Oracle® Cloud

Using the Apache Kafka Adapter with Oracle Integration 3

F45546-21
November 2025

Oracle Cloud Using the Apache Kafka Adapter with Oracle Integration 3,

F45546-21

Primary Author: Oracle Corporation

Contents

About This Content

This guide describes how to configure this adapter as a connection in an integration in Oracle Integration.

Audience

This guide is intended for developers who want to use this adapter in integrations in Oracle Integration.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Resources

See these Oracle resources:

- Oracle Cloud at `http://cloud.oracle.com`
- *Using Integrations in Oracle Integration 3*
- *Using the Oracle Mapper with Oracle Integration 3*
- Oracle Integration documentation on the Oracle Help Center.

Conventions

The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Understand the Apache Kafka Adapter

Review the following conceptual topics to learn about the Apache Kafka Adapter and how to use it as a connection in integrations in Oracle Integration. A typical workflow of adapter and integration tasks is also provided.

Topics

Apache Kafka Adapter Capabilities

The Apache Kafka Adapter enables you to create an integration in Oracle Integration that connects to an Apache Kafka messaging system. The Apache Kafka Adapter connects to the Apache Kafka distributed publish-subscribe messaging system from Oracle Integration and allows for the publishing and consumption of messages from a Kafka topic.

The Apache Kafka Adapter provides the following benefits:

- Establishes a connection to the Apache Kafka messaging system to enable messages to be published and consumed.
- Consumes messages from a Kafka topic and produces messages to a Kafka topic in the invoke (outbound) direction.
- Consumes messages from a topic based on a specified frequency in the trigger (inbound) direction.

> **Note**
>
> Message consumption in the inbound direction is only supported with use of the on-premises connectivity agent.

- Reads data from topics in the Kafka cluster using the Apache Kafka Streaming API.
- Enables you to browse the available metadata using the Adapter Endpoint Configuration Wizard (that is, the topics and partitions to which messages are published and consumed).
- Supports debatching of messages for trigger operations.
- Supports a consumer group.
- Supports headers.
- Supports the following message structures:
  - Sample JSON
  - XML schema (XSD) and schema archive upload

  - Sample XML
  - Avro schema
  - Opaque

  These schemas are applicable for the following scenarios:

  - Producing and consuming messages - Invoke connections (supported with both direct connectivity and the connectivity agent)
  - Consuming messages - Trigger connections (supported with the connectivity agent only)
- Supports connectivity to Amazon MSK using the Mutual TLS security policy with the connectivity agent.
- Supports connecting to private resources that are in your virtual cloud network (VCN) private subnet with a private endpoint. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3* and Configure the Endpoint Access Type. This type of connection does not use the connectivity agent.
- Supports polling of messages for processing in the Apache Kafka Adapter with or without use of the connectivity agent. See Perform Inbound Polling Without the Connectivity Agent.
- Supports the following security policies:
  - SASL OAuth (Client Credentials)
  - SASL Plain over SSL
  - TLS
  - Mutual TLS
  - SASL OAuth (Client Credentials using JWT Client Assertion)
  - Simple Authentication and Security Layer Plain (SASL/PLAIN)
- Supports direct connectivity to an Apache Kafka messaging system over SSL.
- Supports connectivity to an on-premises Apache Kafka messaging system through the connectivity agent.
- Supports integration with the Confluent Kafka platform to produce and consume messages.
- Supports optionally configuring the Kafka producer to be transactional. This enables an application to send messages to multiple partitions atomically. See Topic & Partition Page.
- Supports connectivity to Azure Event Hub using the SASL OAuth (Client Credentials) and SASL OAuth (Client Credentials using JWT Client Assertion) security policies. This connectivity enables existing Apache Kafka Adapter features to be utilized in Azure Event Hub integrations. See Apache Kafka on Azure Event Hubs.

See http://kafka.apache.org/.

You can configure the Apache Kafka Adapter as a trigger connection and an invoke connection in an integration in Oracle Integration. The Apache Kafka Adapter is one of many predefined adapters included with Oracle Integration. See the Adapters page in the Oracle Help Center.

Apache Kafka Adapter Restrictions

Note the following Apache Kafka Adapter restrictions in Oracle Integration.

- There are restrictions when using Confluent Kafka with the Apache Kafka Adapter. See Configure Confluent Kafka with the Apache Kafka Adapter.
- Only GZIP compression (the default) is supported. Other forms of compression are not supported.

