**Part1: AWSで最も早くサーバを立ち上げる方法**

1. AWSのManagement ConsoleでLightsailを検索して選択



1. インスタンスの作成ボタンを押す



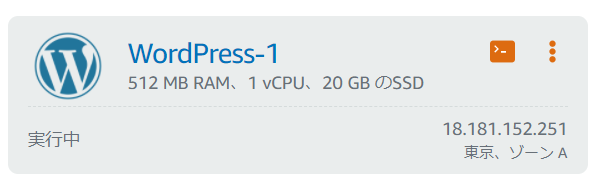
1. Linux、WordPressを選んで画面を下にスクロール。その他設定は全てそのままで一番下の「インスタンスの作成」ボタンを押す。





1. インスタンスが起動中なのでしばらく待ちます。実行中になれば起動完了です。

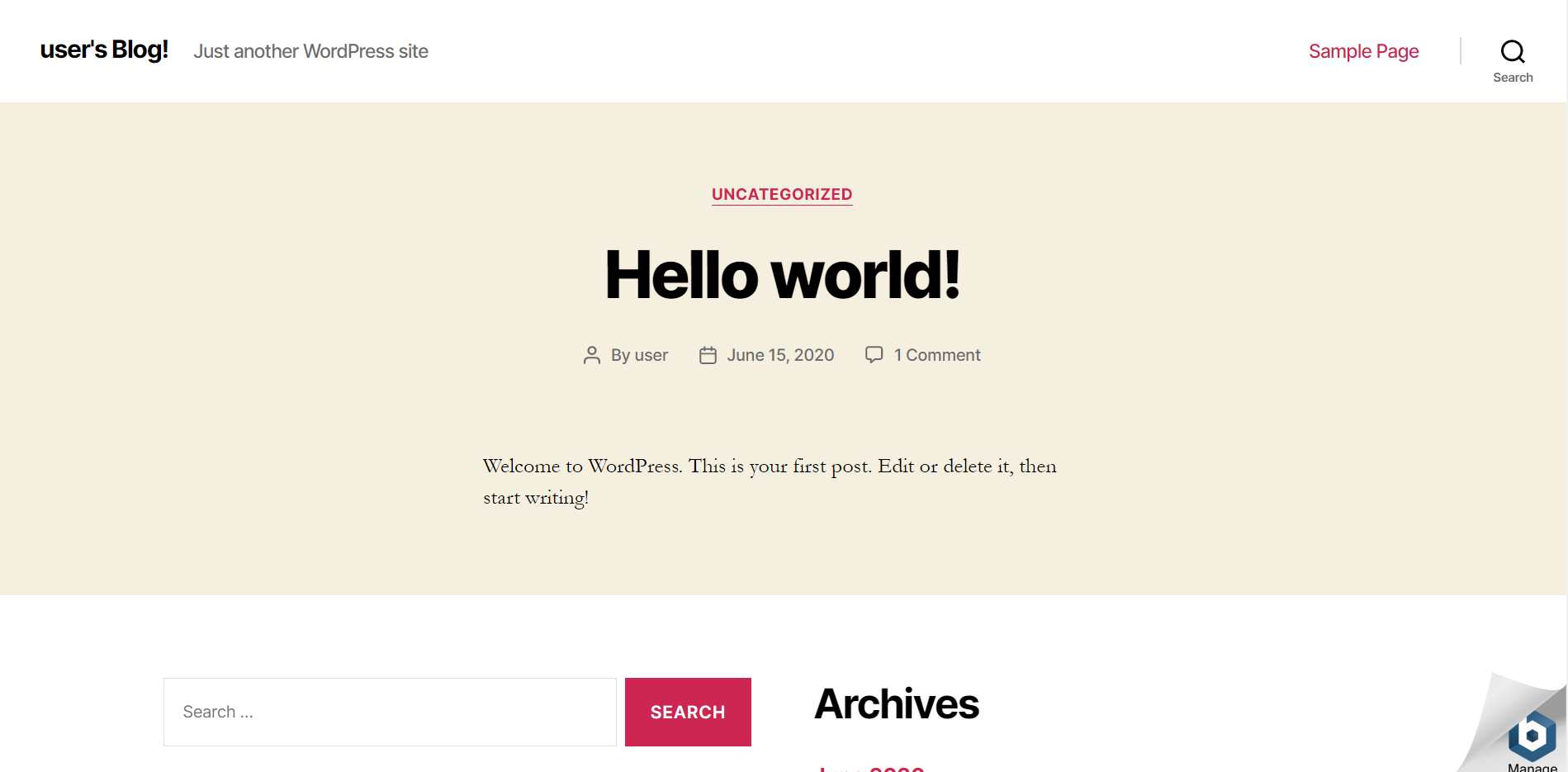




1. コマンドプロンプトを起動するためにオレンジ色の四角いボタンを押します。（[>\_]がオレンジ色の箱の中に記載されているボタンです）以下のダイアログが表示された場合は、まだ内部の初期化処理中ですのでしばらくまちます。何度か「再接続」のボタンを押してみてください。



1. 以下の画面が表示されれば起動は完了です。最新のブラウザではセキュリティ対策のため、この画面で直接コマンド入力ができないケースがあります。その場合、汎用のSSH用ツールを使ってログインし操作することができます。
2. ブラウザでIPアドレスを直接入力しアクセスできていることを確認します。（この例だと18.181.152.251）



