AWS Step Functionsワークショップ

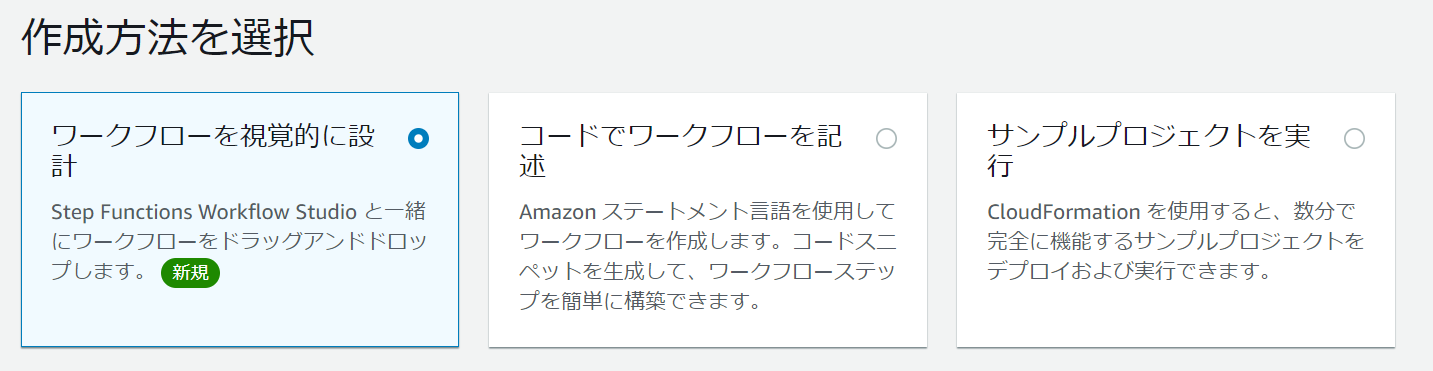
2022/07/28

シニアエバンジェリスト 亀田

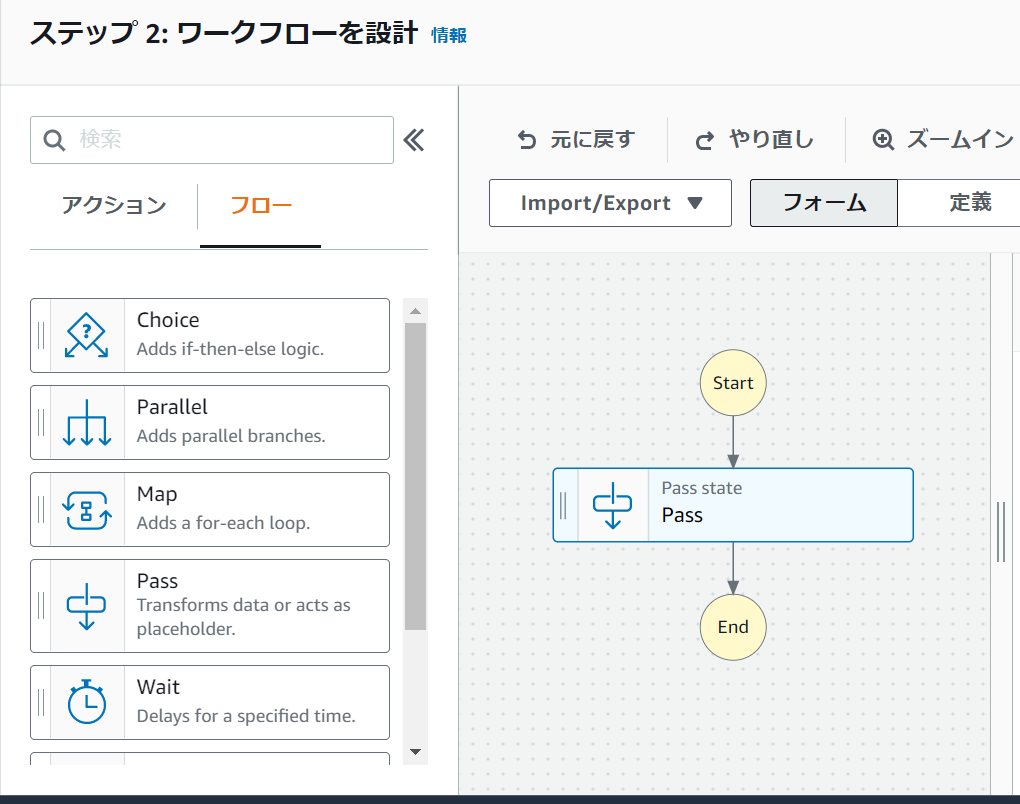
[Hello World]の作成

Lambdaを使わずにStep FunctionsによるHello Worldです

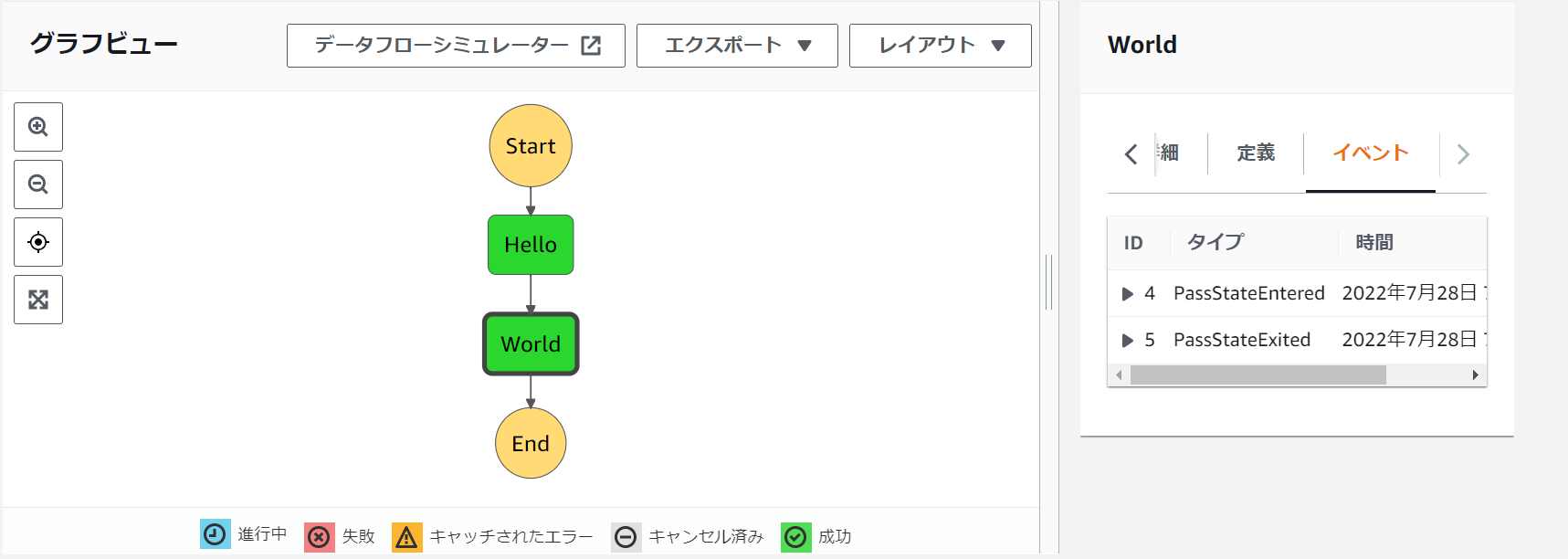
1. Step Functionsのマネージメントコンソールに移動します
2. [ステートマシンの作成]をおします
3. [ワークフローを視覚的に設計]を選びます