> **Note**
>
> There are overall service limits with Oracle Integration. A service limit is the quota or allowance set on a resource. See Service Limits.

What Application Version Is Supported?

For information about which application version is supported by this adapter, see the Connectivity Certification Matrix.

Workflow to Create and Add a Apache Kafka Adapter Connection to an Integration

You follow a very simple workflow to create a connection with an adapter and include the connection in an integration in Oracle Integration.

This table lists the workflow steps for both adapter tasks and overall integration tasks, and provides links to instructions for each step.

Step	Description	More Information
1	Decide where to work	• Work in a project (see why working with projects is preferred in *Using Integrations in Oracle Integration 3*). • Work outside a project.
2	Create the adapter connections for the applications you want to integrate. The connections can be reused in multiple integrations and are typically created by the administrator.	Create an Apache Kafka Adapter Connection
3	Create the integration. When you do this, you add trigger (source) and invoke (target) connections to the integration.	Create Integrations in *Using Integrations in Oracle Integration 3* and Add the Apache Kafka Adapter Connection to an Integration. **Note**: The Apache Kafka Adapter can only be used as an invoke connection to produce and consume operations.
4	Map data between the trigger connection data structure and the invoke connection data structure.	Map Data in *Using Integrations in Oracle Integration 3*
5	(Optional) Create lookups that map the different values used by those applications to identify the same type of object (such as gender codes or country codes).	Manage Lookups in *Using Integrations in Oracle Integration 3*
6	Activate the integration.	Activate Integrations in *Using Integrations in Oracle Integration 3*

Step	Description	More Information
7	Monitor the integration on the dashboard.	Monitor Integrations in *Using Integrations in Oracle Integration 3*
8	Track payload fields in messages during runtime.	Assign Business Identifiers for Tracking Fields in Messages and Manage Business Identifiers for Tracking Fields in Messages in *Using Integrations in Oracle Integration 3*
9	Manage errors at the integration level, connection level, or specific integration instance level.	Manage Errors in *Using Integrations in Oracle Integration 3*

2

Create an Apache Kafka Adapter Connection

A connection is based on an adapter. You define connections to the specific cloud applications that you want to integrate. The following topics describe how to define connections.

Topics

Prerequisites for Creating a Connection

You must satisfy the following prerequisites to create a connection with the Apache Kafka Adapter.

Know the Host and Port of the Bootstrap Server

Know the host and port of the bootstrap server to use to connect to a list of Kafka brokers.

Obtain Security Policy Details

Obtain the following security policy details for the Apache Kafka Adapter.

- If using the Simple Authentication and Security Layer (SASL) Plain over SSL or SASL Plain security policy, know the SASL username and password.
- To use SASL Plain over SSL, TLS, or Mutual TLS policies, have the required certificates.

Configure Confluent Kafka with the Apache Kafka Adapter

To configure Confluent Kafka with the Apache Kafka Adapter, you must obtain the following information to successfully configure the Apache Kafka Adapter on the Connections page.

1. Generate the username and password required for the Connections page at the following location:

   ```
   https://confluent.cloud/environments/env-dvgny/clusters/lkc-rjn91/api-keys
   ```

   - Click **Add Key**.

     You must enter the key in the **SASL Username** field and the secret key in the **SASL Password** field on the Connections page. See Configure Connection Security.
2. Generate the truststore:

a. Generate the certificate.

```
 echo -n | openssl s_client -connect host:port | sed -ne  '/-BEGIN
CERTIFICATE-/,/-END CERTIFICATE-/p' > /tmp/server.cert
```

Where *host*:*port* is the combination of the bootstrap server and port.

b. Generate the truststore from the certificate created in Step a:

```
keytool -keystore conf_2.jks -alias ConfRoot
-import -file conf_server.cert -storetype JKS
```

> **Note**
>
> Specify the `-storetype JKS` option when generating the truststore with a JDK version higher than 8.

For this example, `conf_2.jks` is the name of the truststore file to upload in the **TrustStore** field on the Connections page.

c. When prompted, enter a password. Remember the password because you must enter it in the **Truststore password** field on the Connections page.

