

AmazonConnectによる自動電話通知（7.複数連絡先への電話通知〈構築⑥〉）

AWS



×

ZABBIX

×



AmazonConnectによる自動電話通知 （7.複数連絡先への電話通知〈構築⑥〉）

2021.11.12 2021.11.04

[【前回】 AmazonConnectによる自動電話通知（7.複数連絡先への電話通知〈構築⑤〉）](#)[【次回】 AmazonConnectによる自動電話通知（7.複数連絡先への電話通知〈構築⑦〉）](#)[【簡易版】 AmazonConnectによる自動電話通知（まとめ）](#)

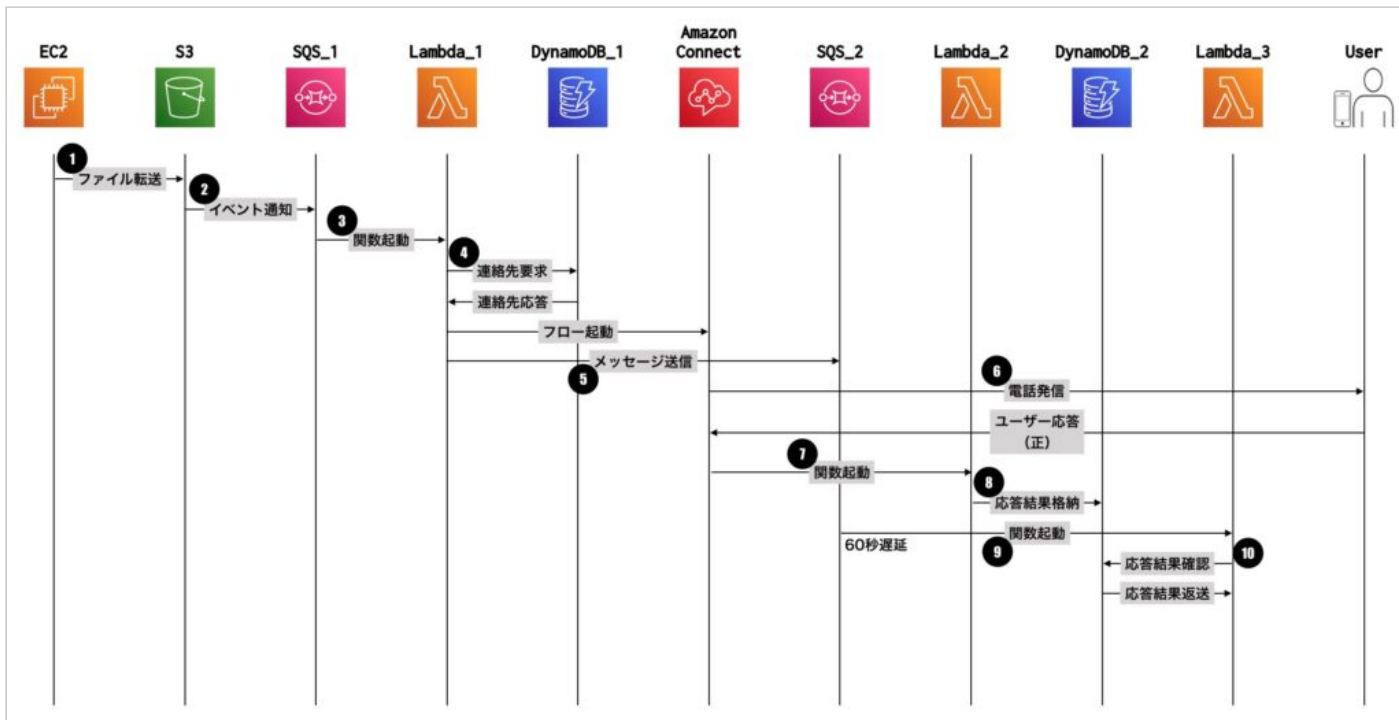
監視サーバーで障害を検知した際に、自動で電話通知できるようにしていきます。ネットワークエンジニアも利用することの多い監視サーバー(Zabbix)で障害検知し、AWS上のAmazonConnectを利用し自動電話を発信します。

今回は下記の条件を満たせるようにAWSの各サービスを利用して自動電話通知の仕組みを導入します。

- 複数の通知先を登録した連絡先リストを持たせる。
- 連絡先リストに優先度(通知順)を設定する。
- 優先度が高い人に最初に電話する。
- 応答が無かった場合、次の優先度の人に順番に電話する。
- 連絡先リストの最後まで電話しても応答が無かった場合、最初に戻って継続する。

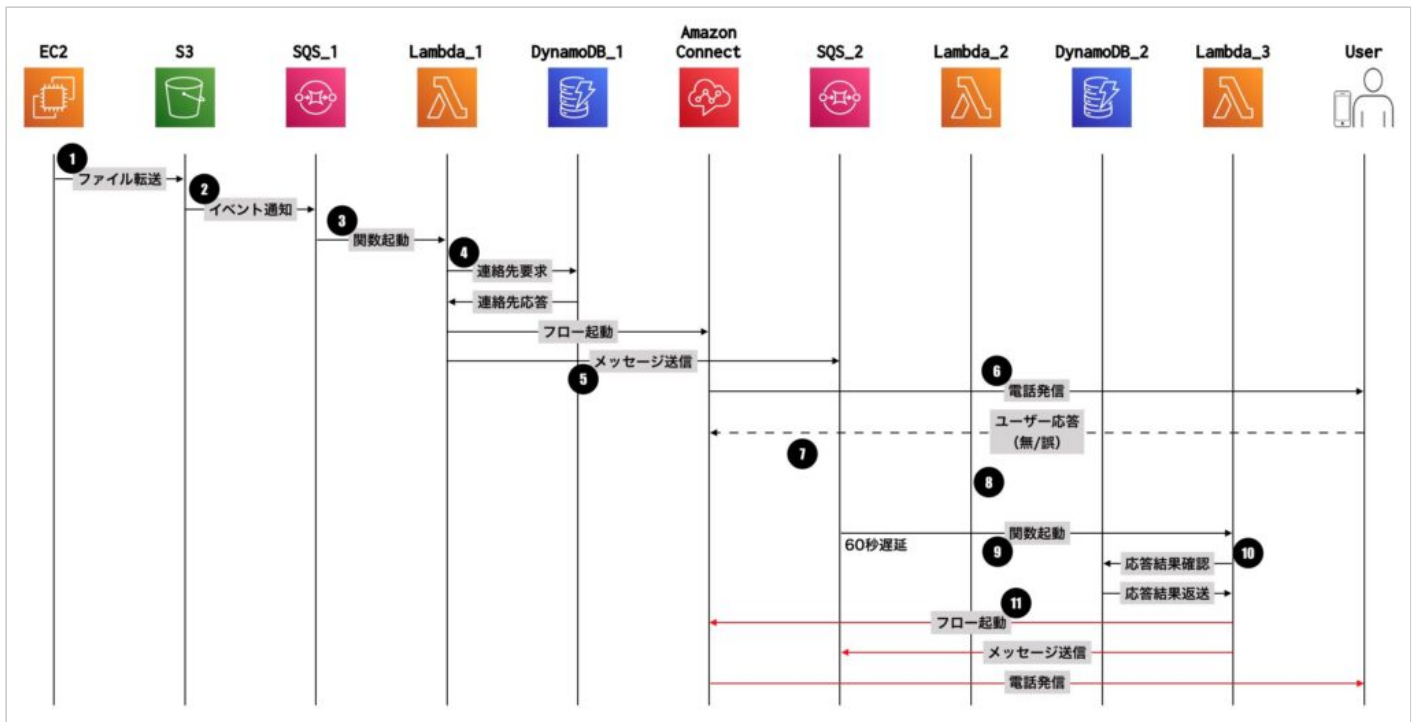
自動電話通知フロー

電話に応答した場合のフロー



1. EC2上の監視サーバーで障害を検知し、S3へトリガーファイルを格納
2. S3のイベント通知機能で、SQS_1にメッセージを送信
3. SQS_1をトリガーとして、Lambda_1を起動
4. Lambda_1がDynamoDB_1から連絡先を取得し、AmazonConnectを起動
5. Lambda_1がAmazonConnectを起動すると同時に、SQS_2へメッセージを送信
6. AmazonConnectがユーザーへ自動電話通知を実施
7. ユーザーが正常応答し、AmazonConnectがLambda_2を起動
8. Lambda_2が応答結果をDynamoDB_2に保存(応答OK)
9. 60秒後にSQS_2をトリガーとしてLambda_3を起動
10. Lambda_3がDynamoDB_2の応答結果を確認(正常応答しているため、何もせずに処理完了)

電話に応答しなかった場合のフロー



1. EC2上の監視サーバーで障害を検知し、S3へトリガーファイルを格納
2. S3のイベント通知機能で、SQS_1にメッセージを送信
3. SQS_1をトリガーとして、Lambda_1を起動
4. Lambda_1がDynamoDB_1から連絡先を取得し、AmazonConnectを起動
5. Lambda_1がAmazonConnectを起動すると同時に、SQS_2へメッセージを送信
6. AmazonConnectがユーザーへ自動電話通知を実施
7. ユーザーが正常応答せず、AmazonConnectがLambda_2を起動
8. Lambda_2が応答結果をDynamoDB_2に保存(応答NG)
9. 60秒後にSQS_2をトリガーとしてLambda_3を起動
10. Lambda_3がDynamoDB_2の応答結果を確認
11. 正常応答していないため、再度AmazonConnectを起動(以降、5から繰り返す)

