The Continuous Dynamic Plugin for Jira® Data Center

Black Duck Software, Inc.

2025-04-14

Table of Contents

Chapter 1. Continuous Dynamic Plugin for Jira Data Center

The Black Duck® Continuous Dynamic™ Plugin for Jira Data Center is designed to integrate the Continuous Dynamic Portal with Jira® Data Center edition. The Plugin provides a seamless interface within Jira that will sync vulnerabilities from the Portal to your Jira application.

As vulnerabilities are discovered and listed in the Portal, they are pulled into Jira and issues are created to get those vulnerabilities into the hands of the individuals responsible for remediation. The vulnerability can be retested and developers can use the Portal's **Ask A Question** feature directly from the issue in Jira as though they were in the Portal.

1.1. Prerequisites

Note the following prerequisites:

- The Continuous Dynamic Plugin for Jira Data Center is compatible with Jira Data Center LTS versions 10.3.0 - 10.3.5.
- You must be logged in as a Jira Administrator with global permissions to install or configure the plugin.

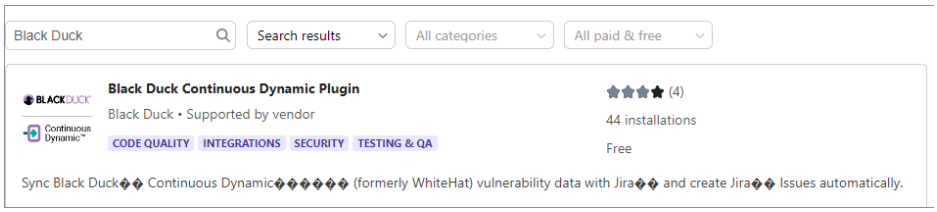
Chapter 2. Installing the Continuous Dynamic Plugin for Jira Data Center

You can install the **Continuous Dynamic Plugin for Jira Data Center** directly from the **Atlassian Marketplace for JIRA** page in Jira Data Center. We strongly recommend installing the plugin this way rather than from the public Atlassian Marketplace website (where the Data Center Plugin is available to download as a .jar file).

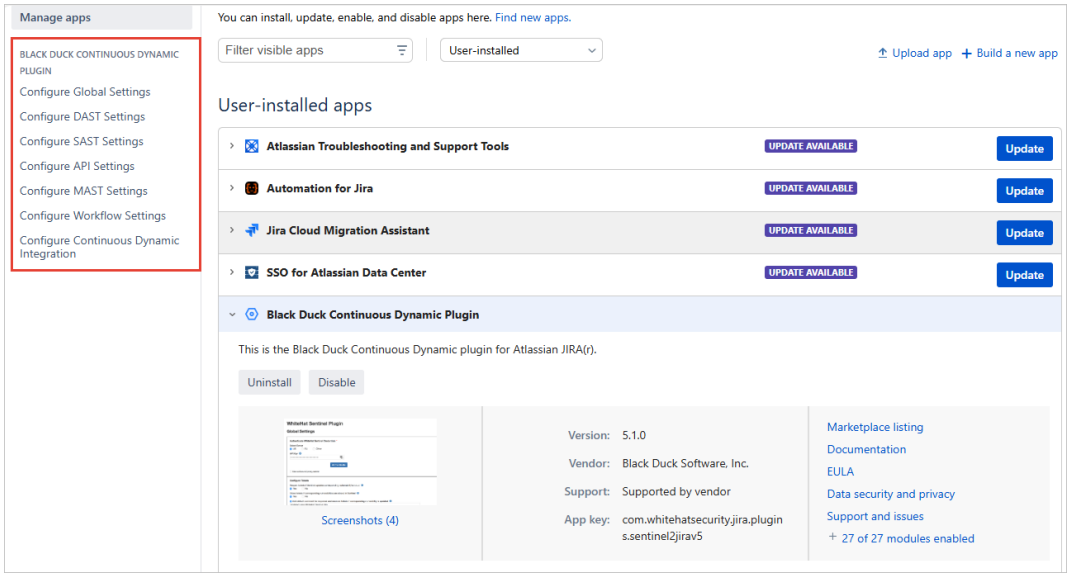
2.1. Installing the Plugin from the Atlassian Marketplace for JIRA

Perform the following steps:

1. Log in to your Jira instance as an admin.
2. From the Jira Administration menu, select **Manage apps**.
   The **Atlassian Marketplace for JIRA** page is displayed.
3. Select **Find new apps** and then search the marketplace for **Black Duck® Continuous Dynamic™**.

