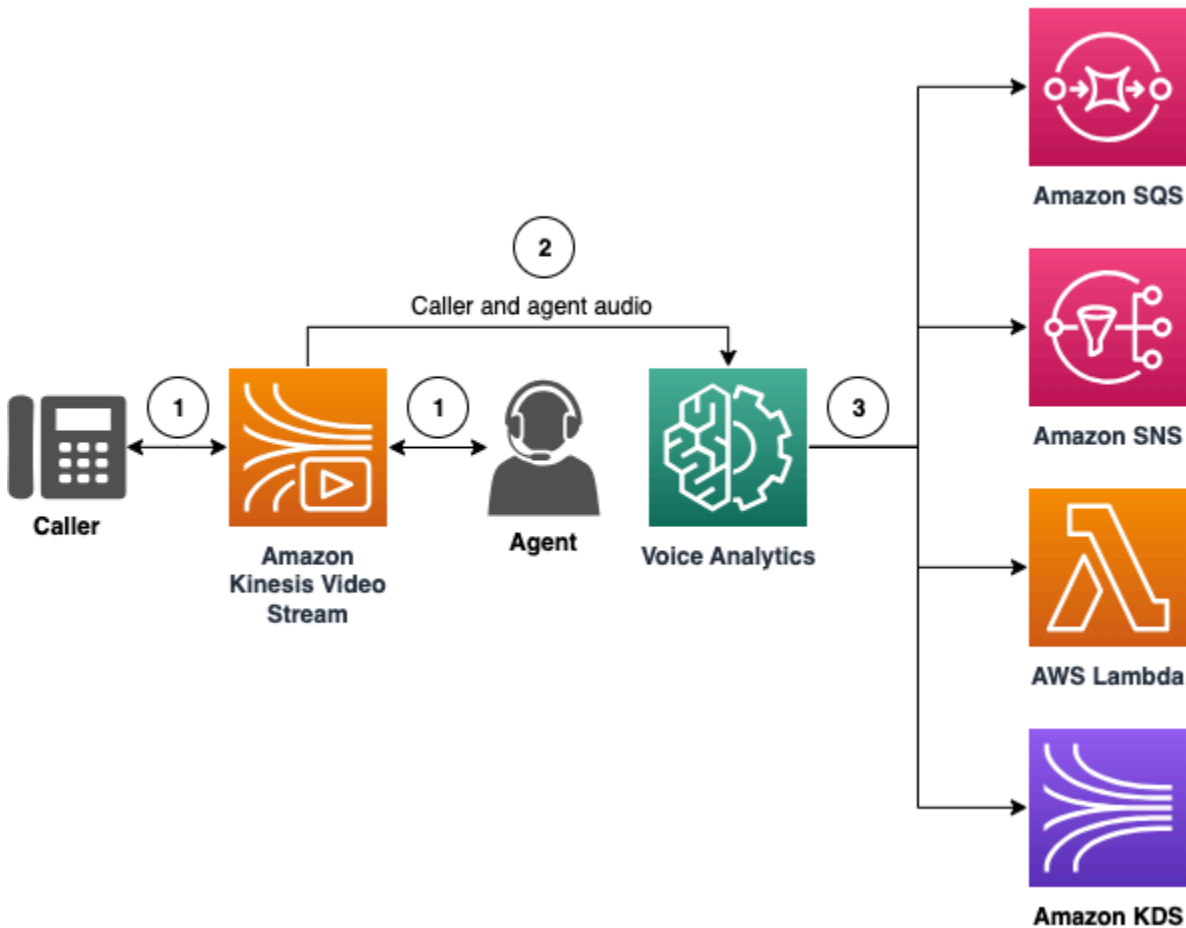
主题

- [语音分析架构](#)
- [发言人搜索工作流程示例](#)
- [语音音调分析工作流程示例](#)
- [轮询任务结果](#)
- [了解通知](#)
- [了解数据存储、选择退出和数据留存策略](#)
- [使用语音 API 来运行语音分析](#)

语音分析架构

本部分中的主题概述了 Amazon Chime SDK 语音分析架构，包括每项功能的数据流。

此图提供数据流如何通过语音分析的概要视图。



在下图中：

1. 音频流式传输到 Kinesis 视频流，供呼叫者和座席使用。您可以使用 Kinesis Video Streams 制作程序或 Amazon Chime SDK 语音连接器流式传输来做到这一点。有关更多信息，请参阅本指南中的 [基于机器学习的分析工作流程](#)，以及 Amazon Chime SDK 管理员指南中的 [Amazon Chime SDK 语音连接器媒体流式传输到 Kinesis](#)。
2. 在呼叫者同意后，应用程序或生成器会触发针对音频流的发言人搜索、语音音调分析，或两者兼而有之。
3. 在呼叫期间，语音分析发送通知至目标，可以是 Amazon Simple Queue Service (SQS)、Amazon Simple Notification Service (SNS)、AWS Lambda 或 Amazon Kinesis Data Streams。

此外，语音分析还提供以下工具来管理自己生成的数据。

语音配置文件

语音嵌入、嵌入的唯一 ID 及其过期日期的组合。出于安全考虑，也因为语音会随着时间变化，语音配置文件将在三年后过期。为避免重新创建语音配置文件，请调用 [UpdateVoiceProfileAPI](#)。有关过期日期的更多信息，请参阅 [数据留存](#)。

要注册语音嵌入或更新已注册的语音嵌入，您必须在通话结束后 24 小时内调用 [CreateVoiceProfile](#) 或 [UpdateVoiceProfileAPI](#)。

语音配置文件域

语音配置文件集合。

发言人搜索工作流程示例

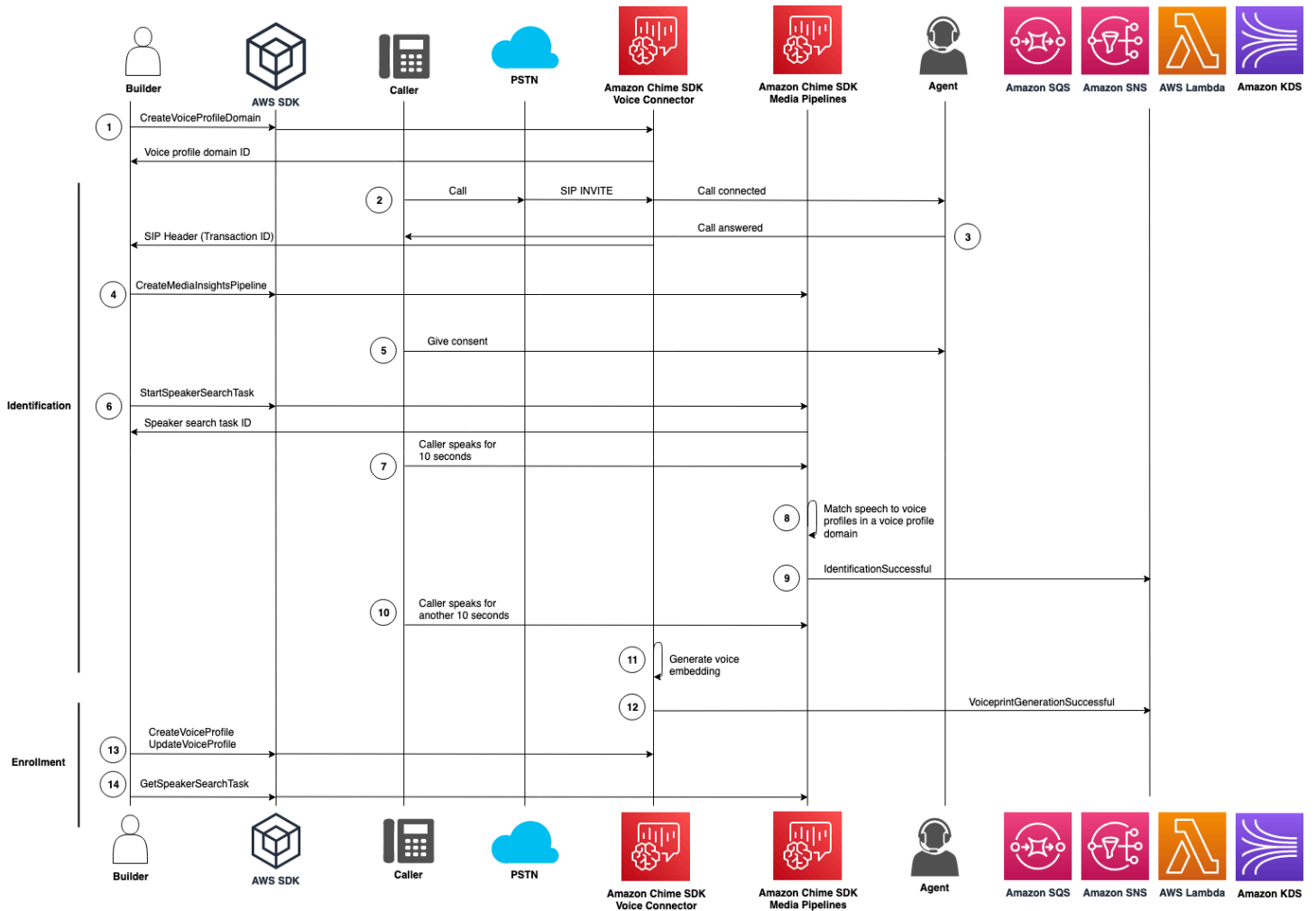
Important

发言人搜索功能涉及创建语音嵌入，该嵌入可用于比较呼叫者的语音与先前存储的语音数据。以数字声纹的形式收集、使用、存储和保留生物识别标识符和生物识别信息，可能需要以书面形式获得呼叫者的知情同意。根据各种州法律，包括伊利诺伊州、德克萨斯州、华盛顿州的生物识别法和其他州的隐私法，都要求获得此类同意。在使用发言人搜索功能之前，您必须根据适用法律以及监督您使用该功能的 [AWS 服务条款](#) 的要求，全面通知并获得一致同意。

下图显示了通过发言人搜索分析任务的数据流示例。图像下方的带编号的文字描述了该过程的每个步骤。

Note

该图假设您已经使用具有 VoiceAnalyticsProcessor 的呼叫分析配置来配置了 Amazon Chime SDK 语音连接器。有关更多信息，请参见 [录制 Voice Connector 通话](#)。



