

Using Bitbucket Cloud Adapter in Oracle Integration

Table of Contents

1.	Learn about the Bitbucket Cloud Adapter	3
1.1.	Features and Capabilities	3
1.2.	What Application Version Is Supported?	4
2.	Creating and configuring a Bitbucket Cloud adapter connection	4
2.1.	Prerequisites for Creating a Connection	4
2.1.1.	Create an OAuth Consumer	4
2.1.2.	Configure a Client Identifier for Bitbucket Trigger Connections	5
2.2.	Steps to Create a Connection	6
2.2.1.	Configure Connection Security	7
2.2.2.	Test the Connection	8
3.	Adding the Adapter to an Integration	9
3.1.	Trigger Basic Info Page	10
3.2.	Trigger Configuration Page	11
3.3.	Invoke Basic Info Pag	12
3.4.	Summary Page Element	13
4.	Implementing business use cases using the Bitbucket Cloud Adapter	13

1. Learn about the Bitbucket Cloud Adapter

Explore the key concepts to understand the Bitbucket Cloud Adapter and how to use it as a connection in integrations in Oracle Integration.

Topics Covered:

- [Features and Capabilities](#)
- [What Application Version Is Supported?](#)

1.1. Features and Capabilities

The Bitbucket Cloud Adapter connects Oracle Integration to Bitbucket Cloud. It can invoke Bitbucket Cloud operations and receive Bitbucket webhook events as an integration trigger.

The Bitbucket Cloud Adapter provides the following capabilities:

- Supports integrating Oracle Integration with Bitbucket Cloud workspaces, projects, repositories, pull requests, source files, branches and tags, commits, and commit statuses.
- Supports repository-level and workspace-level event triggers.
- Enables creation, retrieval, update, listing, comparison, merge, review, and deletion of supported Bitbucket resources.
- Allows text and binary file uploads when creating source commits.
- Returns JSON, binary file or diff, and no-content success responses.
- Validates incoming webhooks using a shared secret.

Supported Modules

Module	Supported Operations
Workspaces	Workspaces are used to organise repositories, members, permissions, and projects.
Projects	Retrieve individual Bitbucket workspace projects.
Repositories	Create, retrieve, update, delete, fork, and list Bitbucket repositories.
Pull Requests	Create, review, update, merge, and decline pull requests.
Source	Read and update repository source files and file history.
Refs	List and manage repository branches and tags.
Commits	List, retrieve, and compare repository commits.
Commit Statuses	Monitor and update build statuses associated with commits and pull requests.
Repository Events	Receive a selected event from a selected repository.
Workspace Events	Receive a selected event for a workspace.

1.2. What Application Version Is Supported?

The adapter supports Bitbucket Cloud REST API 2.0. Use the current Bitbucket Cloud documentation to confirm permissions and service limits for your plan.

2. Creating and configuring a Bitbucket Cloud adapter connection

Before you can use the adapter in an integration, you must create a connection in Oracle Integration. The following section explains how to set up and configure the connection:

- [Prerequisites for Creating a Connection](#)
- [Steps to Create a Connection](#)

2.1. Prerequisites for Creating a Connection

You must complete the steps below before setting up the connection:

- Identify the Bitbucket Cloud workspace to use. To create a workspace, see [Join or create a workspace](#).
- A Bitbucket account with access to the required workspace, project, repository, branch, and pull-request resources.
- Workspace or repository administrator permission when the integration creates and manages webhooks.
- The exact OAuth callback URL shown by Oracle Integration when the Authorization Code is used.
- HTTPS connectivity from Oracle Integration to Bitbucket Cloud.

2.1.1. Create an OAuth Consumer

1. Sign in to **Bitbucket Cloud** and open the target workspace.
2. Open **Workspace settings**.
3. Under **Apps and features**, select **OAuth consumers**.
4. Click **Add consumer**.
5. Enter a meaningful **Name** and **Description**.
6. Enter the exact **Callback URL** provided by Oracle Integration.