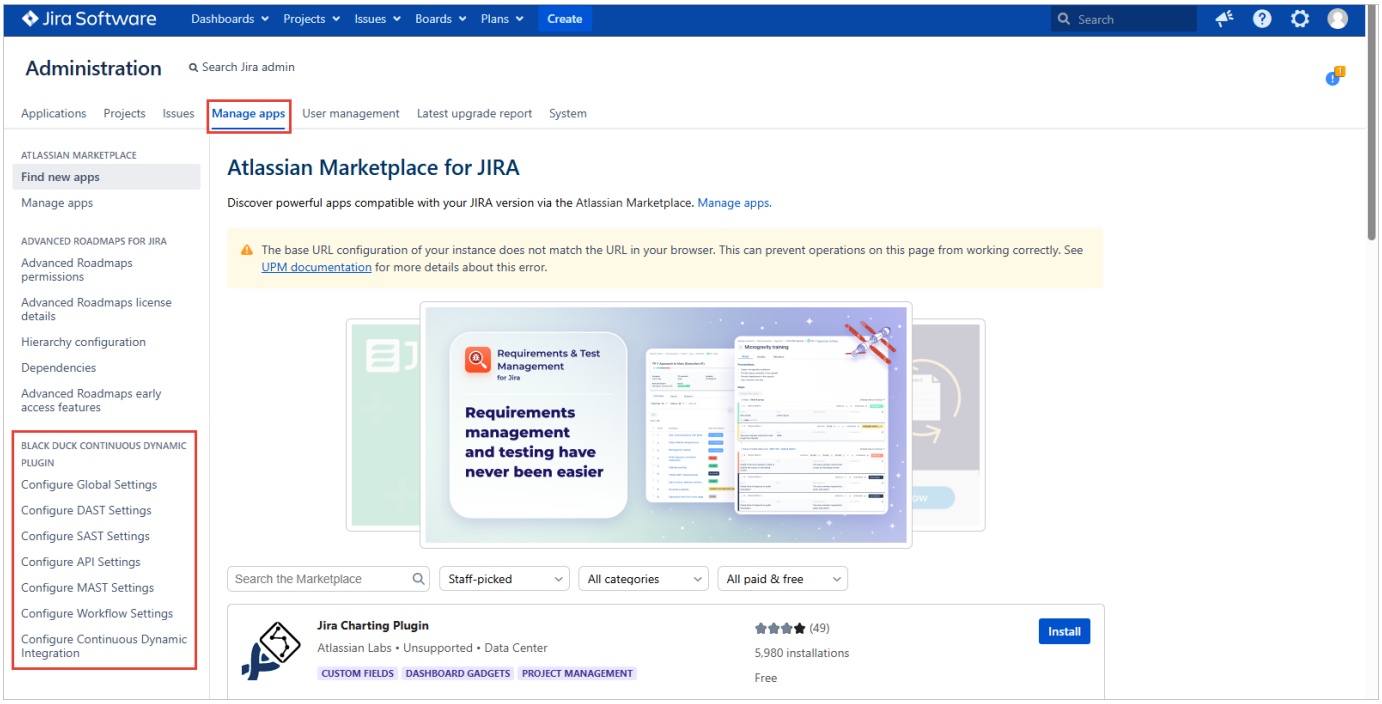


4. Click **Install**.
5. In the **Confirm app Installation** dialog, click **Accept & install**.
6. Select **Manage apps** in the left-hand menu; the plug-in is visible under **User-installed apps**. If necessary, refresh the page to display it.
7. Once this process is complete, new configuration options are added to the **Manage apps** page:


Manage apps
BLACK DUCK CONTINUOUS DYNAMIC PLUGIN
Configure Global Settings
Configure DAST Settings
Configure SAST Settings
Configure API Settings
Configure MAST Settings
Configure Workflow Settings
Configure Continuous Dynamic Integration
You can install, update, enable, and disable apps here. Find new apps.
Filter visible apps
User-installed
Upload app
Build a new app
User-installed apps
Atlassian Troubleshooting and Support Tools
UPDATE AVAILABLE
Update
Automation for Jira
UPDATE AVAILABLE
Update
Jira Cloud Migration Assistant
UPDATE AVAILABLE
Update
SSO for Atlassian Data Center
UPDATE AVAILABLE
Update
Black Duck Continuous Dynamic Plugin
This is the Black Duck Continuous Dynamic plugin for Atlassian JIRA(r).
Uninstall
Disable
Screenshots (4)
Version: 5.1.0
Vendor: Black Duck Software, Inc.
Support: Supported by vendor
App key: com.whitehatsecurity.jira.plugins.sentinel2jirav5
Marketplace listing
Documentation
EULA
Data security and privacy
Support and issues
27 of 27 modules enabled

Chapter 3. Configuring the Continuous Dynamic Plugin for Jira Data Center

After installing the Continuous Dynamic Plugin for Jira Data Center, new configuration options are added to the **Manage apps** page in Jira, as shown below. Refreshing the **Manage apps** page may be necessary to see these options.



NOTE

There are seven possible configurations displayed. For the Plugin to function, the **Global Settings**, **Workflow Settings**, **Continuous Dynamic Integration**, and at least one asset type (**DAST Settings**, **SAST Settings**, **API Settings**, or **MAST Settings**) must be configured.

- Configure Global Settings: Configure the connections between the Plugin and Continuous Dynamic. Also configure how basic issues will be handled.
- Configure DAST Settings: Configure the interactions between the Plugin and Continuous Dynamic DAST services.
- Configure SAST Settings: Configure the SAST interactions between the Plugin and Sentinel Source SAST services.
- Configure API Settings: Configure the interactions between the Plugin and the Continuous Dynamic API services.
- Configure MAST Settings: Configure the interactions between the Plugin and Continuous Dynamic Mobile services.
- Configure Workflow Settings: Configure the Jira workflow to define how issues will be created, reopened and closed.

- Configure Continuous Dynamic Integration: Start and stop the integration, or view the Plugin log files.