Note the following Confluent Kafka restrictions:

- The Apache Kafka Adapter supports Apache Kafka serializers/deserializers (String/ByteArray). It doesn't support Confluent or any other serializers/deserializers.
- Supports only the SASL PLAIN over SSL security policy.
- Supports the XML/JSON and AVRO message structures. Other structures/formats are not supported.
- The schema registry is not supported with the Apache Kafka Adapter.

Create a Connection

Before you can build an integration, you must create the connections to the applications with which you want to share data.

> **Note**
>
> You can also create a connection in the integration canvas. See Define Inbound Triggers, Outbound Invokes, and Actions.

To create a connection in Oracle Integration:

1. Decide where to start:
   - Work in a project (see why working with projects is preferred).
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations** .

d. In the **Connections** section, click **Add** if no connections currently exist or **+** if connections already exist. The Create connection panel opens.

- Work outside a project.
  a. In the navigation pane, click **Design**, then **Connections**.
  b. Click **Create**. The Create connection panel opens.

2. Select the adapter to use for this connection. To find the adapter, scroll through the list, or enter a partial or full name in the **Search** field.
3. Enter the information that describes this connection.

Element	Description
Name	Enter a meaningful name to help others find your connection when they begin to create their own integrations.
Identifier	Automatically displays the name in capital letters that you entered in the **Name** field. If you modify the identifier name, don't include blank spaces (for example, `SALES_OPPORTUNITY`).
Role	Select the role (direction) in which to use this connection. **Note**: *Only* the roles supported by the adapter you selected are displayed for selection. Some adapters support all role combinations (trigger, invoke, or trigger and invoke). Other adapters support fewer role combinations. When you select a role, only the connection properties and security policies appropriate to that role are displayed on the Connections page. If you select an adapter that supports both invoke and trigger, but select only one of those roles, you'll get an error when you try to drag the adapter into the section you didn't select. For example, assume you configure a connection for the Oracle Service Cloud (RightNow) Adapter as only an **invoke**. Dragging the adapter to a **trigger** section in the integration produces an error.
Keywords	Enter optional keywords (tags). You can search on the connection keywords on the Connections page.
Description	Enter an optional description of the connection.

Element	Description
Share with other projects	**Note**: This field only appears if you are creating a connection in a project. Select to make this connection publicly available in other projects. Connection sharing eliminates the need to create and maintain separate connections in different projects. When you configure an adapter connection in a different project, the **Use a shared connection** field is displayed at the top of the Connections page. If the connection you are configuring matches the same type and role as the publicly available connection, you can select that connection to reference (inherit) its resources. See Add and Share a Connection Across a Project.

4. Click **Create**.

   Your connection is created. You're now ready to configure the connection properties, security policies, and (for some connections) access type.

5. Follow the steps to configure a connection.
   The connection property and connection security values are specific to each adapter. Your connection may also require configuration with an access type such as a private endpoint or an agent group.

6. Test the connection.

Configure Connection Properties

Enter connection information so your application can process requests.

1. Go to the **Properties** section.
2. In the **Bootstrap Servers** field, specify the host and port to use to connect to a list of Kafka brokers. A Kafka cluster consists of one or more servers (Kafka brokers) running Kafka. Producers are processes that publish data (push messages) to Kafka topics within the broker. A consumer of topics pulls messages from a Kafka topic.

   You cannot enter multiple bootstrap servers for one Apache Kafka Adapter connection.

Configure Connection Security

Configure security for your Apache Kafka Adapter connection by selecting the security policy and security token.

1. Go to the **Security** section.
2. Select the security policy.

Security Policy	Description
SASL PLAIN over SSL	Simple Authentication and Security Layer (SASL) is a framework for authentication and data security in Internet protocols. It separates authentication mechanisms from application protocols to enable any authentication mechanism supported by SASL to be used in any application protocol that uses SASL. Plain-text authentication assumes that the user name and password are submitted to the server in clear text. Therefore, this authentication method is only considered secure when using an encrypted connection. This security policy enables you to use SASL Plain with SSL encryption.
SASL OAuth (Client Credentials)	SASL can be used as an authentication method in a variety of protocols, including OAuth. OAuth client credentials enable HTTP authorization schemes in the OAuth framework to be used within the SASL framework. A client authenticates to an OAuth authorization server over HTTPS, which issues tokens after successfully authenticating the resource owner. The obtained token may then be presented in an OAuth-authenticated request to the resource server.
TLS	Transport Layer Security (TLS) is a cryptographic protocol that provides end-to-end security of data sent between applications over the Internet.
Mutual TLS	Mutual TLS is a security practice that uses client TLS certificates to provide an additional layer of protection that allows client information to be cryptographically verified. Mutual TLS enables the server to authenticate the identity of the client.
No Security Policy	Do not use any security policy.
SASL OAuth (Client Credentials using JWT Client Assertion)	SASL can be used as an authentication method in a variety of protocols, including OAuth. OAuth JWT client assertions enable HTTP authorization schemes in the OAuth framework to be used within the SASL framework. A JWT client assertion is produced by creating a JSON payload and then signing it with a private key.
SASL PLAIN	Use SASL Plain without SSL encryption.