For example: `https://OIC_instance_URL/icsapis/agent/oauth/callback`

Note: The **Callback URL** must exactly match the value provided by Oracle Integration. Do not modify, shorten, or guess this value.

7. Select only the **Scopes** required by your integrations.
8. Click **Save**, then securely record the generated **Client ID** and **Client Secret**.

Prepare a Webhook Secret for Trigger Connections

Create a strong, random **Webhook Secret** to validate incoming webhook requests. Enter the same secret in the Oracle Integration connection. Do not include the secret in mappings, payload samples, descriptions, or logs.

2.1.2. Configure a Client Identifier for Bitbucket Trigger Connections

For Bitbucket trigger connections, configure the **Client Identifier** in OCI IAM—not in Bitbucket application. This identifier represents an application in the Oracle Integration identity domain that is authorized with the *ServiceInvoker* role.

- [Create and activate a confidential application](#)
- [Assign the Service Invoker role](#)

Oracle Integration uses this identity to authorize incoming repository or workspace events after validating the webhook signature.

Create and activate a confidential application.

1. Sign in to the Oracle Cloud Console with permissions to create applications and assign Oracle Integration application roles.
2. Navigate to **Identity & Security**, then **Domains**.
3. Select the appropriate compartment and open the identity domain associated with your Oracle Integration instance.
4. Select **Integrated applications**, then **Add application**.
5. Select **Confidential Application**, then click **Launch workflow**.
6. Enter a descriptive name, for example, *Bitbucket_Webhook_Client*. Optionally, enter a description.
7. Continue to the configuration pages. See [Oracle confidential-application setup](#).
 - a. For **Client configuration**, select **Skip for later**.
 - b. For **Resource server configuration** or **Resources**, select **Skip for later**.
 - c. Leave web-tier configuration unconfigured.

8. Complete the wizard, select **Activate**, and confirm activation.
9. Copy the generated **Client ID** from the application details.

Note: Console labels and page order can vary by OCI IAM version. See [Oracle identity-service instructions](#).

Assign the ServiceInvoker role.

1. In the same identity domain, open **Oracle Cloud Services**.
2. Select the service application for the target Oracle Integration instance.
3. Open **Application roles**.
4. Locate the *ServiceInvoker* role.
5. Select **Manage applications** or select **Manage** next to **Assigned applications**.
6. Select **Assign applications**.
7. Select the confidential application created in the previous section.
8. Select **Assign** and verify that the application appears in the assigned-applications list.

Assign *ServiceInvoker* to the confidential application itself, not only to an individual user or group. This is the application authorization required by the HMAC policy. See [Oracle HMAC requirements](#).

2.2. Steps to Create a Connection

Follow the steps mentioned below to set up a connection with the applications you want to exchange data with.

1. Navigate to **Design >Connections** in Oracle Integration
2. Click **Create**.
Note: You can also create a connection directly from the integration canvas. Refer [Define Inbound Triggers, Outbound Invokes, and Actions](#). For more details.
3. In the **Create Connection** panel, find the adapter you want to use by either scrolling through the list or typing its name in the search bar.
4. Fill in the connection details:
 - a. **Name:** Provide a clear and descriptive name. The system will automatically turn it into an uppercase Identifier (e.g., BITBUCKETCLOUD_INVOKE). Avoid using spaces.
 - b. **Role:** Select the connection role: invoke or trigger.

- c. **Note:** The list will show only the roles that the adapter will support. If you choose only one role and later try to use the other, you may encounter errors. For example, a Bitbucket Cloud Adapter set to invoke only will not work as a trigger.
 - d. **Optional:** You can add Keywords to identify the connection later and a description to explain its purpose
5. Click **Create**.

Your connection is now created. The next step is to configure connection settings like properties, security, and agent group (if needed).

2.2.1. Configure Connection Security

Enter the necessary information to set up the connection.

