



# Amazon EventBridge Adapter for SAP Integration Suite

Version 1.0.3 – February 2026

# Contents

1	Introduction .....	3
1.1	Objective.....	3
1.2	Coding Samples.....	3
1.3	Internet Hyperlinks .....	3
1.4	Overview .....	3
1.5	Features.....	4
2	Installation and Configuration.....	5
2.1	Adapter Installation on Cloud Foundry.....	5
2.1.1	Prerequisite .....	5
2.1.2	Procedure .....	5
2.1.2.1	Adapter Installation by Creating a New Integration Flow.....	5
2.1.2.2	Adapter Installation without Creating a New Integration Flow .....	6
2.1.3	Monitor the Deployment Status .....	7
3	Getting Started: AmazonEventBridge Adapter .....	8
3.1	Architecture Overview.....	8
3.2	Application Configuration .....	9
3.3	Authentication.....	9
3.3.1	Creating Security Parameters in Security Material.....	9
3.4	Supported Operations .....	10
4	AmazonEventBridge Adapter Configuration .....	11
4.1	Receiver Adapter .....	11
4.1.1	General.....	11
4.1.2	Connection Tab .....	11
4.1.3	Processing Tab.....	12
5	AmazonEventBridge Adapter Operations .....	15
5.1	Basic.....	15
5.2	Advanced.....	16

# 1 Introduction

## 1.1 Objective

This is the official guide for the AmazonEventBridge Adapter for SAP Integration Suite. This guide covers all relevant information for integration developers to start working with the AmazonEventBridge adapter. Read this guide carefully before using the Adapter.

## 1.2 Coding Samples

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. The correctness and completeness of the Code given herein are not guaranteed.

## 1.3 Internet Hyperlinks

The documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. The availability and the correctness of this related information or the ability of this information to serve a particular purpose are not warranted.

## 1.4 Overview


The AmazonEventBridge adapter works in an event-driven integration architecture to push your events to Amazon EventBridge for further consumption. You can use the AmazonEventBridge adapter to send events from variety of data sources to single or multiple event buses in Amazon EventBridge thereby ensuring reliable and quick event ingestion.

## 1.5 Features

- **Amazon EventBridge Operations Support:** You can send events to event buses for further processing in your event driven systems.
- **Ease of Access and Connectivity:** Allows you to save connection related details on the adapter, thereby providing a convenient mode for establishing connections to any external systems.
- **Dynamic configuration with headers and properties:** Assigning dynamic values to different properties allows enhanced flexibility to your integration flows. You can also refer to dynamic parameters using SAP Cloud Integration exchange headers and properties.
- **Multiple Event Bus Support:** You can send events to a single event bus using the basic type of operation whereas the advanced operation type allows you to send events to multiple event buses at a time.

# 2 Installation and Configuration

This section details the prerequisites to install and configure the AmazonEventBridge adapter.

 The AmazonEventBridge adapter is available as part of your SAP Integration Suite license.


## 2.1 Adapter Installation on Cloud Foundry

Before the AmazonEventBridge adapter can be used in the Cloud Foundry environment, it must be deployed to the SAP Integration Suite tenant.

### 2.1.1 Prerequisite

To deploy the adapter, you must have access to “*AmazonEventBridge Adapter for SAP Integration Suite*” as part of your SAP Integration Suite license.

### 2.1.2 Procedure

 The below installation procedure is compatible with Apache Camel 2, Apache Camel 3, and Edge Integration Cell (EIC) platform.

You can deploy the adapter using the following methods:

#### 2.1.2.1 Adapter Installation by Creating a New Integration Flow

The AmazonEventBridge adapter is available for selection in the receiver adapter list and can be deployed in the **Design** tab directly as you use it in an Integration flow.



#### Purpose

To install an adapter for use in your Integration flow.

#### Procedure

Go to **Design** workspace and select the integration package where you want to create a new Integration flow.

1. Click **Edit** to make the package editable.
2. Go to the **Artifacts** tab. Click **Add** and select **Integration Flow**.