3. Based on your security policy selection, enter the following details:

If You Selected...	Specify These Details...
SASL OAuth (Client Credentials)	• **Access Token URI**: Provide the request URI to obtain the access token. • **Client ID**: Provide the client ID to identify the client making the request. • **Client Secret**: Provide the client secret to identify the client making the request. • (Optional) **Client Secret**: Provide the scope. **Note**: Refer to the documentation provided by your service provider for instructions on how to configure these fields.
SASL PLAIN over SSL This option enables you to use direct connectivity and eliminates the need to perform the procedures described Configure the Endpoint Access Type.	• **SASL Username**: Enter the SASL username. • **SASL Password** and **Confirm SASL Password**: Enter the password, then enter it a second time to confirm. • **TrustStore**: Select the check box, then click **Upload** to upload the truststore. • **KeyStore**: Select the check box, then click **Upload** to upload the keystore. • **TrustStore password** and **Confirm TrustStore password**: Enter the password, then enter it a second time to confirm. • **KeyStore password** and **Confirm KeyStore password**: Enter the password, then enter it a second time to confirm.
TLS	• **TrustStore**: Select the check box, then click **Upload** to upload the truststore. • **TrustStore password** and **Confirm TrustStore password**: Enter the password, then enter it a second time to confirm.
Mutual TLS	• **TrustStore**: Select the check box, then click **Upload** to upload the truststore. • **KeyStore**: Select the check box, then click **Upload** to upload the keystore. • **TrustStore password** and **Confirm TrustStore password**: Enter the password, then enter it a second time to confirm. • **KeyStore password** and **Confirm KeyStore password**: Enter the password, then enter it a second time to confirm. • **Key password** and **Confirm Key password**: Enter the password, then enter it a second time to confirm.

If You Selected...	Specify These Details...
SASL OAuth (Client Credentials using JWT Client Assertion)	• **Access Token URI**: Provide the request URI to obtain the access token. • **JWT headers in JSON format**: Upload the JSON-formatted JWT headers. • **JWT payload in JSON format**: Upload the JSON-formatted payload. • **JWT Private Key**: Upload the JWT private key. • (Optional) **JWT Private Key Password**: Enter the JWT password. • (Optional) **Scope**: Enter the scope. **Note**: Refer to the documentation provided by your service provider for instructions on how to configure these fields.
SASL PLAIN	• **SASL Username**: Enter the SASL username. • **SASL Password** and **Confirm SASL Password**: Enter the password, then enter it a second time to confirm.

Configure the Endpoint Access Type

Configure access to your endpoint. Depending on the capabilities of the adapter you are configuring, options may appear to configure access to the public internet, to a private endpoint, or to an on-premises service hosted behind a fire wall.

- Select the Endpoint Access Type
- Ensure Private Endpoint Configuration is Successful

Select the Endpoint Access Type

1. Go to the **Access type** section.
2. Select the option for accessing your endpoint.

Option	This Option Appears If Your Adapter Supports ...
Public gateway	Connections to endpoints using the public internet.
Private endpoint	Connections to endpoints using a private virtual cloud network (VCN). **Note**: To connect to private endpoints, you must complete prerequisite tasks in the Oracle Cloud Console. Failure to do so results in errors when testing the connection. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3* and Troubleshoot Private Endpoints in *Using Integrations in Oracle Integration 3*.

Option	This Option Appears If Your Adapter Supports ...
Connectivity agent	Connections to on-premises endpoints through the connectivity agent. a. Click **Associate agent group**. The Associate agent group panel appears. b. Select the agent group, and click **Use**. To configure an agent group, you must download and install the on-premises connectivity agent. See Download and Run the Connectivity Agent Installer and About Creating Hybrid Integrations Using Oracle Integration in *Using Integrations in Oracle Integration 3*.