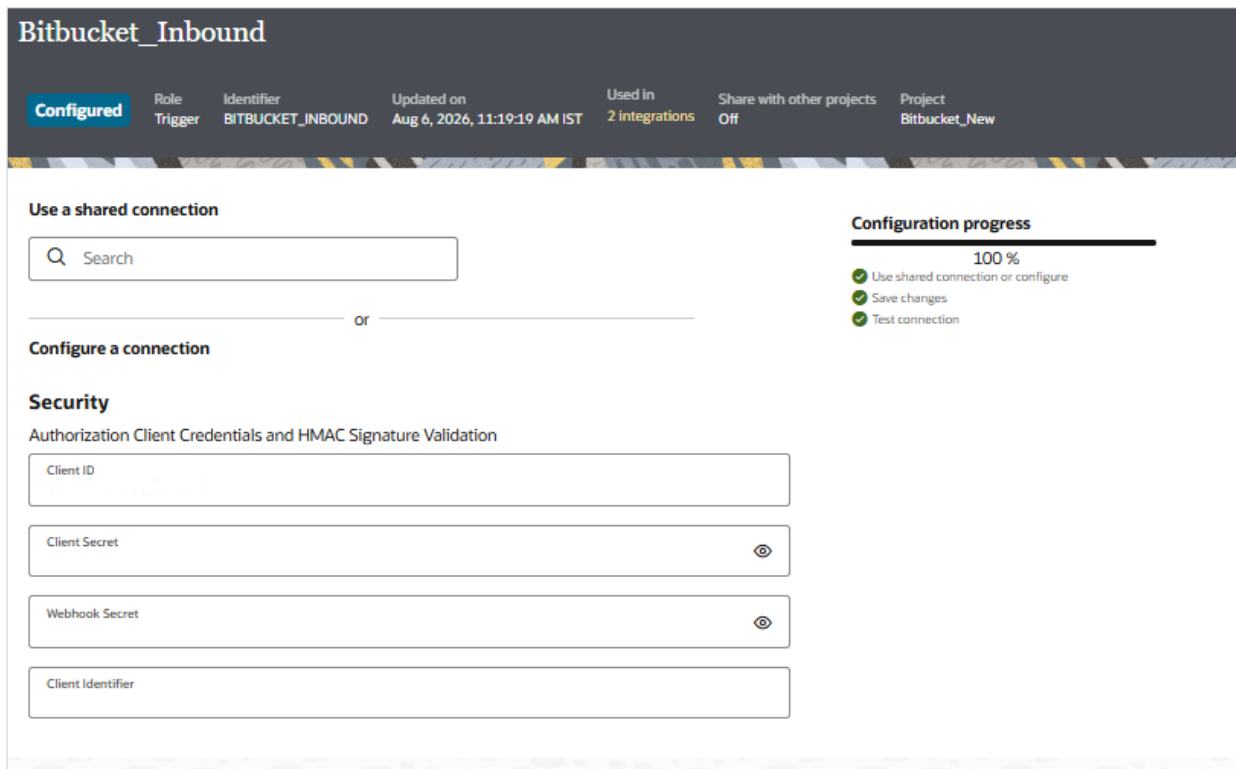
1. Go to the **Security** section.
2. For trigger connections: Use the **Authorization Client Credentials and HMAC Signature Validation** policy for a Bitbucket Cloud trigger connection. This policy authorizes webhook administration and validates incoming Bitbucket Cloud webhook deliveries.
 - a. Enter the **Client ID** and **Client Secret** from the Bitbucket Cloud OAuth consumer. See [Prerequisites for Creating a Connection](#).
 - b. Enter the **Webhook Secret** used to validate incoming Bitbucket Cloud webhook deliveries.
 - c. Enter the **Client Identifier** required for the connection. The Client Identifier is a predefined, instance-specific value. See [Configure a Client Identifier for Bitbucket Trigger Connections](#).
3. For invoke connections: In the **Security** section, select the authentication policy that matches the Bitbucket Cloud access required by the integration.
 - a. **Security Policy-** Bitbucket OAuth Authorization Code
 - I. Enter the **Client ID** and **Client Secret** from the Bitbucket Cloud OAuth consumer. See [Prerequisites for Creating a Connection](#).
 - II. Click **Provide Consent** and sign in to Bitbucket Cloud to authorize the connection.
 - b. **Security Policy-** Bitbucket OAuth Client Credentials
 - I. Enter the **Client ID** and **Client Secret** from the Bitbucket Cloud OAuth consumer. See [Prerequisites for Creating a Connection](#).