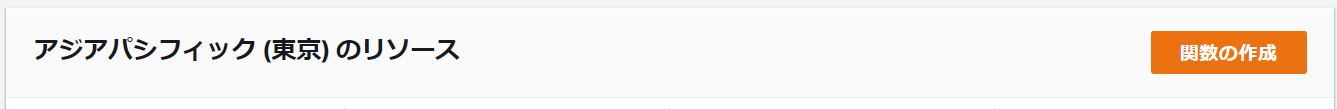
**Part2.サーバレスでHello Worldを実行**

1. AWSのマネージメントコンソールでLambdaを検索して選択。



２．関数の作成ボタンを押す。（初めて起動する方は違う画面が表示されるケースもあります。）その後画面一番下の「関数の作成」を押します。

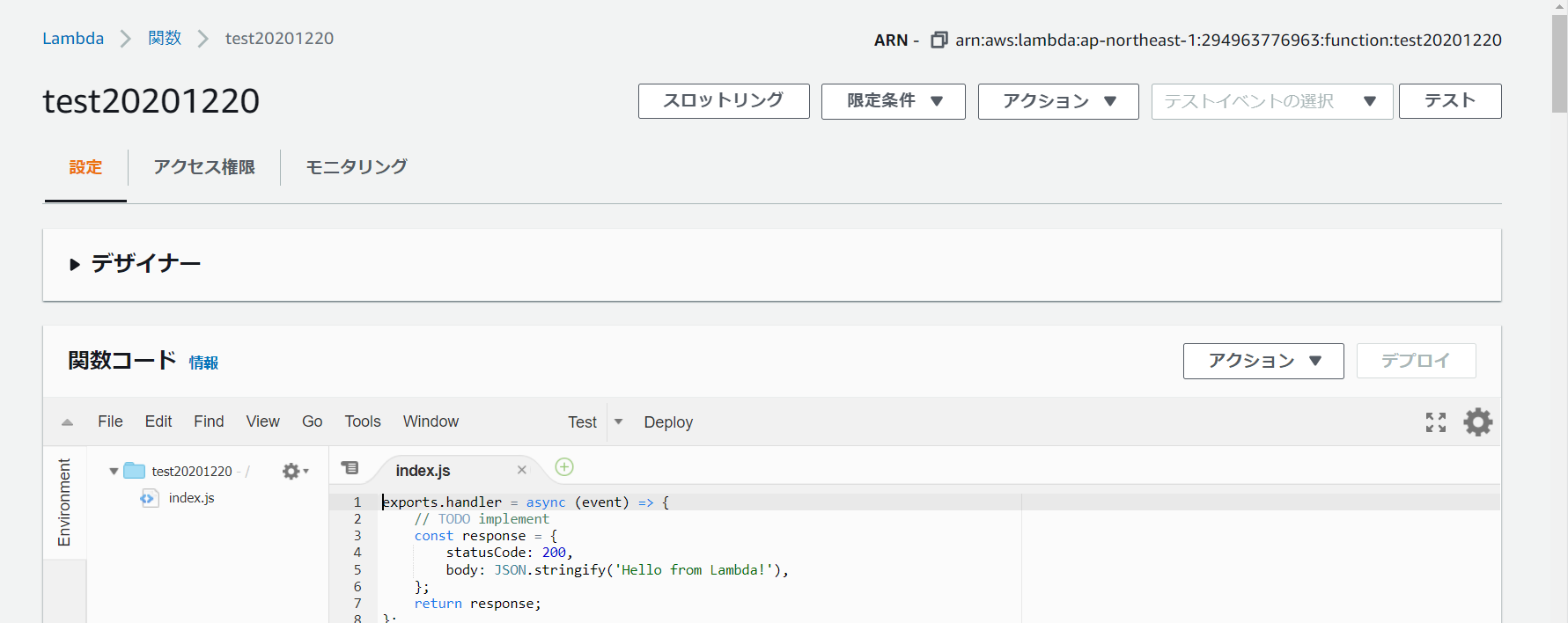




３．「一から作成」を選んで適当な名前を入れます。



４．あらかじめデフォルトでHello Worldを出力するプログラムが入っています。「テスト」ボタンを押します。



５．イベント名に適当な名前を付けて、設定は全てデフォルトのまま画面下の「作成」を押します。



６．もう一度「テスト」ボタンを押します。

実行結果の詳細に「Hello World!」と表示されていれば成功です。

Part3.LambdaからDynamoDBへデータを書き込む

1. AWS マネジメントコンソールのサービス一覧から **[DynamoDB]** を選択します。

**[テーブルの作成]** をクリックします。



1. **以下の通り入力します。**

* **テーブル名:** [*YYYYMMDD*serverless] （*YYYYMMDD*は本日の日付）
* **プライマリーキー:** [Artist] と入力、右側のプルダウンから [文字列] を選択
* **[ソートキーの追加]** にチェックを入れる
* **ソートキー:** [Title] と入力、右側のプルダウンから [文字列] を選択



1. **[作成]** をクリックします。
2. テーブルが作成されるまで1～2分程度かかります。
3. テーブルが作成されましたら、**[項目]** タブをクリックします。

