インターン課題

AmazonConnect

課題 1

Amazon Connect を使って、ソフトフォン（CCP）での通話を体験

課題 2

Amazon Connect の フロー の 発着信 の フロー を作成

課題 3

作成した 営業日判定のLambda を Amazon Connect の フロー に組み込んで、
営業時間内外の判定が分岐されて、

・営業時間内の場合、オペレータ役と通話ができる。

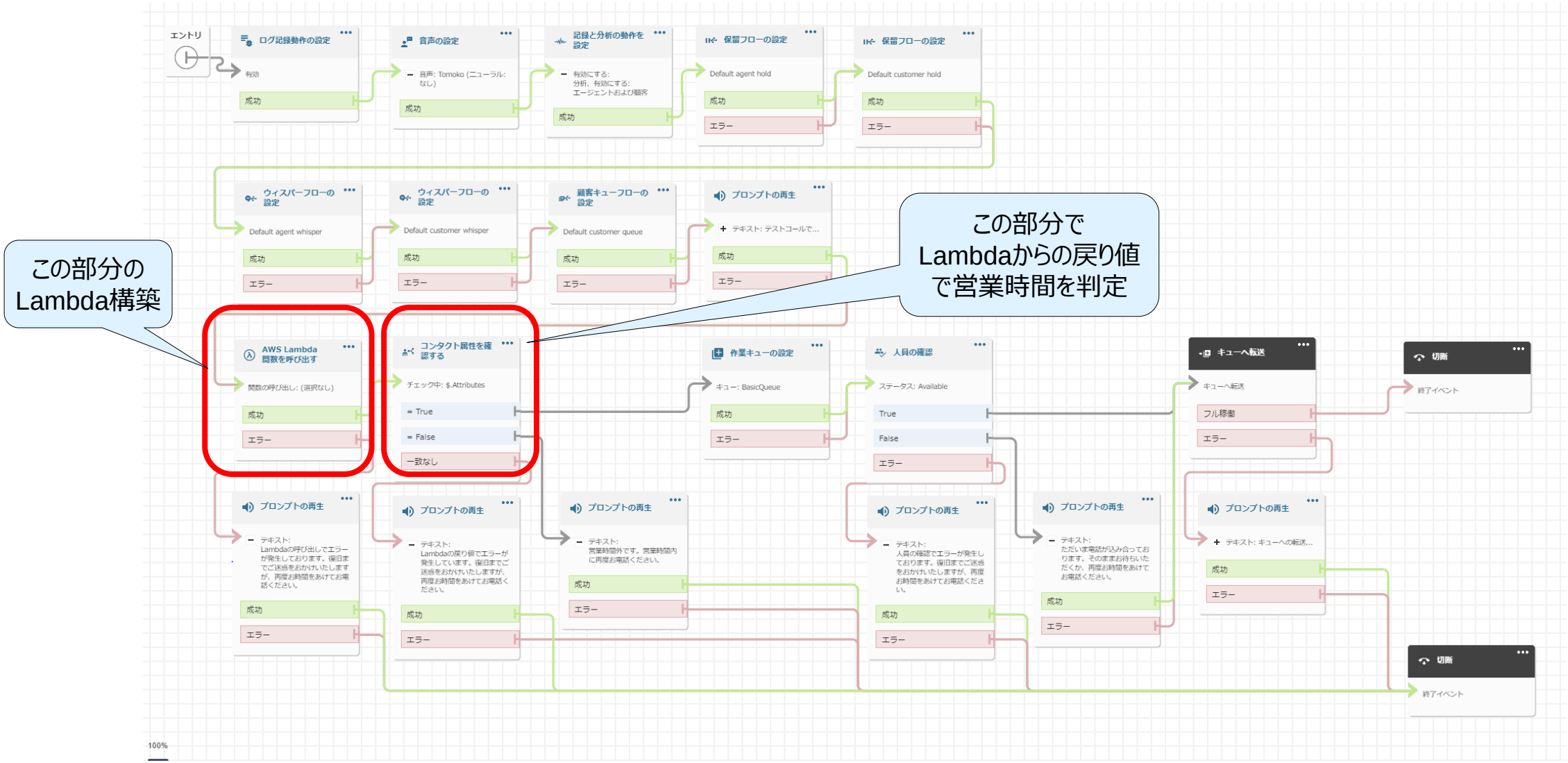
・営業時間外の場合、営業時間外の自動音声ガイドが流れる（オペレータ役との通話はできない）

ことを体験

Lambda課題１：スケジュール管理機能の構築

背景：
AmazonConnectでは、祝日・臨時休業日の設定ができないため、システムでスケジュール登録出来ないかの相談があった。
（週ごとに設定を変える必要があり、運用が煩雑となるため。）
要件：
曜日単位の営業時間判定ができること
2年先まで祝日、臨時休業日、臨時休業時間の判定ができること
祝日、臨時協業日、臨時休業時間の判定が優先されること
アマゾンコネクトインスタンス単位で設定できること
キュー名単位で設定できること
月～金　営業時間10:00～18:30
土日　休み
祝日、臨時休業日、臨時休業時間　例外テーブルのみ
インスタンスごとのテーブル
キューはパーティションキー
日付はソートキー
AWS使用サービス：
Lambda , DynamoDB
言語：
Python or Java
Input：
amazon connect キュー名、インスタンス名
戻り値：
True：営業日、営業時間内　/ False：営業日、営業時間外