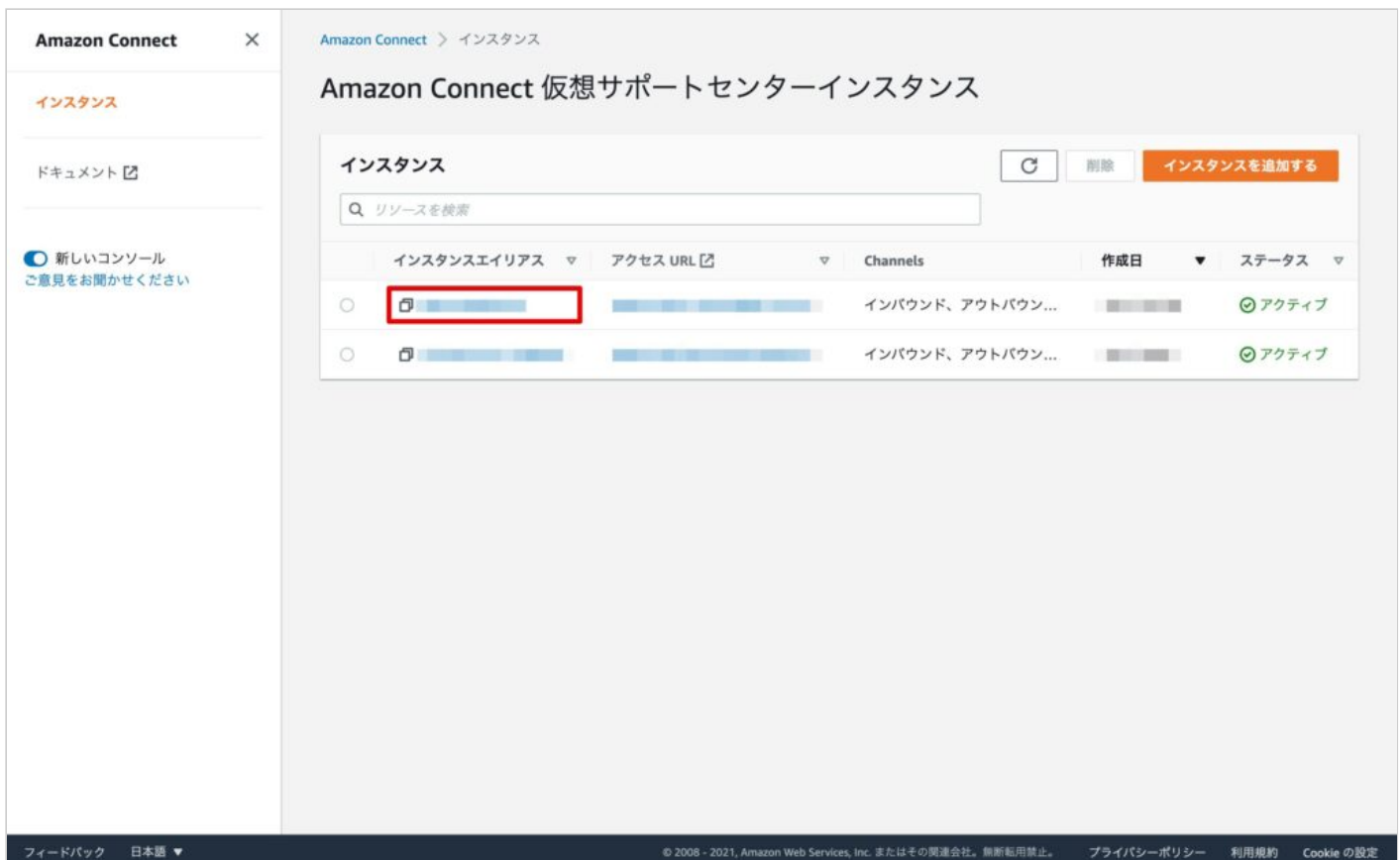
AmazonConnectの問い合わせフロー修正

構築③で作成した問い合わせフローを修正します。

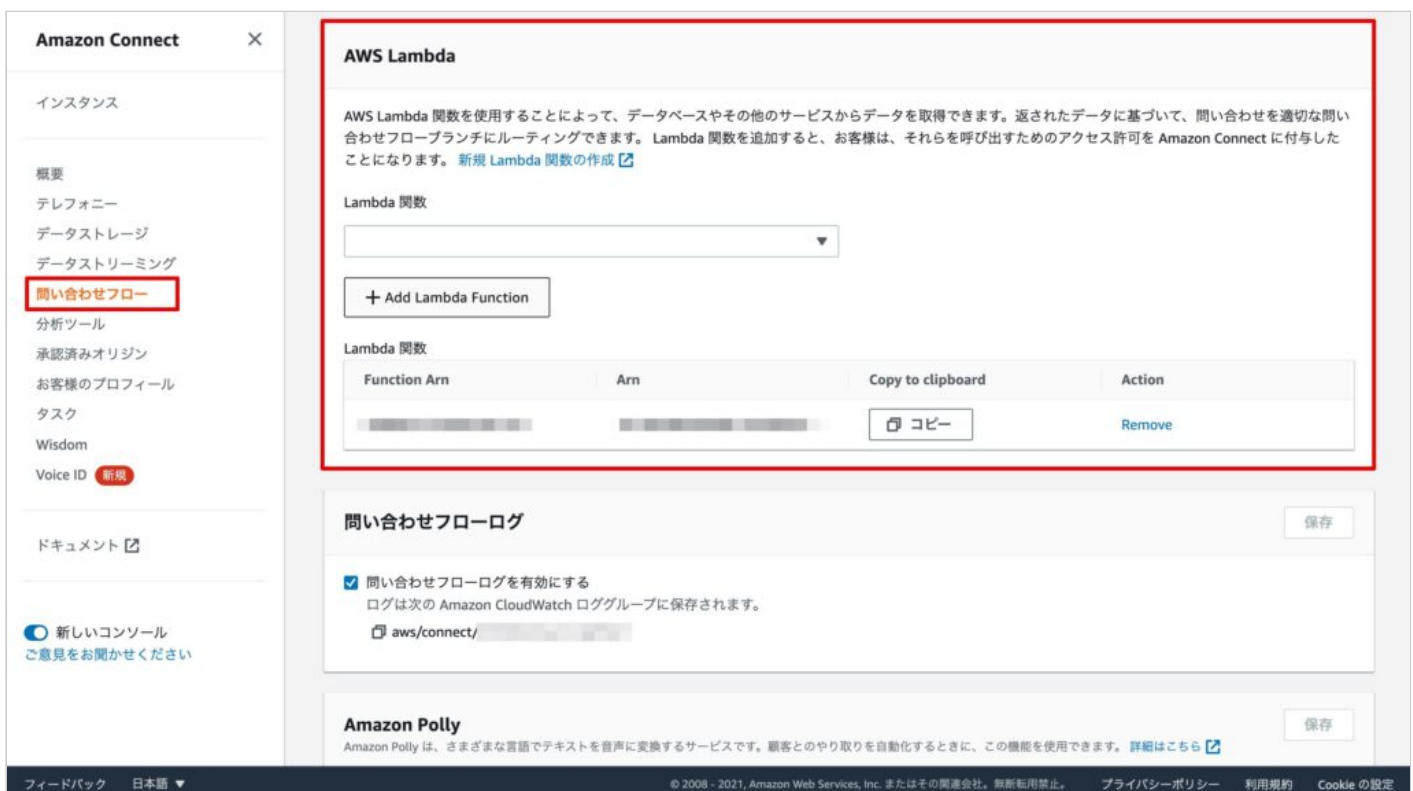
[AmazonConnectによる自動電話通知（7.複数連絡先への電話通知〈構築③〉）](#)

Lambdaの登録

AmazonConnectの画面から対象のインスタンスをクリックします。



左メニューから「問い合わせフロー」を選択し、「AWS Lambda」の登録画面を開きます。



Lambda_2(AmazonConnectの通知結果を保存する関数)を選択し、「+ Add Lambda Function」をクリックします。

Amazon Connect ✕

インスタンス

概要
テレフォニー
データストレージ
データストリーミング
問い合わせフロー
分析ツール
承認済みオリジン
お客様のプロフィール
タスク
Wisdom
Voice ID **新規**

ドキュメント 📄

🔔 新しいコンソール
ご意見をお聞かせください

AWS Lambda

AWS Lambda 関数を使用することによって、データベースやその他のサービスからデータを取得できます。返されたデータに基づいて、問い合わせを適切な問い合わせフローランチにルーティングできます。Lambda 関数を追加すると、お客様は、それら呼び出すためのアクセス許可を Amazon Connect に付与したことになります。 [新規 Lambda 関数の作成](#)

Lambda 関数

amazonconnect-status-save ▼

+ Add Lambda Function

Lambda 関数

Function Arn	Arn	Copy to clipboard	Action
[Redacted]	[Redacted]	📄 コピー	Remove

問い合わせフローログ 保存

☒ 問い合わせフローログを有効にする
ログは次の Amazon CloudWatch ロググループに保存されます。
aws/connect/[Redacted]

Amazon Polly 保存

Amazon Polly は、さまざまな言語でテキストを音声に変換するサービスです。顧客とのやり取りを自動化するときに、この機能を使用できます。 [詳細はこちら](#)

フィードバック 日本語 ▼ © 2008 - 2021, Amazon Web Services, Inc. またはその関連会社。無断転用禁止。 プライバシーポリシー 利用規約 Cookie の設定

対象のLambdaが追加されたことを確認します。

Amazon Connect ✕

インスタンス

概要
テレフォニー
データストレージ
データストリーミング
問い合わせフロー
分析ツール
承認済みオリジン
お客様のプロフィール
タスク
Wisdom
Voice ID **新規**

ドキュメント 📄

🔔 新しいコンソール
ご意見をお聞かせください

AWS Lambda

AWS Lambda 関数を使用することによって、データベースやその他のサービスからデータを取得できます。返されたデータに基づいて、問い合わせを適切な問い合わせフローランチにルーティングできます。Lambda 関数を追加すると、お客様は、それら呼び出すためのアクセス許可を Amazon Connect に付与したことになります。 [新規 Lambda 関数の作成](#)

Lambda 関数

Choose an option ▼

+ Add Lambda Function

Lambda 関数

Function Arn	Arn	Copy to clipboard	Action
[Redacted]	[Redacted]	📄 コピー	Remove
amazonconnect-status-save	arn:aws:lambda:ap-northeast-1:123456789012:function:amazonconnect-status-save	📄 コピー	Remove