在下图中：

- 您或系统管理员创建一个语音配置文件域，用于存储语音嵌入和语音配置文件。有关创建语音配置文件域的更多信息，请参阅 Amazon Chime SDK 管理员指南中的 [创建语音配置文件域](#)。您也可以使用 [CreateVoiceProfileDomainAPI](#)。
- 呼叫者使用分配给 Amazon Chime SDK 语音连接器的电话号码拨号。或者，座席使用语音连接器号码执行出站呼叫。
- Amazon Chime SDK 语音连接器服务会创建一个事务 ID 并将其与呼叫关联。
- 假设您的应用订阅了 EventBridge 事件，则您的应用使用媒体见解管道配置和 Kinesis Video Stream ARN 调用语音连接器调用 [CreateMediaInsightsPipelineAPI](#)。

有关使用的更多信息 EventBridge，请参阅[基于机器学习的分析工作流程](#)。

- 您的应用程序（例如交互式语音应答系统）或座席会向呼叫者提供有关呼叫记录和使用语音嵌入进行语音分析的通知，并征得他们的同意才能参与。

- 一旦来电者表示同意，如果您有语音连接器和交易 ID，则您的应用程序或代理可以通过[语音 SDK](#) 调用 [StartSpeakerSearchTaskAPI](#)。或者，如果您有媒体见解渠道 ID 而不是交易 ID，则可以在[媒体管道 SDK](#) 中调用 [StartSpeakerSearchTaskAPI](#)。

一旦呼叫者表示同意，您的应用程序或座席就会调用 StartSpeakerSearchTask API。您必须将语音连接器 ID、事务 ID 和语音配置文件域 ID 传递给 API。返回发言人搜索任务 ID 以识别异步任务。

Note

在两个 SDK 之一中调用 StartSpeakerSearchTask API 之前，您必须根据法律和 [AWS 服务条款](#) 的要求提供所有必要的通知并获得必要的同意。

- 系统会累积 10 秒钟的呼叫者语音。呼叫者必须至少在这段时间内说话。系统不会捕获或分析静默。
- 媒体见解管线比较语音与域中的语音配置文件，并列出生信度最高的十个匹配项。如果找不到匹配项，语音连接器就会创建语音配置文件。
- 媒体见解管线服务向配置的通知目标发送通知事件。
- 呼叫者继续说话，并额外提供 10 秒钟的非静默语音。
- 媒体见解管线生成注册语音嵌入，您可用于创建语音配置文件或更新现有语音配置文件。
- 媒体见解管线向配置的通知目标发送 VoiceprintGenerationSuccessful 通知。
- 您的应用程序调用 [CreateVoiceProfile](#) 或 [UpdateVoiceProfile](#) API 来创建或更新配置文件。
- 您的应用程序会根据需要调用 [GetSpeakerSearchTask](#) API 以获取扬声器搜索任务的最新状态。

语音音调分析工作流程示例

Important

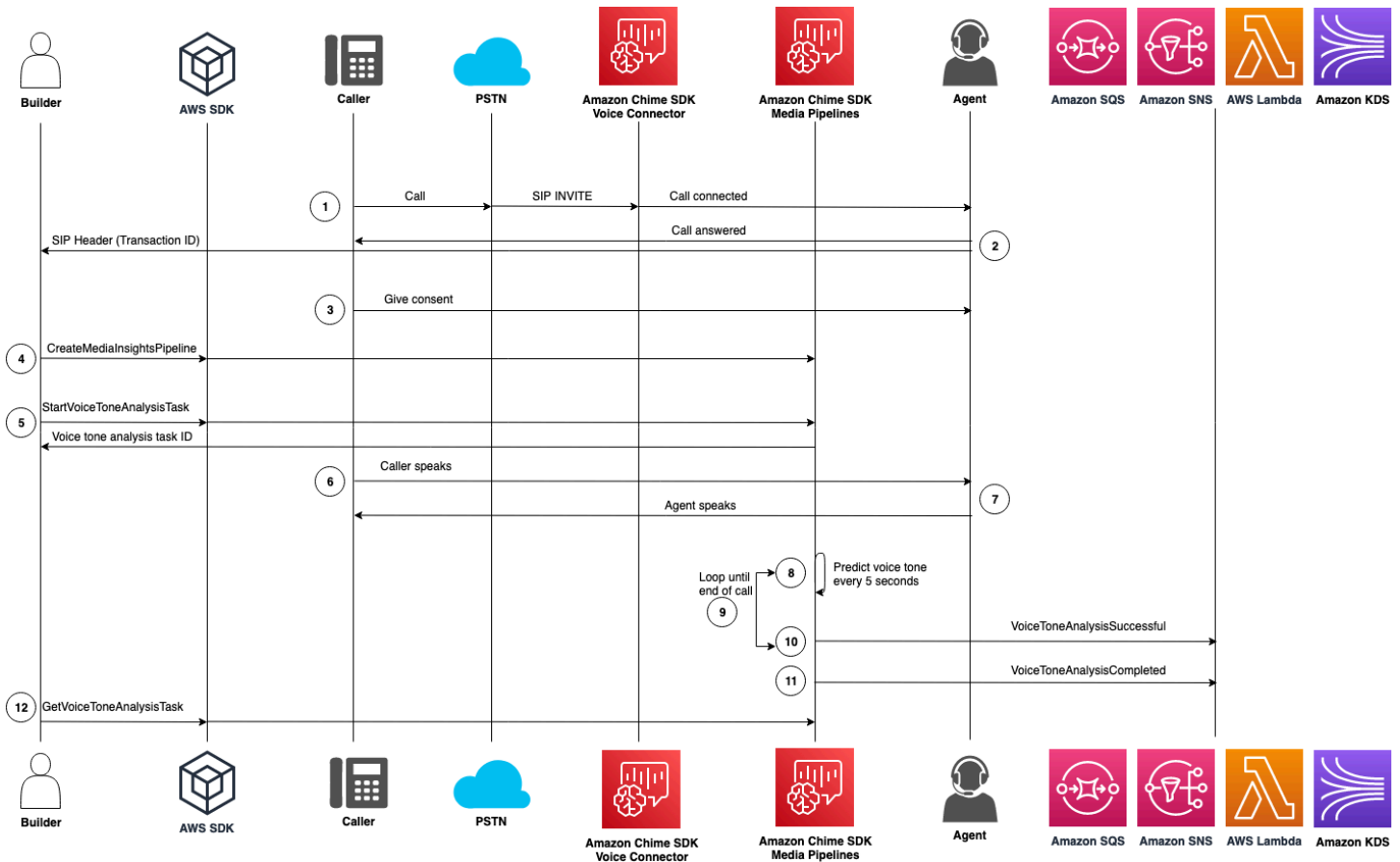
语音音调分析包括根据语言和音调信息来预测发言人的情绪。您不得以任何法律禁止的方式使用情绪分析，包括涉及对个人做出可能对此类个人产生法律或类似重大影响（例如与就业、住房、信用价值或财务报价等有关）的决策。

语音音调分析可以分析人们在呼叫中的语音并预测他们的情绪，可以是 positive、negative 或 neutral。

下图显示了语音音调分析的工作流程示例。图像下方的带编号的内容描述了该过程的每个步骤。

Note

该图假设您已经使用具有 VoiceAnalyticsProcessor 的呼叫分析配置来配置了 Amazon Chime SDK 语音连接器。有关更多信息，请参见 [录制 Voice Connector 通话](#)。



在下图中：

1. 呼叫者使用分配给 Amazon Chime SDK 语音连接器的电话号码拨号。或者，座席使用语音连接器号码执行出站呼叫。
2. 语音连接器服务创建事务 ID 并将其与呼叫关联。
3. 您的应用程序（例如交互式语音应答系统）或座席会向呼叫者提供有关呼叫记录和使用语音嵌入进行语音分析的通知，并征得他们的同意才能参与。
4. 假设您的应用订阅了 EventBridge 事件，则您的应用使用媒体见解管道配置和 Kinesis Video Stream ARN 调用语音连接器调用 [CreateMediaInsightsPipelineAPI](#)。

有关使用的更多信息 EventBridge，请参阅[基于机器学习的分析工作流程](#)。

- 一旦来电者表示同意，如果您有语音连接器和交易 ID，则您的应用程序或代理可以通过[语音 SDK](#) 调用 [StartSpeakerSearchTaskAPI](#)。或者，如果您有媒体见解渠道 ID 而不是交易 ID，则可以在[媒体管道 SDK](#) 中调用 [StartSpeakerSearchTaskAPI](#)。

一旦呼叫者表示同意，您的应用程序或座席就会调用 StartSpeakerSearchTask API。您必须将语音连接器 ID、事务 ID 和语音配置文件域 ID 传递给 API。返回发言人搜索任务 ID 以识别异步任务。

- 用户在整个呼叫过程中说话。
- 座席在整个呼叫过程中说话。
- 每隔 5 秒，媒体见解管线就会使用机器学习模型来分析最后 30 秒的话音，并预测该间隔内呼叫者的音调，以及从首次呼叫 StartVoiceToneAnalysisTask 时起的整个呼叫的音调。
- 媒体见解管线向配置的通知目标发送包含该信息的通知。您可以根据通知的流 ARN 和通道 ID 来识别通知。有关更多信息，请参阅本节后文中的[了解通知](#)。
- 重复步骤 9 和 10，直到呼叫结束。
- 呼叫结束时，媒体见解管线会发送最后一条通知，其中包含过去 30 秒的当前平均音调预测以及整个呼叫的平均音调。
- 您的应用程序根据需要调用 [GetVoiceToneAnalysisTaskAPI](#) 以获取语音分析任务的最新状态。

Note

GetVoiceToneAnalysisTask API 不流式传输音调数据。

Note

[GetVoiceToneAnalysisTaskAPI](#) 不返回语音数据。

轮询任务结果

Important

默认情况下，语音分析会在 7 天内提供结果，然后自动删除数据。如果您想长时间使用任务数据或遵守数据保留法，您必须存储您的任务数据。有关更多信息，请参阅本指南下文中的[数据留存](#)。

语音分析试图确保每个任务结果至少有一次交付。但是，网络问题可能会增加延迟。要解决潜在的问题，或者如果您更喜欢同步处理，您可以在[媒体管道 SDK](#) 或[语音 SDK](#) 中使用以下 API：

