

目的：Amazon Lex で日本語のチャットボックスを作成し、Amplify で Web ページをホスティングを行う

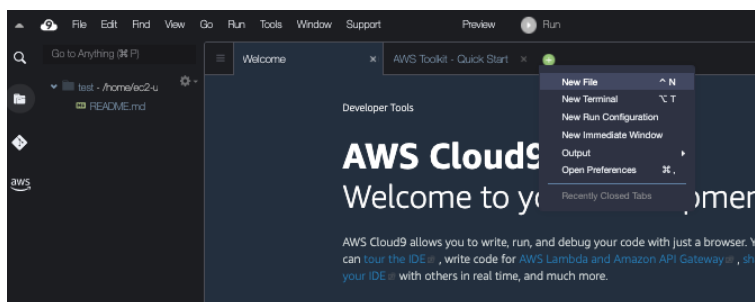
今回の資料や利用するコードなどはこちらのリポジトリにアップしています。

<https://bit.ly/2TO3cUS>

■初期設定 (Amplify 利用環境の設定)

[Cloud9 の起動]

1. マネージメントコンソールで Cloud9 へアクセスします。
2. 画面右上の[Create environment]をおします。
3. 適当な名前を付け[Next step]をおします。
4. [Instance type]で[Other instance type]を選びドロップダウンから[t3.medium]を選びます。
6. [Next step]を押し、次の画面で[Create environment]を押します。
7. 起動中の画面が出ますので、数分間待ちます。
8. アクセスが出来たらディスクを拡張します。Cloud9 はデフォルトで 10GB しか容量がないため、ディスクを拡張します。
9. Welcome タブの右にある緑のプラスボタンをおして、[New File]を選択します。



10. ディスク容量拡張のスクリプトをコピーし、内容を貼り付けます。Resize.sh [<https://bit.ly/3cUgZ36>]

```
#!/bin/bash

# Specify the desired volume size in GiB as a command-line argument. If not specified, default to 20 GiB.
SIZE=${1:-20}

# Get the ID of the environment host Amazon EC2 instance.
INSTANCEID=$(curl http://169.254.169.254/latest/meta-data/instance-id)

# Get the ID of the Amazon EBS volume associated with the instance.
VOLUMEID=$(aws ec2 describe-instances \
  --instance-id $INSTANCEID \
  --query "Reservations[0].Instances[0].BlockDeviceMappings[0].Ebs.VolumeId" \
  --output text)

# Resize the EBS volume.
aws ec2 modify-volume --volume-id $VOLUMEID --size $SIZE

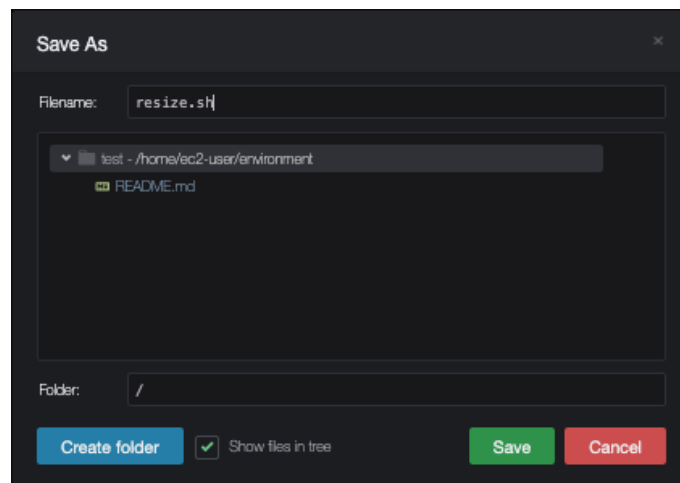
# Wait for the resize to finish.
while [ \
  "$(aws ec2 describe-volumes-modifications \
    --volume-id $VOLUMEID \
    --filters Name=modification-state,Values="optimizing","completed" \
    --query "length(VolumesModifications)" \
    --output text)" != "1" ]; do
  sleep 1
done

# Check if we're on an NVMe filesystem
if [ $(readlink -f /dev/xvda) = "/dev/xvda" ]
then
  # Rewrite the partition table so that the partition takes up all the space that it can.
  sudo growpart /dev/xvda 1

  # Expand the size of the file system.
  # Check if we are on AL2
  STR=$(cat /etc/os-release)
  SUB="VERSION_ID=\"2\""
  if [[ "$STR" == "$SUB" ]]
  then
    sudo xfs_growfs -d /
  else
    sudo resize2fs /dev/xvda1
  fi
else
  # Rewrite the partition table so that the partition takes up all the space that it can.
  sudo growpart /dev/nvme0n1 1

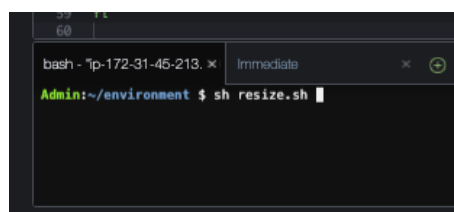
  # Expand the size of the file system.
  # Check if we're on AL2
  STR=$(cat /etc/os-release)
  SUB="VERSION_ID=\"2\""
  if [[ "$STR" == "$SUB" ]]
  then
    sudo xfs_growfs -d /
  else
    sudo resize2fs /dev/nvme0n1p1
  fi
fi
```