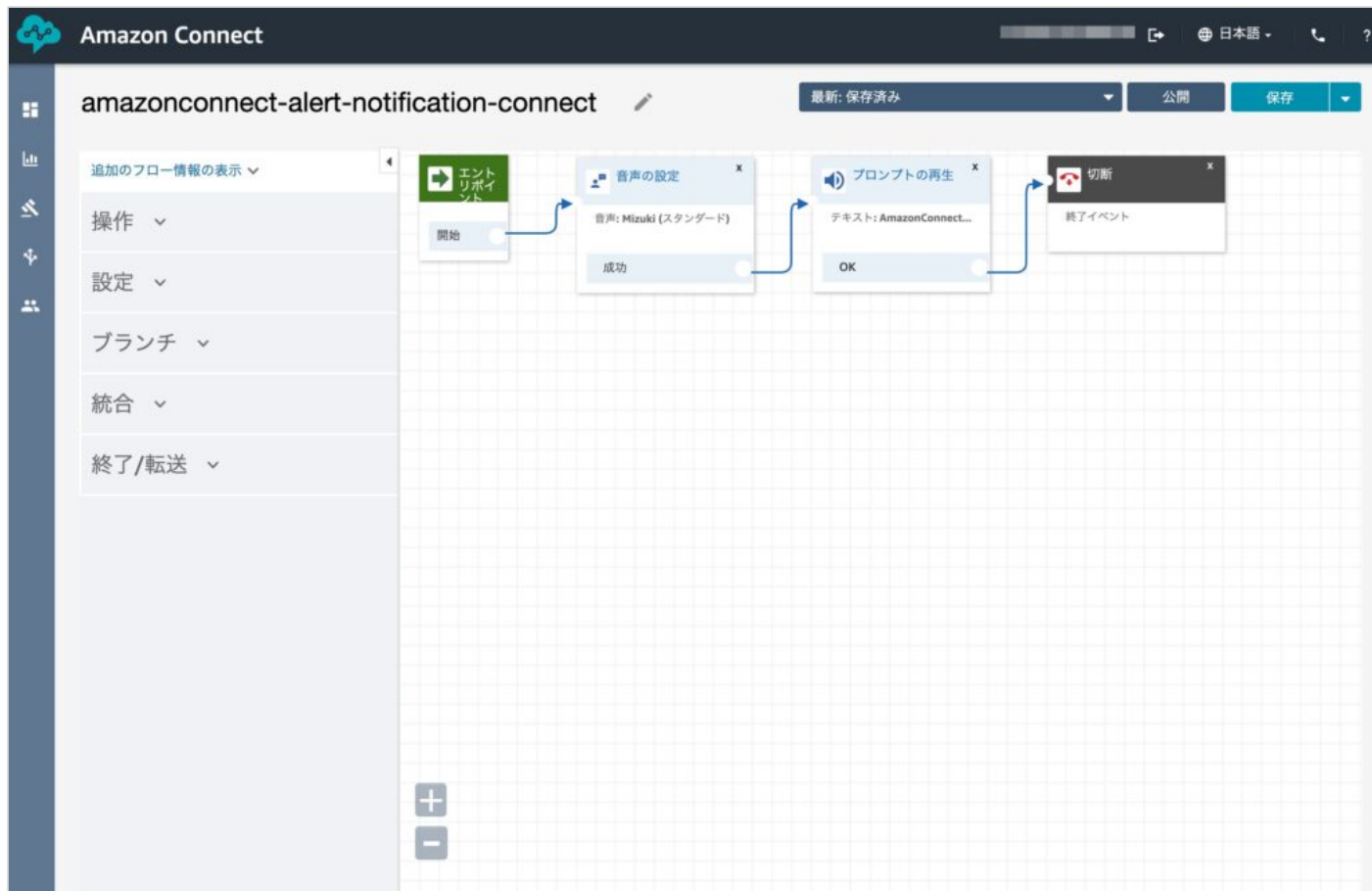
問い合わせフローログ 保存

☒ 問い合わせフローログを有効にする
ログは次の Amazon CloudWatch ロググループに保存されます。
aws/connect/[Redacted]

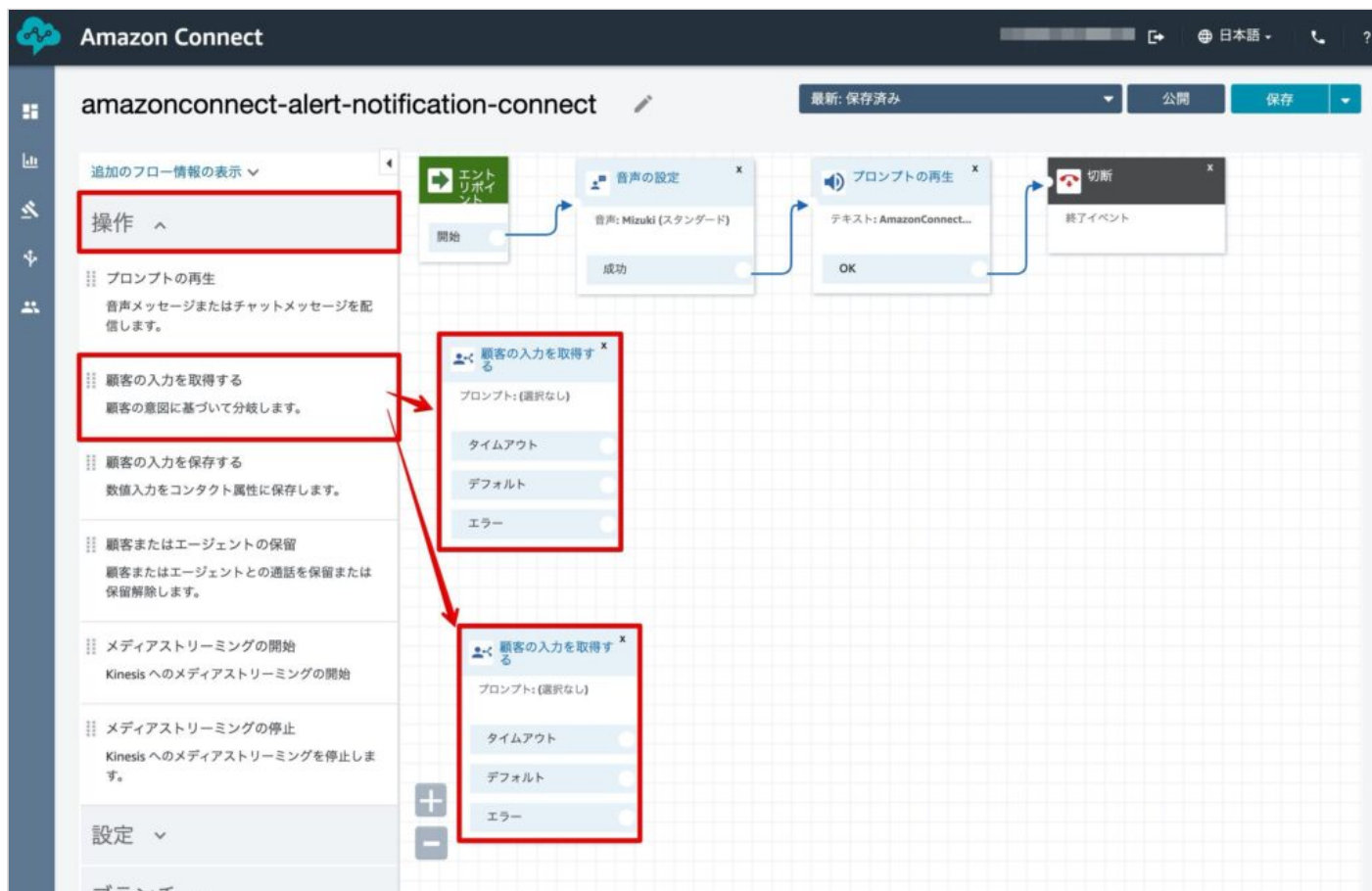
フィードバック 日本語 ▼ © 2008 - 2021, Amazon Web Services, Inc. またはその関連会社。無断転用禁止。 プライバシーポリシー 利用規約 Cookie の設定