Ensure Private Endpoint Configuration is Successful

- To connect to private endpoints, you must complete prerequisite tasks in the Oracle Cloud Console. Failure to do so results in errors when testing the connection. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3*.
- When configuring an adapter on the Connections page to connect to endpoints using a private network, specify the fully-qualified domain name (FQDN) and *not* the IP address. If you enter an IP address, validation fails when you click **Test**.

Test the Connection

Test your connection to ensure that it's configured successfully.

1. In the page title bar, click **Test**. What happens next depends on whether your adapter connection uses a Web Services Description Language (WSDL) file. Only some adapter connections use WSDLs.

If Your Connection...	Then...
Doesn't use a WSDL	The test starts automatically and validates the inputs you provided for the connection.
Uses a WSDL	A dialog prompts you to select the type of connection testing to perform: • **Validate and Test**: Performs a full validation of the WSDL, including processing of the imported schemas and WSDLs. Complete validation can take several minutes depending on the number of imported schemas and WSDLs. No requests are sent to the operations exposed in the WSDL. • **Test**: Connects to the WSDL URL and performs a syntax check on the WSDL. No requests are sent to the operations exposed in the WSDL.

2. Wait for a message about the results of the connection test.
   - If the test was successful, then the connection is configured properly.
   - If the test failed, then edit the configuration details you entered. Check for typos and verify URLs and credentials. Continue to test until the connection is successful.
3. When complete, click **Save**.

3

Add the Apache Kafka Adapter Connection to an Integration

When you drag the Apache Kafka Adapter into the trigger or invoke area of an integration, the Adapter Endpoint Configuration Wizard is invoked. This wizard guides you through configuration of the Apache Kafka Adapter endpoint properties.

The following sections describe the wizard pages that guide you through configuration of the Apache Kafka Adapter as a trigger and an invoke in an integration.

Topics

Basic Info Page

You can enter a name and description on the Basic Info page of each adapter in your integration.

Element	Description
What do you want to call your endpoint?	Provide a meaningful name so that others can understand the responsibilities of this connection. You can include English alphabetic characters, numbers, underscores, and hyphens in the name. You can't include the following characters: • No blank spaces (for example, `My Inbound Connection`) • No special characters (for example, `#;83&` or `righ(t)now4`) except underscores and hyphens • No multibyte characters
What does this endpoint do?	Enter an optional description of the connection's responsibilities. For example: `This connection receives an inbound request to synchronize account information with the cloud application.`
Messaging Type This field only appears when configuring the adapter in the trigger direction.	Select a messaging type: • **Consumer**: Reads data from topics in the Kafka cluster using the Apache Kafka Consumer API. • **Streaming**: Reads data from topics in the Kafka cluster using the Apache Kafka Streaming API.

Operations Page

Select the operation to perform.

Element	Description
What operation do you want to perform on a Kafka topic?	• **Publish records to a Kafka topic** • **Consume records from a Kafka topic** • **Consume records from a Kafka topic by specifying offset** • **Consume records from a Kafka topic by specifying multiple offsets and ranges**

Topic & Partition Page

Select the operation and topic on which to perform the operation and optionally specify the message structure.

- Configure the Apache Kafka Adapter as an Invoke
- Configure the Apache Kafka Adapter as a Trigger

Configure the Apache Kafka Adapter as an Invoke Connection

Element	Description
Select a Topic	Select the topic on which to perform the operation. You can also enter the beginning letters of the topic to filter the display of topics. A topic is a category in which applications can add, process, and reprocess messages. You subscribe to messages in topics.
Specify the Partition (This field is only displayed if you select **Publish records to a Kafka topic** or **Consume records from a Kafka topic**.)	Specify the partition. Kafka topics are divided into partitions that enable you to split data across multiple brokers. If you do not select a specific partition and use the default selection, Kafka considers all available partitions and decides which one to use.
Consumer Group (This field is only displayed if you select **Consume records from a Kafka topic**.)	Specify the consumer group to attach. Consumers join a group by using the same group ID. Kafka assigns the partitions of a topic to the consumers in a group. **Note**: When using a nondefault partition, specify the consumer group specific to the topic partition.