3. Enter the **Name** and **ID** for your flow. Additionally, select **Runtime Profile** from the drop-down and choose **Sender** and **Receiver** systems from the list . Finally, click **Add** to create the integration flow.
4. Go to the newly created integration flow and click **Edit** to make it editable.
5. In the integration flow, click **End** to add a **Connector**  between the **End** and the **Receiver Box**. A drop-down with the available adapters appears. The **AmazonEventBridge** adapter should show up in the list.
6. Select the **AmazonEventBridge** adapter from the list. The adapter is now imported which *triggers* an adapter deployment. Once the AmazonEventBridge Adapter is deployed, a success message is *displayed*.

After the above steps are done, the AmazonEventBridge Adapter is successfully deployed in your Design workspace of the SAP Integration Suite tenant.

### 2.1.2.2 Adapter Installation without Creating a New Integration Flow



The following procedure describes how the AmazonEventBridge adapter is migrated from the Discover workspace to the Design workspace of the SAP Integration tenant.

This method is useful for scenarios where integration flow packages are migrated from development to a higher environment such as Production. The AmazonEventBridge adapter can be imported into the Design workspace without creating an integration flow. Use the Transport Management Service (TMS) to import/transport the AmazonEventBridge adapter to a higher environment. Alternatively, if the TMS is not available in the landscape, the adapter package can be imported into the Design workspace by copying it from the Discover workspace.

#### Purpose

To import the AmazonEventBridge adapter to **Design** workspace by copying the integration package from **Discover** workspace.

#### Procedure

1. Go to **Discover** workspace.
2. In the search box, search for **AmazonEventBridge adapter for SAP Integration Suite** package.
3. Select the package and click **Copy**.  
This copies the package from the Discover workspace to Design workspace.
4. Go to Design workspace and select the copied **AmazonEventBridge adapter for SAP Integration Suite** package.
5. In the **Actions** tab of the selected package, click **Deploy**.

This completes the adapter deployment to **Design** workspace.

## 2.1.3 Monitor the Deployment Status

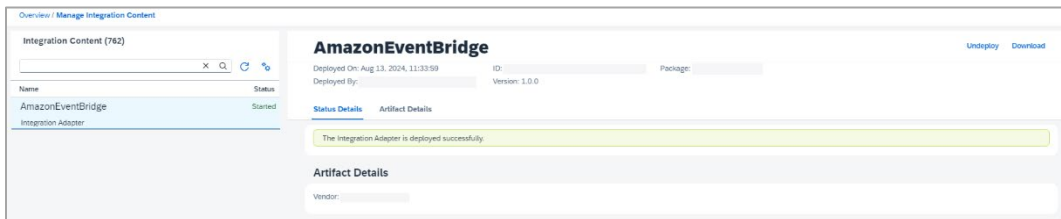
After the adapter deployment is complete, you can check the status in the **Monitor** section.

### Purpose

To check the status of the deployed adapter.

### Procedure

1. Under the **Monitor** tab, click **Integrations and APIs**. This opens the **Overview** page.
2. On the **Overview** page, go to **Manage Integration Content** section and click **All**. This opens **Integration Content** page with a list of all the deployed adapters.
3. Here, you can check and confirm the deployment status of your adapter.