問い合わせフロー修正

問い合わせフローを修正していきます。



「操作」を展開し、「顧客の入力を取得する」を2つ、ドラッグ&ドロップします。



「設定」を展開し、「コンタクト属性の設定」を2つ、ドラッグ&ドロップします。

Amazon Connect

amazonconnect-alert-notification-connect

最新: 保存済み 公開 保存

追加のフロー情報の表示

操作

設定

作業キューの設定
問い合わせの転送先となるキューを指定します。

コンタクト属性の設定
問い合わせ属性として key-value ペアを定義して保存します。

ルーティングの優先度/時間を変更する
キューのコンタクトの優先度を変更します。

ログ記録動作の設定
コンタクトフローログを有効または無効にします。

記録と分析の動作を設定
通話記録動作を指定し、音声分析を設定します。音声分析を使用するには、録音を設定する必要があります。

保留フローの設定
顧客またはエージェントとの通話を保留状態にしたときに呼び出されるフローを設定します。

開始

音声の設定
音声: Mizuki (スタンダード)
成功

プロンプトの再生
テキスト: AmazonConnect...
OK

切断
終了イベント

顧客の入力を取得する
プロンプト: (選択なし)
タイムアウト
デフォルト
エラー

コンタクト属性の設定
出力: =
成功
エラー

顧客の入力を取得する
プロンプト: (選択なし)
タイムアウト
デフォルト
エラー

コンタクト属性の設定
出力: =
成功
エラー

「操作」を展開し、「プロンプトの再生」を2つ、ドラッグ&ドロップします。

Amazon Connect

amazonconnect-alert-notification-connect

最新: 保存済み 公開 保存

追加のフロー情報の表示

操作

プロンプトの再生
音声メッセージまたはチャットメッセージを配信します。

顧客の入力を取得する
顧客の意図に基づいて分岐します。

顧客の入力を保存する
数値入力をコンタクト属性に保存します。

顧客またはエージェントの保留
顧客またはエージェントとの通話を保留または保留解除します。

メディアストリーミングの開始
Kinesis へのメディアストリーミングの開始

メディアストリーミングの停止
Kinesis へのメディアストリーミングを停止します。

設定

開始

音声の設定
音声: Mizuki (スタンダード)
成功

プロンプトの再生
テキスト: AmazonConnect...
OK

切断
終了イベント

顧客の入力を取得する
プロンプト: (選択なし)
タイムアウト
デフォルト
エラー

コンタクト属性の設定
出力: =
成功
エラー

プロンプトの再生
プロンプト: (選択なし)
OK

顧客の入力を取得する
プロンプト: (選択なし)
タイムアウト
デフォルト
エラー

コンタクト属性の設定
出力: =
成功
エラー

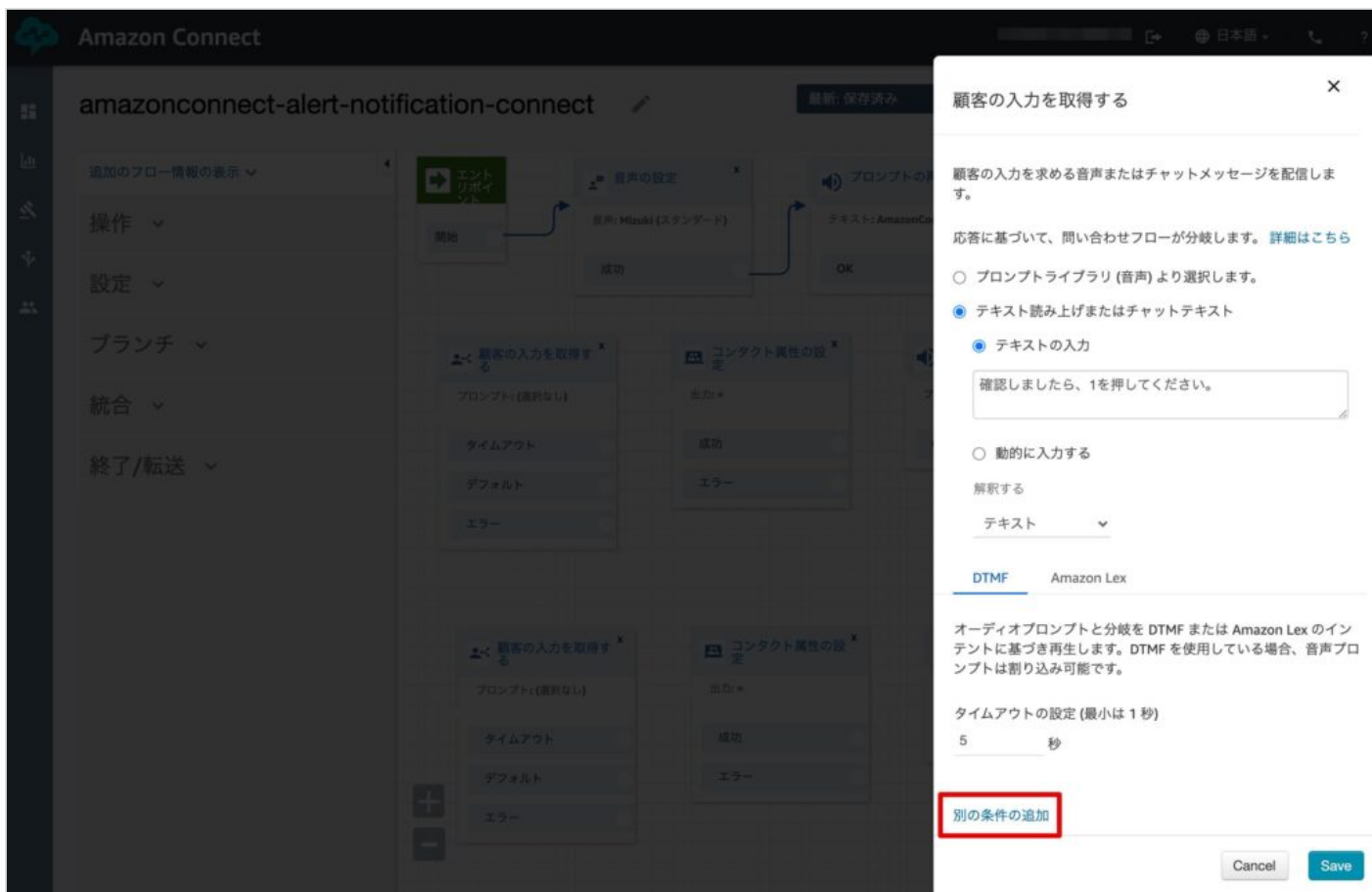
プロンプトの再生
プロンプト: (選択なし)
OK

「統合」を展開し、「AWS Lambda 関数を呼び出す」をドラッグ&ドロップします。

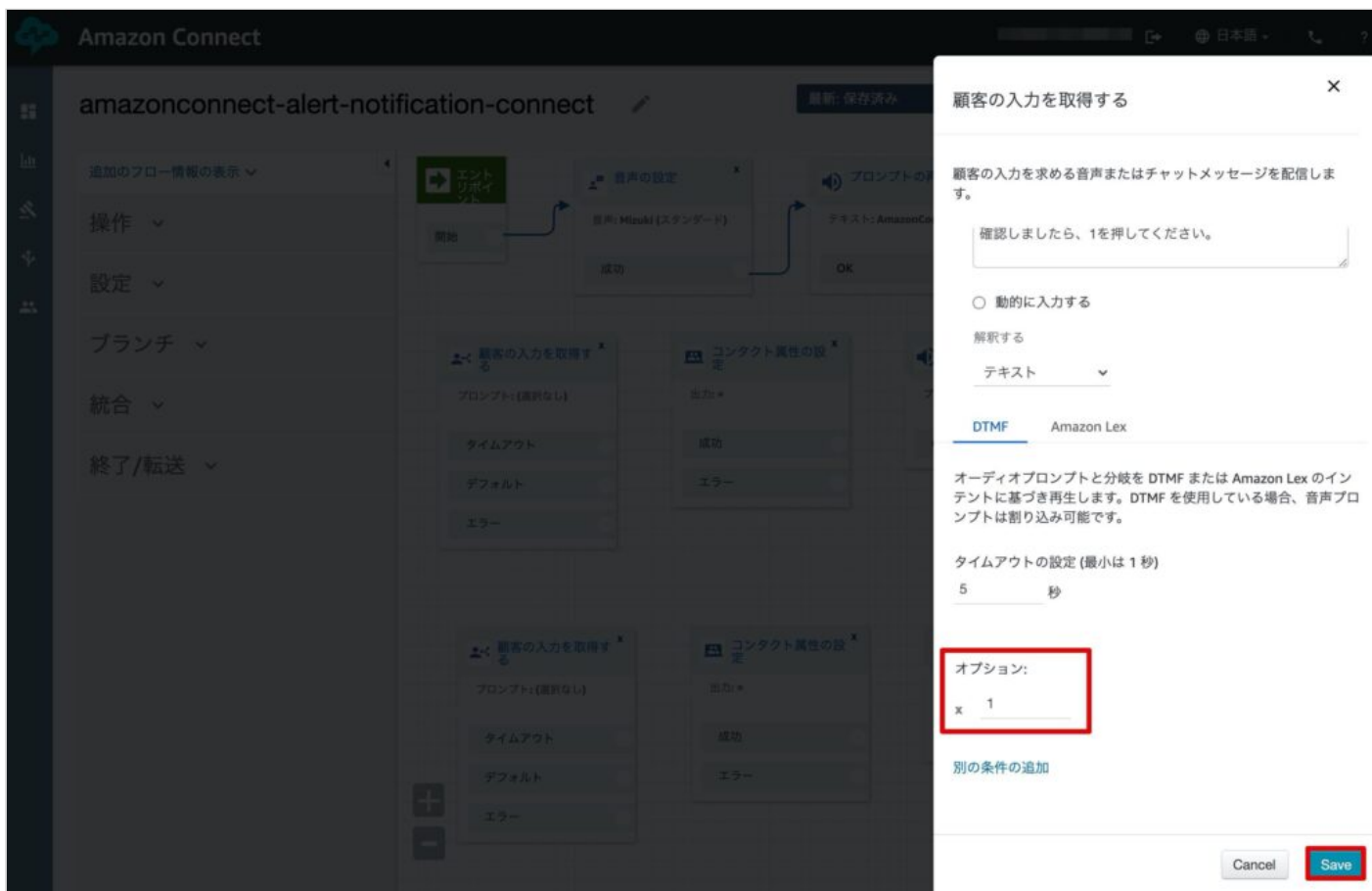
「テキスト読み上げまたはチャットテキスト」→「テキストの入力」を選択し、「確認しましたら、1を押してください」と入力します。

The screenshot displays the Amazon Connect console interface. In the background, a workflow diagram is visible with steps like '開始' (Start), '音声の設定' (Voice settings), and 'プロンプトの入力' (Prompt input). Overlaid on this is a modal window titled '顧客の入力を取得する' (Get customer input). Inside the modal, there are two main options: 'プロンプトライブラリ (音声) より選択します。' (Select from prompt library (voice)) and 'テキスト読み上げまたはチャットテキスト' (Text-to-speech or chat text). The second option is selected, and under it, 'テキストの入力' (Text input) is chosen. A text input field contains the Japanese text '確認しましたら、1を押してください。' (After confirmation, please press 1.). Below this, there are tabs for 'DTMF' and 'Amazon Lex', with 'DTMF' currently active. A section for 'オーディオプロンプトと分岐を DTMF または Amazon Lex のインテントに基づき再生します。' (Play audio prompts and branches based on DTMF or Amazon Lex intent) is visible. At the bottom of the modal, there is a 'タイムアウトの設定 (最小は 1 秒)' (Set timeout (minimum is 1 second)) field with the value '5' and a unit of '秒' (seconds). A link '別条件の追加' (Add other conditions) is at the bottom left, and 'Cancel' and 'Save' buttons are at the bottom right.