下図のように **[Artist]**、**[Title]** の各属性が構成されていることを確認します。

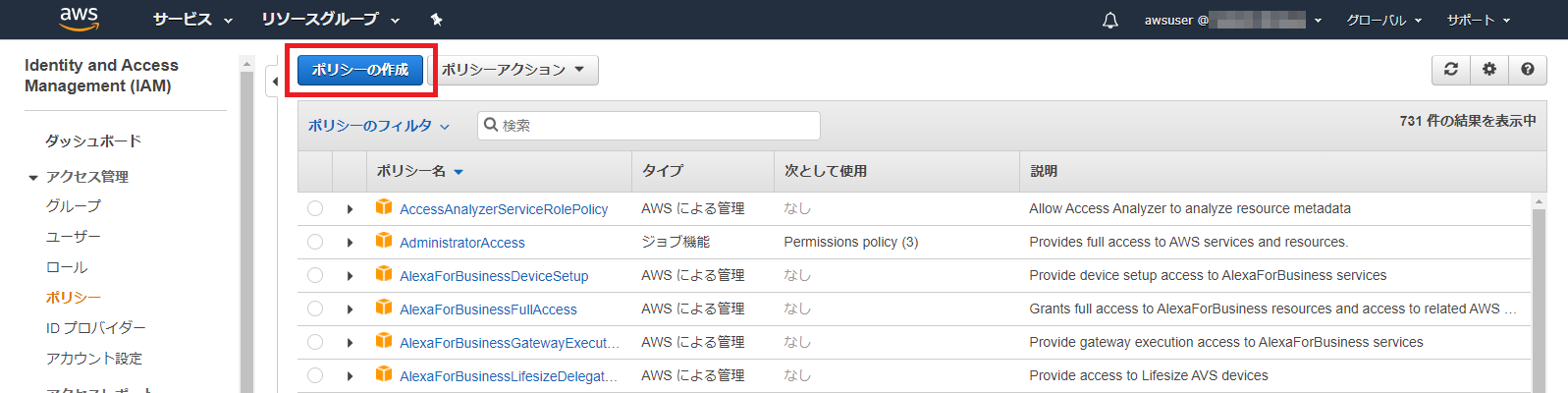


1. AWS マネジメントコンソールのサービス一覧から **[IAM]** を選択します。

画面左側のメニューから **[ポリシー]** を選択します。



1. **[ポリシーの作成]** をクリックします。

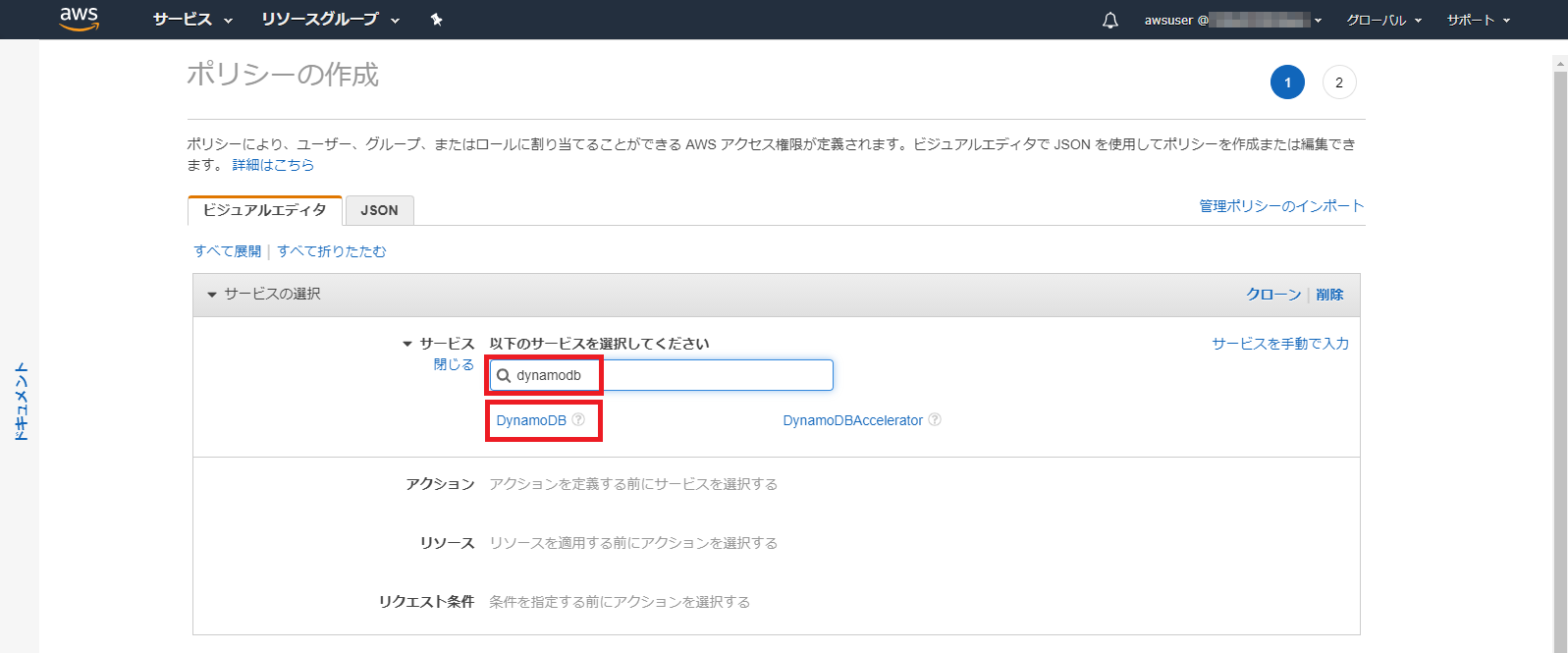


1. **[サービス]** をクリックして展開します。