1. [次へ]をおします
2. 左ペインから[フロー]のタブを選び、[Pass]を真ん中にドラッグします。以下のようになります



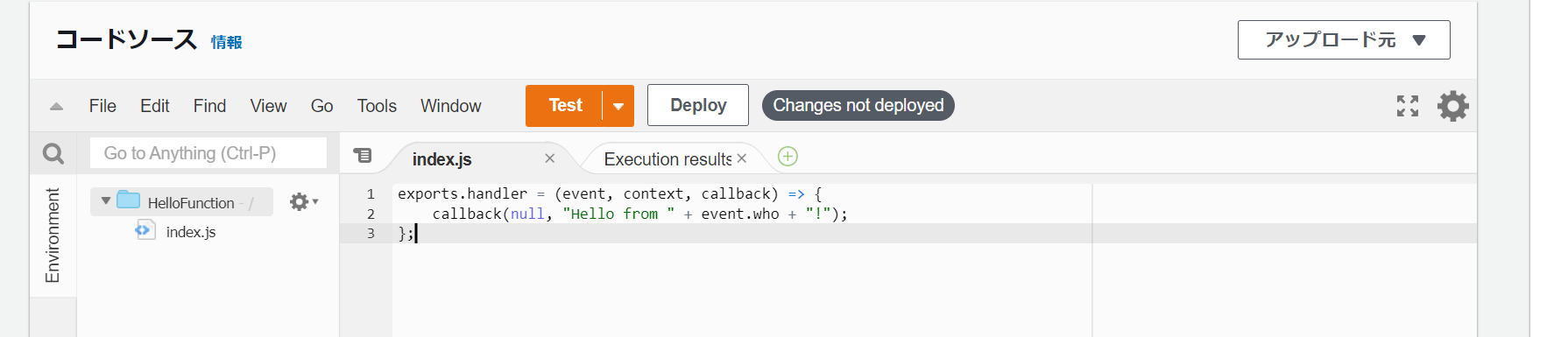
1. [次へ]をおします
2. コードスニペット部分をcommands.txtの1番に置き換えます
3. [次へ]をおします
4. 名前に[HelloWorld]と入力します
5. [ステートマシンの作成]をおします
6. [実行の開始]をおします
7. もう一度[実行の開始]をおします
8. 実行結果が出力されますので、グラフビューなどを眺めてください。また[Table View][Event View]等何が出力されているかも確認してください



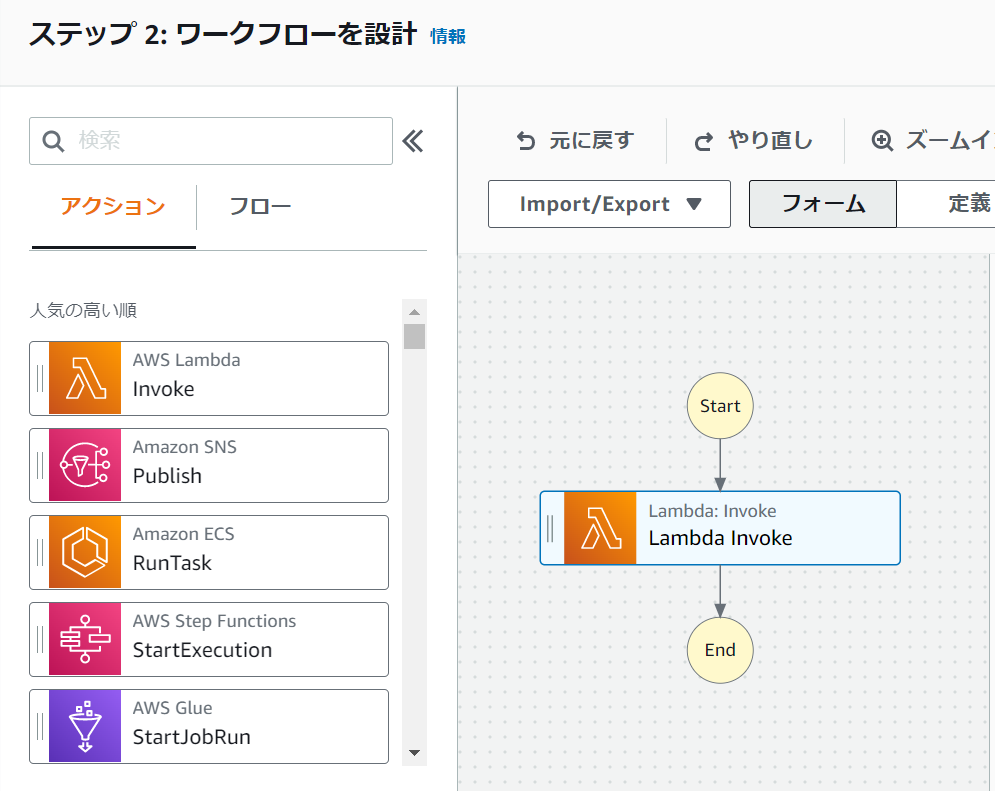
Lambdaの起動

ここからの手順ではStep FunctionsからLambda関数を起動するステートマシンを作成していきます

1. ブラウザ別タブでLambdaのマネージメントコンソールに移動します
2. [関数の作成]をおします
3. [HelloFunction]と関数名に入力します
4. [関数の作成]をおします
5. 関数が作成出来たら、コードソースのindex.jsをcommands.txtの2番に置換して[Deploy]をおします