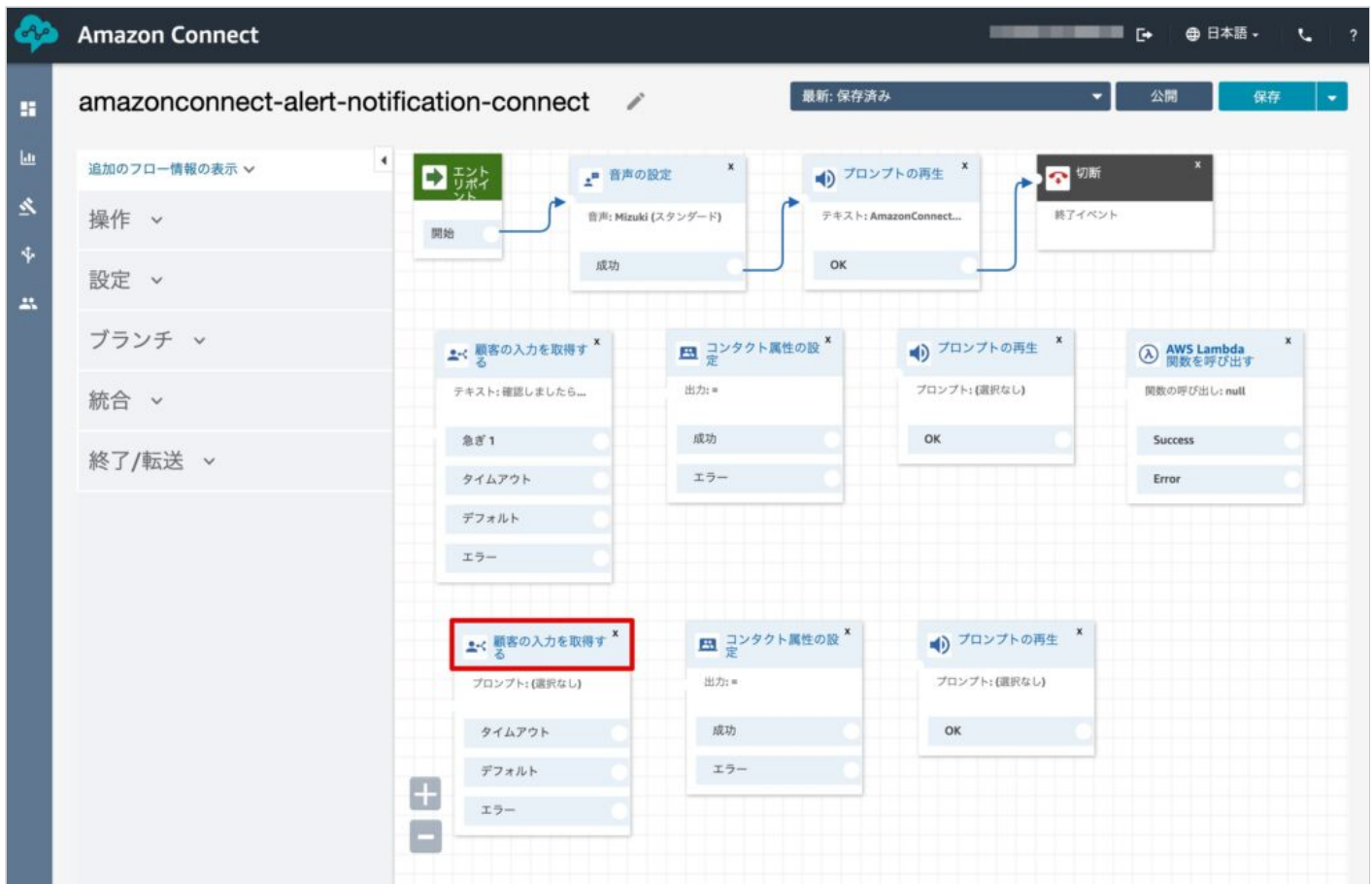
「別の条件の追加」をクリックします。



オプションに「1」を入力し、「Save」をクリックします。



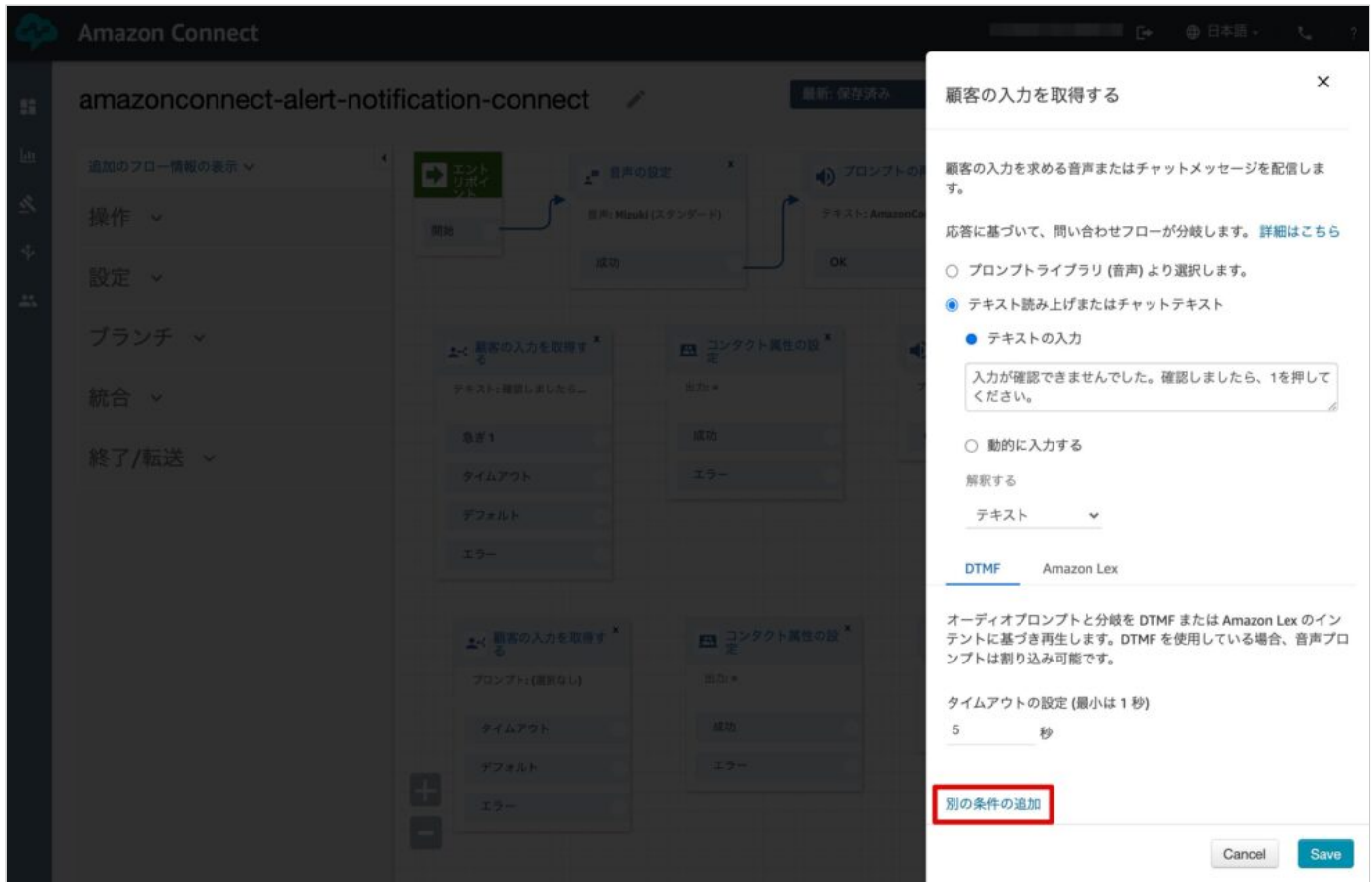
下側の「顧客入力を取得する」をクリックします。



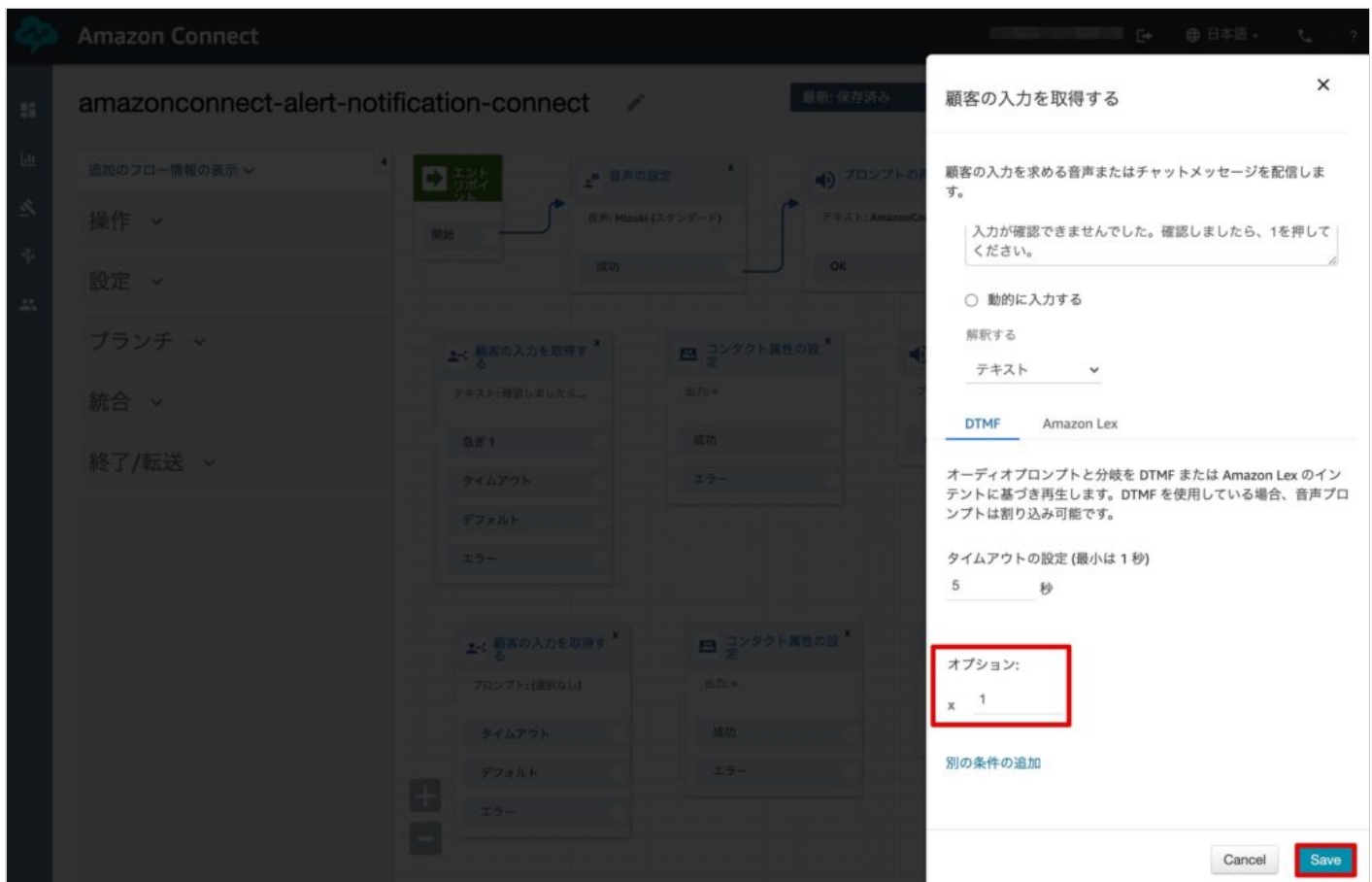
「テキスト読み上げまたはチャットテキスト」→「テキストの入力」を選択し、「入力が確認できませんでした。確認したら、1を押してください」と入力します。※最初のアナウンスで応答が確認できなかった時に、再度確認を行うためのフローです。