11. 貼り付けたあとに保存をします。ファイル名に[resize.sh]と入力し、[Save]を押します。

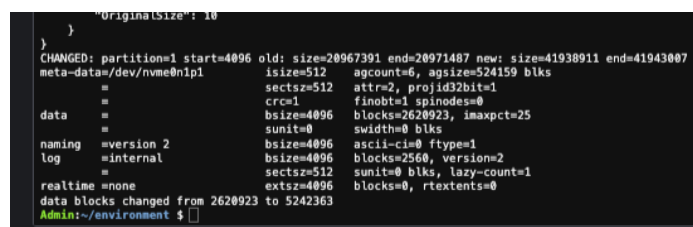



12. スクリプトを保存後、画面下のターミナルウィンドウで以下のコマンドを実行します。

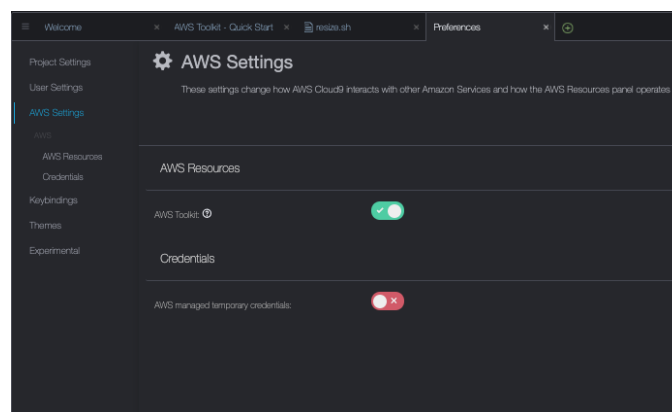
\$ sh resize.sh 30



13. 以下のように実行が完了し、容量が 3 倍になっているメッセージが出ればディスク拡張が完了です。



14. 次に、Cloud9 で初回時に設定されている AWS 認証情報を無効化します。画面右上の  マークを押下し、Setting 情報を開きます。その後、左メニューの[AWS Settings]を選択し、[AWS managed temporary credentials]を無効化します。



15. 続いて、Amplify CLI のインストールと設定を行います。まずは、以下のコマンドを入力し、ホームディレクトリに移動してください。

```
$ cd /home/ec2-user
```

※これからの作業は必ずこのディレクトリにて作業を行ってください。

16.次に Amplify CLI をインストールします。以下のコマンドを入力し、Amplify CLI をインストールしてください。

```
$ npm install -g @aws-amplify/cli
```

インストール処理が開始し、以下のメッセージが表示されればインストール完了です。

```
npm WARN notsup Not compatible with your version of node/npm: amplify-app@3.0.0
npm WARN notsup Unsupported engine for value-or-promise@1.0.6: wanted: {"node": ">=12"} (current: {"node": "10.24.1", "npm": "6.14.12"})
npm WARN notsup Not compatible with your version of node/npm: value-or-promise@1.0.6
npm WARN notsup Unsupported engine for amplify-graphql-types-generator@2.8.3: wanted: {"node": ">=12.0.0"} (current: {"node": "10.24.1", "npm": "6.14.12"})
npm WARN notsup Not compatible with your version of node/npm: amplify-graphql-types-generator@2.8.3
npm WARN notsup Unsupported engine for amplify-graphql-docs-generator@2.2.4: wanted: {"node": ">=12.0.0"} (current: {"node": "10.24.1", "npm": "6.14.12"})
npm WARN notsup Not compatible with your version of node/npm: amplify-graphql-docs-generator@2.2.4
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules/@aws-amplify/cli/node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os": "darwin", "arch": "any"} (current: {"os": "linux", "arch": "x64"})

+ @aws-amplify/cli@5.0.0
added 1375 packages from 781 contributors in 81.282s
Admin:~ $
```

