Atlassian Bitbucket Integrations (New)

Introduction

The Bitbucket integration collects audit logs from the audit log files or the audit API.

Reference: https://developer.atlassian.com/server/bitbucket/reference/rest-api/

Assumptions

The procedures described in Section 3 assume that a Log Collector has already been set up.

Requirements

For more information on auditing in Bitbucket and how it can be configured, see View and configure the audit log on Atlassian's website.

Reference: https://confluence.atlassian.com/bitbucketserver/view-and-configure-the-audit-log-776640417.html

Logs

Audit

The Confluence integration collects audit logs from the audit log files or the audit API from self-hosted Confluence Data Center. It has been tested with Confluence 7.14.2 but is expected to work with newer versions. As of version 1.2.0, this integration added experimental support for Atlassian Confluence Cloud. JIRA Cloud only supports Basic Auth using username and a Personal Access Token.

Atlassian Bitbucket Integration Procedures

Please provide the following information to CyTech:

Collect Bitbucket audit logs via log files

1. Path

2. Preserve Original Event? (Enable Yes/No)

- Preserves a raw copy of the original event, added to the field event.original

3. Tags

4. Processors (Optional)

5. Processors are used to reduce the number of fields in the exported event or to enhance the event with metadata. This executes in the agent before the logs are parsed.

6. Indexing settings (experimental) (Enable Yes/No)

7. Select data streams to configure indexing options. This is an experimental feature and may have effects on other properties.

Collect Bitbucket audit logs via API (Enable Yes/No)

1. API URL - The API URL without the path.

2. Bitbucket Username - JIRA Username. Needs to be used with a Password. Do not fill if you are using a personal access token.

3. Bitbucket Password - JIRA Password. Needs to be used with a Username. Do not fill if you are using a personal access token.

4. Personal Access Token - The Personal Access Token. If set, Username and Password will be ignored.

5. Initial Interval - Initial interval for the first API call. Defaults to 24 hours.

Create Access Token:

Access Tokens are single-purpose access tokens (or passwords) with access to a single workspace with limited permissions (specified at creation time). Use tokens for tasks such as scripting, CI/CD tools, and testing Bitbucket integrations or Marketplace apps while in development.

To create an Access Token:

1. At bitbucket.org, navigate to the target workspace for the Access Token. This workspace is the only one that the Workspace Access Token can access.
2. On the sidebar, select **Settings**.
3. On the sidebar, under **Security**, select **Access tokens**.
4. Select **Create Workspace Access Token**.
5. Give the Workspace Access Token a name, usually related to the app or task that will use the token.
6. Select the permissions the Access Token needs. *Note give the Access Token admin permission.*
7. Select the **Create** button. The page will display the **Workspace Access Token created** dialog.
8. Copy the generated token and either record or paste it into the app that requires access. *The token is only displayed once and can't be retrieved later*.

If you need further assistance, kindly contact our support at info@cytechint.com for prompt assistance and guidance.

Revision #1
Created 11 November 2024 02:57:45
Updated 11 November 2024 05:01:34