2.2.2. Test the Connection

Test your connection to ensure that it is configured successfully.

1. In the page title bar, click **Test**.
2. A confirmation message indicates whether the connection is configured properly or not.
3. If the test was successful, then the connection has been configured.
4. If the test fails, review, cross-check for typos, and confirm the accuracy of URLs and credentials. Update and retest until you can test the connection successfully.
5. When complete, click **Save**, then click .

Trigger Connection page:



Bitbucket_Inbound

Configured	Role Trigger	Identifier BITBUCKET_INBOUND	Updated on Aug 6, 2026, 11:19:19 AM IST	Used in 2 Integrations	Share with other projects Off	Project Bitbucket_New
-------------------	-----------------	---------------------------------	--	---------------------------	----------------------------------	--------------------------

Use a shared connection

or

Configure a connection

Security

Authorization Client Credentials and HMAC Signature Validation

Client ID

Client Secret

Webhook Secret

Client Identifier

Configuration progress

100 %

- ✓ Use shared connection or configure
- ✓ Save changes
- ✓ Test connection

Invoke Connection Page:

The screenshot shows the 'Invoke' configuration page for the Bitbucket Cloud Adapter. At the top, a dark header bar contains the following information: 'Configured' (in a blue box), 'Role: Invoke', 'Identifier: BITBUCKET_AUTHCODE', 'Updated on: Aug 3, 2026, 03:01:06 PM IST', 'Not used in any integrations', 'Share with other projects: Off', and 'Project: Bitbucket_New'. Below the header, the page is divided into two main sections. On the left, under 'Use a shared connection', there is a search bar. Below that, under 'Configure a connection', there is a 'Security' section with a 'Provide Consent' button. The 'Security' section contains several input fields: 'Security policy' (a dropdown menu), 'Bitbucket OAuth Client Credentials', 'Bitbucket OAuth Authorization Code' (highlighted in blue), 'OAuth Client ID', and 'OAuth Client Secret' (with an eye icon for toggling visibility). On the right, a 'Configuration progress' section shows a progress bar at 100% and three steps: 'Use shared connection or configure', 'Save changes', and 'Test connection', all of which are marked with green checkmarks.

3. Adding the Adapter to an Integration

Once the connection is created, you can add it to an integration flow. When you drop the Bitbucket Cloud Adapter onto the canvas, the Adapter Endpoint Configuration Wizard launches.

The following sections describe the wizard pages that guide you through the configuration of the Bitbucket Cloud Adapter as a trigger or an invoke in an integration.

Topics:

- [Trigger Basic Info Page](#)
- [Trigger Configuration Page](#)
- [Invoke Basic Info Page](#)
- [Summary Page Element](#)

3.1. Trigger Basic Info Page

Use the **Edit Basic Info** page to define the Bitbucket Cloud event that starts the integration.

Field	Description
What do you want to call your endpoint?	Provide a clear and descriptive name that helps others understand the purpose of the connection. For instance, if you are setting up a database connection to add new employee data, a suitable name could be CreateEmployeeInDB. Naming rules: - You may use letters (a-z, A-Z), numbers (0–9), dashes (-), and underscores (_) Do not use: - Spaces (e.g., My DB Connection) - Special characters (e.g., #;83& or righ(t)now4) - Multibyte characters
What does this endpoint do?	Optionally, you can enter a description stating the use of this connection.
Select Module	Select the Bitbucket API Module. Available options are: Repository Events: To start the integration from an event in a selected repository. Workspace Events: To start the integration from an event in a workspace.
Trigger	Enter the trigger name to Filter based on the Bitbucket API module selected. Subscribe to Repository Event Subscribe to Workspace Event

