

# AWS Lambda Connected System

## Overview

The AWS Lambda plugin provides the integrations to invoke and list the available functions in the AWS Lambda service in the AWS instance. To connect with the AWS instance, the connected system must be authenticated with the Access Key ID and the Secret Access Key.

## Creating an Access Key in AWS Instance

1. Login to your AWS Console.

2. Click on your username and in the appearing pop up select Security Credentials.

The screenshot shows the AWS Management Console Home page. The top navigation bar includes the AWS logo, a 'Services' menu, a search bar, and the region 'N. Virginia'. The main content area is titled 'Console Home' and features a 'Recently visited' section with links to various services: Lambda, Amazon Connect, S3, AWS Marketplace Subscriptions, CloudFormation, Route 53, IAM, CloudFront, AWS Cost Explorer, Support, and AWS Budgets. A 'View all services' link is located below this section. On the right side, a user menu is open, displaying the account ID and IAM user name, along with links to 'Account', 'Organization', 'Service Quotas', 'Billing Dashboard', 'Security credentials' (highlighted in green), and 'Settings'. At the bottom of the menu are 'Switch role' and 'Sign out' buttons. Below the 'Recently visited' section, there are two main cards: 'Welcome to AWS' with links for 'Getting started with AWS' and 'Training and certification', and 'AWS Health' which shows 'No health data' and a message about permissions. The footer contains 'Feedback', a link for language selection, and 'Privacy', 'Terms', and 'Cookie preferences' links.

3. In the Security credential window, scroll down to the Access keys section and click on the Create Access Key button to create a new one.

The screenshot displays the AWS IAM console interface. On the left is a navigation sidebar for 'Identity and Access Management (IAM)'. The main content area is divided into three sections: 'MFA devices', 'Access keys', and 'Signing certificates'. The 'Access keys' section is currently active and shows one active key with details such as 'Last used 18 hours ago' and 'Last used service lambda'. A 'Create access key' button is visible in the 'Access keys' section.

**Identity and Access Management (IAM)**

Search IAM

Dashboard

**Access management**

- User groups
- Users
- Roles
- Policies
- Identity providers
- Account settings

**Access reports**

- Access analyzer
  - Archive rules
  - Analyzers
  - Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Related consoles

IAM Identity Center **New**

Device type	Identifier	Created on
No MFA devices. Assign an MFA device to improve the security of your AWS environment		

**Assign MFA device**

**Access keys (1)**

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

**Create access key**

[Redacted]		<b>Actions</b> ▼
Description	-	Status Active
Last used	18 hours ago	Created 92 days ago
Last used region	[Redacted]	Last used service lambda

**Signing certificates (X.509) (0)**

Use X.509 certificates to make secure SOAP-protocol requests to some AWS services. You can have a maximum of two X.509 certificates (active or inactive) at a time. [Learn more](#)

**Actions** ▼ | **Upload** | **Create X.509 certificate**

4. In the appearing list of options, select Third-party service and click on Next.

IAM > Security credentials > Create access key

Step 1 of 3

## Access key best practices & alternatives

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

- Command Line Interface (CLI)**  
You plan to use this access key to enable the AWS CLI to access your AWS account.
- Local code**  
You plan to use this access key to enable application code in a local development environment to access your AWS account.
- Application running on an AWS compute service**  
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- Third-party service**  
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- Application running outside AWS**  
You plan to use this access key to enable an application running on an on-premises host, or to use a local AWS client or third-party AWS plugin.
- Other**  
Your use case is not listed here.

Cancel **Next**

5. In the next window, provide a description tag. This is an optional step. Click on Create Access key to proceed.

The screenshot shows the AWS IAM console interface. On the left is a navigation sidebar for 'Identity and Access Management (IAM)' with a search bar and a menu including 'Access management' (User groups, Users, Roles, Policies, Identity providers, Account settings) and 'Access reports' (Access analyzer, Archive rules, Analyzers, Settings, Credential report, Organization activity, Service control policies (SCPs)). Below the sidebar is a 'Related consoles' section with a link to 'IAM Identity Center' marked as 'New'. The main content area is titled 'IAM > Security credentials > Create access key' and shows 'Step 2 of 3: Set description tag - optional'. A sub-header explains: 'The description for this access key will be attached to this user as a tag and shown alongside the access key.' Below this is a text input field for the 'Description tag value' with a placeholder: 'Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.' A note below the field states: 'Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: \_ : / = + - @'. At the bottom of the main area are three buttons: 'Cancel', 'Previous', and 'Create access key'.