1. 検索欄に **[dynamodb]** と入力します。

表示された **[DynamoDB]** をクリックします。



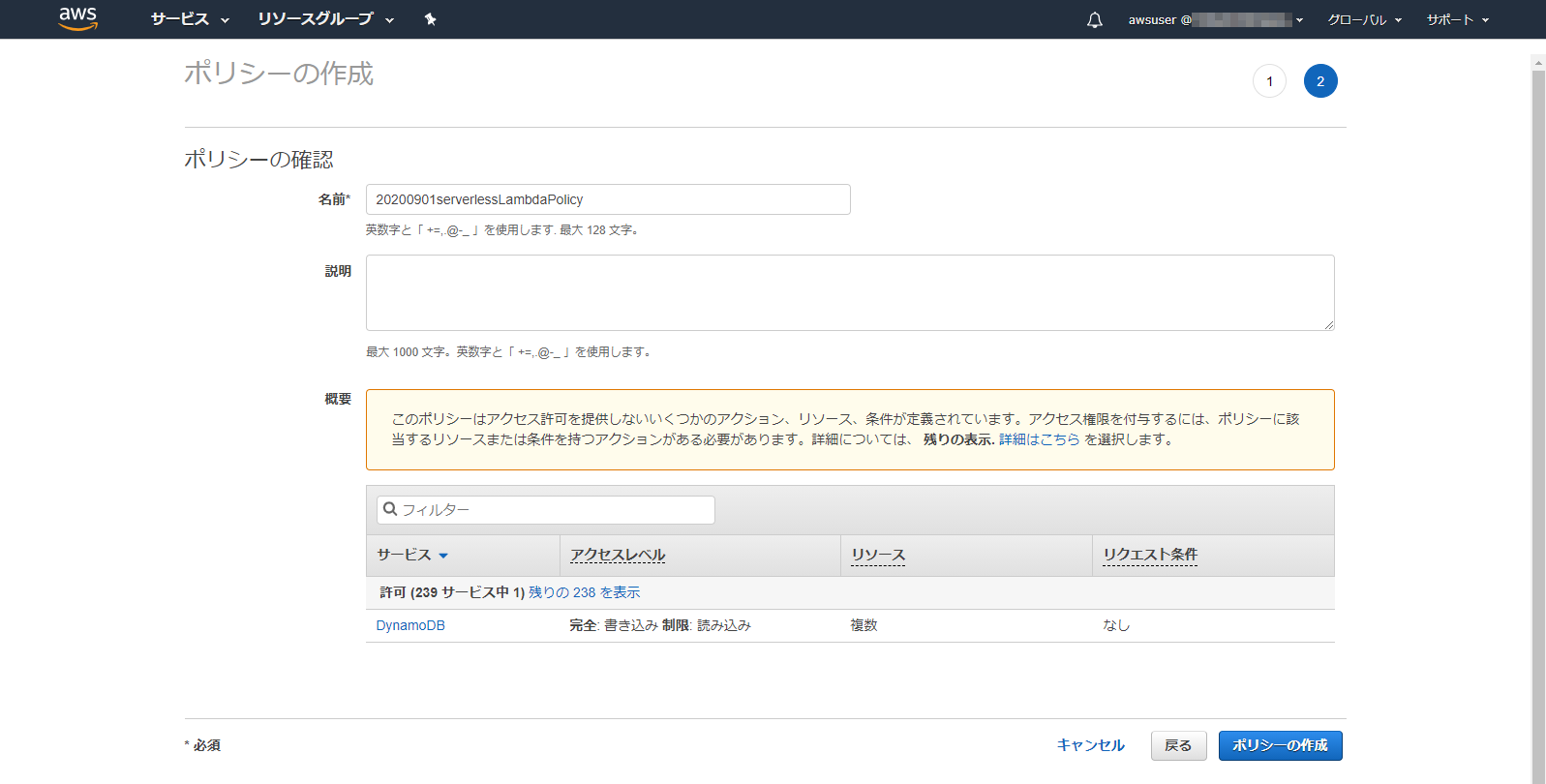
1. **[アクション]** を展開して、**[アクセスレベル]** から **[読み込み]** と **[書き込み]** にチェックを入れます。



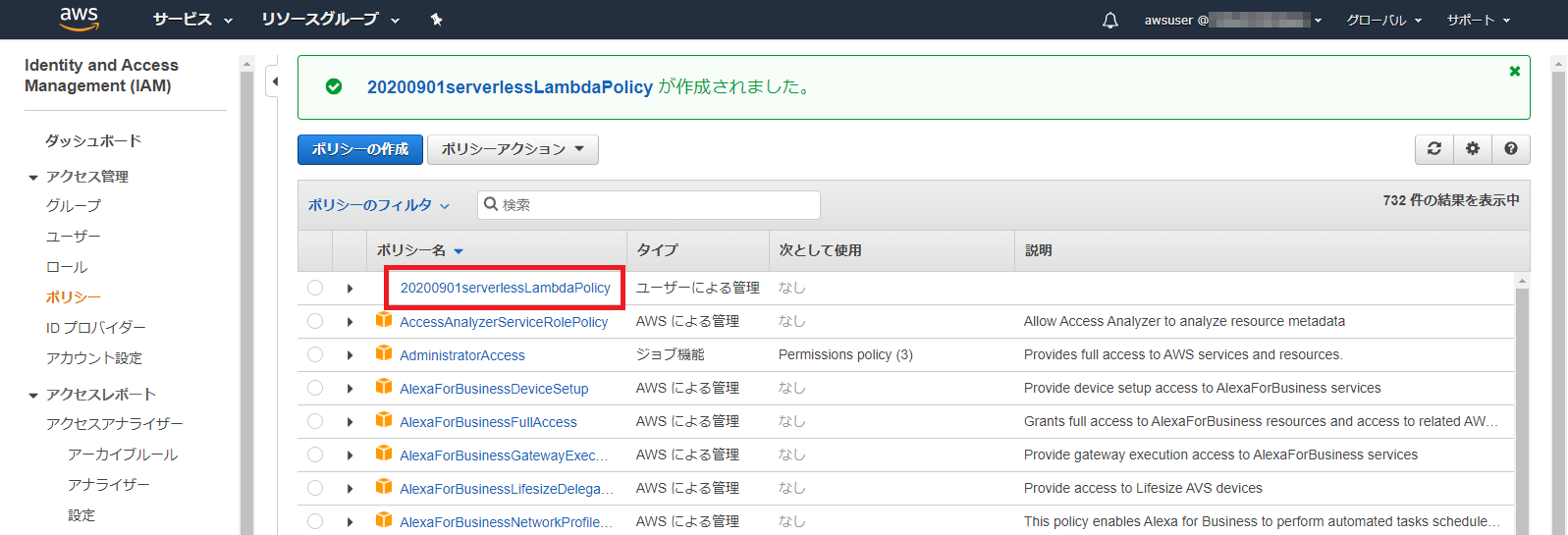
1. **[リソース]** を展開して、**[table]** 列の右端にある **[このアカウント内のいずれか]** にチェックを入れます。