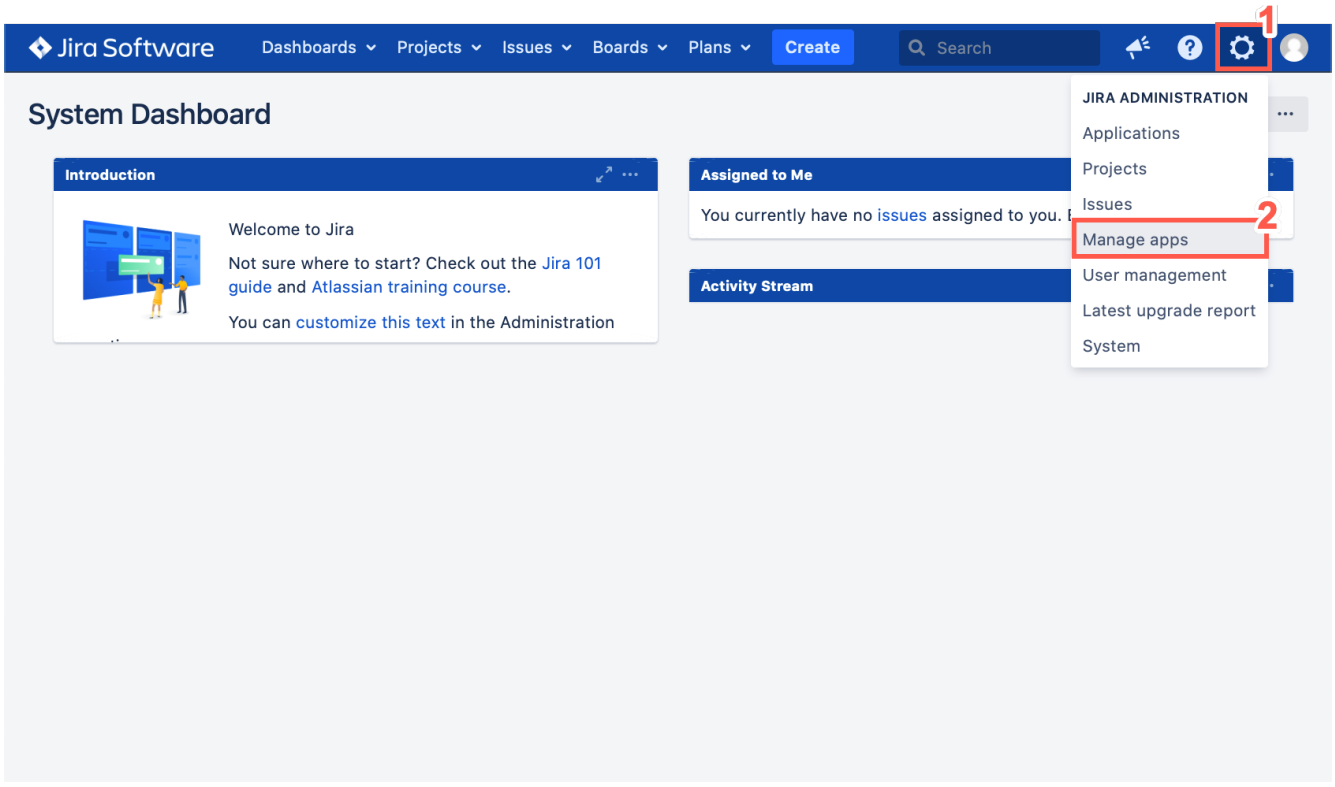
NOTE	In the existing version of the Plugin, if a user has DAST configured for a group in the Continuous Dynamic Portal that contains DAST, API, or Mobile assets, then if a vulnerability is created for any of those assets, a ticket is created in JIRA. For users migrating to the new version of the Plugin, if DAST is configured for that same group, only the DAST assets would sync with JIRA. API and Mobile assets must now be configured separately.

Once this process is complete, the Issues page displays vulnerability information and key Continuous Dynamic features.

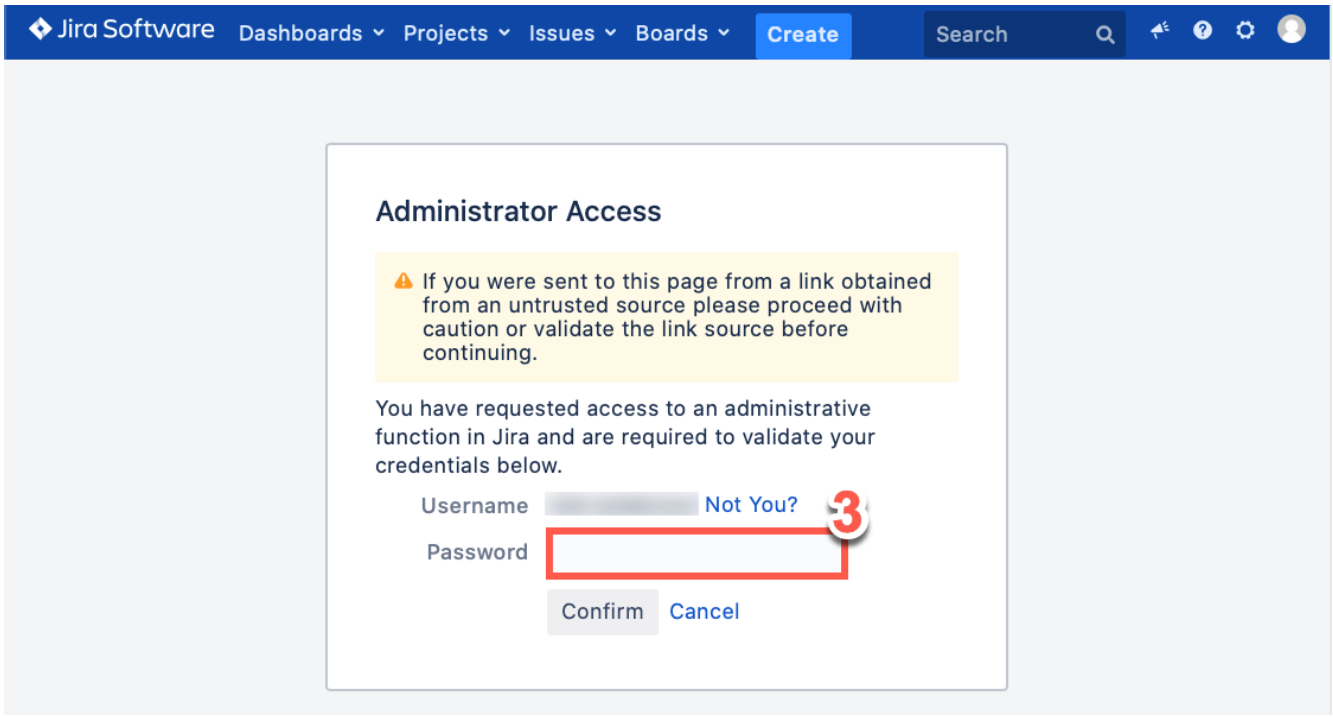
3.1. Configure Global Settings

To configure the Continuous Dynamic Plugin for Jira Data Center perform the following steps:

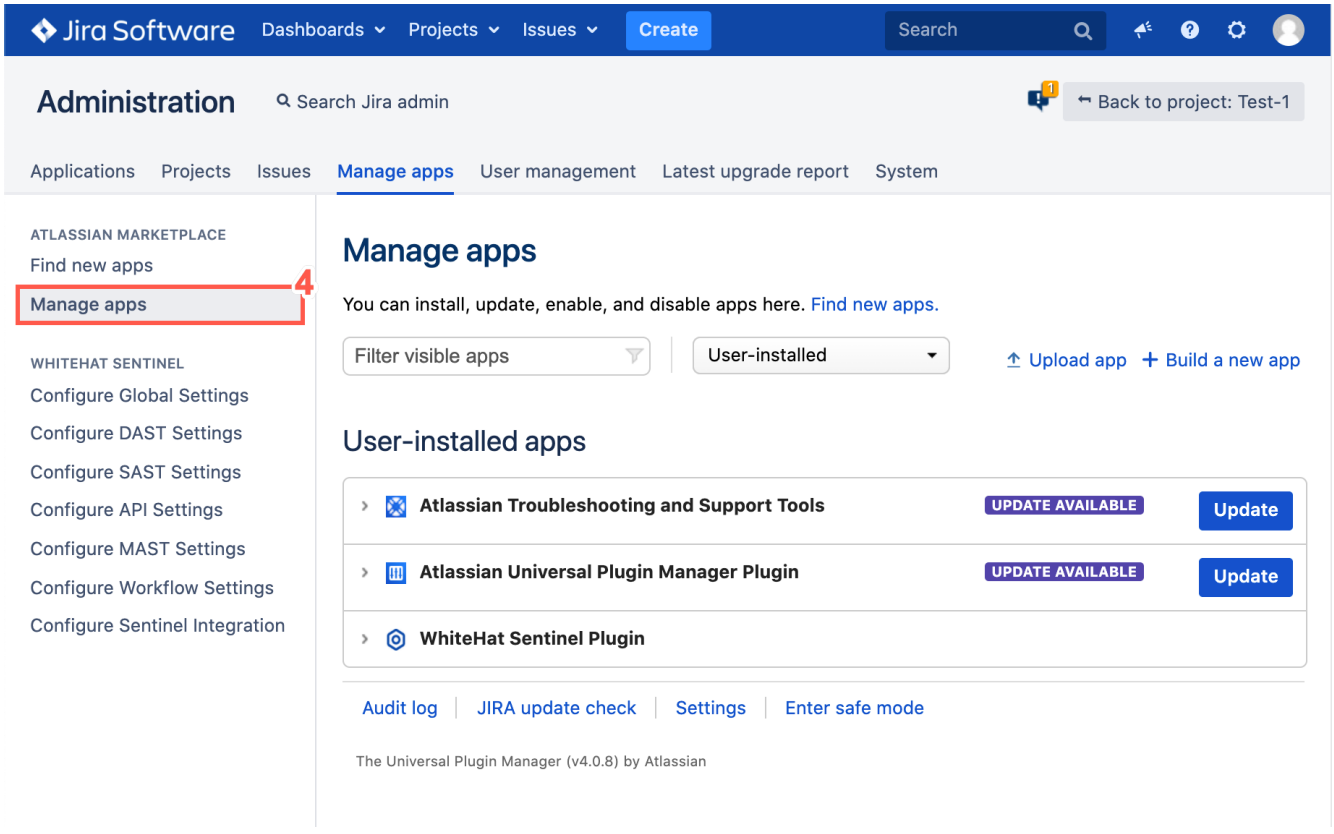
1. From the Jira **System Dashboard** click the gear icon.



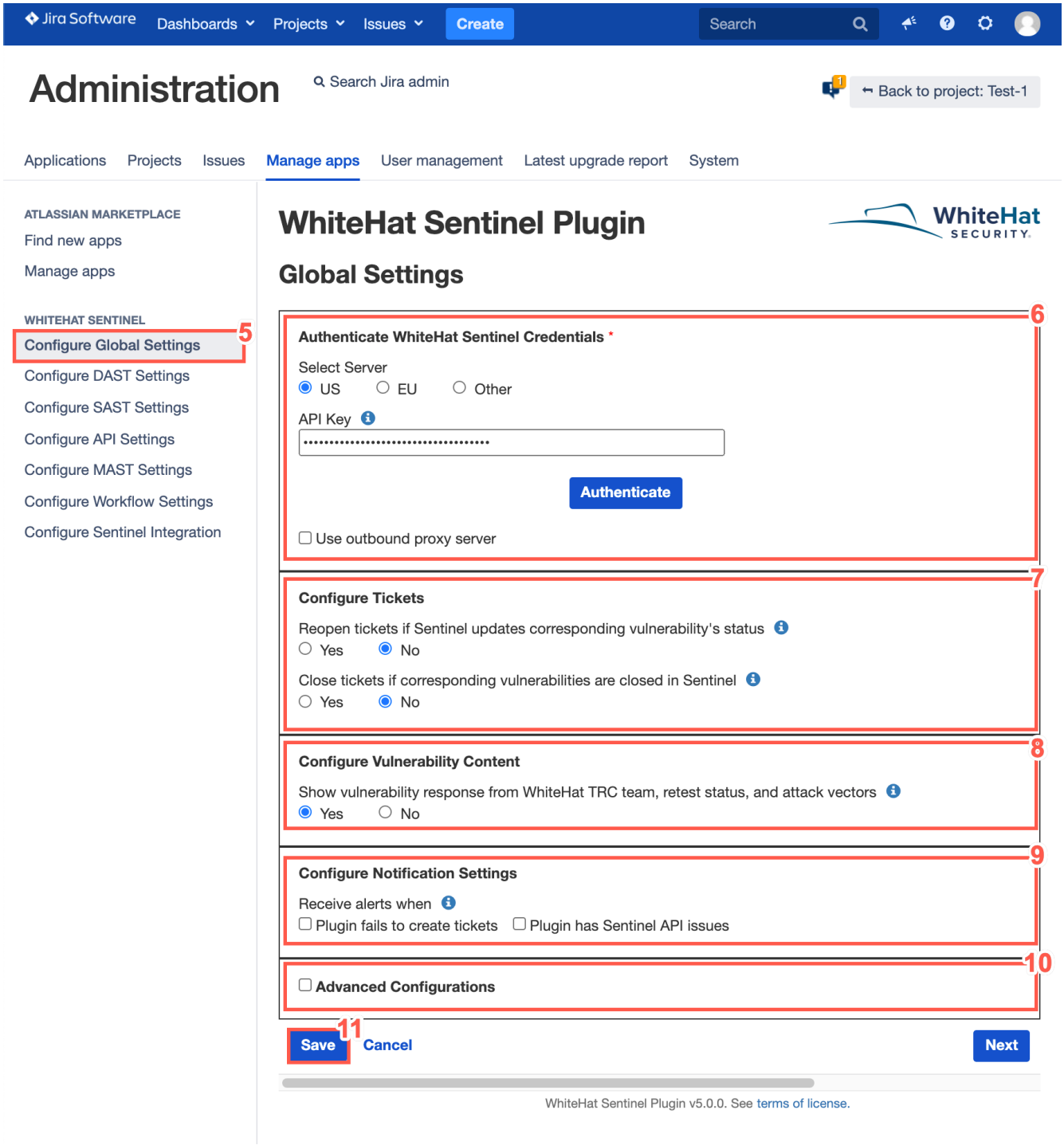
2. Click **Manage apps**.
3. Log into an Admin account to manage the apps installed on Jira.