The image displays two side-by-side screenshots of the 'Edit Basic Info' page for a Bitbucket trigger. Both screenshots show the same form structure with the following fields:

- What do you want to call your endpoint?**: A text input field with the placeholder text 'trigger'.
- What does this endpoint do?**: A text input field.
- Select a module**: A dropdown menu. In the left screenshot, 'Repository Events' is selected. In the right screenshot, 'Workspace Events' is selected. This field is highlighted with a red box in both images.
- Trigger**: A section containing a list of available triggers. In the left screenshot, 'Subscribe to Repository Event' is listed. In the right screenshot, 'Subscribe to Workspace Event' is listed.
- Buttons**: 'Cancel' and 'Continue' buttons at the bottom right of the form.

3.2. Trigger Configuration Page

Use the **Edit Configuration** page to select the Bitbucket Cloud workspace and event that start the integration.

Field	Description
Workspace	Enter the Bitbucket Cloud workspace in which the event occurs.
Repository	Select the repository in which the event occurs. This field is required when you select the module as Repository Events on the Basic Info Page.
Event	Select the workspace event that starts the integration, such as Repository Created, Push, Fork, etc.

 **Edit Configuration**
Bitbucket trigger

Workspace

Repository

▼

Event

▼

Cancel

Go back

Continue

3.3. Invoke Basic Info Pag

Use the **Edit Basic Info** page to define the Bitbucket Cloud operation that the integration invokes.

Field	Description
What do you want to call your endpoint?	Provide a clear and descriptive name that helps others understand the purpose of the connection. For instance, if you're setting up a database connection to add new employee data, a suitable name could be CreateEmployeeInDB. Naming rules: - You may use letters (a-z, A-Z), numbers (0–9), dashes (-), and underscores (_) Do not use: - Spaces (e.g., My DB Connection) - Special characters (e.g., #;83& or righ(t)now4) - Multibyte characters
What does this endpoint do?	Optionally, you can enter a description stating the use of this connection.
Select Module	Select the Bitbucket API Module. Available options are: Workspaces Projects Repositories Pull Requests Source Refs Commits Commit Statuses
Action	Select the action to perform from the selected module. The available actions depend on the selected module.



Edit Basic Info

Bitbucket_test invoke

What do you want to call your endpoint?

invoke

What does this endpoint do?

Select a module

Action

3.4. Summary Page Element

Element	Description
Summary	Displays a summary of the configuration values you entered in the previous steps of the wizard. Summary - To modify any settings, click Go Back. - To discard your changes, click Cancel. - To save and complete the configuration, click Finish.

4. Implementing business use cases using the Bitbucket Cloud Adapter

You can use the Bitbucket Cloud Adapter to implement the following common pattern:

Notify a Slack Channel When a Bitbucket Pull Request Is Created

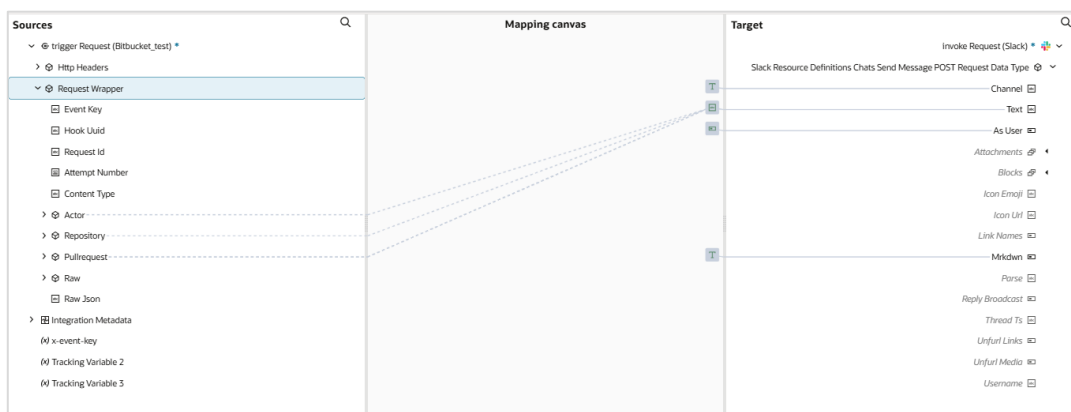
Use this integration to notify a Slack channel when a pull request is created in a Bitbucket Cloud repository. The Bitbucket Adapter receives the pull request event, and the Slack Adapter sends the pull request details to the selected Slack channel.