1. **[ポリシーの確認]** をクリックします。
2. **[名前]** 欄に **[*YYYYMMDD*serverlessLambdaPolicy]** と入力します。（*YYYYMMDD*は本日の日付）

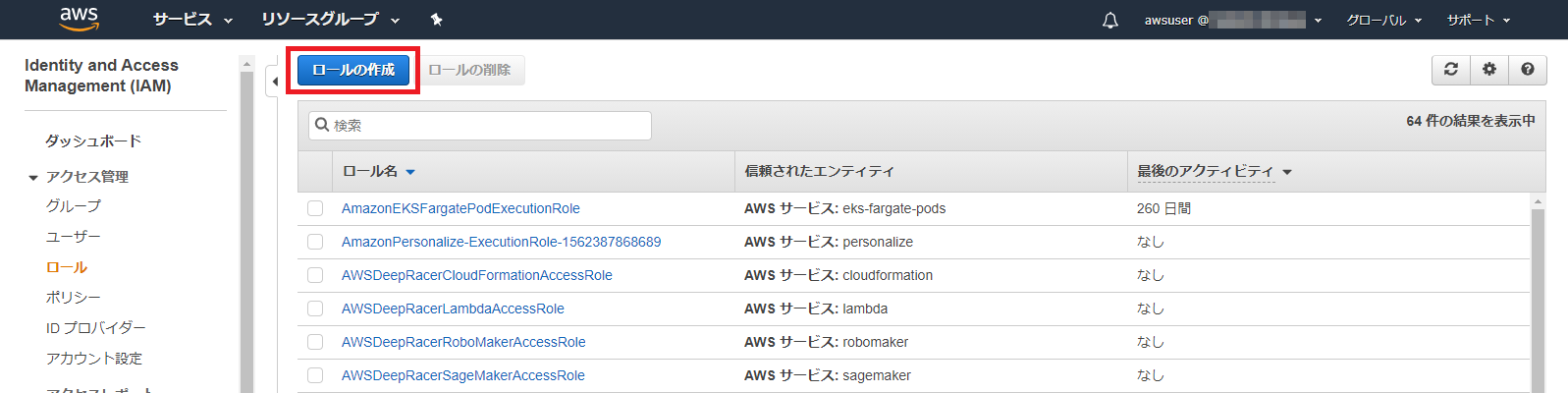


1. **[ポリシーの作成]** をクリックします。
2. ポリシー **[*YYYYMMDD*serverlessLambdaPolicy]** が作成されたことを確認します。