17. 次にインストールされた Amplify CLI の初期設定を行います。以下のコマンドを入力してください。

```
$ amplify configure
```

以下のメッセージが表示されたら Enter を押してください。

```
Admin:~ $ amplify configure
Follow these steps to set up access to your AWS account:

Sign in to your AWS administrator account:
https://console.aws.amazon.com/
Press Enter to continue
```

次に利用するリージョンを選択します。今回は東京リージョンを利用しますので、[ap-northeast-1]を矢印キーを利用し、選択します。

```
Specify the AWS Region
? region:
  eu-west-1
  eu-west-2
  eu-central-1
> ap-northeast-1
  ap-northeast-2
  ap-southeast-1
  ap-southeast-2
(Move up and down to reveal more choices)
```

次に Username の入力となりますので、ここでは[amplify-cli-user]と入力し、Enter を押します。

その後、URL が表示されますので、クリックして[Open]をクリックします。

```
Sign in to your AWS administrator account:
https://console.aws.amazon.com/
Press Enter to continue

Specify the AWS Region
? region: ap-northeast-1
Specify the username of the new IAM user:
? user name: amplify-cli-user
Complete the user creation using the AWS console
https://console.aws.amazon.com/ap-northeast-1/users/amplify-cli-user
Press Enter to continue
```

Open

Open In Preview

Copy

[Open]をクリックすると、AWS マネジメントコンソールが表示されますので、ログイン画面が表示された場合、ログインを実施します。以下、IAM ユーザの作成画面が表示されますので、IAM ユーザの作成を行います。

IAM ユーザ作成では、すでに必要な情報が入力・選択されているため、入力や、選択は不要となりますので、[次のステップ]を押していきます。

ユーザー詳細の設定

同じアクセスの種類とアクセス権限を使用して複数のユーザーを一度に追加できます。 [詳細はこちら](#)

ユーザー名*

[別のユーザーの追加](#)

AWS アクセスの種類を選択

これらのユーザーから AWS にアクセスする方法を選択します。アクセスキーと自動生成パスワードは前のステップで提供されています。 [詳細はこちら](#)

アクセスの種類* ☒ プログラムによるアクセス
AWS API、CLI、SDK などの開発ツールの アクセスキー ID と シークレットアクセスキーを有効にします。

☐ AWS マネジメントコンソールへのアクセス
ユーザーに AWS マネジメントコンソールへのサインインを許可するための パスワードを有効にします。

* 必須 [キャンセル](#) [次のステップ: アクセス権限](#)

IAM ユーザが作成されると、以下の画面が表示されますので、[CSV のダウンロード]を押し、ダウンロードを実施します。その後、ダウンロードした CSV ファイルを開きます。

[.csv のダウンロード](#)

	ユーザー	アクセスキー ID	シークレットアクセスキー
▶	amplify-cli-user	AKIA2DPB5C67YG73OTEE	***** 表示

IAM ユーザの作成が完了しましたので、Cloud9 の画面に戻ります。

先程、作成した IAM ユーザの[アクセスキーID]と[シークレットアクセスキー]をコピーし、Amplify CLI の初期設定にて入力する項目に対し、入力を実施します。最後に Profile Name を入力するため、[amplify-cli-user]と入力し、Enter を押します。こちらで Amplify CLI の初期設定が完了となります。

```
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.3.2 (node_modules/@aws-amplify/cli/node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})

+ @aws-amplify/cli@5.0.0
added 1375 packages from 781 contributors in 81.282s
Admin:~ $ amplify -v
Initializing new Amplify CLI version...
Done initializing new version.
Scanning for plugins...
Plugin scan successful
5.0.0
Admin:~ $ amplify configure
Follow these steps to set up access to your AWS account:

Sign in to your AWS administrator account:
https://console.aws.amazon.com/
Press Enter to continue

Specify the AWS Region
? region: ap-northeast-1
Specify the username of the new IAM user:
? user name: amplify-cli-user
Complete the user creation using the AWS console
https://console.aws.amazon.com/iam/home?region=ap-northeast-1#/users$new?step=final&accessKey&userNames=amplify-cli-user&permissionType=policies
Press Enter to continue

Enter the access key of the newly created user:
? accessKeyId: *****
? secretAccessKey: *****
This would update/create the AWS Profile in your local machine
? Profile Name: amplify-cli-user

Successfully set up the new user.
Admin:~ $
```