6. The Access Key ID and the Secret access key will be displayed. The credentials can also be downloaded as a CSV file by clicking on the Download .csv file button. Please make sure that the credentials are noted which will not be visible again once the Done

button is clicked.

**Identity and Access Management (IAM)**

Search IAM

Dashboard

▼ Access management

- User groups
- Users
- Roles
- Policies
- Identity providers
- Account settings

▼ Access reports

- Access analyzer
  - Archive rules
  - Analyzers
  - Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Related consoles

IAM Identity Center [New](#)



**Access key created**  
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

IAM > Security credentials > Create access key

Step 3 of 3

## Retrieve access keys

**Access key**  
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
 [Redacted]	 ***** <a href="#">Show</a>

**Access key best practices**

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [Best practices for managing AWS access keys](#).

[Download .csv file](#) [Done](#)

7. Make sure that the Access key is active.

# Connected System

# Create Connected System

Search Connected Systems...



HTTP



OpenAPI



ABBYY Cloud OCR



Adobe Connected System



Advanced Rich Text Editor Image Upload And Pdf Export



Amazon Machine Learning



AmazonTranslateCS



Appian RPA



ASANA



Aurora MySQL Data Source



Aurora PostgreSQL Data Source



AWS Lambda

Establish connection with AWS Lambda to invoke and list available functions.



AWS S3



Azure Data Lake Connected System



Azure Face Recognition



Blue Prism

CANCEL



# Create Connected System



## AWS Lambda

Establish connection with AWS Lambda to invoke and list available functions.  
Version: 1

Name \*

Description

### AWS Lambda Configuration

Access Key ID \*

Enter the AWS access key.

Access Secret Key \*

Enter the AWS secret key.

Region \*

Select the region where the AWS instance is located.

TEST CONNECTION

GO BACK

CANCEL

USE IN NEW INTEGRATION

CREATE

Provide the Access Key ID and the Secret Access key obtained from the AWS console. Select the AWS instance region and click on test connection to verify the credentials.

# Connected System Properties



## AWS Lambda

Establish connection with AWS Lambda to invoke and list available functions.  
Version: 1

### Name \*

### Description

### AWS Lambda Configuration

#### Access Key ID

\*\*\*\*\* [\(Clear\)](#)

Enter the AWS access key.

#### Access Secret Key

\*\*\*\*\* [\(Clear\)](#)

Enter the AWS secret key.

#### Region \*

Select the region where the AWS instance is located.

Connection successful

**TEST CONNECTION**

CANCEL

USE IN NEW INTEGRATION

**SAVE**



# Available Integrations

1. List Available Function
2. Invoke Function


## List Available Function

This integration lists all the available functions for the provided credentials in the AWS Lambda service.

**Connected System \***

 PSS CS AWS Lambda 

**Operation \***

List Available Functions 

Lists the available functions in the instance.

Result Request Response

**Success!**

**Time**  
93 ms  
Prepare: < 1 ms - **Execute**: 92 ms (Send/Wait/Receive: 90 ms) - **Transform**: 1 ms

**Value**

- Dictionary
  - success **true** (Boolean)
- result Dictionary
  - functions List of Dictionary - 2 items
    - Dictionary
    - Dictionary
      - statusCode **200** (Number (Integer))
      - status **"success"** (Text)
    - error **null** (Null)
    - authType Diagnostic


**TEST REQUEST**

# Invoke Function

The Invoke Function integration invokes the specified function and returns the output. It has two parameters:

1. **Function Name(required):** the name of the AWS Lambda function to invoke.
2. **Inputs(optional):** the inputs to the function. Must be provided as a JSON object. Use the Appian [a!toJson](#) method.

### Connected System \*

 PSS CS AWS Lambda ✕

### Operation \*

Invoke Function

Invoke the specified AWS Lambda function.

### Function Name \*

public1

Enter the function name to invoke.

### Inputs

```
1 a!toJson(  
2 {  
3   input1: "name",  
4   input2: "test"  
5 }  
6 )
```

Place cursor on function, rule, or constant to display help

Pass the inputs as a dictionary enclosed within toJson function.

TEST REQUEST

Result Request Response

Success!

**Time**  
309 ms  
Prepare: < 1 ms - Execute: 308 ms (Send/Wait/Receive: 305 ms) - Transform: 1 ms

**Value**

- Dictionary
  - success **true** (Boolean)
  - result Dictionary
    - body **""Hello from Lambda! name test""** (Text)
    - statusCode **200** (Number (Integer))
    - error **null** (Null)
    - authType Diagnostic