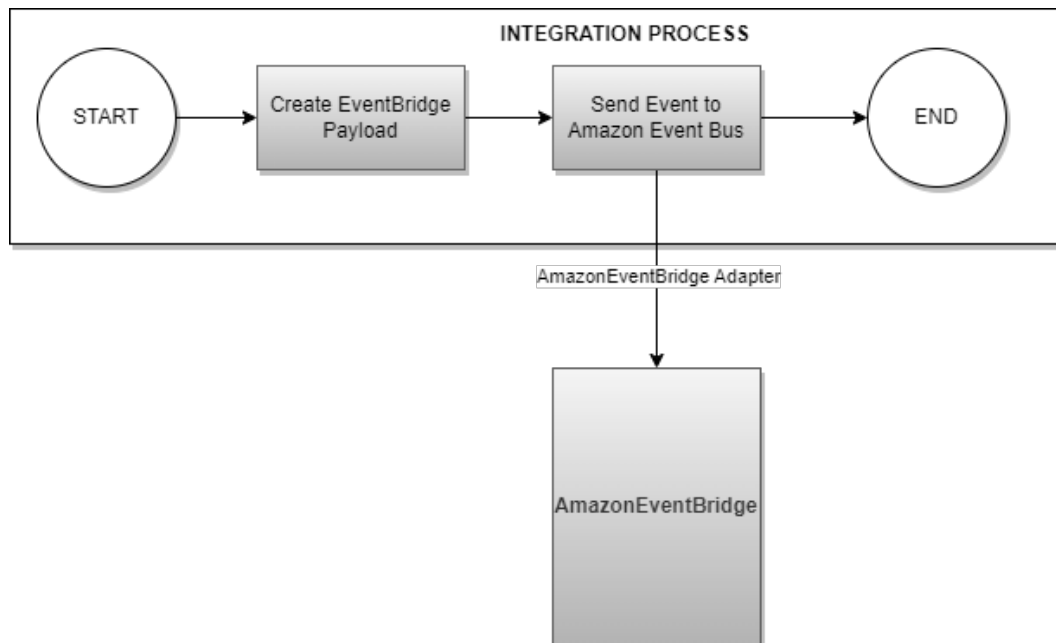
# 3 Getting Started: AmazonEventBridge Adapter

Before you start using the adapter, you can get to know underlying [Architecture](#) for the adapter and [Application Configuration](#) for AmazonEventBridge.

## 3.1 Architecture Overview

The AmazonEventBridge adapter is designed to be employed as a receiver adapter. In such a scenario, SAP Cloud Integration acts as the initiator of the calls. You are provided the basic and advanced operation to send events to event buses located on AmazonEventBridge adapter.

The image below gives a high-level representation of how the adapter works.



For more information about the supported operations, see [Operations Supported in Amazon EventBridge Adapter](#).

## 3.2 Application Configuration

- To start using Amazon EventBridge, see [Getting Started with Amazon EventBridge](#).
- To configure and setup Amazon EventBridge see [Amazon EventBridge setup and prerequisites](#).

## 3.3 Authentication

The AmazonEventBridge Adapter requires creation of credentials for your AWS user. For each user, you can create an Access Key and a Secret Key. The adapter uses common security artifacts stored in SAP Cloud Integration. You must use **Secure Parameter** artifact to safely store Access Key and Secret Key. These security artifacts can then be accessed in the adapter using aliases.

Before setting up the authentication, you must create the Credentials in **Security Material** in the SAP Integration Suite as described in the next section.

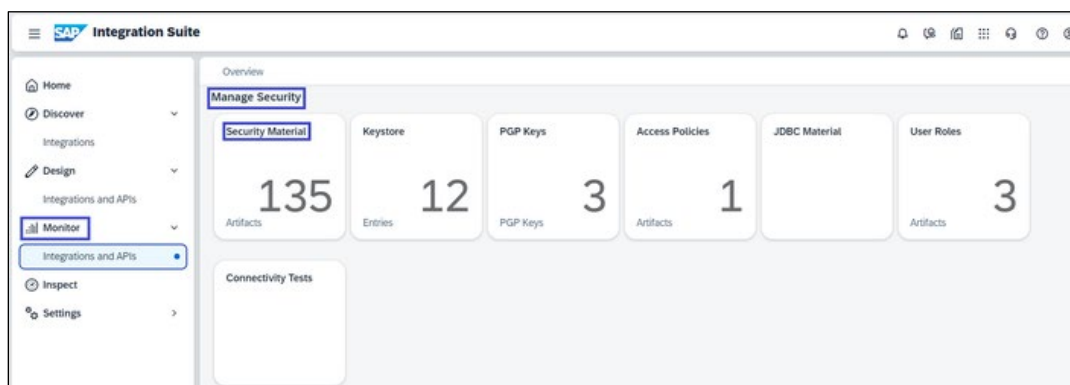
### 3.3.1 Creating Security Parameters in Security Material

#### Purpose

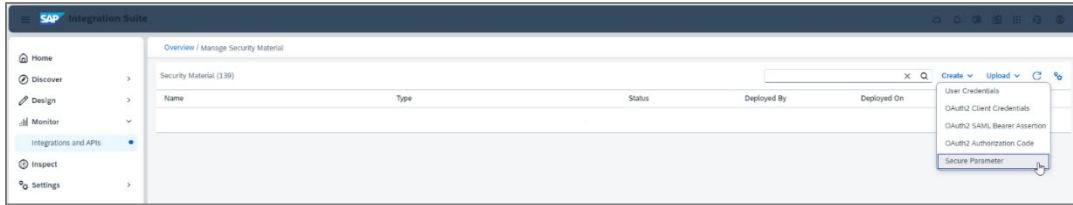
To create credentials for authentication.