- [GetSpeakerSearchTask](#)
- [GetVoiceToneAnalysisTask](#)

Important

GetVoiceToneAnalysisTask API 仅返回任务的状态。它不返回任务结果。要查看结果，请使用 Amazon SQS、Amazon SNS 或 AWS Lambda 通知目标。

GetSpeakerSearchTask API 会同步获取任务 ID、延迟消息或无序到达消息的最新结果。但是，我们建议使用通知目标和异步处理。这样做会消耗更少的计算资源。

了解通知

当发言人搜索或语音音调分析任务开始、运行和完成时，语音分析会自动向目标发送事件。您可以使用通知目标来接收这些事件。如果您的工作流程或应用程序需要高可用性，我们建议使用多个通知目标。

此外，您必须使用具有访问您的通知目标所需策略的 IAM 角色。有关更多信息，请参见[使用通话分析资源访问角色](#)。

Note

对于亚马逊 SQS 和亚马逊 SNS，我们不支持队列。first-in-first-out 因此，消息可能会无序到达。我们建议您检查时间戳以根据需要排序消息，并保留消息在 Amazon DynamoDB 等数据存储中。您也可以使用[轮询任务结果](#) 中所述的“获取”API 来接收最新结果。

下表列出了事件及其相应的细节类型。

通知事件	细节类型
语音分析元数据	VoiceAnalyticsStatus
发言人搜索	SpeakerSearchStatus
语音音调分析	VoiceToneAnalysisStatus

适用于通知目标的 IAM 策略

您必须在呼叫分析配置中使用允许访问您的 Amazon SQS、Amazon SNS、AWS Lambda 或 Amazon KDS 通知目标的 IAM 角色策略。有关更多信息，请参阅本指南中的[使用通话分析资源访问角色](#)。

发言人搜索事件

发言人搜索事件具有 `SpeakerSearchStatus` 细节类型。

Amazon Chime SDK 语音连接器发送以下发言人搜索事件：

- 身份匹配
- 语音嵌入生成

事件可能具有以下状态：

- `IdentificationSuccessful` — 在给定语音配置文件域中成功识别出至少一个具有高置信度分数的匹配语音配置文件 ID。
- `IdentificationFailure` — 无法执行识别。原因：呼叫者至少有 10 秒钟没有说话，音频质量很差。
- `IdentificationNoMatchesFound` — 在给定语音配置文件域中找不到高置信度匹配项。呼叫者可能是新呼叫者，或者他们的语音可能已经改变。
- `VoiceprintGenerationSuccessful` — 系统使用 20 秒的非静默音频生成了语音嵌入。
- `VoiceprintGenerationFailure` — 系统无法生成语音嵌入。原因：呼叫者至少有 20 秒钟没有说话，音频质量很差。

身份匹配

为给定内容调用 [StartSpeakerSearchTask](#) API 后 `transactionId`，Voice Connector 服务会在非静默语音 10 秒钟后返回标识匹配通知。该服务会返回前 10 个匹配项，以及语音配置文件 ID 和范围为 [0, 1] 的置信度分数。置信度分数越高，呼叫中的发言人与语音配置文件 ID 匹配的可能性就越大。如果机器学习模型未找到匹配项，通知的 `detailStatus` 字段就会包含 `IdentificationNoMatchesFound`。

以下示例显示了成功匹配的通知。

```
{
  "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
    }
  }
}

```

语音嵌入生成

另外 10 秒钟的非静默话音后，语音连接器会向通知目标发送语音嵌入生成通知。您可以在语音配置文件中注册新的语音嵌入内容，也可以更新语音配置文件中已有的打印内容。

以下示例显示了成功匹配的通知，这意味着您可以更新相关的语音配置文件。

```
{
```

```

"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
}
}

```

语音音调分析事件

语音音调分析事件具有 `VoiceToneAnalysisStatus` 细节类型。分析可以返回以下状态：

- `VoiceToneAnalysisSuccessful` — 成功地将呼叫者和座席的声音分析为情绪（积极、消极或中性）概率。
- `VoiceToneAnalysisFailure` — 无法执行音调分析。如果呼叫者挂机 10 秒钟不说话，或者音频质量变得太差，就会发生这种情况。
- `VoiceToneAnalysisCompleted` — 成功地将用户和座席的声音分析为整个呼叫的情绪概率。这是最后一个事件，在语音音调分析完成时发送。

以下示例显示了典型的语音音调分析事件。

```

{
  "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"
}
```

呼叫后摘要事件

呼叫后摘要事件在呼叫结束 5 分钟后发送。这些摘要概述了整个呼叫过程中发生的发言人搜索任务。

以下示例显示了呼叫后摘要，其中包含最佳的语音配置文件匹配、已确认的发言人身份，以及在呼叫期间通过调用 `CreateVoiceProfile` 和 `UpdateVoiceProfile` API 而创建或更新的语音配置文件列表。

```
{
  "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",
        }
      ]
    }
  }
}
```

```
    ]  
    }  
  }  
}
```

语音分析示例 Lambda 函数

以下示例中的 Python 代码处理从语音连接器收到的通知。您可以添加代码至 AWS Lambda 函数。您还可以使用它来触发您的 Amazon SQS 队列、Amazon SNS 主题或 Amazon Kinesis 数据流。然后，您可以存储通知到 EventTable 中以备将来处理。有关确切的通知格式，请参阅 [了解通知](#)。

```
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

在调用 [StartSpeakerSearchTask](#) 或 [StartVoiceToneAnalysisAPI](#) 之前，您必须获得同意。我们建议您在获得同意之前，将活动保留在等候区域（例如 Amazon DynamoDB）。

了解数据存储、选择退出和数据留存策略

Amazon Chime SDK 使用语音数据来提供和改进发言人搜索服务。作为其中的一部分，我们使用注册音频（用于创建语音嵌入的记录片段）来训练我们的机器学习和人工智能模型。您可以选择不使用您的数据来训练模型，本部分中的主题将进行解释。

主题

- [用于发言人搜索的数据存储](#)
- [处理适用于发言人搜索的选择退出](#)
- [数据留存](#)

用于发言人搜索的数据存储

Amazon Chime SDK 存储以下用于发言人搜索的数据：

- 附加到语音配置文件中的语音嵌入，我们用于提供发言人搜索功能。
- 注册音频，用于为每个语音配置文件创建语音嵌入的记录语音片段。我们使用注册音频记录来执行以下操作：
 - 使发言人搜索模型保持最新状态，这是提供发言人搜索功能的关键部分。
 - 训练机器学习模型以开发和改进服务。使用注册音频进行训练是可选的，如下一部分所述，您可以选择退出策略，选择不使用此功能。

处理适用于发言人搜索的选择退出

您可以处理适用于最终用户和整个组织的选择退出。选择退出会带来以下影响：

- 在您选择退出后，语音分析将不会使用任何新的注册音频进行模型训练，也不会使用在您选择退出之前收集和存储的任何注册音频。

- 在您选择退出后，语音分析将存储和使用注册音频，以便提供发言人搜索服务。

Warning

以下选择退出操作是不可逆的。您无法恢复已删除的数据。

处理最终用户选择退出

当最终用户想要退出扬声器搜索时，请调用 [DeleteVoiceProfile](#) API。此操作会移除语音配置文件以及语音嵌入和注册音频。

要删除一组语音嵌入内容，请调用 [DeleteVoiceProfileDomain](#) API 删除该域名。此操作会删除域中的所有语音配置文件。

处理组织级别的选择退出

要处理整个组织的选择退出，请使用 AWS 组织选择退出策略。使用 `chimesdkvoiceanalytics` 服务名称。有关这些策略的信息，请参阅 AWS 组织用户指南中的 [AI 服务选择退出策略](#)。

Note

要使用选择退出策略，您的 AWS 账户必须由 AWS 组织集中管理。如果您尚未为自己的 AWS 账户创建组织，请参阅 AWS 组织用户指南中的 [创建和管理组织](#)。

数据留存

默认情况下，Amazon Chime SDK 语音分析会在 3 年后删除语音嵌入。我们之所以这样做，是因为人们的语音会随着时间变化，也是为了安全。您可以使用 [UpdateVoiceProfile](#) API 更新过期的语音嵌入内容。

来自其各自的结果 [StartSpeakerSearchTask](#) [GetSpeakerSearchTask](#) 和 [GetVoiceToneAnalytisTask](#) API 的结果也 [StartVoiceToneAnalysisTask](#) 将公布，最长为 7 天。

从 a 生成的语音嵌入 [StartSpeakerSearchTask](#) 可通过 [CreateVoiceProfile](#) 和 [UpdateVoiceProfile](#) API 保留 24 小时，之后它们将被删除且不可用。

要删除结果并处理您的客户撤回同意的情况，请参阅上一部分。

使用语音 API 来运行语音分析

为了向后兼容，您可以使用 Amazon Chime SDK 语音 API 来启动和管理语音分析。但是，只有用于语音分析的媒体见解管线 API 才提供新功能，因此我们强烈建议改用它们。

以下各部分说明了语音与媒体见解管线 API 之间的区别。

停止任务

如果您使用语音连接器启动语音分析任务，然后使用该 [UpdateMediaInsightsPipelineStatus](#) API 暂停管道，则这些任务将继续运行。要停止任务，必须调用 [StopSpeakerSearchTask](#) 和 [StopVoiceToneAnalysisTask](#) API。

了解通知差异

当您使用语音 API 来运行语音分析时，通知与媒体见解管线生成的通知会有差异。

- 语音分析就绪事件仅适用于使用语音 API 启动的任务。
- 您需要使用您通知中的字段 `voiceConnectorId`、`transactionId` 或 `callId` 来关联语音分析任务与呼叫。如果您使用媒体见解管线来运行语音分析，则可以使用字段 `mediaInsightsPipelineId` 和 `streamArn` 或 `channelId` 来关联任务与呼叫。

以下主题说明如何结合使用通知与语音 API。

主题

- [语音分析就绪事件](#)
- [发言人搜索事件](#)
- [语音音调分析事件](#)

语音分析就绪事件

语音分析就绪事件具有 `VoiceAnalyticsStatus` 细节类型。

您可以使用 Amazon Chime SDK 语音连接器来启动分析任务。当您收到语音分析就绪事件时，您可以触发呼叫的发言人搜索或语音分析任务，该任务由以下属性标识：

- `voiceConnectorId`
- `transactionId`

Note

仅当您的媒体见解管线配置启用了语音分析与语音连接器关联时，才会提供此通知。当客户调用 `CreateMediaInsightsPipeline` API 并通过媒体管道 SDK 启动发言人搜索任务或语音分析任务时，不提供此通知。

语音连接器返回的 SIP 标头包含 `transactionId`。如果您无权访问 SIP 标头，则 `AnalyticsReady` 通知事件还包含 `voiceConnectorId` 和 `transactionId`。这允许您以编程方式接收信息并调用 [StartSpeakerSearchTask](#)、或 [StartVoiceToneAnalysisTaskAPI](#)。

当语音分析准备好进行处理时，语音连接器会以 JSON 正文的形式向通知目标发送包含 `"detailStatus": "AnalyticsReady"` 的事件。如果您使用 Amazon SNS 或 Amazon SQS，则该正文显示在 Amazon SNS 或 Amazon SQS 有效负载的“记录”字段中。

下面的示例显示了典型 JSON 正文。