■アプリケーションの作成

1. Amplify を使い、アプリケーションを作成していきます。今回は React を利用し作成を行います。まずは以下のコマンドを Cloud9 で実行し、プロジェクトの作成を行います。

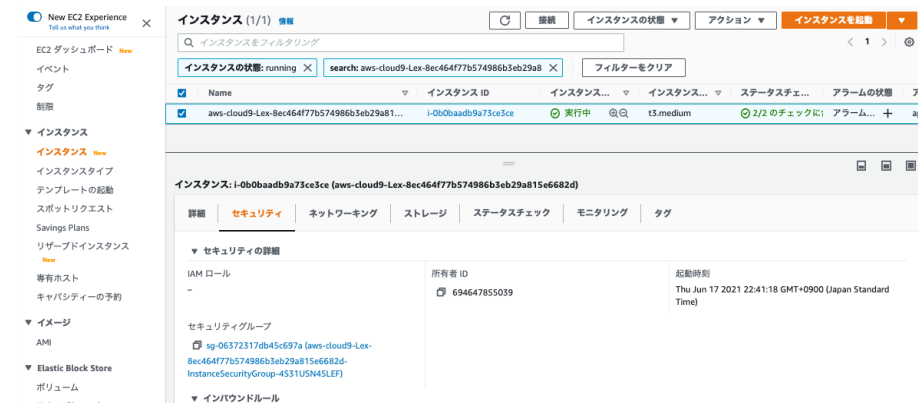
```
$ npm create-react-app lex-amplify
```

2. 次に作成されたプロジェクトのフォルダに移動します。(プロジェクト名 : lex-amplify)

```
$ cd lex-amplify
```

3. ここで動作確認を行いますが、Cloud9 の設定上、外部からのアクセスを許可していない状態なので、動作確認を行うためにセキュリティグループの設定を行います。

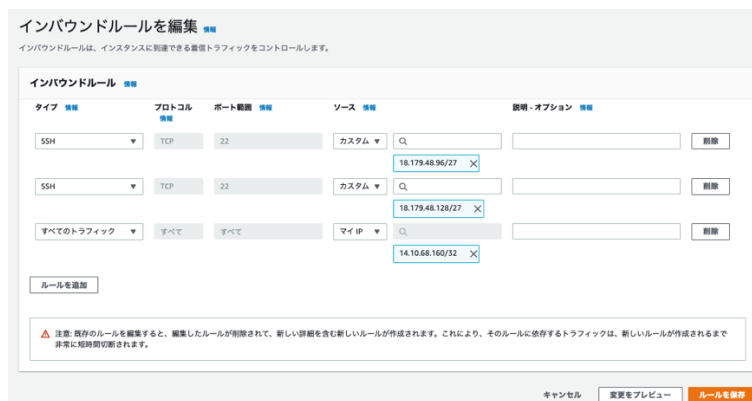
[EC2]の画面に移動し、左メニュー[インスタンス]を開き、現在稼働している Cloud9 のインスタンスを選択し、真ん中にある[セキュリティ]タブを選択し、設定されている[セキュリティグループ]を選択します。



[セキュリティグループ]を選択後、以下の画面になりますので、[インバウンドルールを編集]をクリックします。



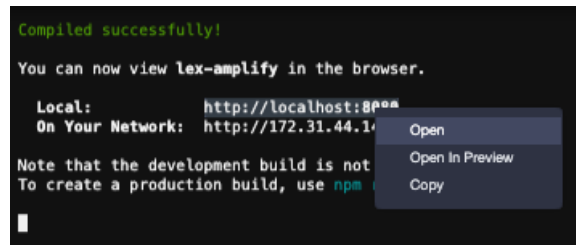
通信を許可するサービスポートや IP アドレスを入力できる画面になりますので、以下のように追加でタイプを[すべてのトラフィック]、ソースを[マイ IP]で選択をし、[ルールを保存]を押下する。これで動作確認の準備ができました。



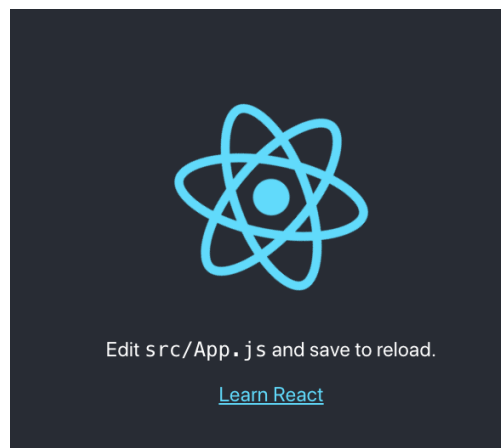
4. さっそく、先程作成したプロジェクトの動作確認をおこないましょう。Cloud9のエディタの画面に戻り、以下のコマンドを入力します。