1. **[IAM]** 画面で、画面左側のメニューから **[ロール]** を選択します。

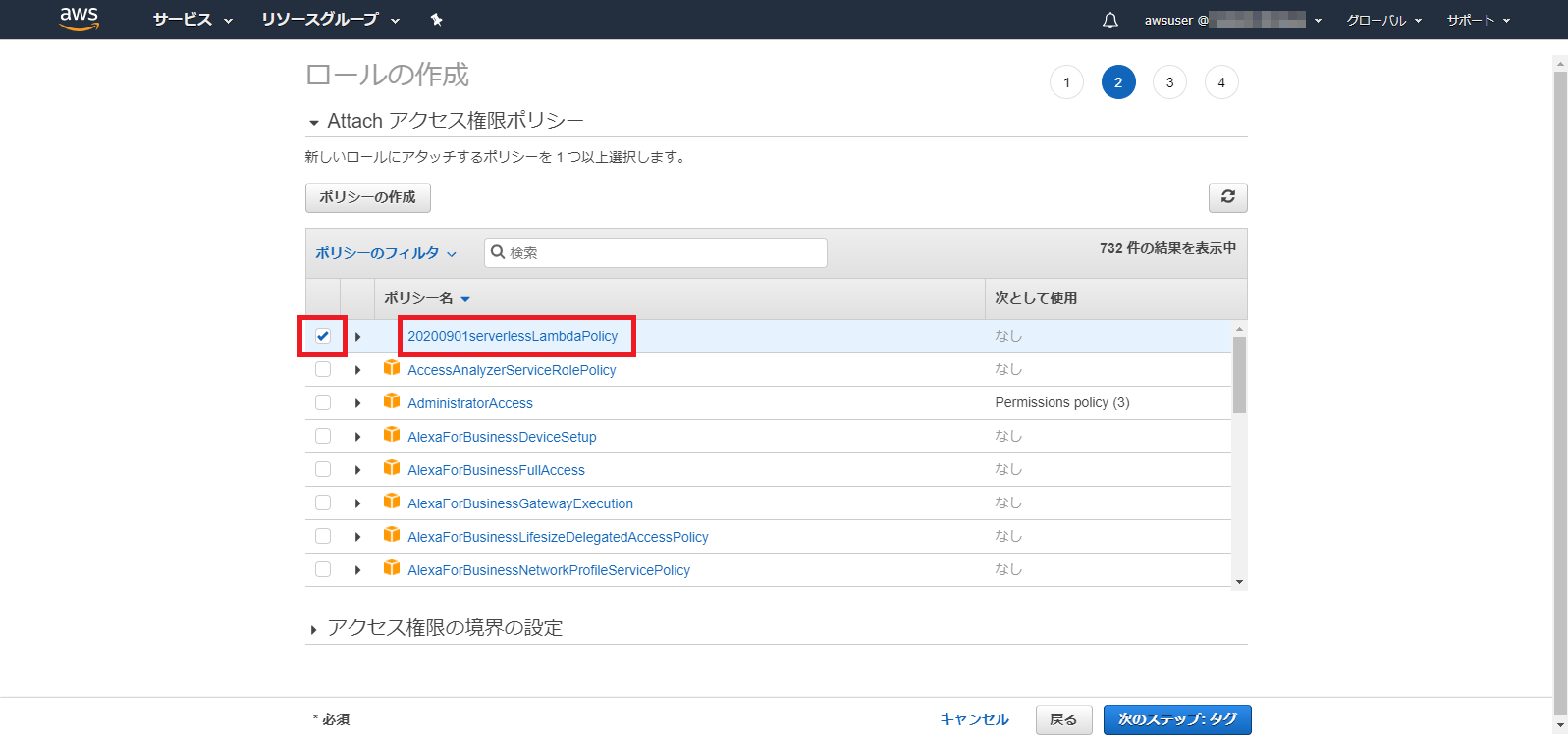


1. **[ロールの作成]** をクリックします。
2. **[信頼されたエンティティの種類]** で **[AWSサービス]** が選択されていることを確認します。

**[ユースケースの選択]** で **[Lambda]** をクリックして選択状態にします。



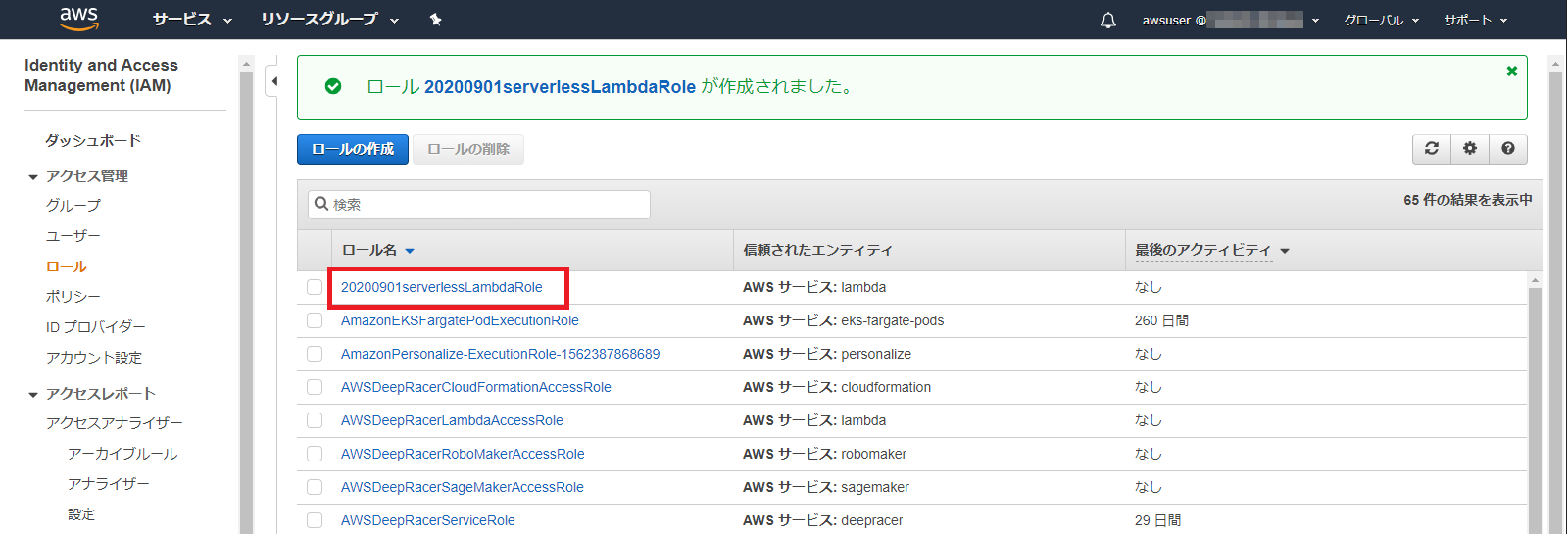
1. **[次のステップ: アクセス権限]** をクリックします。
2. ポリシーの一覧から、前手順で作成した **[*YYYYMMDD*serverlessLambdaPolicy]** を探して、左側のチェックボックスにチェックを入れます。