```
{
  "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"
    }
  }
}
```

此通知允许您在调用语音分析任务 API 之前触发对您应用程序的额外回调，并处理任何法律要求，例如通知和同意。

发言人搜索事件

发言人搜索事件具有 `SpeakerSearchStatus` 细节类型。

Amazon Chime SDK 语音连接器发送以下发言人搜索事件：

- 身份匹配
- 语音嵌入生成

事件可能具有以下状态：

- `IdentificationSuccessful` — 在给定语音配置文件域中成功识别出至少一个具有高置信度分数的匹配语音配置文件 ID。
- `IdentificationFailure` — 无法执行识别。原因：呼叫者至少有 10 秒钟没有说话，音频质量很差。
- `IdentificationNoMatchesFound` — 在给定语音配置文件域中找不到高置信度匹配项。呼叫者可能是新呼叫者，或者他们的语音可能已经改变。
- `VoiceprintGenerationSuccessful` — 系统使用 20 秒的非静默音频生成了语音嵌入。
- `VoiceprintGenerationFailure` — 系统无法生成语音嵌入。原因：呼叫者至少有 20 秒钟没有说话，音频质量很差。

身份匹配

为给定内容调用 [StartSpeakerSearchTask](#) API 后 `transactionId`，Voice Connector 服务会在非静默语音 10 秒钟后返回标识匹配通知。该服务会返回前 10 个匹配项，以及语音配置文件 ID 和范围为 [0, 1] 的置信度分数。置信度分数越高，呼叫中的发言人与语音配置文件 ID 匹配的可能性就越大。如果机器学习模型未找到匹配项，通知的 `detailStatus` 字段就会包含 `IdentificationNoMatchesFound`。

以下示例显示了成功匹配的通知。

```
{
  "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"
  }
}

```

语音嵌入生成

另外 10 秒钟的非静默语音后，语音连接器会向通知目标发送语音嵌入生成通知。您可以在语音配置文件中注册新的语音嵌入内容，也可以更新语音配置文件中已有的打印内容。

以下示例显示了成功匹配的通知，这意味着您可以更新相关的语音配置文件。

```

{
  "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": "abcdef1ghij2klmno3pqr"
  }
}
```

语音音调分析事件

语音音调分析事件具有 `VoiceToneAnalysisStatus` 细节类型。分析可以返回以下状态：

- `VoiceToneAnalysisSuccessful` — 成功地将呼叫者和座席的声音分析为情绪（积极、消极或中性）概率。
- `VoiceToneAnalysisFailure` — 无法执行音调分析。如果呼叫者挂机 10 秒钟不说话，或者音频质量变得太差，就会发生这种情况。
- `VoiceToneAnalysisCompleted` — 成功地将用户和座席的声音分析为整个呼叫的情绪概率。这是最后一个事件，在语音音调分析完成时发送。

以下示例显示了典型的语音音调分析事件。