\$ npm start

以下のようなメッセージが出るので[Local]をクリックするとポップアップが出ますので、[Open]を押してください。



ブラウザで別タブが立ち上がり、以下のような画面が表示されれば、アプリケーションの初期設定は完了です。続いて Lex の設定に移っていきます。



■Lex の設定

1. ここでは Lex を利用し、チャットボットを作成します。以下の URL より Lex のサンプルファイルをダウンロードします。

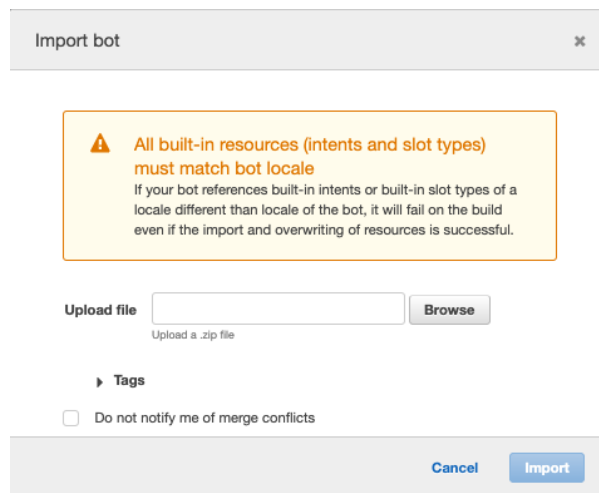
<https://bit.ly/2Uaxy4c>

ファイル名：「BookTrip_jaJP_LEX_V1.zip」

2. ダウンロードしたファイルは解凍せずに、[Lex]の画面からインポートを行います。[Actions]から[Import]を選択します。



インポートを行うファイルのアップロードを行う画面が表示されるので、先程ダウンロードしたファイルを選択し、[Import]をクリックします。



正常にインポートが完了しましたら、このように Status が[NOT_BUILT]の状態ではボットが追加されます。



3. 早速、ボットの中身を触っていきます。今回はサンプルとして、予約に関するボットに機能を追加します。

■インポートしたボット「BookTrip_jaJP」の中身について

- ・ホテルの予約
- ・レンタカーの予約

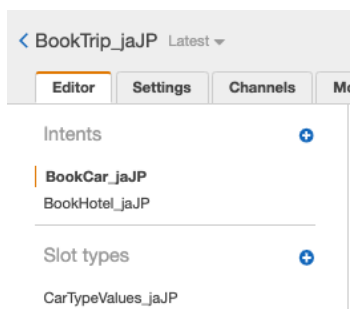
今回はここに[レストランの予約]を追加します。

4. 今回は、追加機能の Intent をインポートします。まず、以下の URL より、インポートを行う Intent のファイルをダウンロードします。

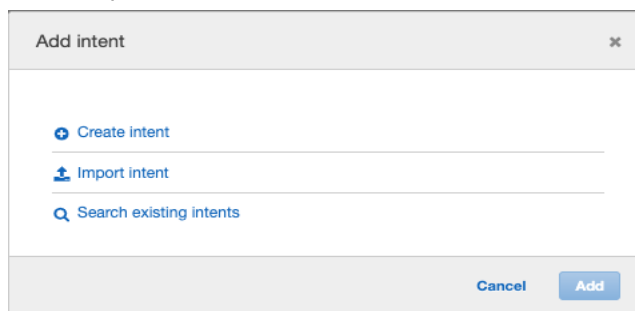
<https://bit.ly/2SLLFg6>

ファイル名：「MakeAppointment_jaJP_Intent_LEX_V1.zip」

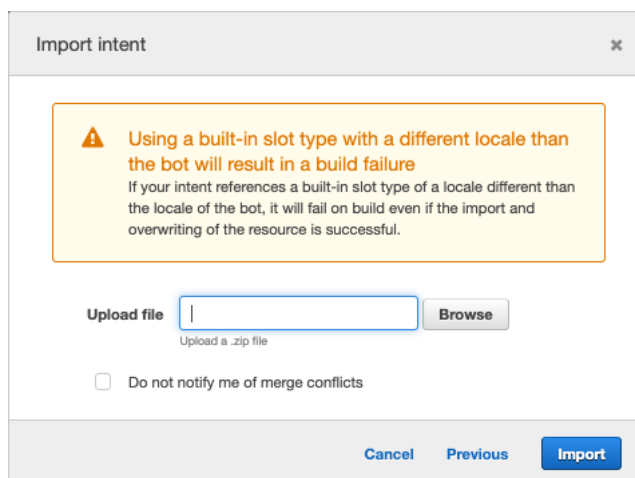
5. 次に、[BookTrip_jaJP]を選択すると、設定画面に移ります。画面左上の[Intents]が各予約のパターンの種類となるため、ここに[レストランの予約]に関する Intent を追加します。まず、Intents の[+]ボタンをクリックすると、Intent の追加画面が表示されます。



6. 次に Intent を追加する画面で[Import intent]を選択し、Intent をインポートします。(今回は時間の都合上、インポートとなりますが、手動で作成が可能です。)



7. Intent ファイルを選択し、アップロードを行う画面が表示されるので、先程ダウンロードしたファイルを Zip のまま選択し、[Import]をクリックします。



8. 上書きして続けるかの画面が表示されるので、[Overwrite and continue]をクリックし、設定を反映すると、Intents に [MakeAppointment_jaJP]が追加されます。

Import intent

⚠ Using a built-in slot type with a different locale than the bot will result in a build failure
If your intent references a built-in slot type of a locale different than the locale of the bot, it will fail on build even if the import and overwriting of the resource is successful.

Choosing Overwrite and continue will update the Latest version of these resources, and current data will be lost. The update will affect other bots that use these resources.

Intent	Slot type
MakeAppointment_jaJP	ReserveTypeValue

[Cancel](#) [Overwrite and continue](#)