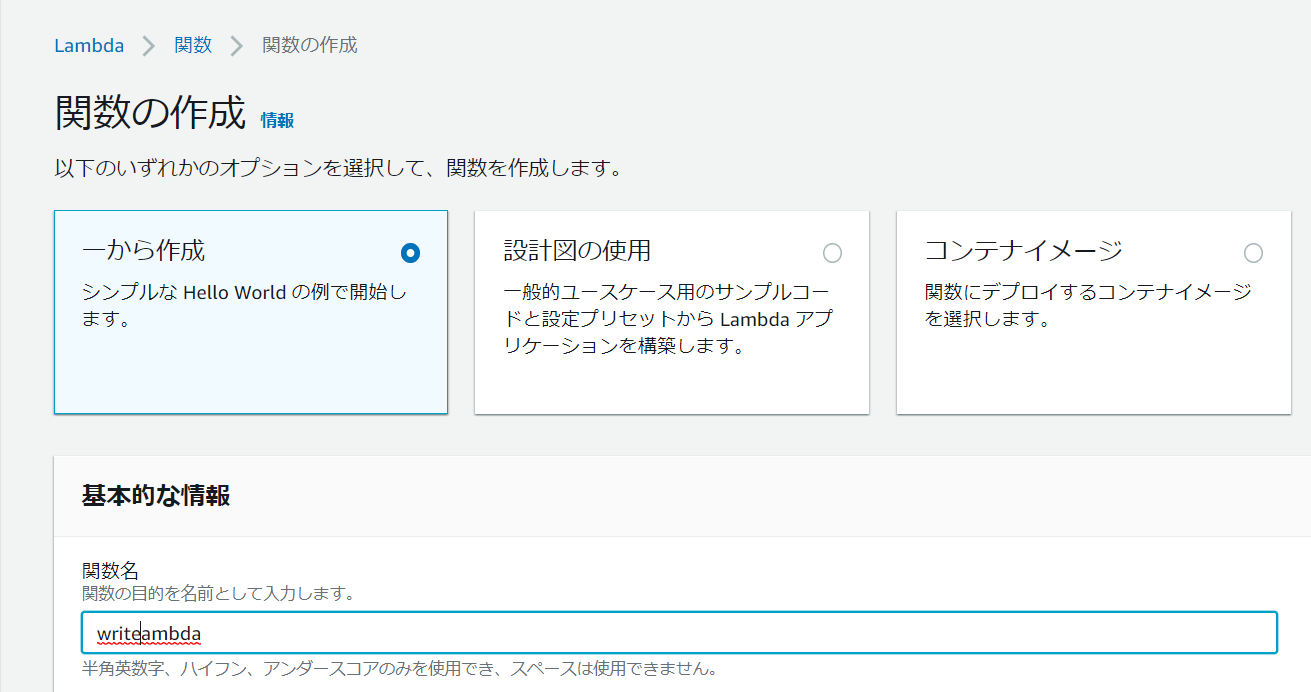
1. **[次のステップ: タグ]** をクリックします。
2. **[タグの追加]** ページでは何も入力せず、**[次のステップ: 確認]** をクリックします。
3. **[ロール名]** 欄に **[*YYYYMMDD*serverlessLambdaRole]** と入力します。（*YYYYMMDD*は本日の日付）



1. **[ロールの作成]** をクリックします。
2. ロール **[*YYYYMMDD*serverlessLambdaRole]** が作成されたことを確認します。



1. Lambdaの画面に戻って新しく関数の作成ボタンを押します。名前には「writelambda」と選んで「関数の作成」ボタンを押します。



1. 以下の内容をプログラム部分に貼り付けます。

console.log('Loading function');

const AWS = require('aws-sdk');

const docClient = new AWS.DynamoDB.DocumentClient();

exports.handler = (event, context, callback) => {

console.log('Received event:', JSON.stringify(event, null, 2));

const params = {

// TODO: 作成したDynamoDBテーブルの名前に書き換えてください

TableName: 'YYYYMMDDserverless',

Item: {

Artist: event.artist,

Title: event.title

}

};

docClient.put(params, function(err, data) {

if (err) {

console.log('Error', err);

callback(Error(err));

} else {

console.log('Success', data);

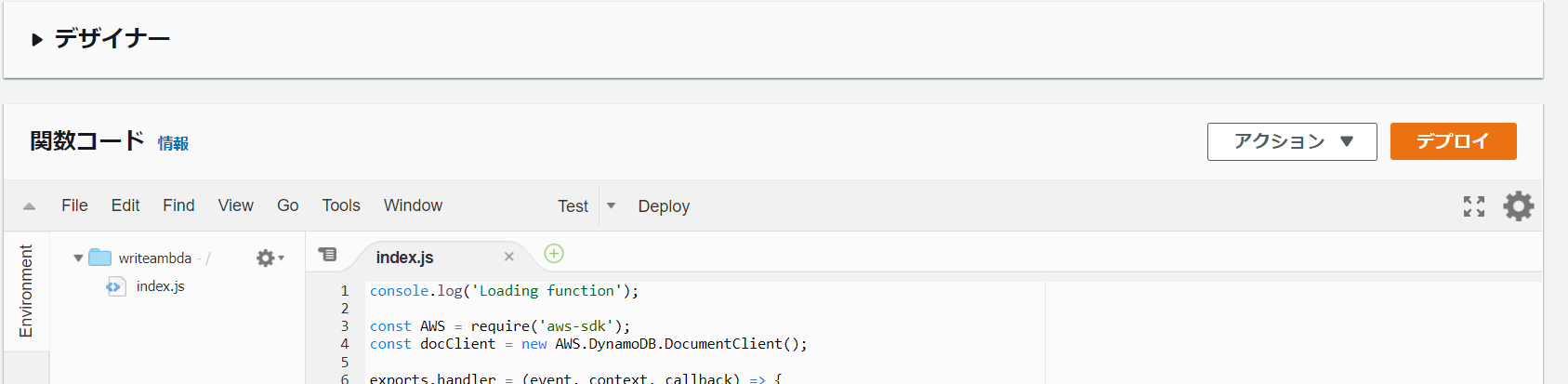
callback(null, data);