```
{
  "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"
}

```

呼叫分析服务限额

本节中的表格列出了 Amazon Chime SDK 呼叫分析的服务限额。

有关呼叫分析区域的更多信息，请参阅本指南前文中的 [可用区](#)。

Amazon Chime SDK 呼叫分析和语音分析具有以下服务限额。

资源	默认限制	可调整
每个区域的媒体见解管道配置	100	是
每个区域的活跃媒体见解管道	20	是
每个区域的语音配置文件域	3	是
每个语音配置文件域的语音配置文件	20	是
每个区域的活跃发言人搜索任务	25	是
每个区域的主动语调分析任务	25	是

资源	默认限制	可调整
活跃的 Voice Connector 呼叫以及每个区域的语音分析	25	是
每个交易 ID 的每个 Voice Connector 呼叫的活动发言人搜索任务	1	否
每个交易 ID 的每个 Voice Connector 呼叫的活动语调分析任务	1	否
每个语音配置文件域的最大并发 API 调用次数	1	是
每个语音配置文件的最大并发 API 调用次数	1	是
每个发言人搜索任务的最大并发 API 调用次数	1	是
每个语音分析任务的最大并发 API 调用次数	1	是

有关 API 费率和限额的更多信息，请参阅《AWS 一般参考》中的 [Amazon Chime SDK 终端节点和限额](#)。

Note

如果您超过任何区域的限额，则会收到超出资源限制异常。您可以使用 AWS 控制台中的服务限额页面申请增加限额，也可以联系您的[客户支持代表](#)。

一些呼叫分析 API 会与其他 AWS 服务创建资源和 API 请求。这些额外费用将计入您账户的限额。如果您申请通话分析配额或 transactions-per-second 增加配额，则还必须申请增加其他 AWS 服务的配额。否则，您的请求可能会受到限制并失败。

使用安卓版 Amazon Chime SDK 客户端库

目前，您可以在 GitHub 上找到安卓版 Amazon Chime SDK 客户端库。前往 <https://github.com/aws/amazon-chime-sdk-android>。

使用 iOS 版 Amazon Chime SDK 客户端库

目前，您可以在 GitHub 上找到 iOS 版 Amazon Chime SDK 客户端库。转到 <https://github.com/aws/amazon-chime-sdk-ios>。

使用 Amazon Chime 软件开发工具包客户端库用于 JavaScript

本指南提供了 Amazon Chime SDK 客户端库的概念性概述 JavaScript，以及关键服务器和客户端组件的示例代码。

主题

- [Amazon Chime SDK 应用程序的组件](#)
- [重要概念](#)
- [服务架构](#)
- [Web 应用程序架构](#)
- [服务器应用程序架构](#)
- [Amazon Chime SDK 媒体控制面板](#)
- [Amazon Chime SDK 媒体数据面板](#)
- [Web 应用程序组件架构](#)
- [构建服务器应用程序](#)
- [构建客户端应用程序](#)
- [将背景筛选器集成至客户端应用程序](#)

Amazon Chime SDK 应用程序的组件

要将实时音频、视频和屏幕共享功能嵌入到您的 Amazon Chime SDK 应用程序中，您可以使用以下组件：

- Amazon Chime SDK 客户端库 JavaScript，即你集成到浏览器或 Electron 网络应用程序中的客户端 SDK。你可以通过添加适用于 [JavaScript NPM 包的 Amazon Chime 软件开发工具包作为依赖项](#)来做到这一点。此套餐利用 [MediaDevices](#)和 [WebRTC](#) API 加入会议、交换音频、视频以及与其他与会者共享内容。它为您提供了管理不同类型媒体的控制界面，并能够将这些资源绑定到应用程序的用户界面。
- AWS 软件开发工具包，Amazon Chime SDK API，您的服务器应用程序使用它来对来自您的网络应用程序的会议请求进行身份验证和授权。该 AWS 软件开发工具包为您提供诸如 [chime:CreateMeeting](#) 和 [chime:CreateAttendee](#) 之类的 API 操作，用于创建和管理会议和与会者资源。

与任何其他 AWS 资源一样，亚马逊身份和访问管理 (IAM) 服务配置对这些操作的访问权限。该 AWS 软件开发工具包有[多种编程语言版本](#)，可减少从服务器应用程序调用 AWS SDK Chime API 的复杂性。如果您的应用程序当前未使用服务器应用程序，则可以从 [demos/serverless](#) 文件夹中包含的 AWS CloudFormation 模板开始。该演示向您展示了如何使用 S AWS DK Chime API 构建 AWS Lambda 基于服务器的无服务器应用程序。

- Amazon Chime SDK 媒体服务提供 Amazon Chime SDK 客户端库中的音频、视频和信号，JavaScript 用于连接会议。媒体服务可在全球范围内使用，支持音频混合、视频转发和使用 TURN 中继的 NAT 遍历。Amazon Chime 服务团队可部署、监控和管理这些服务。媒体服务托管在单个 IP 地址范围 (99.77.128.0/18) 中，并使用端口 TCP/443 和 UDP/3478 简化 IT 管理员的防火墙配置。最后，这些服务利用 [AWS 全球云基础架构](#)。

重要概念

要完全了解如何创建和管理会议和用户，您需要了解以下概念：

[会议](#) — 多方媒体会话。每个会议都具有唯一的会议标识符。您可以在其中一个支持的 AWS 区域创建会议。创建会议时，将返回媒体 URL 列表。这些是加入会议所需数据的关键部分，您需要将这些数据传播给所有尝试加入会议的用户。

[与会者](#) — 试图加入多方媒体会话的用户。每个与会者都有一个唯一的标识符，一个可以传入该标识符以将与与会者映射到开发人员系统中用户的外部用户标识符，以及一个授予其会议访问权限的签名加入令牌。

[MeetingSession](#) 和 [\(DefaultMeetingSession\)](#) — Amazon Chime 软件开发工具包客户端库的根对象 JavaScript，它代表每位用户在会议中的会话。Web 应用程序首先实例化 MeetingSession 并使用正确的会议和与会者信息对其进行配置。

[MeetingSessionConfiguration](#) — 存储加入会议会话所需的会议和与会者数据。该数据是服务器应用程序发出的 CreateMeeting 和 CreateAttendee API 调用的响应。服务器应用程序将这些数据传递给 Web 应用程序，Web 应用程序使用它来实例化 MeetingSession

[DeviceController](#)(DefaultDeviceController)-用于枚举用户系统上可用的音频和视频设备列表。您还可以在会议期间使用设备控制器切换活动设备。

[AudioVideoFacade](#) (DefaultAudioVideoFacade) — 为会议提供动力的关键接口。它提供开始、控制和结束会议的 API。它还提供 API 用于通过跟踪用户加入或离开、静音或取消静音、积极发言或连接不畅等情况，侦听驱动用户体验发生变化的关键事件（例如与会者名单）。您还可以使用这些 API 将音频控制 HTML 元素绑定到会议的音频输出，然后通过选定音频输出设备进行播放。

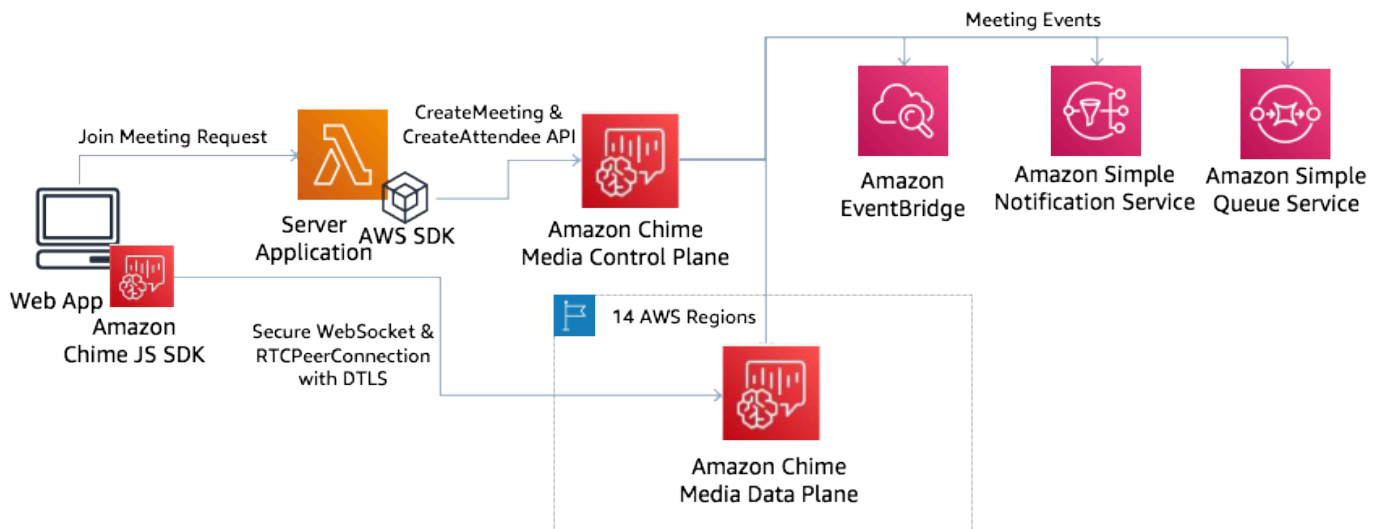
[ActiveSpeakerDetectorFacade\(DefaultActiveSpeakerDetector\)](#) — 订阅活跃演讲者活动的 API。定期返回一段时间内按麦克风音量排序的与会者列表。您可以根据需要覆盖和调整当前发言人策略。

[ContentShareController\(DefaultContentShareController\)](#) — 启动/停止和暂停/取消暂停内容共享的 API。它还提供用于侦听生命周期事件的 API 以跟踪内容共享状态。

[Logger \(ConsoleLogger\)](#) — 该接口用于利用控制台日志，或传入记录器对象来覆盖当前的日志实现并从 Amazon Chime SDK 获取不同级别的日志。

服务架构

此高级架构图显示了中列出的组件如何与其他 AWS 服务 [重要概念](#) 交互和协作：



Web 应用程序架构

您可以从内容分发网络提供您的 Web 应用程序，并当用户在浏览器中导航至 URL 时加载该应用程序。您也可以将其封装在平台自带的 Electron 应用程序中，由用户安装在他们的设备上。

要加入新的或现有会议，Web 应用程序会向服务器应用程序发出 REST 请求。通常，这些请求会携带授权令牌或 Cookie，您的应用程序将其用于其他 API 请求。您还可以将 Web 客户端设计为向服务器发送区域提示，服务器在提供 **MediaRegion** 参数给 [chime 时可以使用该提示:CreateMeeting](#)。您的 Web 应用程序可以通过向 <https://nearest-media-region.l.chime.aws> 终端节点发出 HTTP GET 请求来确定最近的媒体服务区域。

服务器应用程序架构

当服务器收到来自客户端的请求时，它首先确保用户有权限开始或加入会议。服务器使用所选语言的嵌入式 AWS SDK 向全局媒体控制平面发出 [chime: CreateMeeting](#) 和 [chime: CreateAttendee](#) API 调用。这样做旨在受支持的 AWS 区域之一创建会议和与会者。要提出这些请求，该服务需要相应的 IAM 用户或角色。反过来，IAM 用户和角色需要 [AmazonChime 软件开发工具包策略](#)。

Amazon Chime SDK 媒体控制面板

Amazon Chime SDK 媒体控制平面是全球性的，由 us-east-1 托管，它提供了 [chime: CreateMeeting and chime: CreateAttendee](#) 用于在数据平面上创建和管理会议和与会者资源 CreateAttendee 的 API。它会验证凭证并确保会话在请求区域的数据面板被引导。

控制平面还会触发通知机制的 [Amazon Chime SDK 事件](#)，例如亚马逊、亚马逊简单队列服务 (SQS) 或 EventBridge 亚马逊简单通知服务 (SNS) Simple Notification Service。AWS 持续监控服务，它们会随着负载的增加而自动扩展。这些 API 旨在仅接受不透明的用户标识符而非用户数据，因此它们符合数据主权要求。

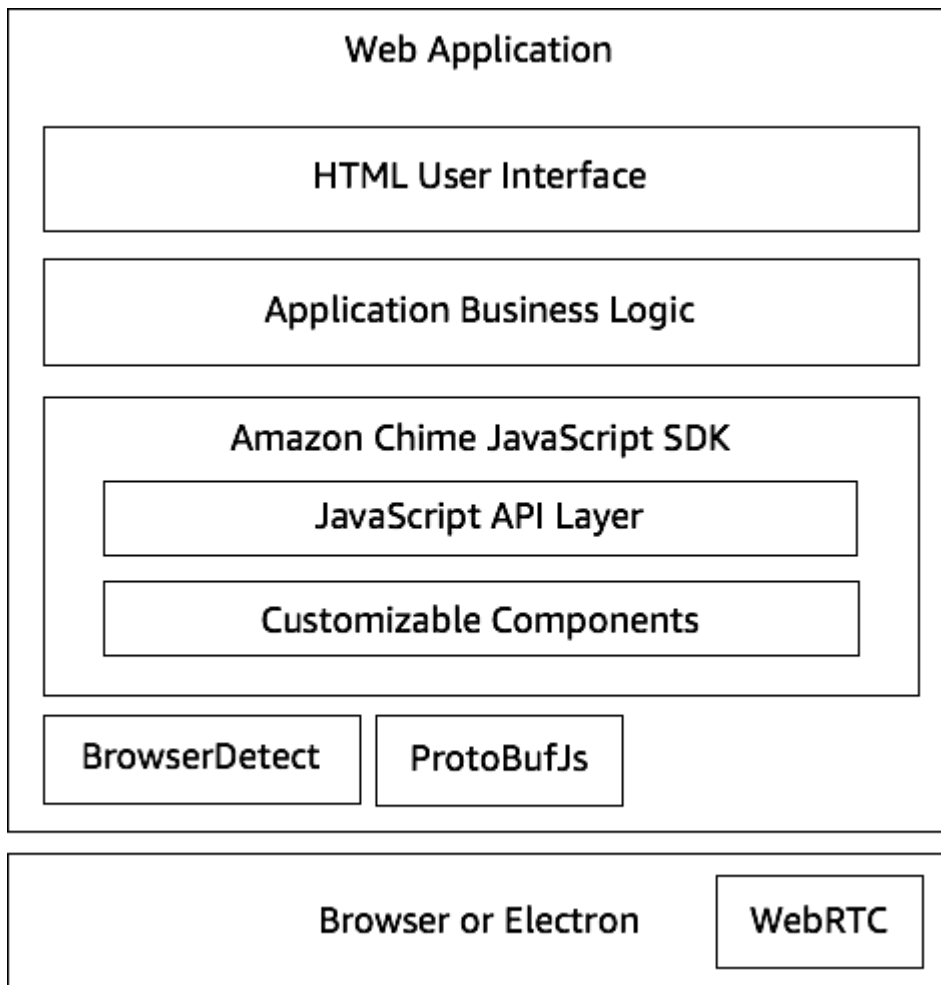
Amazon Chime SDK 媒体数据面板

您可以使用任何控制平面区域在所有 AWS 区域创建会议。媒体数据平面在所有 AWS 地区都可用。它包括音频混合服务、视频转发服务、TURN 服务和会话初始协议 (SIP) 互操作性服务。这些服务受到持续监控，旨在随着负载的增加而自动扩展。要了解更多信息，请参阅 [Amazon Chime SDK 媒体区域](#)。

有关区域和可用区的现有列表，请参阅 [区域和可用区](#)。

Web 应用程序组件架构

显示 Amazon Chime SDK Web 客户端应用程序的架构的图表。



Web 应用程序通常由应用程序业务逻辑层提供支持的 HTML 和 CSS 用户界面层组成。你可以用纯 HTML 来构建 Web 应用程序 JavaScript，也可以使用 React 和 Angular 等用户界面框架。

Web 应用程序的业务逻辑层通过一组 JavaScript API 与 Amazon Chime 软件开发工具包客户端库 JavaScript 进行交互。[DefaultMeetingSession](#) 是 SDK 的根对象。在构建服务器应用程序时，您可以使用 [MeetingSessionConfiguration](#) 会议和与会者信息对其进行初始化并加入会议。[DefaultMeetingSession](#) 还公开了 [AudioVideoFacade](#)，它使业务逻辑层能够采取行动，并注册在会话底层状态发生变化时更新用户界面的回调。

适用于 Amazon Chime SDK 的客户端库 JavaScript 是开源的，具有一组可自定义的组件，您可以根据需要覆盖这些组件。默认实现允许您构建完整的统一通信应用程序，例如我们演示 MeetingV2 应用程序。适用的 Amazon Chime 软件开发工具包客户端库 JavaScript 依赖于另外两个库：

- [Browser-Detect](#) 用于识别浏览器类型和功能。
- [ProtoBufJs](#) 对加入媒体会话所需的信号命令和响应进行编码和解码。

Amazon Chime SDK 还依赖浏览器或 Electron 应用程序为音频视频会话提供设备管理 API 和 WebRTC 实现。

Amazon Chime SDK 的源代码客户端库 JavaScript 已在 TypeScript，但您可以使用编译器将其编译为 JavaScript。然后，您可以使用诸如 Webpack 等模块捆绑器将其捆绑。作为最佳实践，请从 NPM 注册表中安装适用的 Amazon Chime SDK 客户端库，然后在 CommonJS 环境中使用它。AWS 还提供了一个汇总脚本，用于将 Amazon Chime SDK 捆绑到缩小的 JS 文件中，以便您想将其 [作为脚本](#) 标签直接包含在 HTML 中。

构建服务器应用程序

以下部分中的信息说明了如何构建 Amazon Chime SDK 服务器应用程序。每个部分都根据需要提供了示例代码，您可以调整该代码以满足您的需求。

主题

- [创建 IAM 用户或角色](#)
- [配置 AWS 软件开发工具包以调用 API](#)
- [创建会议](#)
- [创建与会者](#)
- [向客户端发送响应](#)

创建 IAM 用户或角色

您可以将用户创建为 IAM 用户或适合您的使用案例的角色。然后，您可以为其分配以下策略。这样可以确保您拥有服务器应用程序中嵌入的 AWS SDK 的必要权限。反过来，这允许您对会议和与会者资源执行生命周期操作。