9. 追加された Intent では、レストランの予約のため、以下の設定を確認します。また、今回はハンズオンですので、試しに設定変更を行っても問題ありませんので、ぜひ好きな値に変更してみてください。この際、設定を反映させるために画面下にある[Save Intent]のクリックは忘れないで実施してください。

[Sample utterances] : サンプル発話

[Slots] : チャットでやり取りを行う順序や値

[Slot types] : 選択肢の定義

[Confirmation prompt] : 確認時のメッセージ

[Response] : 最終メッセージ

10. 設定変更が完了したら、最後に公開を行います。画面右上[Build]をクリックし、Build が完了したら[Publish]をクリックします。Alias を聞かれたら、好きな名前を Alias に指定してください。これで Lex でチャットボットの公開が完了です。

Publish BookTrip_jaJP

Publishing is the last step before you can connect your bot to your mobile app or chatbot.

Choose an alias ⓘ

[Create a new alias](#)

[Cancel](#) [Publish](#)

■Lex と Amplify の統合

1. 最後に先程作成した Lex チャットボットを Amplify に統合していきます。まずは Cloud9 を起動し、エディタの画面にて操作します。作業場所はプロジェクト[lex-amplify]のフォルダで実施します。※もし、作業場所がわからない場合、以下のコマンドを入力してください。

```
$ cd /home/ec2-user/lex-amplify
```

2. 次に Amplify でプロジェクトの初期化を行います。プロジェクト初期化は以下のコマンドを入力します。

```
$ amplify init
```

3. 次にプロジェクト名を聞かれますので、デフォルトのまま Enter を押下する。すると、プロジェクト情報の一覧が表示されるので、ここで Enter を押下する。

```
Admin:~/lex-amplify (master) $ amplify init
Note: It is recommended to run this command from the root of your app directory
? Enter a name for the project lexmlamplify
The following configuration will be applied:

Project information
| Name: lexmlamplify
| Environment: dev
| Default editor: Visual Studio Code
| App type: javascript
| Javascript framework: react
| Source Directory Path: src
| Distribution Directory Path: build
| Build Command: npm run-script build
| Start Command: npm run-script start
? Initialize the project with the above configuration? (Y/n) █
```

次に、どの認証情報で初期化作業を実施するかを選択する必要があるので、[AWS Profile]を選択します。

ユーザーは設定時に作成した[amplify-cli-user]を選択してください。すると CloudFormation が起動し、デプロイが開始されます。

```
? Initialize the project with the above configuration? Yes
Using default provider awscloudformation
? Select the authentication method you want to use: AWS profile

For more information on AWS Profiles, see:
https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-profiles.html

? Please choose the profile you want to use (Use arrow keys)
> amplify-cli-user █
```