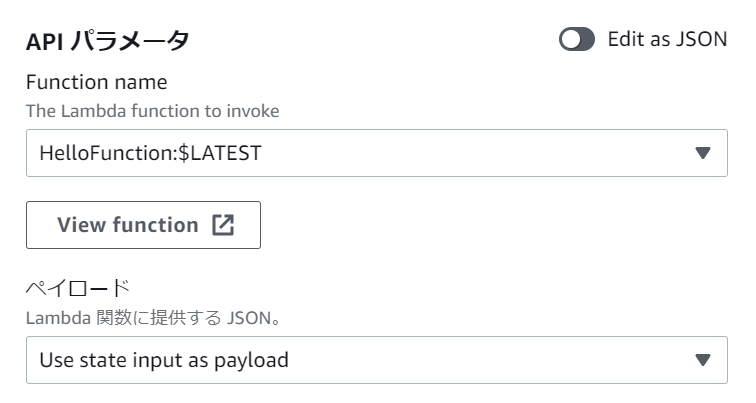


1. [Test]をおし、イベント名に[test]と入力します。[イベントJSON]にcommands.txt3番をコピペして、[保存]をおします
2. もう一度[Test]をおします。"Hello from AWS Step Functions!"と表示されたら成功です。
3. Step Functionsの画面に戻り[ステートマシンの作成]をおします
4. [次へ]をおします
5. アクションから[Lambda Invoke]を真ん中にドラッグします



ついでに、サポートされているAWSサービスを眺めてください。主要サービスがかなり多くサポートされていることがわかります。

1. 右の設定タブから[Function name]に先程作成したLambda関数をセットします



1. [次へ]をおします
2. [次へ]をおします
3. 名前に[LambdaStateMachine]と入力し[ステートマシンの作成]をおします
4. [実行の開始]をおします。次のダイアログでも同様に[実行の開始]をおします

グラフビューの[Lambda Invoke]の個所が進行中を表す水色になっていますが、ほどなく成功を表す緑色に代わります。

1. 出力が[“Hello from undefined”]になっています。Lambda関数にInputを渡さなかったせいです



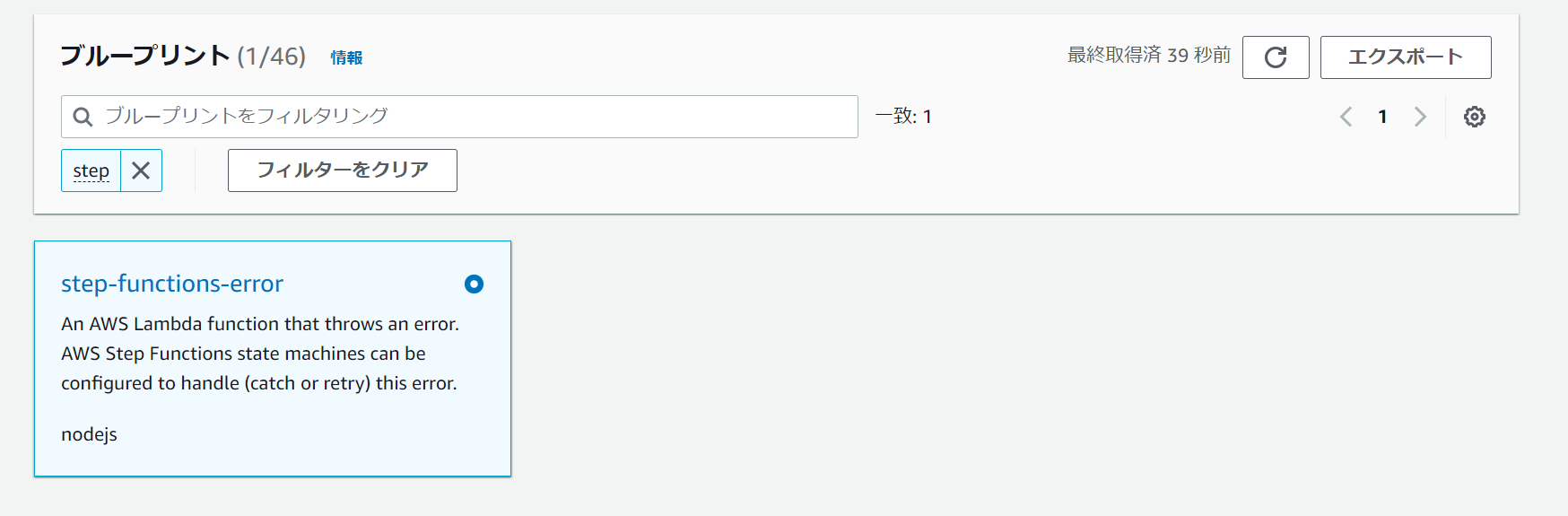
1. [新しい実行]のボタンをおします
2. 入力のcommands.txt3番で置換します（先ほどLambdaのテストにセットしたものと同じものです）
3. [実行の開始]をおします
4. 今度は正しく出力されています