Element	Description
Specify the option for consuming messages (This field is only displayed if you select **Consume records from a Kafka topic**.)	• **Read latest**: Reads the latest messages starting at the time at which the integration was activated. • **Read from beginning**: Select to read messages from the beginning. As an example, if you select to read from the beginning and have activated the integration, the first scheduled run picks up 20 records and the next scheduled run picks up the next 20 records. If the integration is then deactivated, edited, and reactivated, the next 20 records are picked up. • **Read from specific offset**: Select to reset the consumer to a specific offset (partition-specific). Click **Add** to add the partition and the offset. Once configured, if you edit it again, the following options are displayed: • **Continue reading**: Reads the next message from where it left off before deactivation. • **Reset to beginning**: Reads the message again from the beginning (may cause duplication). • **Read from specific offset**: Select to reset the consumer to a specific offset (partition-specific). Click **Add** to add the partition and the offset (may cause duplication). **Notes**: • When you select **Read from beginning**, **Read latest**, or **Read from specific offset**, not all messages are guaranteed to be consumed in a single run. However, any remaining messages are consumed in subsequent runs. • Selecting **Read from beginning**, **Reset to beginning**, or **Read from specific offset** resets the consumer group offset upon completing the wizard.
Maximum Number of Records to be fetched (This field is only displayed if you select **Consume records from a Kafka topic** or **Consume records from a Kafka topic by specifying offset**.)	Specify the number of messages to read. The threshold for the complete message payload is 10 MB. **Note**: This field specifies the upper boundary of records to fetch. It does not guarantee the specified amount of records to retrieve from the stream in a single run. Remaining messages are fetched in subsequent runs.
Do you want to specify the message structure?	Select **Yes** if you want to define the message structure to use on the Message Structure page of the wizard. Otherwise, select **No**. **Note**: If **No** is selected, specify the **Message Type** value as **Opaque(Base64binary)** under **Advanced Options** for an opaque schema.
Do you want to specify the headers for the message?	Select **Yes** if you want to define the message headers to use on the Headers page of the wizard. Otherwise, select **No**.

Element	Description
Review and update advanced configurations	Click **Edit** to open the Advanced Options section to enable or disable the transactional producer. 1. **Transaction Producer**: This field is only displayed if you select **Publish records to a Kafka topic**. This option provides the following capabilities: a. If selected, the transactional producer enables an application to send messages to multiple partitions atomically. b. If not selected, the Apache Kafka Adapter is configured as a nontransactional producer. 2. **Message Type**: This option defines the message type. Available options are **String** or **Bytes**. It defines the serializers to use for the message. This selection is applicable for the message key and value.

Configure the Apache Kafka Adapter as a Trigger Connection

Element	Description
Select a Topic	Select the topic on which to perform the operation. You can also enter the beginning letters of the topic to filter the display of topics. A topic is a category in which applications can add, process, and reprocess messages. You subscribe to messages in topics.
Specify the Partition (This field is only displayed if you are reading data from topics using the Kafka Consumer API.)	Specify the partition to which to push the selected topic. Kafka topics are divided into partitions that enable you to split data across multiple brokers. If you do not select a specific partition and use the **Default** selection, Kafka considers all available partitions and decides which one to use.
Consumer Group (This field is only displayed if you are reading data from topics using the Kafka Consumer API.)	Specify the consumer group to attach. Consumers join a group by using the same group ID. Kafka assigns the partitions of a topic to the consumers in a group. **Note**: When using a nondefault partition, specify the consumer group specific to the topic partition.