}

});

};

1. YYYYMMDDserverlssの部分を今日作成したDynamoDBのテーブル名に書き換えます。その後「デプロイ」のボタンを押します。



1. テストボタンを押し、適当なイベント名を入力し、以下をコピペします。その後「作成」ボタンをおします。

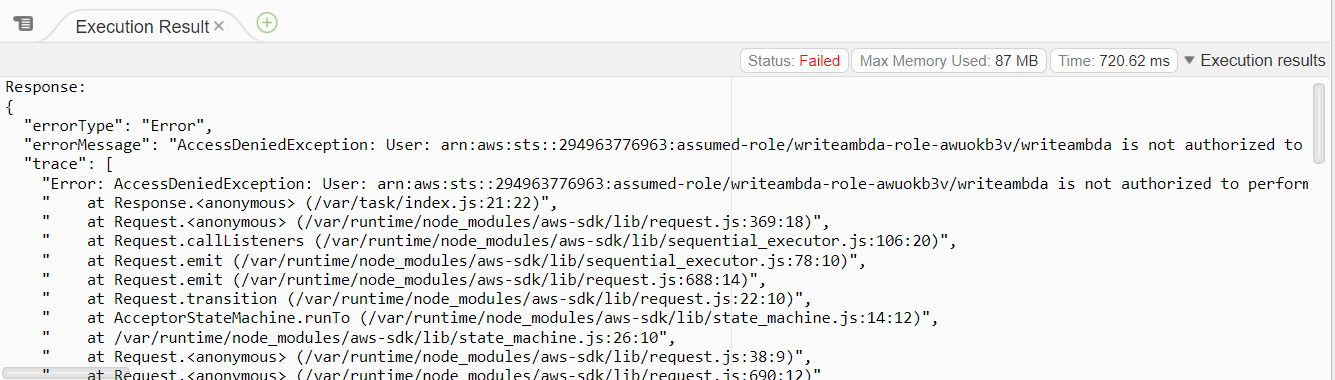
{

"artist": "Michael Jackson",

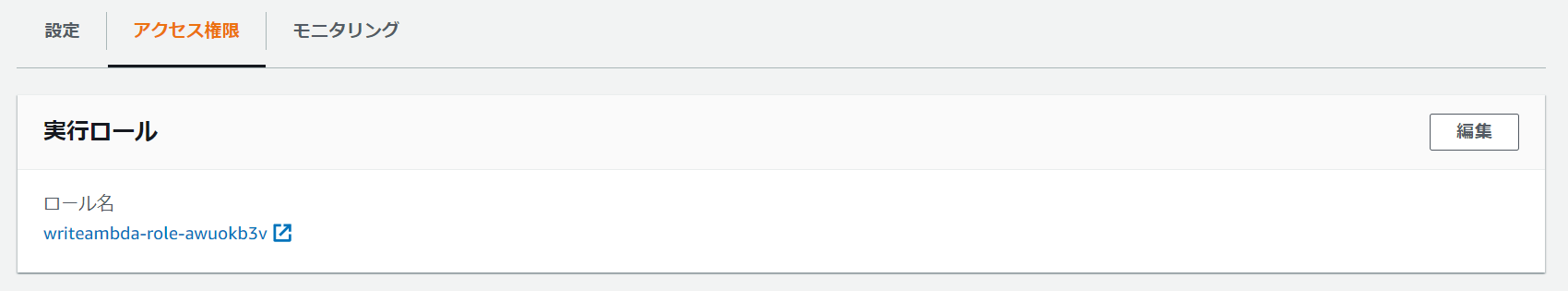
"title": "Thriller"

}

1. テストボタンを押すと画面下にErrorがでます。これは先ほど作成した適切なIAMロール（権限）が関数に付与されていないため、DynamoDBを操作する権限を持たないためです。



1. アクセス権限のタブを押し、「編集」ボタンを押します。



1. 先ほど作成した**[*YYYYMMDD*serverlessLambdaRole]**を[既存のロール]から選び「保存」ボタンを押します。
2. 再度「テスト」ボタンを押すと今度は実行が成功します。



1. DynamoDBの画面へ戻り、「項目」のタブを選ぶと正しくデータが投入されています。



1. 全く同じ要領で今度は新しくDynamoDBからデータを呼び出す関数を作ってみましょう。プログラムは以下をコピペして使います。

[プログラム]

console.log('Loading function');

const AWS = require('aws-sdk');

const docClient = new AWS.DynamoDB.DocumentClient();

exports.handler = (event, context, callback) => {

console.log('Received event:', JSON.stringify(event, null, 2));

const params = {

// TODO: 作成したDynamoDBテーブルの名前に書き換えてください

TableName: 'YYYYMMDDserverless',

KeyConditionExpression: 'Artist = :artist',

ExpressionAttributeValues: {

':artist': event.artist

}

};

docClient.query(params, function(err, data) {

if (err) {

console.log('Error', err);

callback(Error(err));

} else {

console.log('Success', data);

callback(null, data.Items);

}

});

};

[テスト]

{

"artist": "Michael Jackson"

}

より複雑な手順は以下のURLからダウンロードできます。興味のある方は挑戦してみましょう。

<https://github.com/harunobukameda/AWS-Amplify-AWS-Lambda-Amazon-DynamoDB-AWS-API-Gateway-Amazon-Cognito>