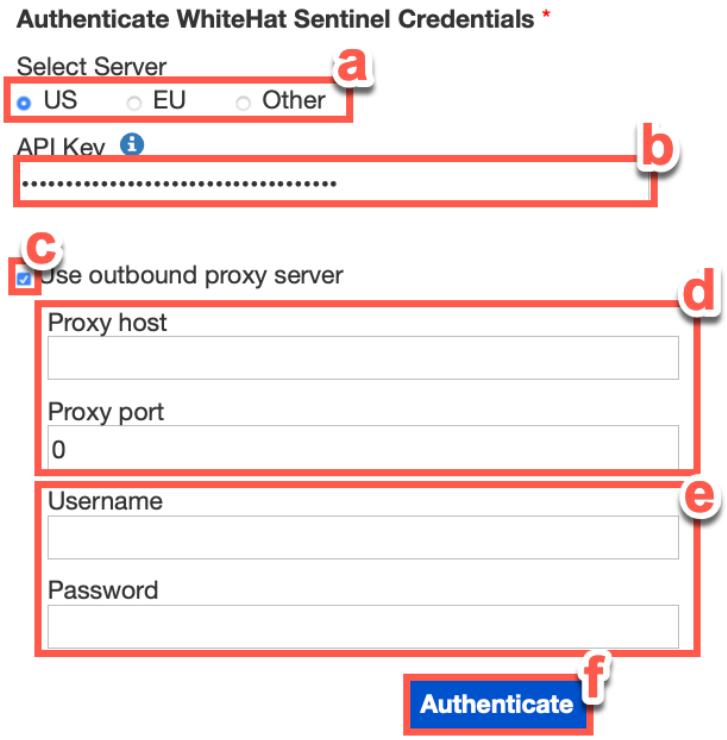
4. Click **Manage apps**.



5. Select **Configure Global Settings** to configure the plugin connection to Continuous Dynamic. The following page is displayed.



6. Authenticate **Continuous Dynamic Credentials**.



a. Select a US, EU, or Other server.

b. Enter your API key.

> **NOTE** A valid API key must be entered before the plugin can be used.

To learn how to generate an API key in the Continuous Dynamic Portal, see Your Continuous Dynamic Portal Profile.

c. To use an outbound proxy server, click the radio button.

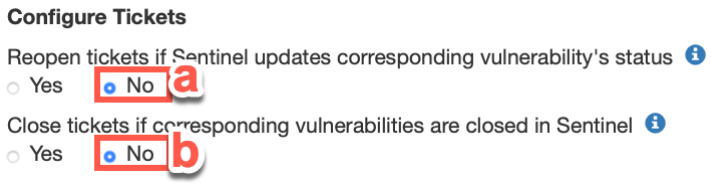
d. Type in the **Proxy Host** name and the **Proxy Port** number.

e. Type in your **Username** and **Password** to log into the proxy server.

f. Click **Authenticate** to check your connection.

7. Optionally, **Configure Jira Tickets**. Select the relevant radio buttons to configure default updates for your tickets.

a. To reopen closed tickets whenever a vulnerability's status is updated in the Portal.



b. To close your existing tickets automatically if corresponding vulnerabilities are closed in the Portal.

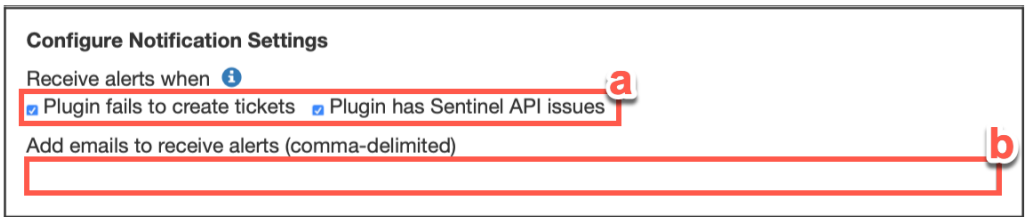
8. Optionally, configure the **Vulnerability Content**.

   Additionally, you can configure your tickets to view the retest status and notes or tags associated with a vulnerability.

   a. Select the relevant radio button to show responses from the Black Duck TRC, when your team members ask questions about a vulnerability.

9. Optionally, configure **Notification Settings**.

   a. Select the relevant radio button to determine when alerts should be sent.