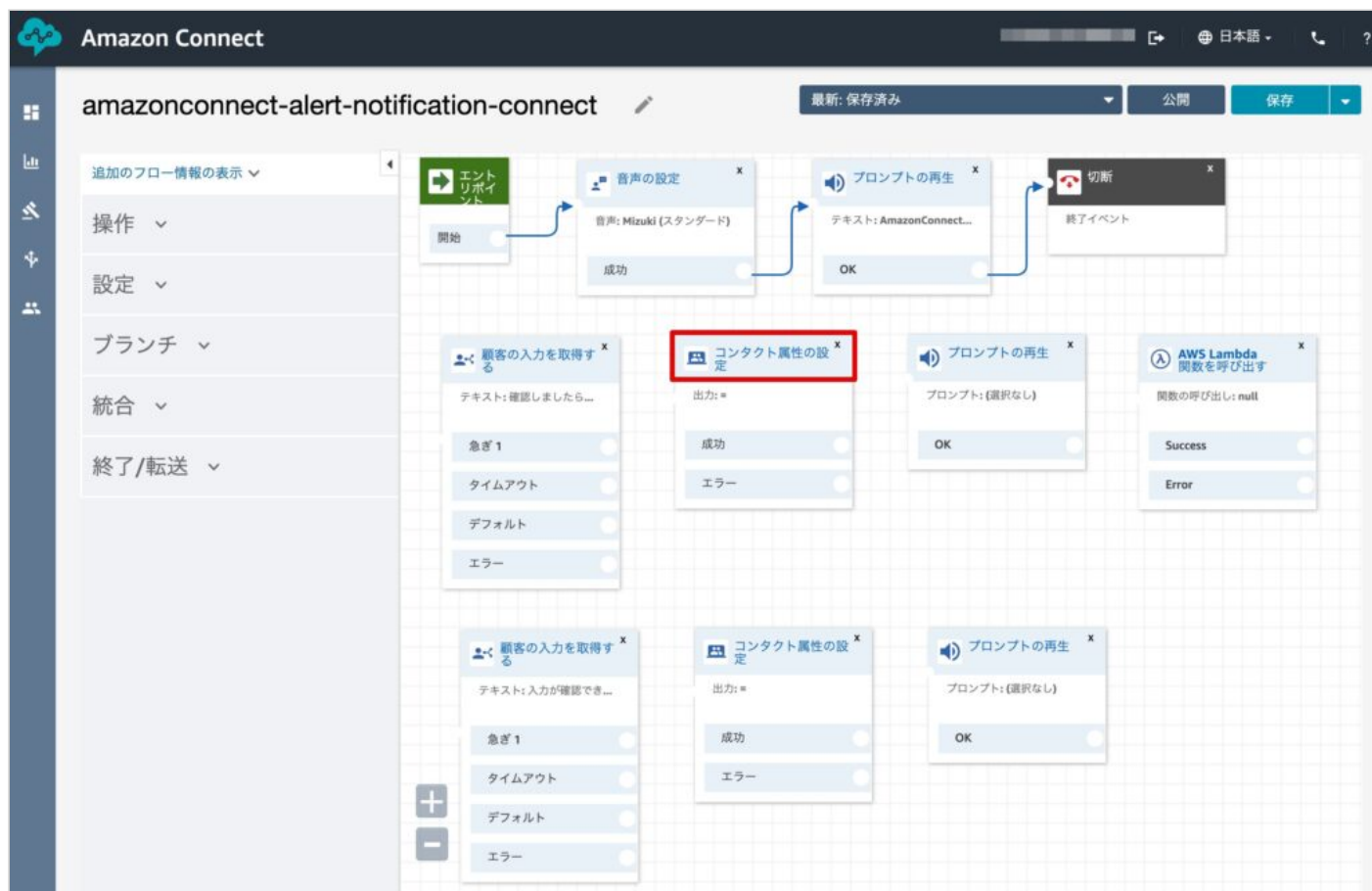
「別の条件の追加」をクリックします。



オプションに「1」を入力し、「Save」をクリックします。



上側の「コンタクト属性の設定」をクリックします。



下記の通り入力し、「Save」をクリックします。

宛先タイプ：ユーザー定義を選択

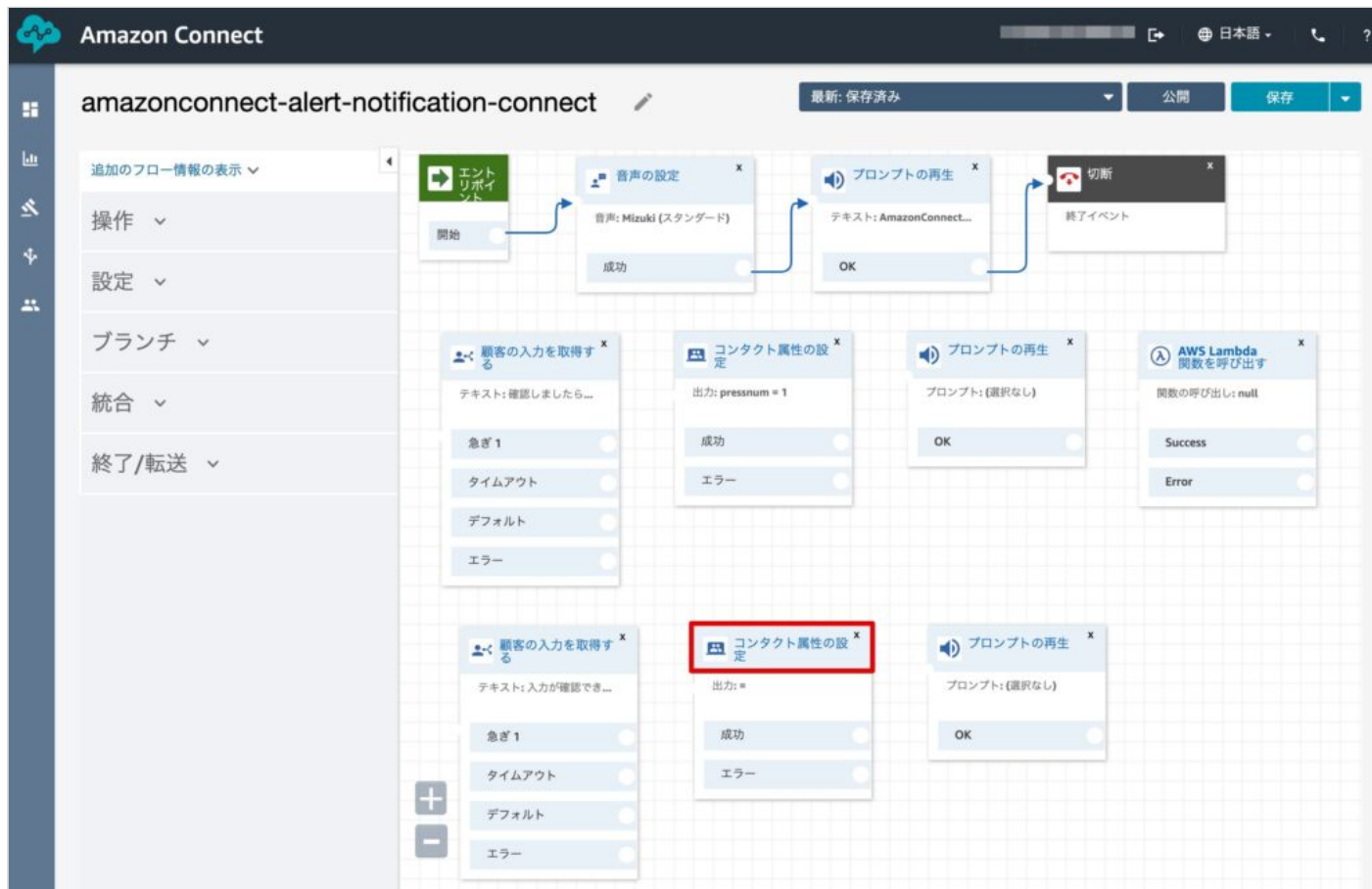
宛先属性：“pressnum”と入力 ※任意の名前を入力します。

テキストの使用：チェックを入れる

値：“1”を入力



下側の「コンタクト属性の設定」をクリックします。



下記の通り入力し、「Save」をクリックします。

宛先タイプ：ユーザー定義を選択

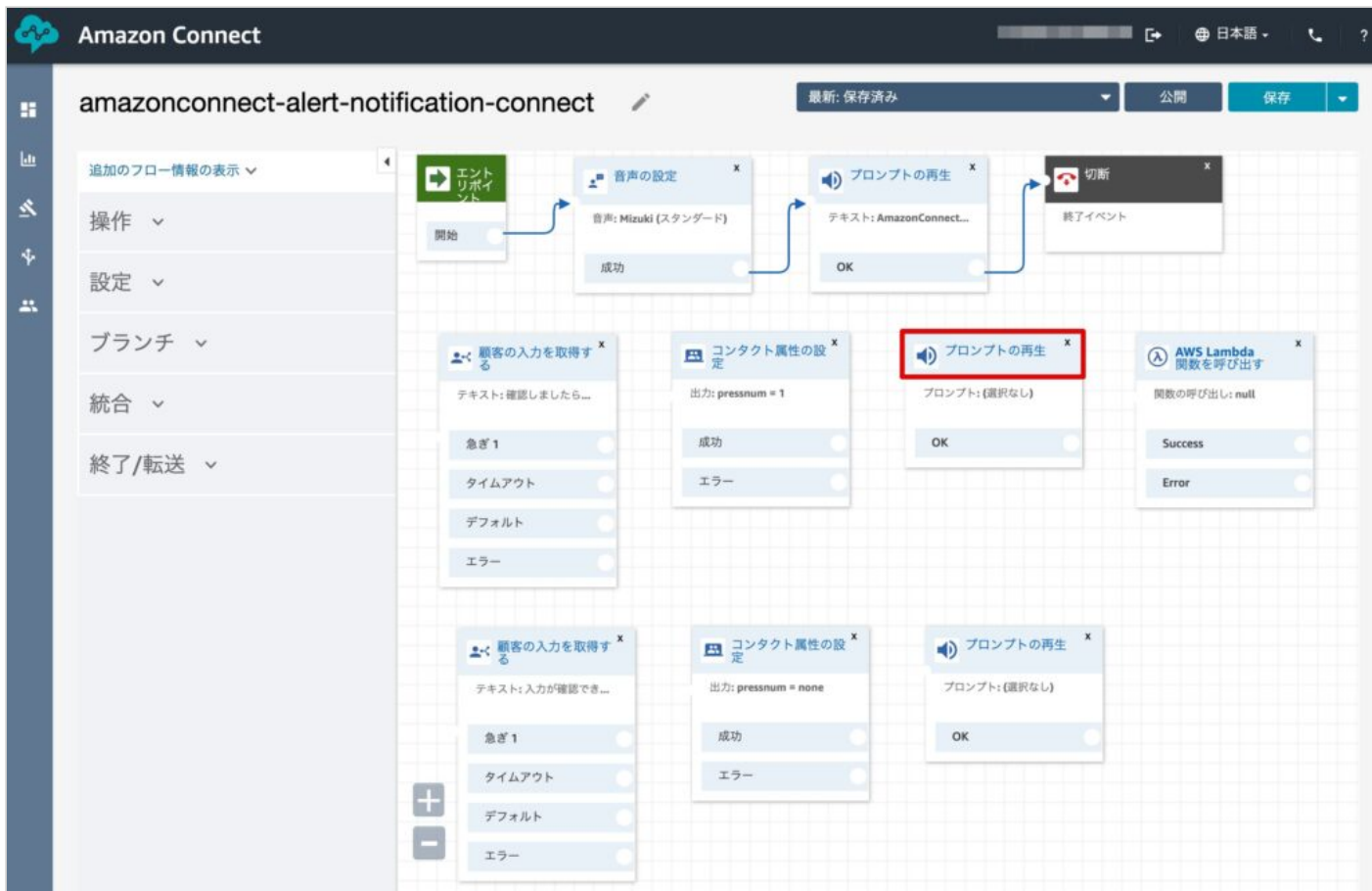