Element	Description
Specify the option for consuming messages	• **Read latest**: Reads the latest messages starting at the time at which the integration was activated. • **Read from beginning**: Select to read messages from the beginning. • **Read from specific offset**: Select to reset the consumer to a specific offset (partition-specific). Click **Add** to add the partition and the offset. Once configured, if you edit it again, the following options are displayed: • **Continue reading**: Reads the next message from where it left off before deactivation. • **Reset to beginning**: Reads the message again from the beginning (may cause duplication). • **Read from specific offset**: Select to reset the consumer to a specific offset (partition-specific). Click **Add** to add the partition and the offset (may cause duplication). **Note**: • When you select **Read from beginning**, **Read latest**, or **Read from specific offset**, not all messages are guaranteed to be consumed in a single run. However, any remaining messages are consumed in subsequent runs. • Selecting **Read from beginning**, **Reset to beginning**, or **Read from specific offset** resets the consumer group offset upon completing the wizard.
Application ID (This field is only displayed if you are reading data from topics using the Kafka Streaming API.)	Specify the application ID to attach. The stream must have a unique ID.
Polling Frequency (Sec) (This field is only displayed if you are reading data from topics using the Kafka Consumer API.)	Specify the frequency at which to fetch records. **Note**: Trigger-based polling supports a maximum frequency interval of 10 minutes.
Do you want to Debatch the messages?	Select **Yes** if you want to debatch the messages. Debatch allows you to select the maximum number of messages to be raised per tracking instance. Otherwise, select **No**.
Maximum Number of Records to be fetched (This field is only displayed if you are reading data from topics using the Kafka Consumer API.)	Specify the number of messages to read. The threshold for the complete message payload is 10 MB. **Note**: This field specifies the upper boundary of records to fetch. It does not guarantee the specified amount of records to retrieve from the stream in a single run. Remaining messages are fetched in subsequent runs.
Maximum Number of Records to be raised per Instance (This field is only displayed when **Debatch** is selected.)	Specify the maximum number of records to be raised per tracking instance in Oracle Integration.
Do you want to specify the message structure?	Select **Yes** if you want to define the message structure to use on the Message Structure page of the wizard. Otherwise, select **No**.
Do you want to specify the headers for the message?	Select **Yes** if you want to define the message headers to use on the Headers page of the wizard. Otherwise, select **No**.

Element	Description
Review and update advanced configurations	Click **Edit** to open the Advanced Options section. 1. **Message Type**: This option defines the message type. Available options are **String** or **Bytes**. It defines the serializers to use for the message. This selection is applicable for the message key and value.
Review and update rejected message (DLQ) configurations	Define the topic for rejected messages. You can select the topic to which the rejected message is copied.

Message Structure Page

Select the message structure to use. This page is displayed if you selected **Yes** for the **Do you want to specify the message structure?** field on the Topic & Partition page.

Element	Description
How would you like to specify the message structure?	• **Avro Schema document** • **Sample JSON document** • **XML schema (XSD) document** • **Sample XML document**
Select File	Click **Browse** to select the file. Once selected, the file name is displayed in the **File Name** field.
Element	Select the element if you specified an XSD or Avro file.

Headers Page

Define the message headers structure to attach to the message. This page is displayed if you selected **Yes** for the **Do you want to specify the headers for the message?** field on the Topic & Partition page.

Element	Description
Specify Message Headers	Specify headers and optional descriptions.

Summary Page

You can review the specified adapter configuration values on the Summary page.

Element	Description
Summary	Displays a summary of the configuration values you defined on previous pages of the wizard. The information that is displayed can vary by adapter. For some adapters, the selected business objects and operation name are displayed. For adapters for which a generated XSD file is provided, click the XSD link to view a read-only version of the file. To return to a previous page to update any values, click the appropriate tab in the left panel or click **Go back**. To cancel your configuration details, click **Cancel**.

4

Implement Common Patterns Using the Apache Kafka Adapter

You can use the Apache Kafka Adapter to implement the following common patterns.

Topics:

Consume Messages from an Apache Kafka Topic by Specifying Multiple Offsets and Ranges

You can configure an integration to use the Apache Kafka Adapter to consume messages from an Apache Kafka topic by specifying multiple offsets and ranges.

The following integration provides one example of how to implement this pattern:

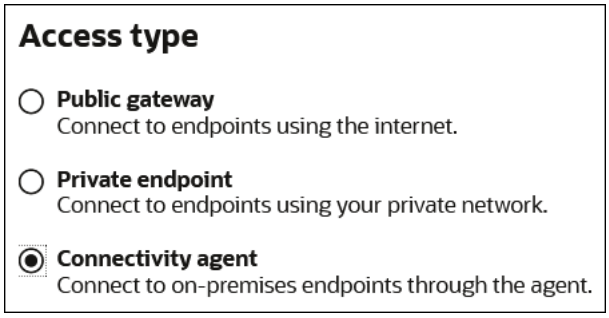
- Create an application integration with a SOAP Adapter or REST Adapter as a trigger connection.
- Configure the Apache Kafka Adapter to:
  - Consume records from a Kafka topic by specifying multiple offsets and ranges.
  - Specify the topic and message structure to use. For example, specify an XML schema (XSD) document and the headers to use for the message.
- Configure the mapper to perform appropriate source-to-target mappings between the SOAP Adapter or REST Adapter and the Apache Kafka Adapter.
- Map or specify the partition and its offset range in the mapping (for example, the **Partition** can be `0` and the **Offset Range** format can be `6,10-11,16`).