This use case involves the following adapters and their respective operations:

- **Bitbucket Adapter:** Used as a trigger with **Repository Events** and **Subscribe to Repository Event**.
- **Slack Adapter:** Used as an invoke with the Send Message operation.

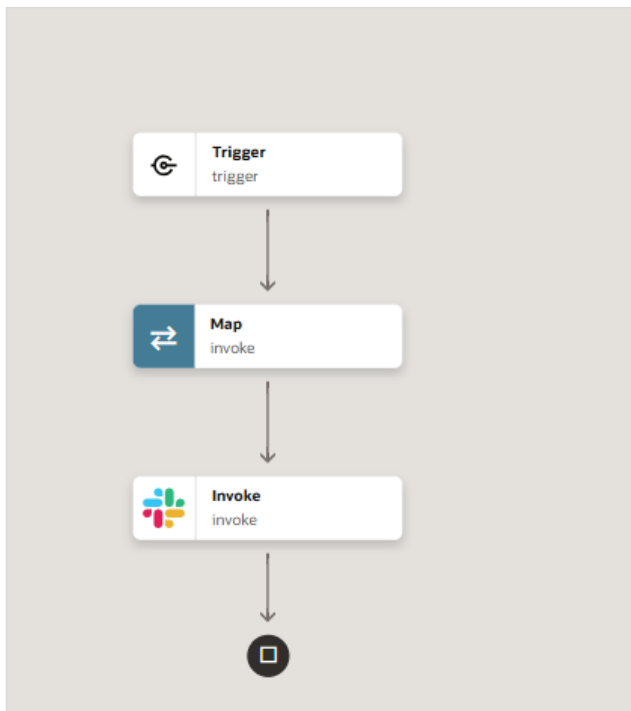
Follow the steps below to implement the use case:

1. Create an application integration.
2. Add the Bitbucket Adapter connection as the trigger.
3. Configure the Bitbucket adapter as follows:
 - On the **Basic Info** page, enter an endpoint name, such as *BitbucketPRCreated*.
 - Select **Repository Events** as the module.
 - Select **Subscribe to Repository Event** as the trigger, and then click **Continue**.
 - On the **Configuration** page:
 - Enter the Bitbucket workspace.
 - Select the repository.
 - Select **Pull Request Created** as the event.
 - Click **Continue**.
 - Review the configuration on the **Summary** page, and then click **Finish**.
4. Add the Slack Adapter connection as an invoke.
5. Configure the Slack adapter as follows:
 - On the Basic Info page, enter an endpoint name, such as *SendPRNotification*.
 - Select the **Send Message** operation, review the summary, and click **Finish**.
6. Open the mapper between the Bitbucket trigger and Slack invoke.
7. Map the Slack request fields.



8. Click **Validate** and resolve any mapping errors.

The completed integration looks as follows.



9. Click **Save** and activate the integration.
10. Create a pull request in the configured Bitbucket repository to test the integration.

Integration Flow Summary

When a pull request is created, Bitbucket Cloud sends the selected repository event to Oracle Integration. The integration maps the pull request details and sends a formatted notification to the configured Slack channel.

Sample Slack Notification

New Bitbucket pull request created.

Repository: <repository full name>

PR: #<pull request ID> - <pull request title>

Created by: <actor display name>

Source: <source branch>

Destination: <destination branch>

Open pull request: <pull request URL>