宛先属性：“pressnum”と入力 ※任意の名前を入力します。

テキストの使用：チェックを入れる

値：“none”を入力



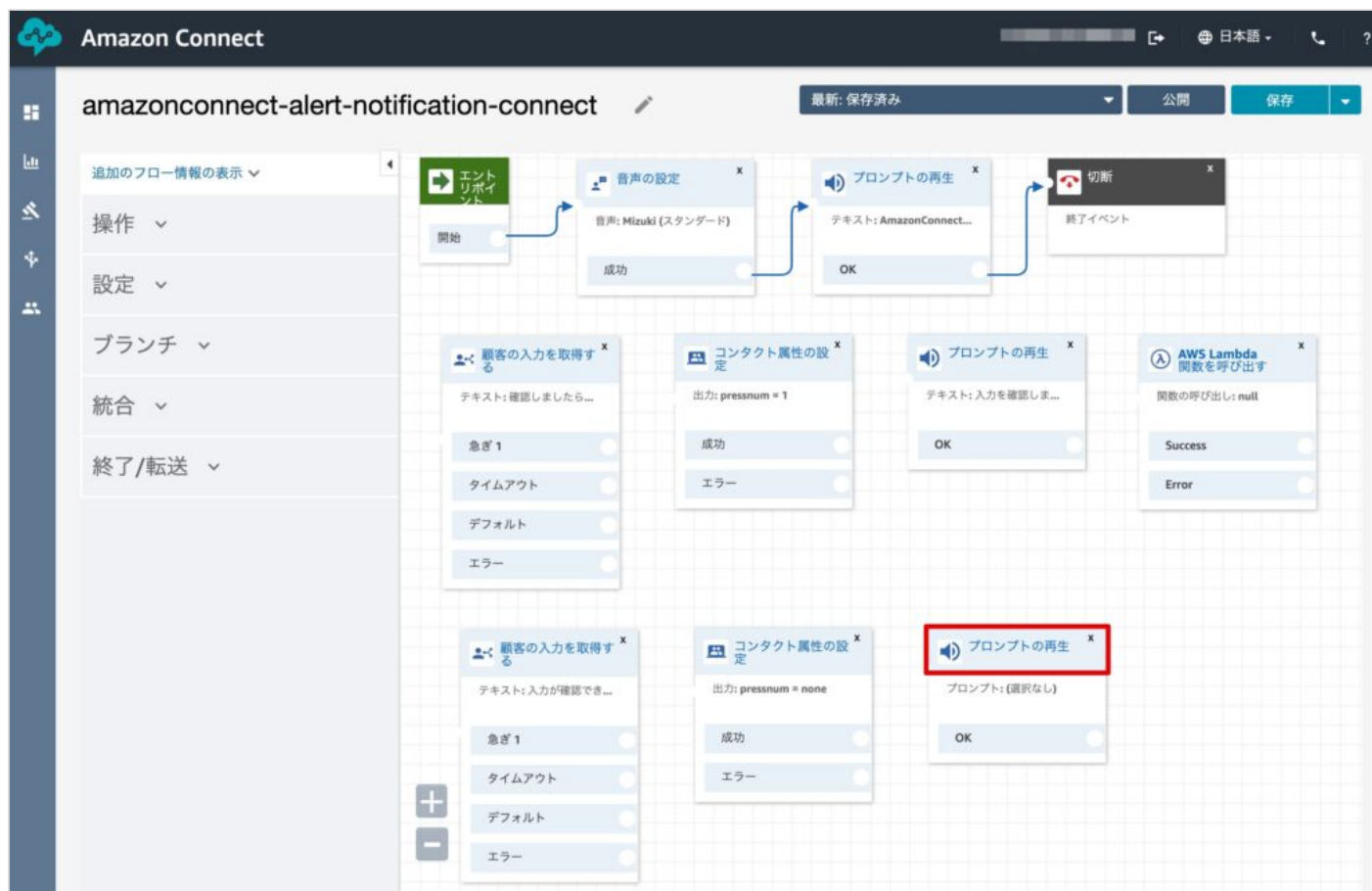
上側の「プロンプトの再生」をクリックします。



「テキスト読み上げまたはチャットテキスト」→「テキストの入力」を選択し、「入力を確認しました。障害発生通知完了です。」と入力します。※正常に"1"が押された場合のメッセージ内容です。



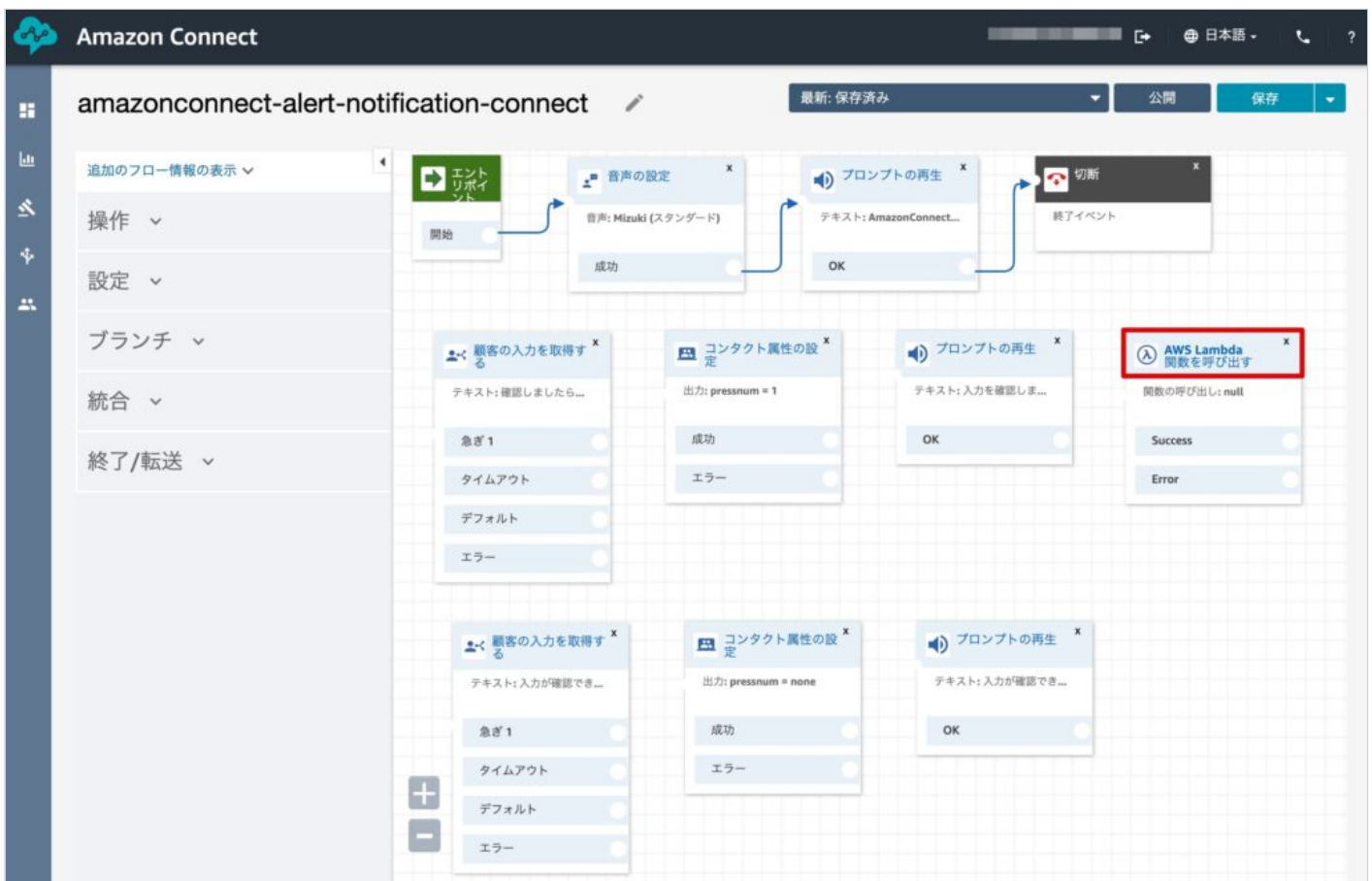
下側の「プロンプトの再生」をクリックします。



「テキスト読み上げまたはチャットテキスト」→「テキストの入力」を選択し、「入力を確認しました。障害発生通知完了です。」と入力します。※正常に"1"が押されなかった場合のメッセージ内容です。