```
// 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": "*"
}
}}
```

配置 AWS 软件开发工具包以调用 API

此代码示例向您展示如何向 AWS SDK 传递凭证以及如何设置区域和终端节点。

```
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');
```

创建会议

[CreateMeeting](#) API 调用接受必需的参数 `ClientRequestToken`，该参数允许开发人员在唯一性上下文中进行传递。它还接受可选参数，例如代表要为会议选择的媒体服务数据面板区域的 `MediaRegion`、用于传递不透明标识符来代表会议主机的 `MeetingHostId`（如果适用），以及用于接收会议生命周期事件的 `NotificationsConfiguration`。默认情况下，Amazon 会 `EventBridge` 交付事件。或者，您也可以通过在 `NotificationsConfiguration` 中传递 SQS 队列 ARN 或 SNS 主题 ARN 来接收事件。API 会返回一个会议对象，该对象包含唯一的 `MeetingId`、`MediaRegion` 和带有一组媒体 URL 的 `MediaPlacement` 对象。

```
meeting = await chime.createMeeting({
  ClientRequestToken: clientRequestToken,
  MediaRegion: mediaRegion,
  MeetingHostId: meetingHostId,
  NotificationsConfiguration: {
    SqsQueueArn: sqsQueueArn,
    SnsTopicArn: snsTopicArn
  }
});
```

```
    }  
  }).promise();
```

创建与会者

创建会议后，您可以创建代表每个尝试加入媒体会话的用户的与会者资源。该 [CreateAttendee](#) API 采用以下内容：

- 要向其中添加用户的会议的 MeetingId。
- ExternalUserId，可以是身份系统中任何不透明用户标识符。

例如，如果您使用 Active Directory (AD)，则可以是 AD 中用户的 Object ID。ExternalUserId 很有用，因为从客户端 SDK 收到与会者事件时，它会传回至客户端应用程序。这使客户端应用程序了解谁加入或离开会议，并从服务器应用程序中检索有关该用户的其他信息，例如显示名称、电子邮件或图片。

调用 CreateAttendee API 会生成一个 Attendee 对象。该对象包含由服务生成的唯一 AttendeeId 值、传入 ExternalUserId 的唯一值，以及允许 JoinToken 与会者在会议期间或在 [DeleteAttendee](#) API 删除与会者之前访问会议的签名。

```
attendee = await chime.createAttendee({  
  MeetingId: meeting.MeetingId,  
  ExternalUserId: externalUserId,  
}).promise();
```

向客户端发送响应

创建会议和与会者资源后，服务器应用程序应对会议和与会者对象进行编码并将其发送回客户端应用程序。客户需要这些信息来引导 Amazon Chime SDK 客户端库，并使与会者能够通过基 JavaScript 于 Web 或 Electron 的应用程序成功加入会议。

构建客户端应用程序

要构建客户端应用程序，请按照 [Amazon Chime JavaScript 软件开发工具包 API 概述](#) 中列出的步骤进行操作。GitHub 概述提供所需的示例代码。

将背景筛选器集成至客户端应用程序

本部分介绍如何使用背景模糊 2.0 和背景替换 2.0 以编程方式筛选视频背景。要将背景筛选器添加至视频流，创建包含 VideoFxConfig 对象的 VideoFxProcessor。然后将该处理器插入到 VideoTransformDevice。

后台过滤器处理器使用 TensorFlow 精简版机器学习模型 JavaScript Web Workers，并 WebAssembly 对视频流中每帧的背景应用过滤器。这些资产是在您创建 VideoFxProcessor 时的运行过程中下载。

[上的浏览器演示应用程序 GitHub](#)使用了新的背景模糊和替换滤镜。若要试用，使用 `npm run start` 启动演示，加入会议，然后点击摄像头启用视频。打开应用筛选器菜单



然后选择背景模糊 2.0或背景替换 2.0选项。

主题

- [关于使用背景筛选器](#)
- [使用 content-security 策略](#)
- [将背景筛选器添加到您的应用程序](#)
- [背景筛选器示例](#)

关于使用背景筛选器

背景筛选器可以是 CPU 密集型，也可以是 GPU 密集型。某些移动设备和低规格笔记本电脑或台式机可能无法同时运行筛选器和多个视频流。

SIMD 支持

在支持单指令、多数据 (SIMD) 的环境中，背景筛选器的效率更高。启用 SIMD 后，在给定的复杂度级别下，筛选器占用的 CPU 资源会更少。运行不支持 SIMD 的浏览器的低功耗设备可能无法运行背景筛选器。

WebGL2 支持

`VideoFxProcessor` 对象需要支持 WebGL2 的浏览器才能访问客户端设备上的 GPU。

内容分发和带宽

Amazon 内容分发网络会在运行时加载背景过滤 `machine-learning-model` 文件。这可提供低延迟的全局分发，而无需提供一整套文件作为应用程序的一部分。但是，加载模型文件可能会增加应用程序某些部分的延迟。为了帮助减轻这种影响，浏览器会无限期缓存模型文件。该缓存使后续加载速度大大加快。作为最佳实践，请检查受支持的浏览器，然后在用户可能没有注意到任何延迟时创建背景筛选器资源。例如，您可以在用户在大厅等候或使用设备选择器时下载模型文件。

您的应用程序必须连接以下应用：

- Amazon Chime SDK 媒体服务。
- 亚马逊 CloudFront 通过 HTTPS (端口 443) 。

所有请求都发送到 `sdkassets.chime.aws` 的子域。无法访问内容分发网络或[内容安全策略](#)中未包含正确域名的应用程序将不能通过支持检查且无法使用筛选器。

有关的 IP 地址范围 CloudFront 的更多信息，请参阅 Amazon CloudFront 开发者指南中的 [CloudFront 边缘服务器的位置和 IP 地址范围](#)。

浏览器兼容性

下表列出了支持背景筛选器的浏览器和版本。

浏览器	支持的最低版本
Firefox	76+
基于 Chromium 的浏览器和环境，包括 Edge 和 Electron	78+

浏览器	支持的最低版本
Android Chrome	110+
macOS 上的 Safari	16.3+
iOS (iPhone、 iPad) 上的 Safari	16.x
iOS 上的 Chrome	110.0.0.x.x
iOS 上的 Firefox (iPhone iPad)	16.x

VideoFxProcessor 对象的 3.14 版本支持安卓系统。要获得 3.14 之前版本的 Android 设备支持，请使用 BackgroundBlurVideoFrameProcessor 和 BackgroundReplacementVideoFrameProcessor 对象。有关使用它们的更多信息，请参阅上的 [backgroundfilter_video_processor](#) 页面。GitHub

使用 content-security 策略

现代 Web 应用程序使用内容安全策略保护用户免受某些类别的攻击。使用 VideoFxProcessor 必须包含以下策略指令的应用程序。这些指令允许 Amazon Chime SDK 在运行时访问其所需的资源。

主题

- [所需的内容安全策略指令](#)
- [使用跨源开启器策略](#)

所需的内容安全策略指令

您必须使用以下内容安全策略指令。

- `script-src`: 添加 `blob: https://*.sdkassets.chime.aws` 以加载视频处理代码，并 `wasm-unsafe-eval` 允许运行它。
- `script-src-elem`: 添加 `blob: https://*.sdkassets.chime.aws` 以从来源加载视频处理代码。
- `worker-src`: 添加 `blob: https://*.sdkassets.chime.aws` 以 JavaScript 跨源加载工作程序。

如果您忽略这些条目中的任何一个，或者您使用 HTTP 标头和 `http-equiv` 元标签指定策略，但意外通过交叉排除其中任何一个条目，则背景筛选器将无法初始化。该筛选器似乎不受支持，或者创建了一个无操作视频帧处理器。您将在浏览器控制台中看到错误，例如：

```
Refused to connect to
'https://static.sdkassets.chime.aws/bgblur/workers/worker.js...'
because it violates the document's content security policy.
```

必需的脚本策略指令

要正常运行，该 `VideoFxProcessor` 类必须在运行时从 Amazon 内容分发网络加载 JavaScript 类。这些类别使用 WebGL2 实现视频的后期处理。要允许应用程序获取和运行这些类别，必须包含以下指令：

- `script-src 'self' blob: https://*.sdkassets.chime.aws`
- `script-src-elem 'self' blob: https://*.sdkassets.chime.aws`

Note

要在 Safari 和 Firefox 上获得全面支持，必须使用 `script-src` 和 `script-src-elem` 指令。

工作线程策略指令

将 JavaScript 类作为 `blob VideoFxProcessor` 加载以运行 Web 工作线程。该线程使用机器学习模型处理视频。要授予应用程序获取和使用此工作线程的访问权限，请包含以下指令：