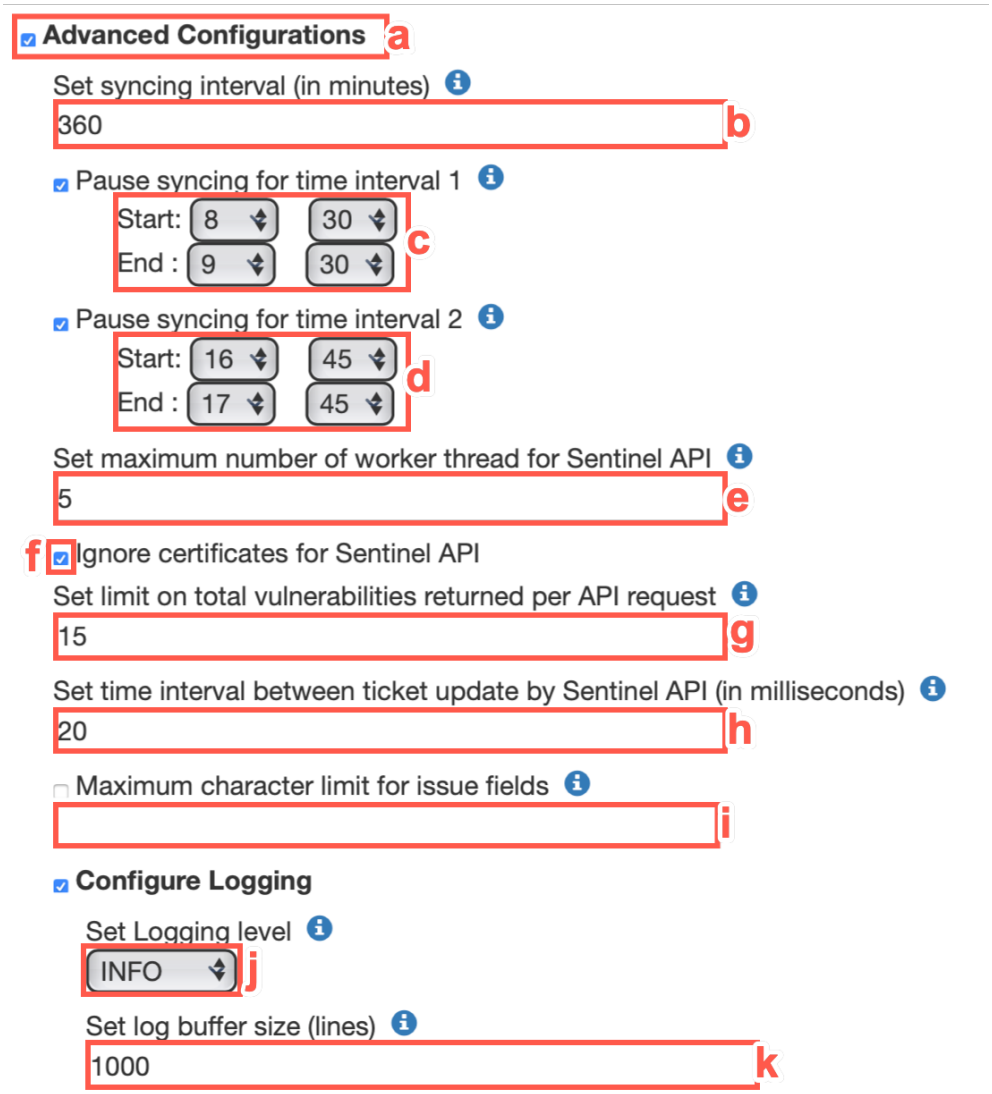
   b. Then enter a comma-delimited list of the emails that should receive alerts.

10. Optionally, make changes to the **Advanced Configurations**.

WARNING	Each of the following actions can cause the Plugin to consume more system resources to the expense of the Jira server and its other components or plugins:

   - Setting too small a syncing interval.
   - Setting too high a value for the maximum number of worker threads for the Continuous Dynamic API.
   - Setting too high a limit on total vulnerabilities returned per API request.
   - Setting too small a time interval between ticket updates.
   - Setting too high a maximum character limit for issue fields.

     a. Select the **Advanced Configurations** checkbox.



b. **Set Synching Interval (in minutes)** set the interval at which Continuous Dynamic and Jira will be synched.

c. **Pause Synching for time interval** set one or two daily time intervals during which synching will be paused. Select the **Pause syncing for time interval 1** checkbox, to see the start and end time boxes for the first and, optionally, second period during which you want to pause synchronization.

d. To pause synchronization between 8:30 and 9:30 a.m. and between 4:45 and 5:45 p.m., your entry should look like this.

NOTE	Each start and end time should be entered in hours (0-23) and minutes (0-59).

e. **Set Maximum Number of Worker Threads** cap the number of API requests that can be in process at once.

f. **Ignore Certification for Continuous Dynamic API** allows you to ignore any cert issues between your server and Continuous Dynamic.

WARNING	Do **not** enable the option to **Ignore Certification for Continuous Dynamic API**, unless troubleshooting connectivity issues with the Continuous Dynamic server.

g. **Set Limit on Total Vulnerabilities Returned per API Request** limit the number of tickets that can be created from a single request.

h. **Set Time Interval Between Ticket Updates** set the minimum interval (in milliseconds) between ticket updates from the Continuous Dynamic API.

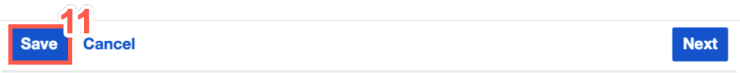
i. **Maximum character limit for issue fields** set a maximum character length for your Summary, Description, and Comments fields. Tick the checkbox to set your preferred maximum. The default value for this field will match **Jira's** global character limit, which is stored in 'jira.text.field.character.limit' and can be accessed from Administration>System>Advanced Settings. If you customize this value, it must be set to a value between zero and the maximum character limit set in 'jira.text.field.character.limit'.

j. **Configure Logging** set the Logging level to the detail level desired from:

- **INFO**
- **DEBUG**
- **TRACE**

a. **Log buffer size (lines)** set the number of lines that will be retained in the log buffer.

NOTE	The **Configure Logging** default values are a **Logging Level** of INFO and a **log buffer size** of 1000 lines.



