

```

import type { PostConfirmationTriggerHandler } from "aws-lambda";
import { type Schema } from "../../data/resource";
import { Amplify } from "aws-amplify";
import { generateClient } from "aws-amplify/data";
import { env } from "$amplify/env/post-confirmation";
import { createUserProfile } from "../graphql/mutations";

Amplify.configure(
  {
    API: {
      GraphQL: {
        endpoint: env.AMPLIFY_DATA_GRAPHQL_ENDPOINT,
        region: env.AWS_REGION,
        defaultAuthMode: "iam",
      },
    },
  },
  {
    Auth: {
      credentialsProvider: {
        getCredentialsAndIdentityId: async () => ({
          credentials: {
            accessKeyId: env.AWS_ACCESS_KEY_ID,
            secretAccessKey: env.AWS_SECRET_ACCESS_KEY,
            sessionToken: env.AWS_SESSION_TOKEN,
          },
        }),
        clearCredentialsAndIdentityId: () => {
          /* noop */
        },
      },
    },
  },
);

const client = generateClient<Schema>({
  authMode: "iam",
});

export const handler: PostConfirmationTriggerHandler = async (event) => {
  await client.graphql({
    query: createUserProfile,
    variables: {
      input: {
        email: event.request.userAttributes.email,
        profileOwner: `${event.request.userAttributes.sub}::${event.userName}`,
      },
    },
  },
);

```

```
});  
    return event;  
};
```