```
worker-src 'self' blob: https://*.sdkassets.chime.aws
```

WebAssembly 政策

从亚马逊拥有的同一个内容交付网络 `VideoFxProcessor` 加载一个 WebAssembly (WASM) 模块。在 Chrome 95 及更高版本中，编译后的 WASM 模块无法跨越多个模块边界。要允许获取和实例化这些模块，请在 `script-src` 指令中包含 `'wasm-unsafe-eval'`。

有关内容安全政策文档的更多信息 WebAssembly，请参阅上的 [WebAssembly 内容安全政策](#) GitHub。

(可选) 背景图片策略

要将动态加载的背景图像与背景替换筛选器一起使用，VideoFxProcessor 必须能够访问该图像。为此，请在托管图像的域中加入 connect-src 指令。

内容安全策略示例

以下示例策略允许使用 VideoFxProcessor。connect-src 定义并不特定于 VideoFxProcessor。相反，它们与 Amazon Chime SDK 会议中的音频和视频有关。

```
<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>
```

内容安全策略错误

如果省略任何必需指令，则 VideoFxProcessor 不会实例化且将不受支持。在这种情况下，浏览器控制台中会出现以下 (或类似) 错误：

```
Refused to connect to
'https://static.sdkassets.chime.aws/ml_media_fx/otherassets/worker.js'
because it violates the document's content security policy.
```

使用跨源开启器策略

为了限制内存使用量，该模块倾向于使用 SharedArrayBuffer 进行处理。但是，这需要您仔细配置 Web 安全。在为应用程序 HTML 提供服务时，必须设置以下标头：

```
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Embedder-Policy: require-corp
```

服务器必须设置这些，因为它们没有等效元标签。如果您不设置这些标头，则背景筛选器可能会占用稍多的 RAM。

背景筛选器可以是 CPU 密集型，也可以是 GPU 密集型。某些移动设备和低规格笔记本电脑或台式机可能无法同时运行筛选器和多个视频流。

将背景筛选器添加到您的应用程序

添加背景筛选器的过程要遵循以下主要步骤：

- 检查是否有支持的浏览器。
- 使用您要使用的配置创建 VideoFxConfig 对象。
- 使用配置对象创建 VideoFxProcessor 对象。
- 将 VideoFxProcessor 对象包含在 VideoTransformDevice。
- 使用 VideoTransformDevice 启动视频输入。

Note

要完成这些步骤，您必须先执行以下操作：

- 创建 Logger。
- 选择 MediaDeviceInfo 类视频设备。
- 成功加入 MeetingSession。

以下部分介绍如何完成流程的每个步骤。

主题

- [在提供筛选器之前先检查是否受支持](#)
- [创建 VideoFxConfig 对象](#)
- [创建 VideoFxProcessor 对象](#)
- [配置 VideoFxProcessor 对象](#)
- [创建 VideoTransformDevice 对象](#)
- [开始视频输入](#)
- [微调资源利用率](#)

在提供筛选器之前先检查是否受支持

Amazon Chime SDK 提供了一种异步静态方法，用于检查受支持的浏览器并尝试下载所需资产。但是，它不检查设备性能。作为最佳实践，在提供筛选器之前，始终确保用户的浏览器和设备可以支持筛选器。

```
import {
  VideoFxProcessor
} from 'amazon-chime-sdk-js';

if (!await VideoFxProcessor.isSupported(logger)) {
  // logger is optional for isSupported
}
```

创建 VideoFxConfig 对象

您可以在同一个对象中为 `backgroundBlur` 和 `backgroundReplacement` 定义配置。但是，不能同时为两个筛选器将 `isEnabled` 设置为 `true`。这是一个无效配置。

`VideoFxConfig` 类本身不进行验证。验证将在下一步骤中进行。

以下示例显示有效的 `VideoFxConfig`。

```
const videoFxConfig: VideoFxConfig = {
  backgroundBlur: {
    isEnabled: false,
    strength: 'medium'
  },
  backgroundReplacement: {
    isEnabled: false,
    backgroundImageURL: 'space.jpg',
    defaultColor: undefined,
  }
}
```

下表列出了您可以在 `VideoFxConfig` 对象中指定的 `VideoFxProcessor` 属性。

背景模糊筛选器属性

属性	Type	描述
<code>isEnabled</code>	<code>boolean</code>	当为 <code>true</code> 时，筛选器会模糊背景。
<code>strength</code>	<code>string</code>	确定模糊程度。有效值： <code>low</code> <code>medium</code> <code>high</code> 。

背景替换筛选器属性

属性	Type	描述
<code>isEnabled</code>	<code>boolean</code>	当为 <code>true</code> 时，筛选器会替换背景。
<code>backgroundImageURL</code>	<code>string</code>	背景图片的 URL。筛选器根据当前屏幕的尺寸动态调整图像大小。您可以使用诸如 <code>https://...</code> 等字符串或诸如 <code>data:image/jpeg;base64</code> 等 URL。
<code>defaultColor</code>	<code>string</code>	十六进制颜色字符串，例如 <code>000000</code> 或 <code>FFFFFF</code> ，或 <code>black</code> 或 <code>white</code> 等字符串。如果未指定图像 URL，则处理器会使用 <code>defaultColor</code> 作为背景。如果未指定 <code>defaultColor</code> ，则处理器默认为黑色。

创建 VideoFxProcessor 对象

创建 `VideoFxProcessor` 对象时，AWS 服务器会下载运行时资产，或者浏览器缓存加载资产。如果网络或 CSP 配置阻止访问资产，则 `VideoFx.create` 操作会引发异常。`VideoFxProcessor` 由此产生的处理器被配置为无操作处理器，这不会影响视频流。

```
let videoFxProcessor: VideoFxProcessor | undefined = undefined;
try {
  videoFxProcessor = await VideoFxProcessor.create(logger, videoFxConfig);
} catch (error) {
  logger.warn(error.toString());
}
```

`VideoFxProcessor.create` 还会尝试从 `backgroundReplacement.backgroundImageURL` 中加载图像。如果图像加载失败，处理器会抛出异常。处理器还会出于其他原因抛出异常，例如配置无效、浏览器不受支持或硬件性能不足。

配置 VideoFxProcessor 对象

下表列出可以配置的 `VideoFxProcessor` 属性。表下方的示例显示典型的运行时系统配置。

背景模糊

背景模糊具有以下属性：

属性	Type	描述
<code>isEnabled</code>	<code>boolean</code>	当为 <code>true</code> 时，筛选器会模糊背景。
<code>strength</code>	<code>string</code>	确定模糊程度。有效值： <code>low</code> <code>medium</code> <code>high</code> 。

背景替换

背景替换使用以下参数：

属性	Type	描述
<code>isEnabled</code>	<code>boolean</code>	当为 <code>true</code> 时，筛选器会替换背景。
<code>backgroundImageURL</code>	<code>string</code>	背景图片的 URL。筛选器根据当前屏幕的尺寸动态调整

属性	Type	描述
		图像大小。您可以使用诸如 <code>https://...</code> 等字符串或诸如 <code>data:image/jpeg;base64</code> 等 URL。
<code>defaultColor</code>	<code>string</code>	十六进制颜色字符串，例如 <code>000000</code> 或 <code>FFFFFF</code> ，或 <code>black</code> 或 <code>white</code> 等字符串。如果未指定图像 URL，则处理器会使用 <code>defaultColor</code> 作为背景。如果未指定 <code>defaultColor</code> ，则处理器默认为黑色。

运行时更改配置

您可以使用 `videoFxProcessor.setEffectConfig` 参数在运行时更改 `VideoFxProcessor` 配置。以下示例显示如何启用背景替换和禁用背景模糊。

Note

一次只能指定一种类型的背景替换。指定 `backgroundImageURL` 或 `defaultColor` 的值，但不能同时指定两者。

```
videoFxConfig.backgroundBlur.isEnabled = false;
videoFxConfig.backgroundReplacement.isEnabled = true;
try {
  await videoFxProcessor.setEffectConfig(videoFxConfig);
} catch(error) {
  logger.error(error.toString())
}
```

如果 `setEffectConfig` 抛出异常，则之前的配置仍然有效。`setEffectConfig` 会在与导致 `VideoFxProcessor.create` 抛出异常的类似条件下抛出异常。

以下示例显示如何在视频运行时更改背景图像。

```
videoFxConfig.backgroundReplacement.backgroundImageUrl = "https://my-domain.com/my-  
other-image.jpg";  
try {  
  await videoFxProcessor.setEffectConfig(videoFxConfig);  
} catch(error) {  
  logger.error(error.toString())  
}
```

创建 VideoTransformDevice 对象

以下示例显示如何创建包含 VideoFxProcessor 的 VideoTransformDevice 对象。

```
// assuming that logger and videoInputDevice have already been set  
const videoTransformDevice = new DefaultVideoTransformDevice(  
  logger,  
  videoInputDevice,  
  [videoFxProcessor]  
);
```

开始视频输入

以下示例演示如何使用 VideoTransformDevice 对象开始视频输入。

```
// assuming that meetingSession has already been created  
await meetingSession.audioVideo.startVideoInput(videoTransformDevice);  
meetingSession.audioVideo.start();  
meetingSession.audioVideo.startLocalVideoTile();
```

微调资源利用率

创建 VideoFxProcessor 时，您可以提供可选 processingBudgetPerFrame 参数并控制筛选器占用 CPU 和 GPU 量。