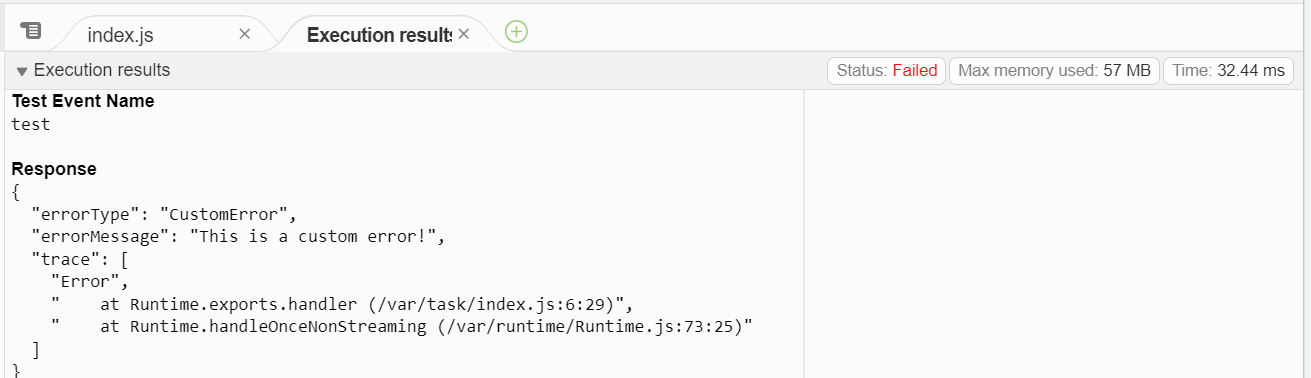
Catchによるエラー処理

次のステップでは意図的に実行が失敗するLambda関数を作成しエラーイベントを発生させ、Catch処理により次のLambda関数を実行させます

1. Lambdaマネージメントコンソールから[関数の作成]をおします
2. [設計図の使用]を選びます
3. [step-functions-error]のブループリントを選び[設定]をおします



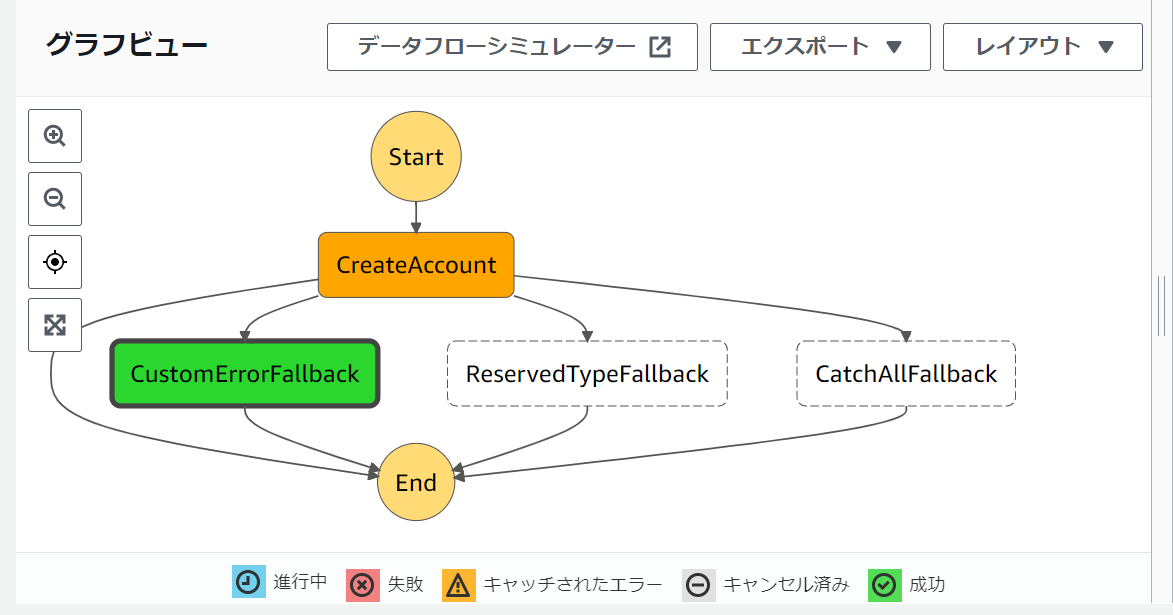
1. 名前に[FailFunction]とつけます
2. [関数の作成]をおします
3. [Test]をおします
4. イベント名に[test]と入力し、[保存]をおします
5. 再度[Test]を実行します。以下のように意図したエラーが表示されます



1. Lambda関数のARNをコピーしておきます



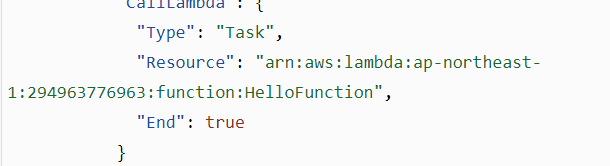
1. Step Functionsのマネージメントコンソールから[ステートマシンの作成]をおします
2. [コードでワークフローを記述]をおします
3. Commands.txtの4番の内容に置換します。その際必ずARNの値を先程コピーしたものに置き換えてください
4. このステートマシンでは[CreateAccount]をまず実行し、失敗したらエラーに応じて3つの処理のどれかを条件に応じて実行させます。[CreateAccount]を作成しておらず当然実行が失敗しますがこのサンプルでは[CustomErrorFallback]が先程のLambda関数にイベントとして引き渡され関数が実行されます。
5. [次へ]を押して[Catchfailure]を名前をつけ[ステートマシンの作成]をおします
6. [実行の開始]をおします。もう一度[実行の開始]をおします
7. 以下のようにエラーがキャッチされLambda関数が実行されたことがわかります