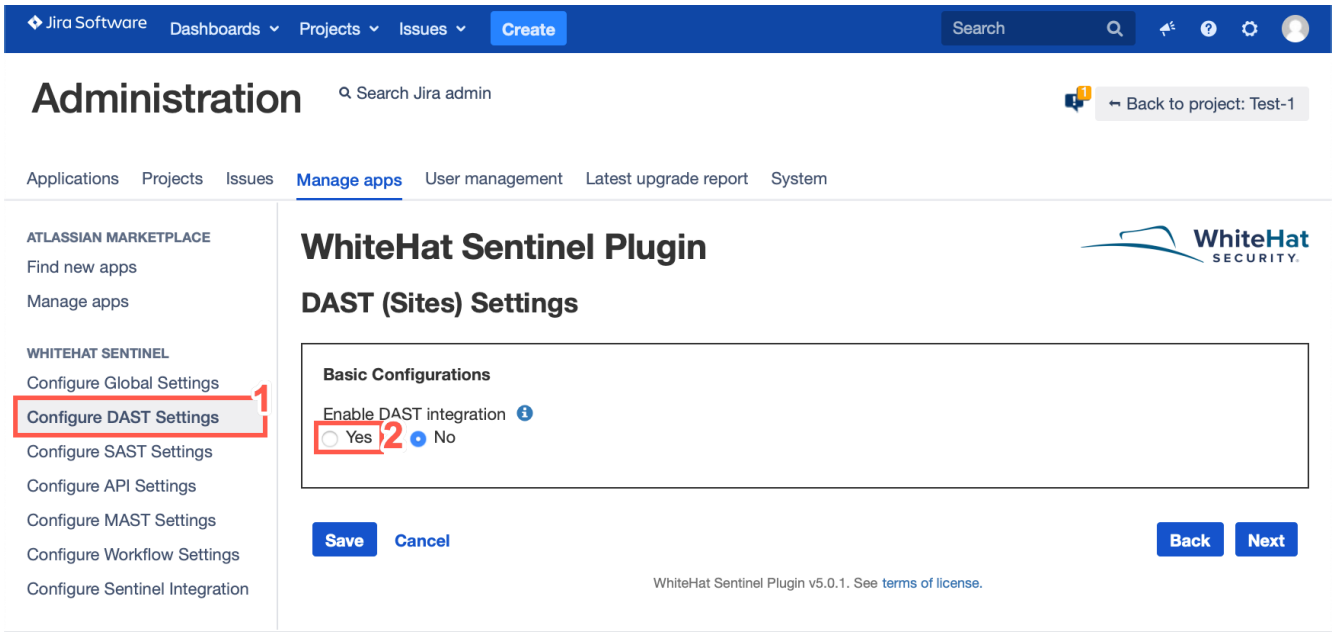
11. When you have completed the **Global Configuration** process, click **Save**.

3.2. Configure DAST or SAST Settings

Select **Configure DAST Settings** or **Configure SAST Settings** to set your default reporter and assignee, map assets or groups to Jira projects, and map ticket priority to Continuous Dynamic ratings. The settings for each are broadly comparable, but some specifics will be called out where relevant.

1. To **Configure DAST/SAST Settings** select **Enable DAST integration** or **Enable SAST integration**.

NOTE	Until the radio button is set to **Yes**, no other configuration options will be available.

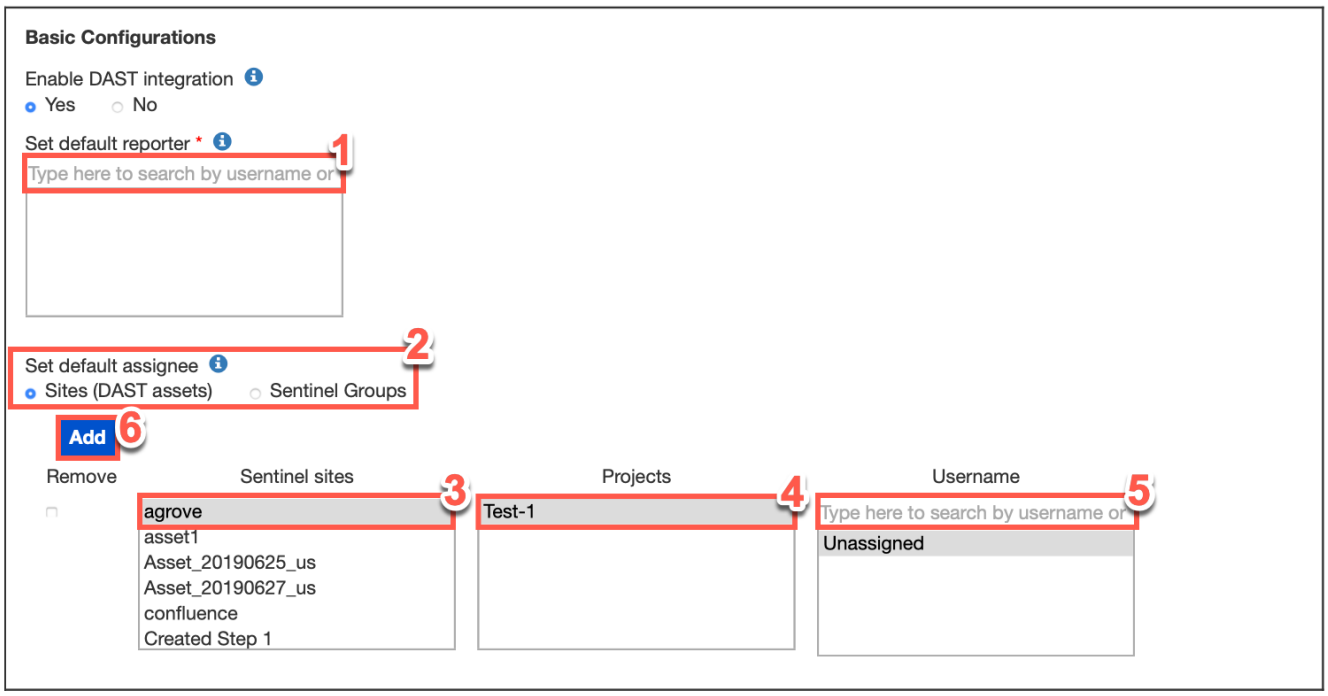


2. Select the **Yes** radio button. Once you have selected **Yes**, you will see the configuration settings. This will allow you to:
    - Set the default Reporter for Jira tickets generated by the Plugin
    - Set the default Assignee for Jira tickets generated by the Plugin based on asset-and-project combinations
    - Map Vulnerability Ratings to Jira Priorities
    - Configure Jira Tickets

3.2.1. Basic Configuration

NOTE	This shows the screen for selecting **Sites** using DAST. If configuring SAST, you have the option to select **Applications**.