フローイメージ


この部分の
Lambda構築
この部分で
Lambdaからの戻り値
で営業時間を判定
エントリ
ログ記録動作の設定
有効
成功
音声の設定
音声: Tomoko (ニューラル: なし)
成功
記録と分析の動作を設定
有効にする: 分析、有効にする: エージェントおよび顧客
成功
保留フローの設定
Default agent hold
成功
エラー
保留フローの設定
Default customer hold
成功
エラー
ウィスパーフローの設定
Default agent whisper
成功
エラー
ウィスパーフローの設定
Default customer whisper
成功
エラー
顧客キューフローの設定
Default customer queue
成功
エラー
プロンプトの再生
テキスト: テストコールで...
成功
エラー
AWS Lambda 関数を呼び出す
関数の呼び出し: (選択なし)
成功
エラー
コンタクト属性を確認する
チェック中: $.Attributes
= True
= False
一致なし
作業キューの設定
キュー: BasicQueue
成功
エラー
人員の確認
ステータス: Available
True
False
エラー
キューへ転送
キューへ転送
フル稼働
エラー
切断
終了イベント
プロンプトの再生
テキスト: Lambdaの呼び出しでエラーが発生しております。復旧までご迷惑をおかけいたしますが、再度お時間をあけてお電話ください。
成功
エラー
プロンプトの再生
テキスト: Lambdaの戻り値でエラーが発生しています。復旧までご迷惑をおかけいたしますが、再度お時間をあけてお電話ください。
成功
エラー
プロンプトの再生
テキスト: 営業時間外です。営業時間内に再度お電話ください。
成功
エラー
プロンプトの再生
テキスト: 人員の確認でエラーが発生しております。復旧までご迷惑をおかけいたしますが、再度お時間をあけてお電話ください。
成功
エラー
プロンプトの再生
テキスト: ただいま電話が込み合っております。そのままお待ちいただくか、再度お時間をあけてお電話ください。
成功
エラー
プロンプトの再生
テキスト: キューへの転送...
成功
エラー
切断
終了イベント
100%

ログ出力用に組み込んでほしいサブモジュール

[環境変数]
LOG_FORMAT：[%(levelname)s] %(message)s
LOG_LEVEL ：10

* LOG_LEVEL：0:NOTSET 10:DEBUG 20:INFO 30:WARNING 40:ERROR 50:CRITICAL

[Global定義]
```
import logging
_logger = logging.getLogger()
```

[local定義]
```
# Loggerの初期設定
initLogger()
```

[sample]
```
_logger.debug('xxx:yyy¥n')
_logger.info('xxx:yyy¥n')
_logger.warning('xxx:yyy¥n')
_logger.error('xxx:yyy¥n')
```

[sub module]
```
#【Logger初期設定処理】
#【INPUT】なし
#【OUTPUT】なし
#【処理概要】
#　loggerのLOGレベルとフォーマットを環境変数から設定する
def initLogger():
    # 環境変数からLOG_LEVELを取得して設定する
    # 0:NOTSET 10:DEBUG 20:INFO 30:WARNING 40:ERROR 50:CRITICAL
    try:
        wLogLevel = os.environ['LOG_LEVEL']
    except KeyError as err:
        # LOG_LEVELが未定義の場合【INFO】を適用する
        _logger.warning('LOG_LEVEL Not Configuration¥n')
        wLogLevel = '20'
    try:
        iLogLevel = int(wLogLevel)
    except:
        # LOG_LEVELが数字以外の場合【INFO】を適用する
        _logger.warning('LOG_LEVEL Bad Configuration:' +wLogLevel +'¥n')
        iLogLevel = 20

    _logger.setLevel(iLogLevel)

    # 環境変数からフォーマットを設定する
    try:
        wLogFormat = os.environ['LOG_FORMAT']
        formatter = logging.Formatter(
            wLogFormat
        )
        for handler in _logger.handlers:
            handler.setFormatter(formatter)
    except KeyError as err:
        # LOG_FORMATが未定義の場合[%(levelname)s] %(message)sを適用する
        _logger.warning('LOG_FORMAT Not Configuration¥n')
        wLogFormat = '[%(levelname)s] %(message)s'
        formatter = logging.Formatter(
            wLogFormat
        )
        for handler in _logger.handlers:
            handler.setFormatter(formatter)

    return
```

Lambda課題２：スケジュール管理機能の構築（余力があれば･･･）

変更背景：
　DBは、システム担当者でないと更新ができないため、コールセンター担当者でも更新ができるようにCSVで
　スケジュール登録出来ないかの相談があった。それを受けて、CSV でスケジュール登録したものを
　夜間バッチ（EventBridge）でDynamoDBに更新をかけるように修正することになった
要件：
　2年先以上の祝日、臨時休業日の判定ができること
　曜日単位の営業時間判定ができること
　祝日、臨時協業日、臨時休業時間の判定が優先されること
　アマゾンコネクトインスタンス単位で設定できること
　キュー名単位で設定できること
　レコードの削除
　S3に格納されたCSVのスケジュール情報を夜間バッチでDynamoDBのスケジュール情報に更新をかける。
AWS使用サービス：
　Lambda , DynamoDB , S3 , EventBridge
言語：
　Python or Java
戻り値：
　True：営業日、営業時間内　/ False：営業日、営業時間外

CloudFormation

課題１：

（Cloud Formation）

作成した Lambda,DynamoDB,EventBridge,S3,AmazonConnect(コールフローなど） を 東京リージョン から
東京リージョンへ展開するテンプレート（yaml）を作成
リソース名の後ろに、_biz （商用） をつけて、yamlを使って、東京リージョンへリソースを展開

確認事項１：

展開したリソースがあること（S3,DynamoDB etc）、正常に動作すること(Lambda,EventBridge)を確認

確認事項２：

展開した 環境 での 発着信・営業日判定が動くことを確認(Amazon Connect)

発表資料とりまとめ

課題１：

　インターン期間の成果についてまとめる。