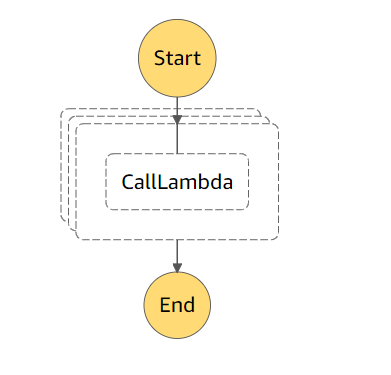


Map機能にするLambda関数の複数回実行

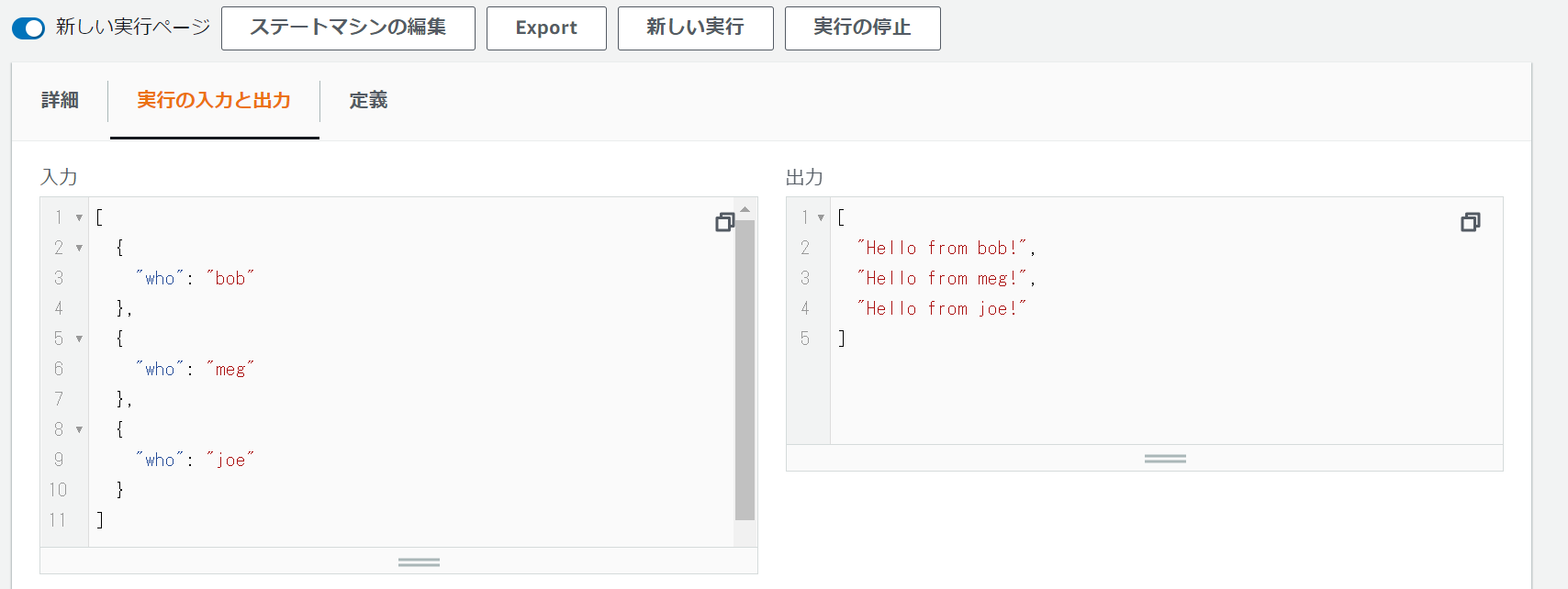
ステートマシンではあらかじめ実行回数がわからず、与えられたパラメータの数（行数）により任意の複数回Lambda関数を実行したいケースがあります。Map機能ではLambdaを任意の複数回実行し結果を結合して表示させることが可能です

1. [ステートマシンの作成]をおします
2. [コードでワークフローを記述]を選びます
3. Commands.txtの5番でコードを置換します。この際Lambda関数[Hellofunction]のARNを必ず置換します





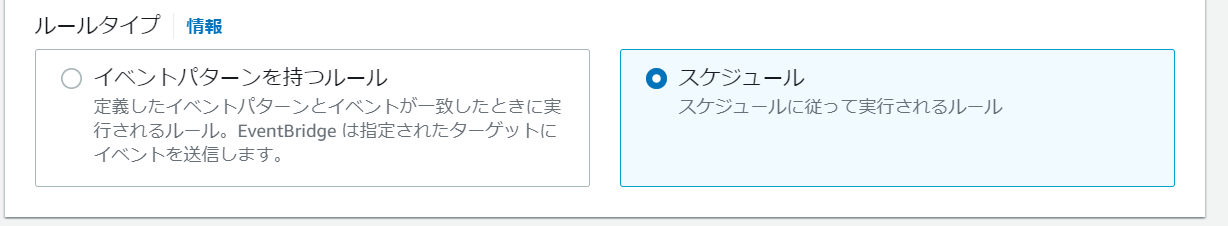
1. [次へ]をおします
2. 名前に[ExampleMapState]と入力し[ステートマシンの作成]をおします
3. [実行の開始]をおします。JSONの内容をcommands.txt 6番で置換し、[実行の開始]をおします
4. Lambda関数が3回実行され出力が3行行われています



EvenBridge連携

Step FunctionsはEventBridgeと連携することで仇められた時間での実行化が可能です

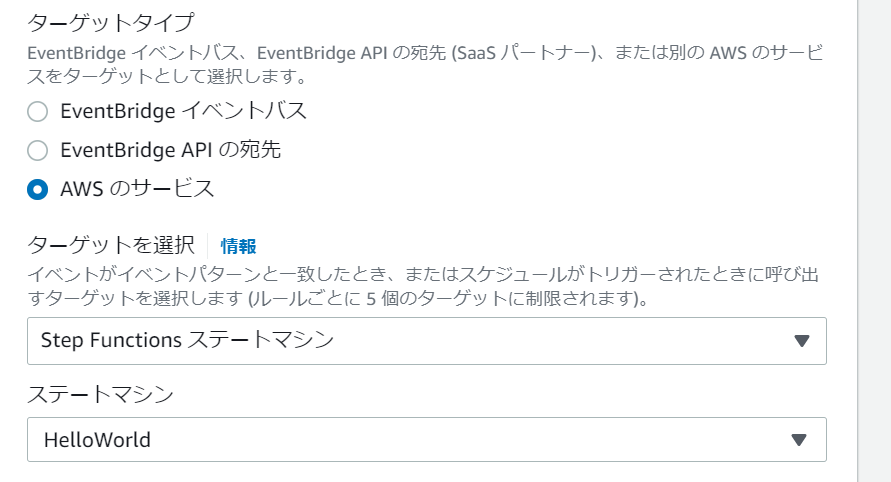
1. ブラウザの新しいタブでEvntBridgeを開きます
2. [ルールを作成]をおします
3. 名前に適当な文字を入力します
4. ルールタイプでスケジュールを選びます



