Practical: Installing and using the AWS SDK (Javascript)

Table of Contents

- Prerequisites and references
- Creating a basic app
  - 1. Create an EC2 instance
  - 2. Create a skeleton node app
  - 3. Install the SDK with npm
  - 4. Create the demo app
  - 5. Run the app
- Authentication

The AWS SDK is the main way of interacting with AWS services programmatically. It can be used for managing services (eg. launching EC2 instances) and also for interacting with them (eg. writing to an S3 bucket). In this practical we'll see how to get started using the SDK and do some basic operations.

Prerequisites and references

- **AWS SDK for Javascript version 3 documentation (https://docs.aws.amazon.com/sdk-for-javascript/v3/developer-guide/welcome.html)**
- **AWS SDK for Javascript version 3 code samples (https://github.com/awsdocs/aws-doc-sdk-examples/tree/main/javascriptv3/example_code)**

Creating a basic app

1. Create an EC2 instance

If you already have an EC2 instance up and running that you want to use then you can skip this step.

- Create an EC2 instance and follow the instructions in **Practical: Creating an EC2 Instance (https://canvas.qut.edu.au/courses/20367/pages/practical-creating-an-ec2-instance)**. **Important:** make sure that you add the `CAB432-Instance-Role` IAM role to the instance.
- Install Node. See **Practical: Node.js application on EC2 (https://canvas.qut.edu.au/courses/20367/pages/practical-node-dot-js-application-on-ec2)** for details.

Do the remaining steps on the command line for this EC2 instance. If using the Session manager to log in, change to the `ubuntu` user first: `sudo -iu ubuntu`

2. Create a skeleton node app

- Create a new folder called `sdkdemo` and change to it
- `npm init -y` to initialise the codebase

3. Install the SDK with npm

The SDK has multiple packages. We'll install the one for S3, and also the `dotenv` package, which allows us to read environment variables.

- run: `npm i @aws-sdk/client-s3`
- run: `npm i dotenv`

4. Create the demo app

We'll create a simple app that grabs the list of all S3 buckets on the QUT AWS account.

- create `index.js` with the following contents:

```
require("dotenv").config();
const S3 = require("@aws-sdk/client-s3");

async function main() {
    // Creating a client for sending commands to EC2
    client = new S3.S3Client({ region: 'ap-southeast-2' });

    // Example of a command for finding out information on EC2 instan
ces
    const command = new S3.ListBucketsCommand({});

    // It's good practice to use try/catch for error handling
    try {
        // Sending the command
        const response = await client.send(command);
        console.log(response);

    } catch (err) {
        console.log(err);
    }
}

main();
```

5. Run the app

- `node index.js`

If all goes well this will print out the response from the SDK, which should include a list of all S3 buckets on the QUT account. If you run into problems with authentication, check that your EC2 instance has the correct IAM role or check out the alternative methods for authentication below.

Authentication

Provided that you have set the IAM role correctly for the EC2 instance, the SDK will automatically fetch the required credentials from the cloud. However, in some situations (for example, developing locally on your own computer) this approach will not work. Additional methods of authentication are discussed in **Authenticating to the AWS SDK (https://canvas.qut.edu.au/courses/20367/pages/authenticating-to-the-aws-sdk)**

Complete code: **sdkdemo.zip (https://canvas.qut.edu.au/courses/20367/files/6583674/download)**

TEQSA PRV12079 | CRICOS 00213J | ABN 83 791 724 622