4. デプロイが完了したら、次に Amplify に認証のサービスを追加します。(Amplify でチャットボットなどのやり取りは Cognito 認証の機能が必要となるためです。)

```
$ amplify add auth
```

設定内容を聞かれるので、[Default configuration]→[Username]→[No, I am done.]の順にデフォルトのまま選択する。

```
Admin:~/lex-amplify (master) $ amplify add auth
Using service: Cognito, provided by: awscloudformation

The current configured provider is Amazon Cognito.

Do you want to use the default authentication and security configuration? Default configuration
Warning: you will not be able to edit these selections.
How do you want users to be able to sign in? Username
Do you want to configure advanced settings? No, I am done.
Successfully added auth resource lexmlamplifyadd8cb41 locally
```

5. 無事、Amplify に認証の機能設定が完了されると、以下のコマンドでデプロイを実施します。

```
$ amplify push
```

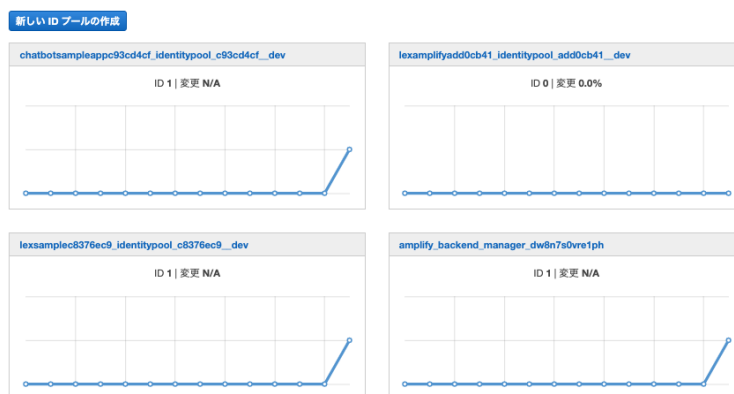
[Are you sure you want to continue?]

というメッセージが表示されるので、Yes と入力し、Enter を押下する。するとデプロイが開始されます。

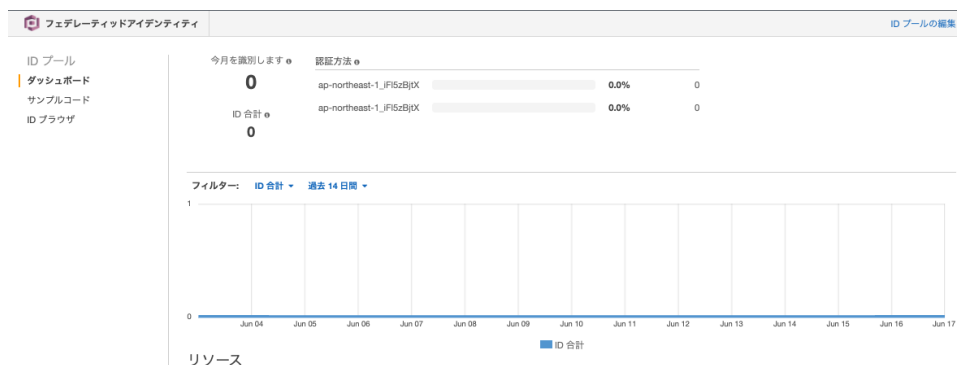
6. 続いて、マネジメントコンソールから[Cognito]の画面を開いて[ID プールの管理]をクリックします。



Amplify のプロジェクト名がついている ID プールをクリックします。(lexamplify から始まる ID プール名となるかと思います。)



右上の[ID プールの編集]をクリックします。



[認証されていない ID]をクリックすると、[認証されていない ID に対してアクセスを有効にする]のチェックボックスが表示されますのでチェックを入れて、一番下にある[変更の保存]をクリックします。また、上に表示されている[認証されていないロール]に対して、アクセス権限を付与しますので、ロール名をメモしておきます。この場合、[**amplify-lexamplify-dev-142236-unauthRole**]が対象となります。また、最後に **ID プールの ID** も利用するので、このタイミングでメモしておく

と後で楽に作業が実施できます。

ID プールの編集

このページからは、ID プールの詳細を変更できます。ID プールには、一意の名前と、認証されたロールおよび認証されていないロールのセットが必要です。ロールは ID プールで保存され、ユーザーの承認リクエストを受け取るたびに、ここで指定されるロールが自動的に利用されます。Amazon Cognito クライアント SDK を初期化するとき、このページから ID プールの ID を指定する必要があります。Amazon Cognito での IAM ロールの使用に関する詳細を参照してください。