1. [次へ]をおします
2. Cronを以下のように10-分後ぐらいに実行されるように設定します



1. ターゲットを以下のように設定します。ステートマシン名は任意のもので大丈夫です

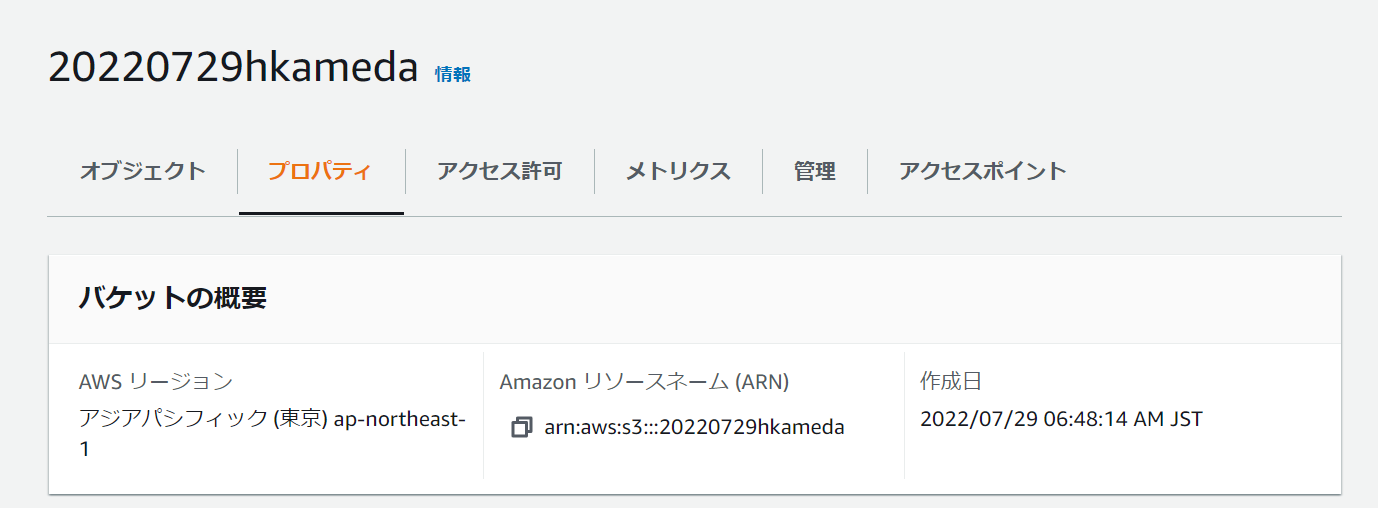


1. [次へ]を2回押します。その後[ルールの作成]をおします
2. ルールのモニタリングタブを開き実行されるのを待ちます。最大1分程度のずれがあります。あらにそこからグラウが表示されるまで数分の遅延が発生します。
3. 実行が確認出来たら[削除]をおしルールを削除しておきます

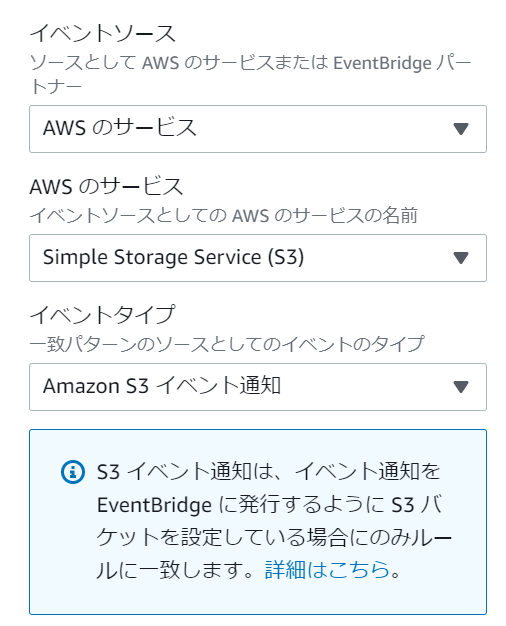
S3との連携

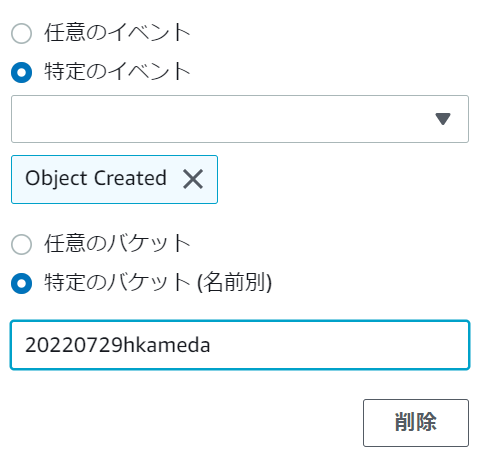
次にS3バケットのイベントをEventBridgeで取得しStateMachineを起動します

1. S3のマネージメントコンソールに移動し、[バケットを作成]をおし、適当名前でバケットを作成します
2. 先程作成したバケットのプロパティ画面に移動します

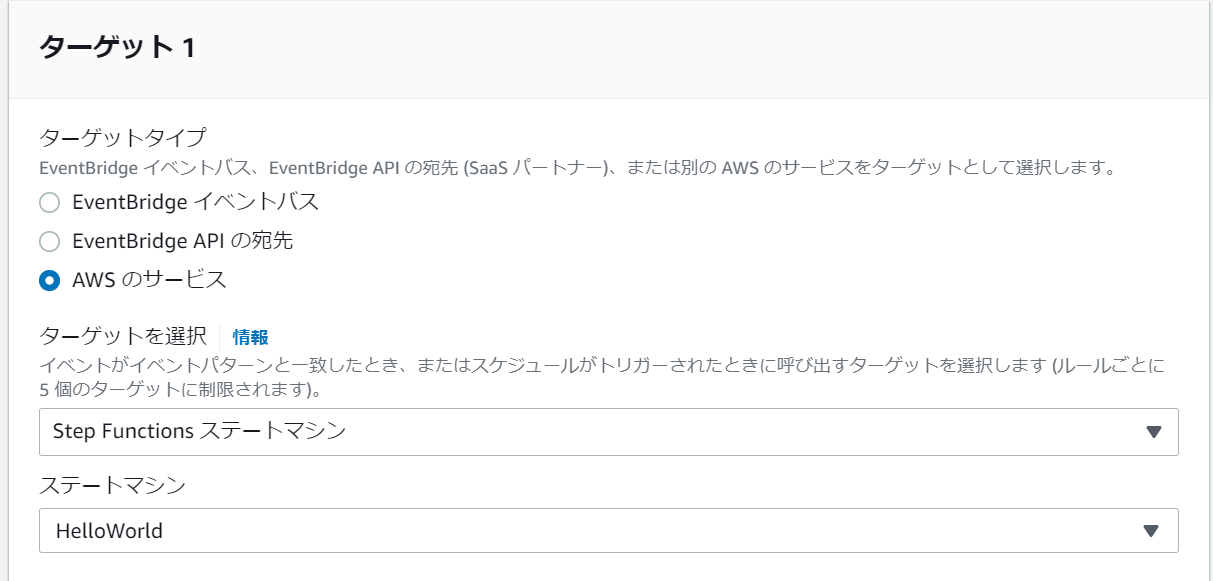


1. イベント通知の個所で、[編集]ボタンをおします
2. [オン]を選び[変更の保存]をおします
3. EventBridgeのマネージメントコンソールで[ルールを作成]をおします
4. 名前に[S3StepFunctions]と入力します
5. [次へ]をおします
6. イベントパターンの個所を以下のように設定します





1. [次へ]をおします
2. ターゲットの個所で、任意のStep Functionsステートマシンを選びます



1. [次へ]を2回押し、[ルールの作成]をおします
2. 何でもよいのでファイルをS3にブラウザ経由マネージメントコンソールからアップすると、EventBridgeを介してStepFunctionsのステートマシンが実行されます。