Perform Inbound Polling Without the Connectivity Agent

You can perform inbound polling without the connectivity agent. You can create new connections or update existing connections to bypass the connectivity agent. Depending on your database or messaging service location, you can use the public internet or a private endpoint for polling. If you want, you can continue to use the connectivity agent for inbound polling.

Perform the following steps to create a new connection or update an existing connection to bypass the connectivity agent.

1. Go to the **Access type** section of the Connections page.



2. Select an option based on the location of your database or messaging service.
   - If the database or messaging service to access is in a private network, select **Private endpoint** to poll without the connectivity agent. This selection requires that you first configure private endpoint support in the Oracle Cloud Console. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3*.
   - If the database or messaging service to access is public, select **Public gateway** to poll without the connectivity agent. Private endpoint configuration is not required in these scenarios and the adapter polls directly without the connectivity agent.
3. Deactivate and then reactivate the integration.

> **Note**
>
> After completing these steps, the connectivity agent is no longer used for inbound polling. However, it still continues to run. If you no longer need to use the connectivity agent, you can manually stop it.

Several adapters support inbound polling without the connectivity agent. Watch a video to learn more about how to use this feature with one such adapter.

Video

Produce Messages to an Apache Kafka Topic

You can configure a scheduled orchestrated integration to read records using the FTP Adapter and stage file read action and publish them to an Apache Kafka topic using the Apache Kafka Adapter produce operation.

The following integration provides one example of how to implement this pattern:

- A scheduled orchestrated integration that runs once.
- A mapper to perform appropriate source-to-target mappings between the schedule and an FTP Adapter.
- An FTP Adapter to fetch files (records) from an input directory and put them in a download directory.
- A stage file action configured to:
  - Perform a **Read File in Segments** operation on each file (record) in the download directory.
  - Specify the structure for the contents of the message to use (for this example, an XML schema (XSD) document).

  - Perform appropriate source-to-target mappings between the stage file action and an Apache Kafka Adapter.
- An Apache Kafka Adapter configured to:
  - Publish records to a Kafka topic.
  - Specify the message structure to use (for this example, an XML schema (XSD) document) and the headers to use for the message.
- A mapper to perform appropriate source-to-target mappings between the Apache Kafka Adapter and FTP Adapter.
- An FTP Adapter to delete files from the download directory when processing is complete.

Consume Messages from an Apache Kafka Topic

You can configure a scheduled orchestrated integration to use the Apache Kafka Adapter to consume messages from an Apache Kafka topic at specific intervals and invoke a child integration (for example, an Oracle Database-based integration for insertion of records into a database table) to process the consumed messages.

The following integration provides one example of how to implement this pattern:

- A scheduled orchestrated integration that runs every ten minutes.
- A mapper to perform appropriate source-to-target mappings between the schedule and an Apache Kafka Adapter.
- An Apache Kafka Adapter configured to:
  - Consume records from a Kafka topic.
  - Specify the consumer group to attach. Kafka assigns the partitions of a topic to the consumers in a group.
  - Specify **Read latest** as the option for consuming messages. The latest messages are read starting at the time at which the integration is activated.
  - Specify the message structure to use (for this example, an XML schema (XSD) document) and the headers to use for the message.
- A mapper for performing appropriate source-to-target mappings between the Apache Kafka Adapter and a SOAP Adapter.
- A SOAP Adapter invoked to insert the Kafka topic record into an Oracle Database table.

5

Troubleshoot the Apache Kafka Adapter

Review the following topics to learn about troubleshooting issues with the Apache Kafka Adapter.

Topics

Additional integration troubleshooting information is provided. See Troubleshoot Oracle Integration in *Using Integrations in Oracle Integration 3* and the Oracle Integration Troubleshooting page on the Oracle Help Center.

Recover from a CLOUD-0005: Unable to Establish Connection Error

If you receive a `CASDK-0005 : A connector specific exception was raised by the application. java.util.concurrent.TimeoutException` error, ensure that the bootstrap URL, port, and security policy are correct and the Apache Kafka server is up and running.