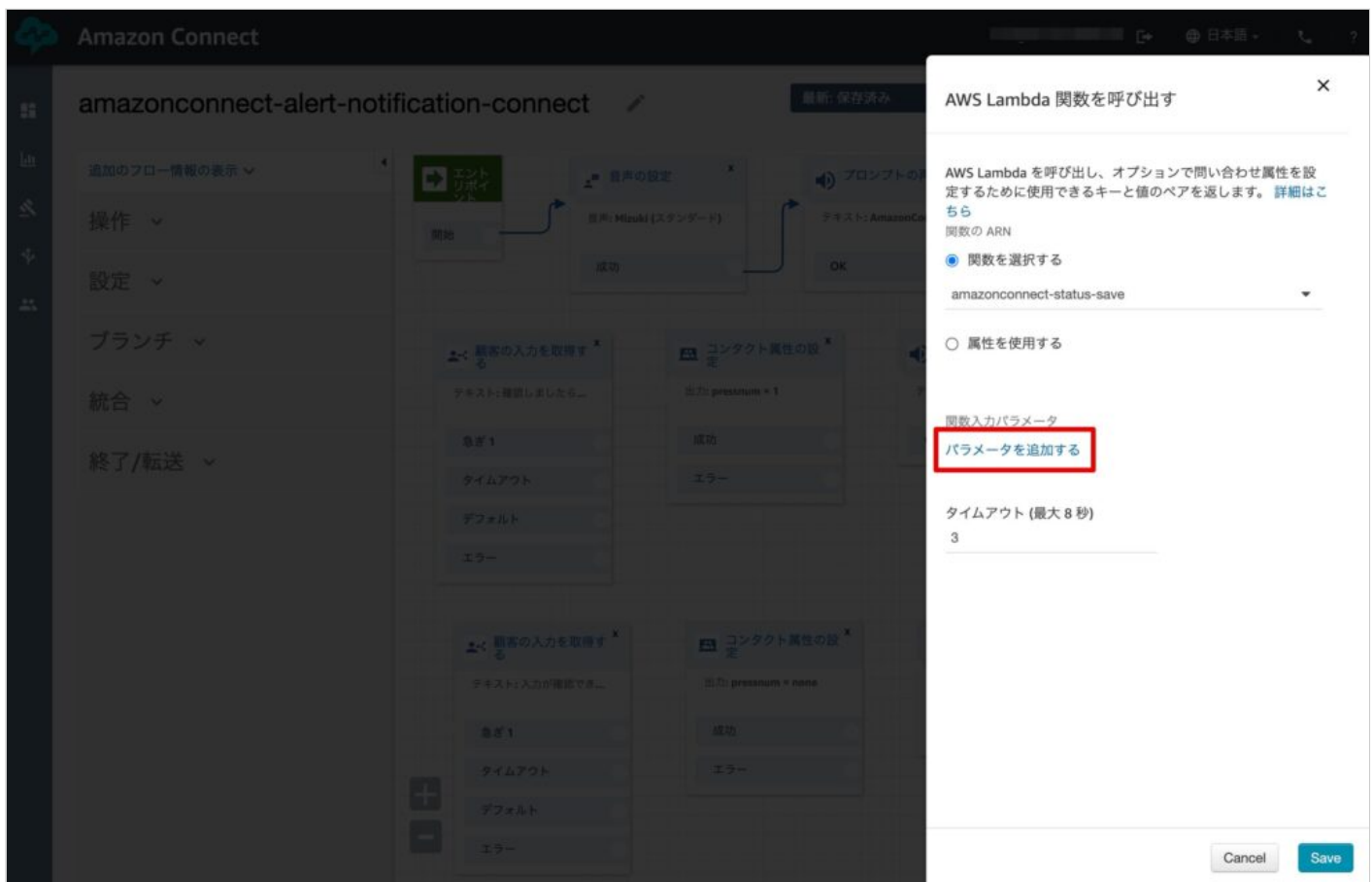
「AWS Lambda 関数を呼び出す」をクリックします。



「関数を選択する」にチェックを入れ、作成したLambda関数を選択します。



「パラメータを追加する」をクリックします。



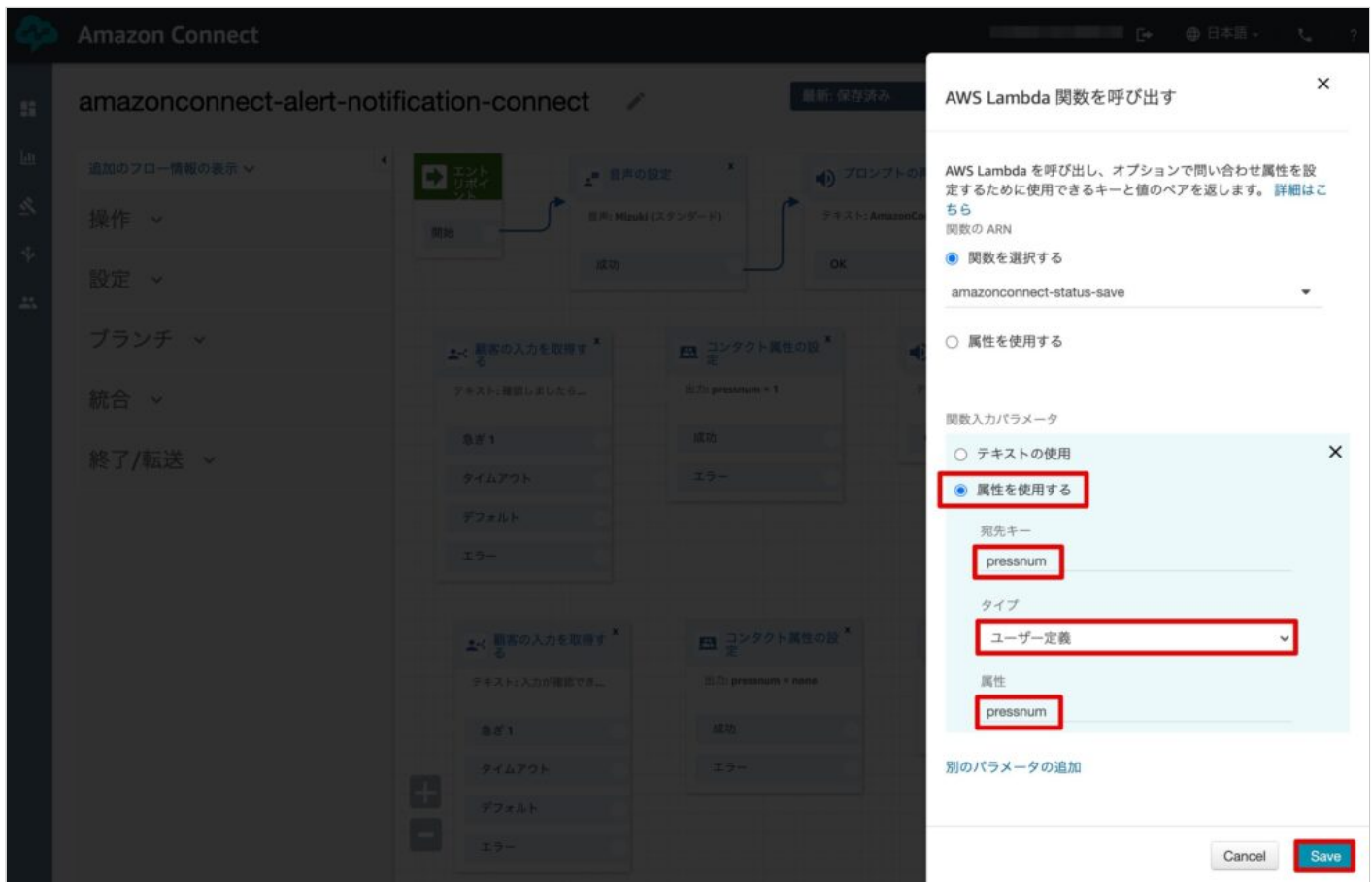
下記の通り入力し、「Save」をクリックします。

属性を使用する：チェックを入れる

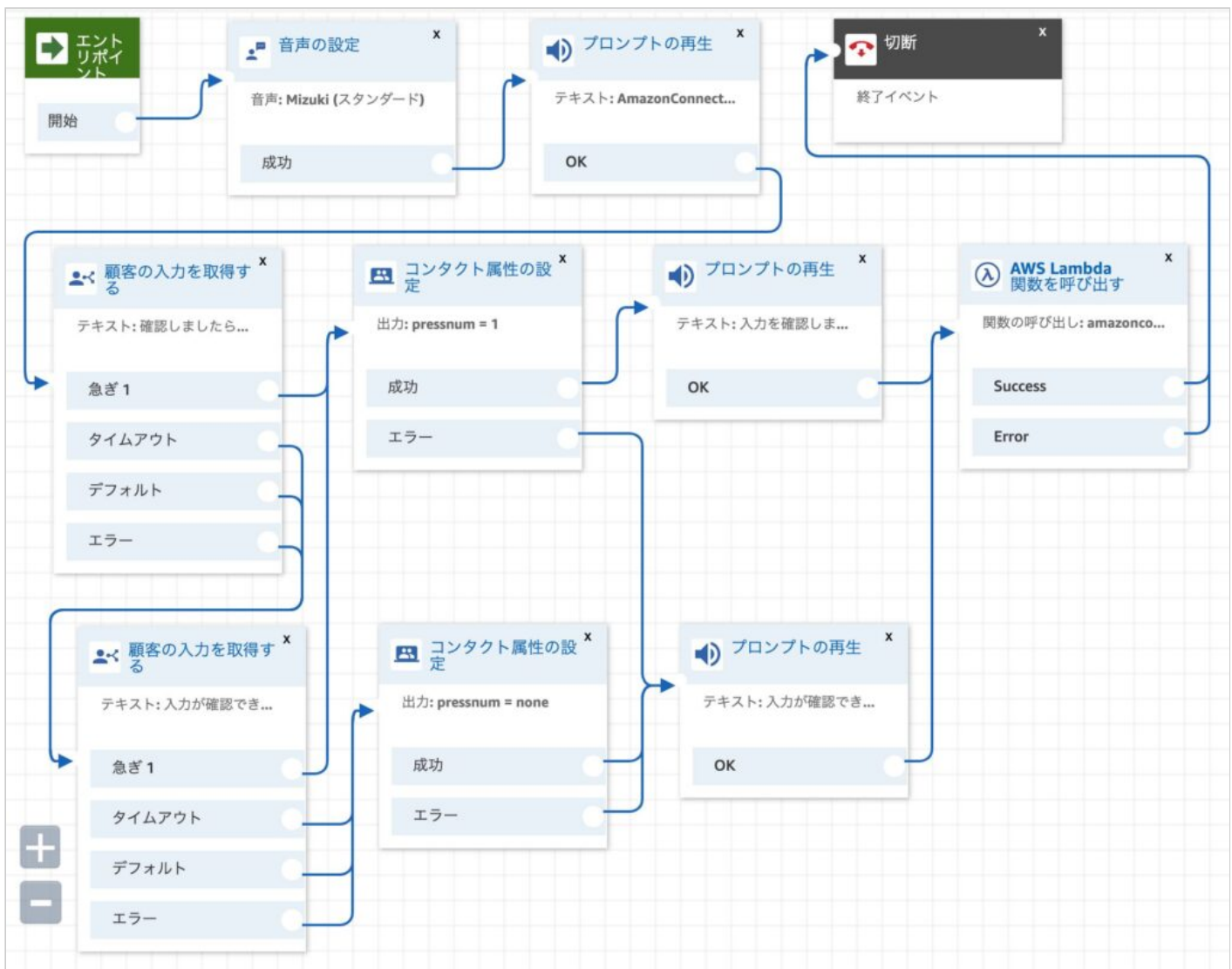
宛先キー：“pressnum”と入力 ※任意の名前を入力します。

タイプ：ユーザー定義

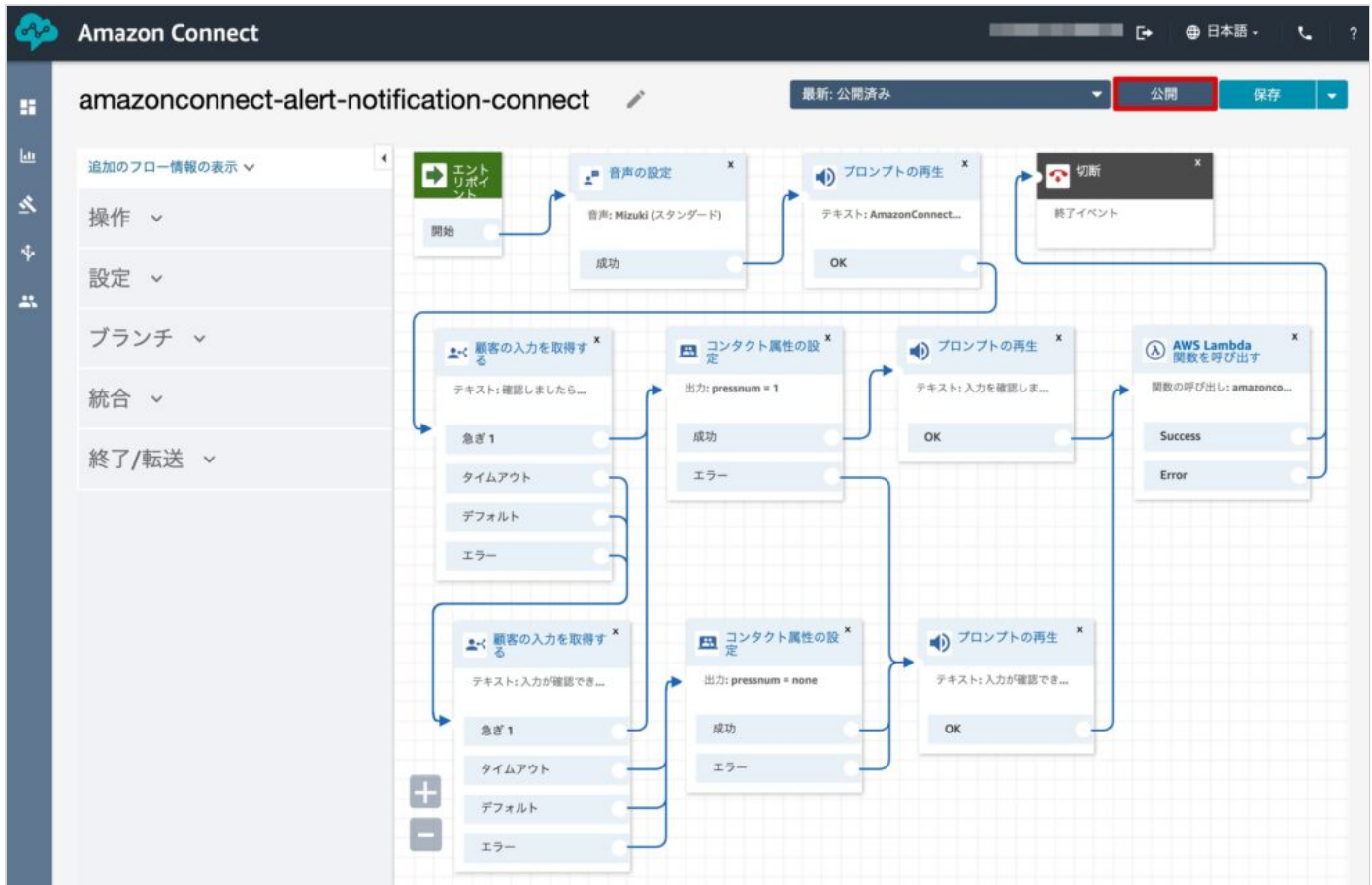
属性：“pressnum”と入力 ※コンタクト属性の設定と合わせます。



下記の通りの接続に変更します。



「公開」をクリックし、問い合わせフローを保存します。



以上で、AmazonConnectによる自動電話通知（7.複数連絡先への電話通知〈構築⑥〉）の説明は完了です！