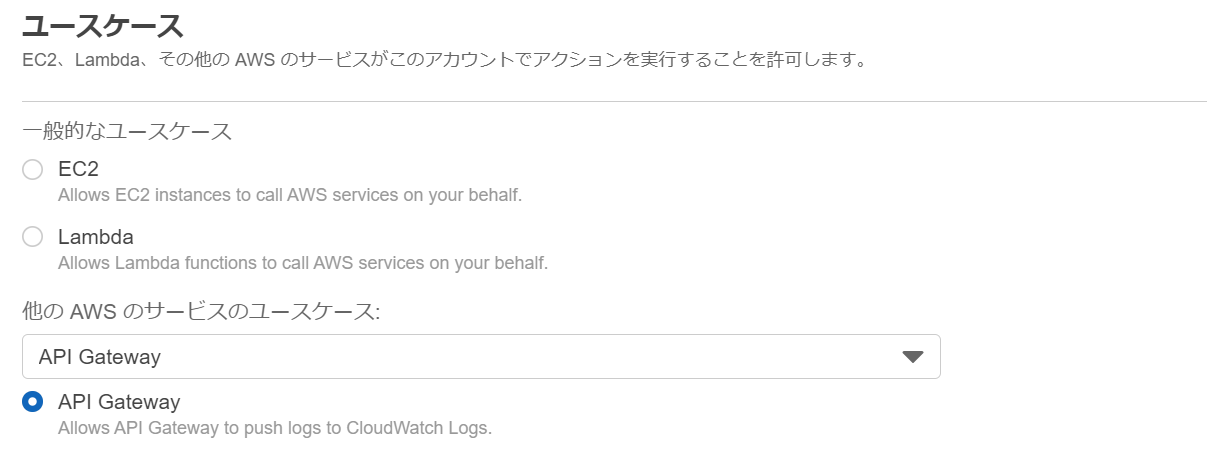
StepFunctionsステートマシンの画面で最新実行日時が確認できます



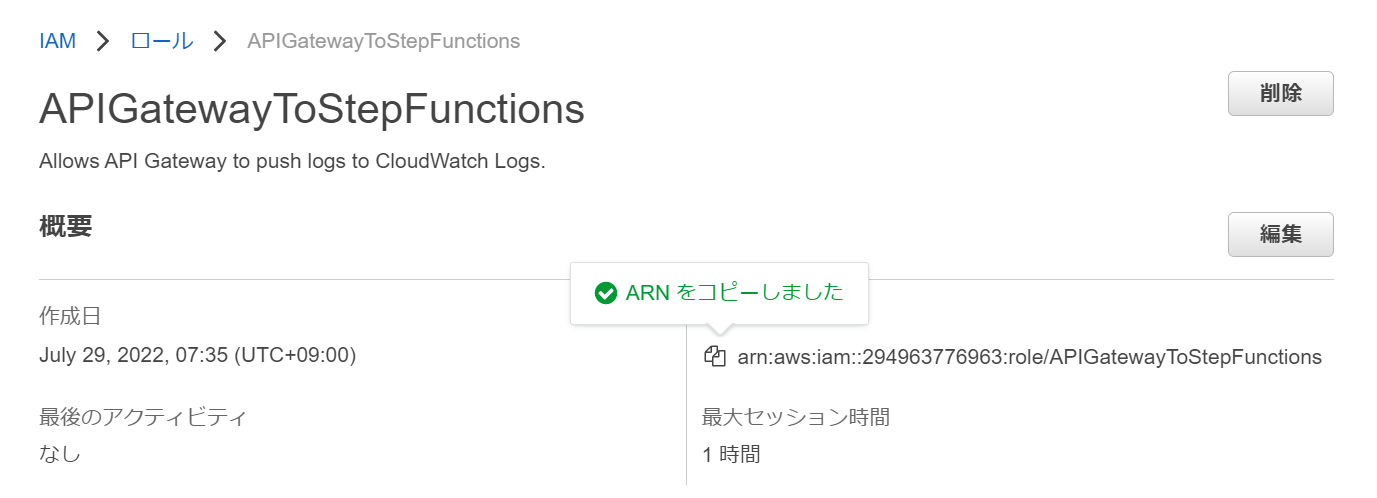
EventBridgeのモニタリングは5分程度遅延したのち反映されます

次にAPI Gateway経由でStepFunctionsのステートマシンを起動してみます

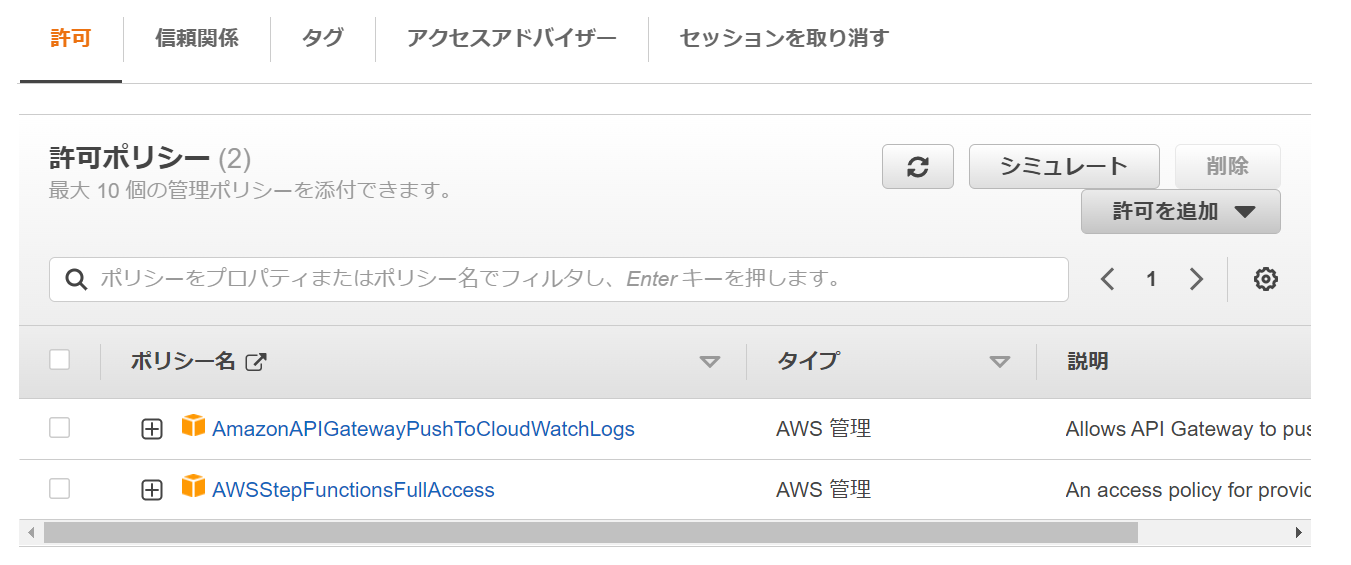
1. ブラウザ別タブでIAMマネージメントコンソールに移動します
2. [ロールを作成]をおします
3. ユースケースでAPI Gatewayを選び、[次へ]を2回おします