#### Procedure

1. In SAP Integration Suite, navigate to **Monitor > Integrations and APIs**. This opens the **Overview** page.
2. On the **Overview** page, go to **Manage Security** section and click **Security Material**.



3. On the **Manage Security Material** page, click **Create** to select **Secure Parameter** from the dropdown.



4. In the **Create Secure Parameter** popup, provide the below details.

Field	Description
Name	Specify the name of the security artifact. The artifact name is used as an alias for the confidential data.
Description	Enter a description for the artifact (optional).
Secure Parameter	Enter the confidential value of the attribute. <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #add8e6;"> <span style="font-size: 1.2em; color: #0070c0; font-weight: bold;">i</span> The permissible length of the secure parameter for Cloud Foundry is a maximum of 4096 characters.         </div>
Repeat Secure Parameter	Repeat the confidential value of the attribute.

5. Click **Deploy** to complete the process.

## 3.4 Supported Operations

### Requirements:

- Amazon EventBridge server should be reachable via the SAP Integration Suite.
- User should have permissions to perform operations on AmazonEventBridge Server.

The following operations supported by the AmazonEventBridge adapter have been described in detail.

- [Basic](#)
- [Advanced](#)

# 4 AmazonEventBridge Adapter Configuration

This section describes the parameters to be configured for your AmazonEventBridge adapter. You need to configure the **Connection** and **Processing** tabs. A description and example usage for each field has been added.

## 4.1 Receiver Adapter

In this section, you will learn how to configure the AmazonEventBridge receiver adapter. After selecting the AmazonEventBridge adapter from the list of adapters, you must configure the **General**, **Connection**, and **Processing** tabs.

### 4.1.1 General

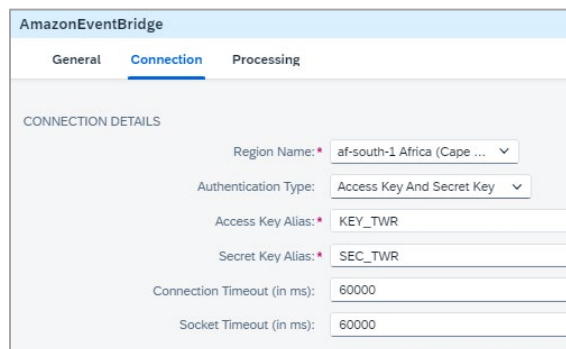
The General tab provides an overview of basic adapter information including **Channel** and **Adapter** details.

Field	Description
Name	Specify the name of the adapter flow.
Description	Specify the description of the adapter.

### 4.1.2 Connection Tab

The Connection tab contains connection and authentication parameters for AmazonEventBridge. Before you set the connection details, see [Create Secure Parameter](#).

The connection tab contains the following fields:



The screenshot shows the 'AmazonEventBridge' configuration interface with the 'Connection' tab selected. Under the 'CONNECTION DETAILS' section, the following fields are visible:

- Region Name: af-south-1 Africa (Cape ...)
- Authentication Type: Access Key And Secret Key
- Access Key Alias: KEY\_TWR
- Secret Key Alias: SEC\_TWR
- Connection Timeout (in ms): 60000
- Socket Timeout (in ms): 60000

Field	Description
Region Name	Select the AWS Region where the S3 Bucket resides.  <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <span style="font-size: 1.2em; color: #007bff; float: left; margin-right: 5px;">i</span>           If you are manually specifying this value, use <code>eu-central-1</code> instead of <code>eu-central-1 Europe (Frankfurt)</code>.         </div>
Authentication	The authentication method used for connection to Amazon EventBridge. Currently, Access Key and Secret Key is used for authentication.
Access Key Alias	Specify the name of the Secure Parameter which stores the AWS Access Key.
Secret Key Alias	Specify the name of the Secure Parameter which stores the AWS Secret Key.
Connection Timeout (in ms)	Specify the maximum waiting time (in milliseconds) for the connection to be established. Example: 6000
Socket Timeout (in ms)	Specify the maximum waiting time (in milliseconds) for a response message. Example: 6000

### 4.1.3 Processing Tab

The Processing tab contains all the operational configurations for the AmazonEventBridge adapter.