```
let videoFxProcessor: VideoFxProcessor | undefined = undefined;  
const processingBudgetPerFrame = 50;  
try {  
  videoFxProcessor = await VideoFxProcessor.create(logger, videoFxConfig,  
    processingBudgetPerFrame);  
} catch (error) {  
  logger.warn(error.toString());  
}
```

VideoFxProcessor 需要时间处理帧。时间长短取决于设备、浏览器以及浏览器或设备上正在运行的其他程序。处理器使用预算的概念确定处理和渲染每帧所用的时间。

处理时间以毫秒为单位。举个如何使用预算的示例，1 秒有 1000 毫秒。将每秒 15 帧的视频捕获作为目标，总预算为 $1000 \text{ 毫秒} / 15 \text{ fps} = 66 \text{ 毫秒}$ 。通过在 `processingBudgetPerFrame` 参数中提供值 50，您可以将预算设置为其中的 50% 或 33ms，如上例所示。

然后 VideoFxProcessor 尝试在指定的预算范围内处理帧。如果处理超出预算，处理器会降低视觉质量使其保持在预算范围内。处理器继续将视觉质量降低到最低限度后，它会停止降低。这种处理持续时间是持续测量的，因此，如果有更多资源可用，例如关闭另一个应用程序并释放 CPU，处理器会再次提高视觉质量，直到达到预算或者达到最高的视觉质量。

如果您未向 `processingBudgetPerFrame` 提供值，则 VideoFxProcessor 默认为 50。

背景筛选器示例

以下示例显示如何实施筛选器

```
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);
}
```

使用适用于 Windows 的 Amazon Chime SDK 客户端库

目前，您可以在 GitHub 上找到以 C++ 编写的 Windows 版 Amazon Chime SDK 客户端库。转至 <https://github.com/aws/amazon-chime-sdk-cpp>。

文档历史记录

下表介绍了自 2019 年 9 月起对《Amazon Chime 开发者指南》的一些重要更改。如需有关此文档更新的通知，您可以订阅 RSS 源。

变更	说明	日期
Alexa 技能内通话已移除	由于亚马逊 Alexa 团队的更改，您无法再向 SIP 媒体应用程序添加 Alexa 呼叫。有关更多信息，请参阅 Alexa 智能属性 页面。	2024年4月1日
新的会议区域	开发者现在可以使用几个新的会议区域。有关更多信息，请参阅本指南中的 可用区域 ，以及《AWS General Reference》中的 Amazon Chime SDK 终端节点和限额 。	2023 年 9 月 25 日
语音增强	开发者现在可以启用通话录音并将录制的通话存储在 Amazon S3 存储桶中。有关更多信息，请参阅本指南中的 了解语音增强 。	2023 年 8 月 31 日
已更新的区域	使用 Amazon Chime SDK 的开发者现在可以使用更多区域。有关更多信息，请参阅 可用区域 。	2023 年 8 月 29 日
呼叫分析和语音分析	开发者现在可以在他们的解决方案中添加低代码分析和转录功能。有关更多信息，请参阅本指南中的 使用 Amazon Chime SDK 呼叫分析 。	2023 年 3 月 27 日

适用于 Windows 的客户端库	开发者现在可以使用采用 C++ 编写的、适用于 Windows 的 Amazon Chime SDK 客户端库。有关更多信息，请参阅本指南中的 适用于 Windows 的 Amazon Chime SDK 客户端库 。	2023 年 2 月 2 日
已更新的区域	使用 Amazon Chime SDK 的开发者现在可以使用更多区域。有关更多信息，请参阅 可用区域 。	2022 年 11 月 18 日
C++ 客户端库已开启 GitHub	使用 Amazon Chime SDK Meetings 的开发者现在可以在上与 C++ 信号客户端库集成。GitHub 有关更多信息，请参阅 与客户端库集成 。	2022 年 8 月 19 日
媒体管道	使用 Amazon Chime SDK 会议的开发者现在可以创建 Media Pipelines。反过来，媒体管道由媒体捕获管道、媒体串联管道和实时连接器管道组成。有关更多信息，请参阅 创建 Amazon Chime SDK 媒体管道 。	2022 年 8 月 18 日
弹性频道	使用 Amazon Chime SDK 消息传递的开发者现在可以在聊天解决方案中使用弹性频道。弹性频道最多可容纳 100 万用户。有关更多信息，请参阅 使用弹性频道托管实时事件 。	2022 年 8 月 12 日

紧急 911 地址验证	使用 Amazon Chime SDK 会议的开发者可以通过编程方式验证紧急呼叫的来源地址。有关更多信息，请参阅《Amazon Chime API 参考》中的 ValidateE911Address 以及《Amazon Chime SDK 管理指南》中的 验证紧急呼叫地址 。	2022 年 8 月 11 日
会议标签更新	开发者现在可以在 Chime 和 ChimeSDKMeetings 命名空间中使用会议标签。有关更多信息，请参阅 迁移到 Amazon Chime SDK 会议命名空间 。	2022 年 8 月 4 日
CallAndBridge 语音连接器和语音连接器组的操作	使用 Amazon Chime SDK 音频服务的开发者可以使用该 CallAndBridge 操作向配置为 Voice Connector 或 Voice Connector 组的 SIP 中继发出外拨呼叫。有关更多信息，请参阅 CallAndBridge 。	2022 年 7 月 14 日
AppKeys 和 TenantTid	使用 Amazon Chime SDK 会议的开发者现在可以使用 AppKeys 和 TenantIDs 控制其客户网络对 WebRTC 媒体会话的访问。有关更多信息，请参阅 使用 AppKeys 和租户 ID 。	2022 年 7 月 7 日

[Connect API](#)

使用 Amazon Chime SDK Messaging 的开发者现在可以使用 WebSockets 用来连接后端服务器并接收消息。AppInstanceUser 有关更多信息，请参阅[使用连接 API](#) 和[使用 WebSockets 接收消息](#)。

2022 年 6 月 6 日

[与会者能力](#)

开发者现在可以使用功能来控制与会者在 Amazon Chime SDK 会议期间对音频、视频和内容的访问权限。有关更多信息，请参阅https://docs.aws.amazon.com/chime-sdk/latest/APIReference/API_meeting-chime_AttendeeCapabilities.html。

2022 年 6 月 2 日

[亚马逊 CloudWatch 指标](#)

开发人员现在可以利用 Amazon Chime 软件开发工具包发布到的服务和使用指标。CloudWatch 这些指标使您能够使用 CloudWatch 图表和控制面板来监控您使用 Amazon Chime SDK 服务的情况。有关更多信息，请参阅[Amazon CloudWatch 指标](#)。

2022 年 6 月 1 日

[回声抑制](#)

开发者现在可以实现减少回声，这有助于防止回声（用户发言人发出的声音被麦克风拾取）循环回会议音频并使讨论陷入停顿。有关更多信息，请参阅[使用减少回声](#)。

2021 年 11 月 23 日

[通话录音](#)

开发者现在可以在 Amazon Chime SDK SIP 媒体应用程序调用的一条或多条路段上实现录音。有关更多信息，请参阅[使用通话录音](#)。此外，该 RecordAudio 操作现在还包
括新参数，包括 SilenceDurationInSeconds 和 RecordingTerminationUsed 。有关更多信息，请参阅[RecordAudio](#)。

2021 年 10 月 28 日

[背景模糊](#)

开发者现在可以为他们的 Amazon Chime SDK 应用程序添加背景模糊功能。有关更多信息，请参阅[使用背景模糊](#)。

2021 年 10 月 21 日

[已更新 IAM 策略](#)

开发者现在有了更新后的 IAM 策略，该策略支持对 Amazon Chime SDK 会议进行实时转录。有关更多信息，请参阅[使用 Chime SDK 策略创建 IAM 用户或角色](#)。

2021 年 9 月 22 日

[SIP 标头](#)

开发人员现在可以在其 AWS Lambda 函数中发送和接收用户对用户标头、转移标头和自定义 SIP 标头。有关更多信息，请参阅[使用 SIP 标头](#)。

2021 年 9 月 13 日

[Amazon Chime SDK 会议](#)

开发者现在可以使用 Amazon Chime SDK 实时转录了。有关更多信息，请参阅[使用 Amazon Chime SDK 实时转录](#)。

2021 年 8 月 11 日

Amazon Chime SDK 会议	开发者现在可以创建媒体管道。有关更多信息，请参阅 创建 Amazon Chime SDK 媒体捕获管道 。	2021 年 7 月 7 日
使用具有 AWS Lambda 功能的 SIP 媒体应用程序	标题已从“Lambda 函数 SDK”改为“Lambda 函数 SDK”，所有主题中的内容都经过了修改，CallAndBridge 部分已添加。有关更多信息，请参阅 使用 PSTN 音频服务 和 CallAndBridge 。	2021 年 6 月 17 日
Lambda 函数 SDK	开发者可以构建自定义 Lambda 函数，用于 Amazon Chime SDK 管理员创建的 Amazon Chime SDK SIP 媒体应用程序。有关更多信息，请参阅 Amazon Chime 开发者指南中的 使用 PSTN 音频服务 。	2020 年 11 月 17 日
JavaScript SDK	开发人员可以 JavaScript 用来构建 Amazon Chime 软件开发工具包应用程序。有关更多信息，请参阅 《亚马逊 Chime 开发者指南》JavaScript 中的使用亚马逊 Chime 软件开发工具包 。	2020 年 11 月 17 日

Android 和 iOS 客户端库	开发者可以在更短的时间内和更少的点击次数找到适用于 Android、iOS 和 Windows 的客户端库。有关更多信息，请参阅《Amazon Chime 开发者指南》中的 使用适用于 Android 的 Amazon Chime SDK 客户端库 和 使用适用于 iOS 的 Amazon Chime SDK 客户端库 。	2020 年 11 月 17 日
代理电话会话	开发者可以创建代理电话会话，以便与 Amazon Chime SDK Voice Connector 配合使用。有关更多信息，请参阅《 亚马逊 Chime 开发者指南 》JavaScript 中的 使用亚马逊 Chime 软件开发工具包 。	2020 年 4 月 7 日
Amazon Chime SDK 内容共享	Amazon Chime SDK 支持内容共享。有关更多信息，请参阅《Amazon Chime 开发者指南》中的 Amazon Chime SDK 架构 。	2020 年 3 月 31 日
适用于 Android 和 iOS 的 Amazon Chime SDK	适用于 Android 和 iOS 的 Amazon Chime SDK 已发布。有关更多信息，请参阅《Amazon Chime 开发者指南》中的 与客户端库集成 。	2020 年 3 月 24 日
Amazon Chime SDK	Amazon Chime SDK 已发布。有关更多信息，请参阅《Amazon Chime 开发者指南》中的 使用 Amazon Chime SDK 。	2019 年 11 月 20 日

[Amazon Chime 开发者指南](#)

《Amazon Chime 开发者指南》已发布。

2019 年 9 月 11 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。