ID プール名

ID プールの ID (ARN の表示)

認証されていないロール [新しいロールの作成](#)

認証されたロール [新しいロールの作成](#)

▼ 認証されていない ID

Amazon Cognito は、ID プロバイダーを使って認証を行わないユーザー用に、一意の識別子や AWS 認証権限を提供して、認証されていない ID をサポートできます。アプリケーションで、ログインすることなくユーザーにアプリケーションの使用を許可している場合、認証されていない ID に対してアクセスを有効にできます。[認証されていない ID の詳細を参照してください。](#)

☒ 認証されていない ID に対してアクセスを有効にする

7. 次に[IAM]に移動し、[ロール]より、上でメモをしたロールに対し、ポリシーを付与します。

ロール ARN [編集](#)

ロールの説明 [編集](#)

インスタンスプロファイル ARN

パス

作成時刻 2021-06-17 23:22 UTC+0900

最後のアクティビティ 追跡期間中はアクセスされません

最大セッション時間 1 時間 [編集](#)

アクセス権限 信頼関係 タグ (2) アクセスアドバイザー セッションの無効化

▼ Permissions policies (1 適用済みポリシー)

[ポリシーをアタッチします](#) [インラインポリシーの追加](#)

ポリシー名	ポリシータイプ
Order	インラインポリシー

ポリシー概要 {} JSON ポリシーの編集 ポリシーのシミュレート

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "lex:PostText"
8       ],
9       "Resource": "arn:aws:lex:ap-northeast-1:*:bot:Order:$LATEST"
10    }
11  ]
12 }
```

IAM ポリシーは以下を参考にしてください。チャットボット名を入力し、必要最低限のボットしか実行できないようにしています。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lex:PostText"
      ],
      "Resource": "arn:aws:lex:ap-northeast-1:*:bot:{チャットボットの名前}:$LATEST"
    }
  ]
}
```

8. 次に、Cloud9 に戻り、src/app.js と src/index.js を以下の Github に掲載している内容に書き換え、src/app.js で XXX となっている箇所に入力が必要となるので、以下の項目の入力を行います。

■ Github に掲載している app.js / index.js

<https://bit.ly/3iVUpee>

identityPoolId : Cognito の画面でメモをした[ID プールの ID]を入力

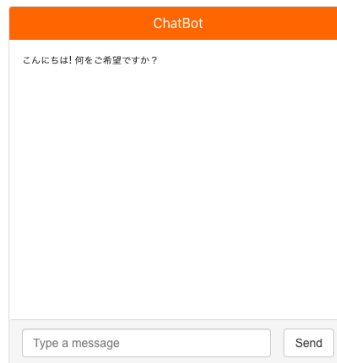
その他の箇所 : Lex のチャットボット名

9. 最後の Amplify ホスティングの前に以下のコマンドで動作確認を行います。

```
$ npm install aws-amplify aws-amplify-react
```

```
$ npm start
```

動作確認の画面で、以下のようなチャットボックスが表示され、実際にテキストボックスに内容を入力し、やり取りができれば、動作確認完了です。



10. 最後に Amplify ホスティングを行い、外部に公開しましょう。以下のコマンドを入力し、ホスティングの機能を追加します。ここもデフォルトの値で Enter を押下します。

```
$ amplify add hosting
```

```
Admin:~/lex-amplify (master) $ amplify add hosting
? Select the plugin module to execute Hosting with Amplify Console (Managed hosting with custom domains, Continuous deployment)
? Choose a type Manual deployment

You can now publish your app using the following command:
Command: amplify publish
Admin:~/lex-amplify (master) $
```

そして、以下のコマンドで公開を実施できます。

```
$ amplify publish
```

ビルドとデブロイが走りますので暫く待つと、完了とともに、URL が表示されますので、その URL に対し、アクセスを行うと、チャットボックスのページが表示され、Amplify にてホスティングが完了していることが確認できます。

```
✓ Zipping artifacts completed.
✓ Deployment complete!
https://dev.d4fmegndam870.amplifyapp.com
Admin:~/lex-amplify (master) $
```

■ リソースの削除

1. 最後に作成したリソースの削除を行しましょう。削除については、以下のリソースが削除対象となります。

- Amplify
- Lex
- Cloud9
- IAM ロール

これですべての作業が終了となります。お疲れさまでした！