**AmazonEventBridge**

General   Connection   Processing

---

PROCESSING DETAILS

Operation Type: Basic ▼

Source: \*

Resources:

Detail Type: \*

Bus Name: \*

Detail: \*

Check if Bus Exists:

Throw Exception on Failed Entry:



---


HEADER DETAILS

Request Headers:

Response Headers:

The processing tab contains the following fields:

Field	Description
Operation Type	<p>Specify the type of operation to be performed. You can select from either of the two available options:</p> <ul style="list-style-type: none"> <li>• <b>Basic</b> allows you to send a single event to a single event bus.</li> <li>• <b>Advanced</b> allows you to send multiple events to multiple event buses.</li> </ul>
Source	<p>Specify the source of the event.</p> <p>Example: <code>com.mycompany.myapp</code></p>
Resources	<p>Specify the resources to be tagged to this event.</p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #cfe2f3;"> <p> Use comma separated values for multiple resources. These are AWS resources, identified by Amazon Resource Name (ARN), which the event primarily concerns. Any number, including zero, may be present.</p> </div> <p>Example: <code>resource1, resource2</code></p>
Detail Type	<p>Specify the type of event.</p> <p>Example: <code>PrimaryLogin</code></p>
Bus Name	<p>Specify the name of the event bus.</p>
Detail	<p>Specify the detail that needs to be sent to the event bus.</p> <p>Example: <code>\${in.body}</code></p> <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #cfe2f3;"> <p>This value can also be read dynamically using an exchange header or property. Example: <code>\${in.body}</code>.</p> <p> However, if you choose to specify the payload here, ensure your new line and tab characters are escaped using <code>\n</code> and <code>\t</code> respectively. Modify accordingly for any other special characters you use.</p> </div>
Check if Bus Exists	<p>Enable to validate the bus name for each entry in your payload.</p>
Throw Exception on Failed Entry	<p>Enable to check for the failed entries. In case of failures, an exception is thrown with failure count and failed entries.</p>

Field	Description
Entries	<p>Specify the entries that need to be sent to the event bus. This must be a JSON element.</p> <div data-bbox="397 331 1377 520" style="background-color: #e6f2ff; padding: 10px;"> <p> This value can also be read dynamically using an exchange header or property. Example: <code>\${in.body}</code>. However, if you choose to specify the payload here, ensure your new line and tab characters are escaped using <code>\n</code> and <code>\t</code> respectively. Modify accordingly for any other special characters you use.</p> </div>
Request Headers	<p>Enter a list of headers coming from the target system's response, separated by a pipe ( ), to be received in the message. Use an asterisk (*) to receive all the headers from the target system, which is also the default value.</p>
Response Headers	<p>Enter a list of custom headers, separated by a pipe ( ), to send to the target system. By default, no custom headers are sent. Use an asterisk (*) to send all custom headers to the target system. Alternatively, you can dynamically pass on the values by defining a property that includes a list of headers.</p> <p>Default: *</p>

# 5 AmazonEventBridge Adapter Operations

This section describes the operation types available in AmazonEventBridge Adapter.

## 5.1 Basic

This operation allows you to send a single event to a single event bus.