1. 名前に[APIGatewayToStepFunctions]と入力し[ロールを作成]をおします。以下に表示されるARNをコピーしておいてください



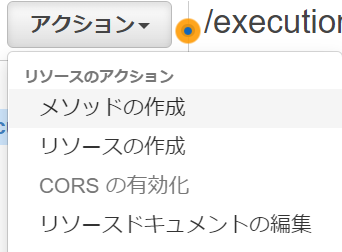
1. [許可の追加]から[AWSStepFunctionsFullAccess]を付与します



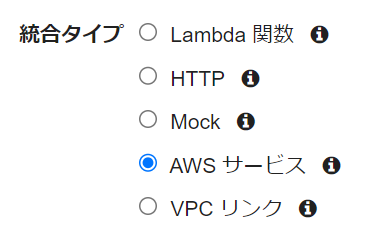
1. ブラウザの別タブでAPI Gatewayにアクセスします
2. [APIの作成]をおします
3. REST APIの[構築]をおします
4. 名前に[StartExecutionAPI]と入力します
5. [APIの作成]をおします
6. [アクション][リソースの作成]をおします



1. リソース名に[execution]と入力します
2. [リソースの作成]をおします
3. 今度はメソッドの作成を選びPOSTメソッドを作成します

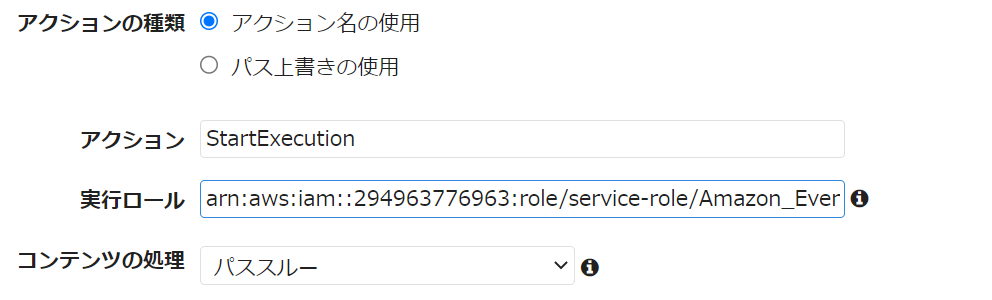


1. 統合タイプで[AWSサービス]を選びます



1. 以下の画面に従い同じ設定を行います



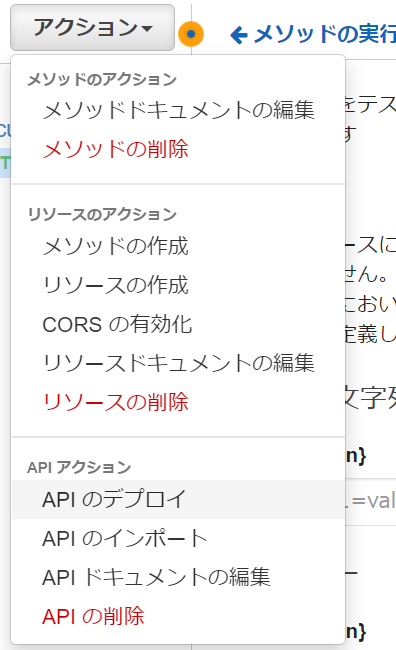


IAMロールARNは先程作成したものを 貼り付けてください

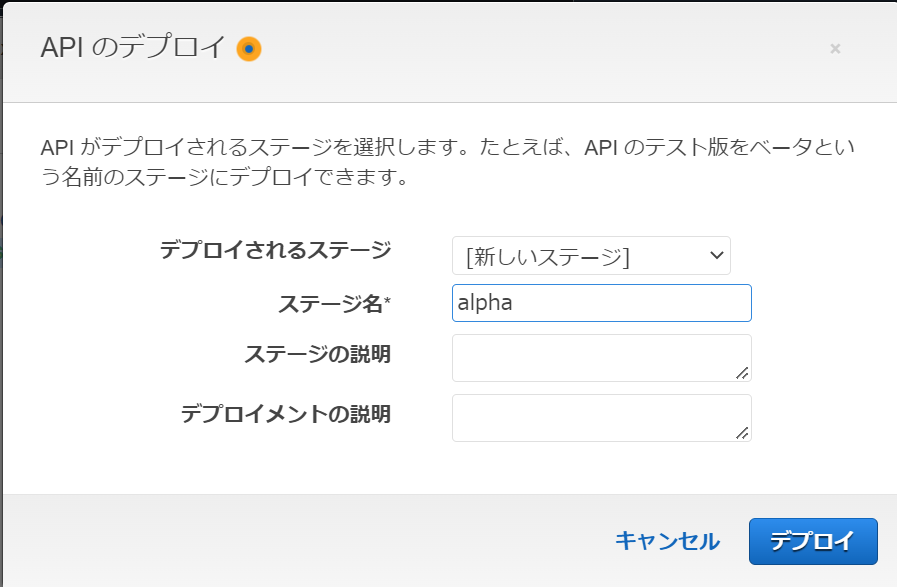
1. [保存]をおします
2. Commands.txtの7番を[リクエスト本文]に貼り付けて[テスト]をおします。この際ARNはStepFunctionsのHelloWorld SatemachineのARNを入力してください
3. テストが以下のように戻りになれば設定完了です



1. [アクション]から[APIのデプロイ]をおします



1. 以下のように設定し[デプロイ]をおします



1. Curlでcommands.txt 8番を実行してみます。ARN、HTTPS二つの値は皆さんの環境の値で置き換えてください（Already existsのエラーが戻ってきた人はMyExecution2の値を適当に変えてください

おつかれさまでした！：

削除は以下を行ってください

S3バケット

Step Functionsステートマシン

IAMロール

EventBridgeルール

Lambda関数

API Gateway