The screenshot shows the 'AmazonEventBridge' configuration interface with the 'Processing' tab selected. The 'PROCESSING DETAILS' section includes the following fields:

- Operation Type: Basic (dropdown menu)
- Source: ecommerce
- Resources: (empty text field)
- Detail Type: orderPlaced
- Bus Name: CustomerBus
- Detail: {"url":"www.startapps.com","email":"smith.evan@outlook.com"}

Below these fields are two checkboxes:

- Check if Bus Exists:
- Throw Exception on Failed Entry:

The 'HEADER DETAILS' section includes:

- Request Headers: (empty text field)
- Response Headers: \*

Field	Description
Operation Type	Select Basic from the available dropdown.
Source	Specify the source of the event. Example: com.mycompany.myapp
Resources	Specify the resources to be tagged to this event. Example: resource3, resource4
Detail Type	Specify the type of event. Example: FirstEvent

Field	Description
Bus Name	<p>Specify the name of the event bus.</p> <p>This is a mandatory field.</p> <p><b>i</b> For Advanced operation type, even if the json doesn't contain Bus Name, event is sent to the Default event bus. However, Basic requires you to populate this field.</p> <p>Example: <code>central_bus</code></p>
Detail	<p>Specify the detail that needs to be sent to the event bus.</p> <p>Example: <code>{"url": "www.startapps.com", "email": "smith.evans@outlook.com"}</code></p>
Check if Bus Exists	<p>Enable to validate the bus name for each entry in your payload.</p>
Throw Exception on Failed Entry	<p>Enable to check for the failed entries. In case of failures, an exception is thrown with failure count and failed entries.</p>

## 5.2 Advanced

The **Processing** tab contains all the operational configurations for the AmazonEventBridge adapter.

The screenshot shows the AmazonEventBridge configuration interface, specifically the Processing tab. The interface is divided into several sections:

- General:** Contains tabs for General, Connection, and Processing (which is currently selected).
- PROCESSING DETAILS:**
  - Operation Type:** A dropdown menu set to "Advanced".
  - Entries:** A text area containing a JSON payload:

```

{
  "Entries": [
    {
      "Detail": {
        "orderId": "45154987945",
        "orderDate": "2024-06-21T10:15:00Z",
        "customerName": "SUJ",
        "totalAmount": 150.75,
        "items": [
          { "itemId": "SKU001", "itemName": "Product A", "quantity": 2, "unitPrice": 50.25 },
          { "itemId": "SKU002", "itemName": "Product B", "quantity": 1, "unitPrice": 50.25 }
        ]
      },
      "DetailType": "orderPlaced",
      "Source": "ecommerceSystem",
      "EventBusName": "Customer-Bus"
    }
  ]
}

```
  - Check if Bus Exists:** A checkbox that is currently unchecked.
  - Throw Exception on Failed Entry:** A checkbox that is currently unchecked.
- HEADER DETAILS:**
  - Request Headers:** A text input field.
  - Response Headers:** A text input field containing an asterisk (\*).

The processing tab contains the following fields:

Field	Description
Operation Type	Select Advanced from the available dropdown.
Entries	<p>Specify the entries that need to be sent to the event bus. This must be a JSON element.</p> <p>Example: <code>\${in.body}</code></p> <p>Sample payload that goes in body:</p> <pre data-bbox="399 527 1406 1507"> {   "Entries": [     {       "Detail": {         "orderId": "45154987945",         "orderDate": "2024-06-21T10:15:00Z",         "customerName": "SU",         "totalAmount": 150.75,         "items": [           { "itemId": "SKU001", "itemName": "Product A", "quantity": 2, "unitPrice": 50.25 },           { "itemId": "SKU002", "itemName": "Product B", "quantity": 1, "unitPrice": 50.25 }         ]       },       "DetailType": "orderPlaced",       "Source": "ecommerceSystem",       "EventBusName": "Customer-Bus"     },     {       "Detail": {         "orderId": "987654321",         "orderDate": "2024-06-22T11:30:00Z",         "customerName": "KC",         "totalAmount": 200.00,         "items": [           { "itemId": "SKU003", "itemName": "Product C", "quantity": 1, "unitPrice": 100.00 },           { "itemId": "SKU004", "itemName": "Product D", "quantity": 2, "unitPrice": 50.00 }         ]       },       "DetailType": "orderPlaced",       "Source": "ecommerceSystem",       "EventBusName": "Order-Bus"     }   ] } </pre>
Check if Bus Exists	Enable to validate the bus name for each entry in your payload.
Throw Exception on Failed Entry	Enable to check for the failed entries. In case of failures, an exception is thrown with failure count and failed entries.