1. Type the name or email of your default reporter in the search bar provided and then select your default reporter.



2. Set the default Jira assignee for a given asset (site or application) and associated Jira project. (This will map these assets to the Jira project(s) in question.) To set default assignees by group rather than asset, select the **Continuous Dynamic Groups** radio button. In this case, all assets in a group will be associated to the Jira project selected.
3. Select the asset from the list of **Continuous Dynamic sites**.
4. Select a project from the **Projects** list to assign.
5. Type the name or email of your default assignee in the search bar provided and then select them from the list.
6. To create additional default assignees and asset-to-project mappings, click **Add**.

NOTE	Only one user can be selected as the default reporter in Jira. Only one user can be set as the default assignee to any given asset-project mapping. If **Unassigned** is selected for the **Username** field, any tickets generated will show the default assignee for that project.

Reporter Permissions Required

A reporter must have the following privileges for the project:

- Assign Issue
- Close Issue
- Create Issue
- Edit Issue
- Modify Reporter

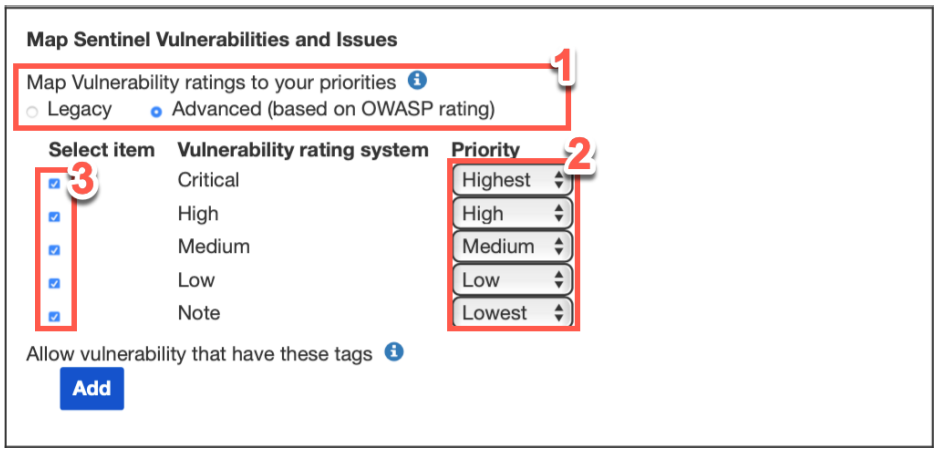
- Resolve Issue
- Transition Issue
- Comment Issue

If you attempt to assign a user as reporter who does not have these permissions for the appropriate project, you will receive an error message.

3.2.2. Mapping Vulnerability Ratings to Jira Priorities

Vulnerability ratings for Source (SAST) vulnerabilities will all automatically use the **Advanced Rating Methodology**, which is based on **OWASP** ratings. Vulnerability ratings for Dynamic (DAST) Vulnerabilities may use either the **Advanced Rating Methodology** or the **Legacy Methodology**.

1. Select the vulnerability rating to use for DAST configuration.

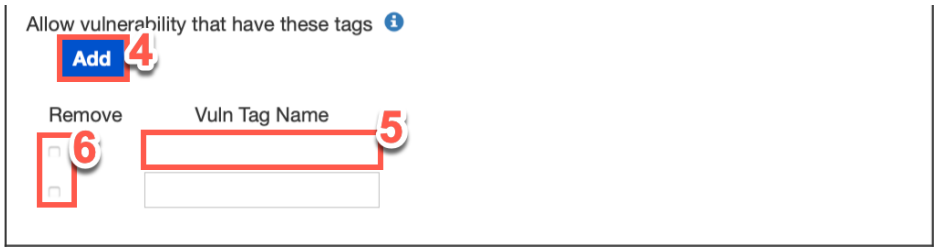


   For more information on choosing Legacy Ratings or Advanced Ratings, see Understanding the Rating Methodologies.

2. The default mapping will associate the most severe rating with the highest Jira priority. You can change this mapping using the drop-down lists.
3. Select the vulnerability ratings that should (checked) or should not (unchecked) be used to create Jira tickets.

   Jira tickets will now be created for vulnerabilities rated **Critical**, **High**, or **Medium**. **Critical** vulnerabilities will receive the **Highest** Jira priority, **High** risk vulnerabilities will receive a Jira priority of **High**, and **Medium** risk vulnerabilities will receive a Jira priority of **Medium**. Jira tickets will not be created for vulnerabilities with a rating of **Low** or **Note**. In addition it is also possible to limit vulnerabilities that will result in Jira tickets based on the Continuous Dynamic tags associated to the vulnerability.

4. Click **Add** under **Allow vulnerabilities that have these tags:**



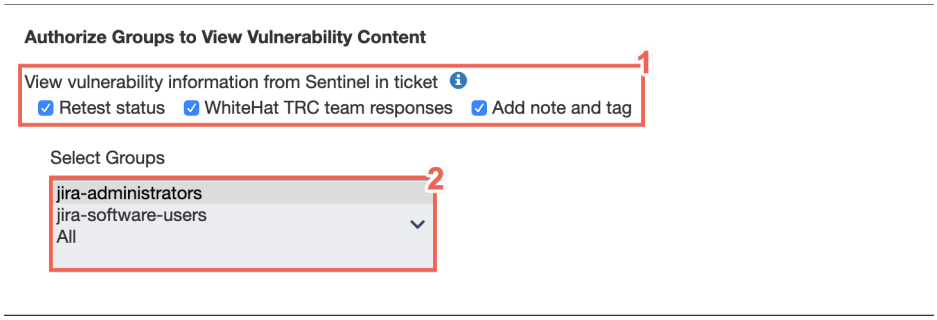
5. Type the tag name in the **Vuln Tag Name** text field.
6. To remove tags, click the checkbox next to the tag in question.

NOTE	If you select any tags to be used to create Jira tickets, only vulnerabilities that have at least one of the listed tags in the Continuous Dynamic Portal will be used to create Jira issues.

3.2.3. Set Vulnerability Viewing Authorizations for Dynamic (DAST) Vulns

You can authorize Jira groups to view content from the vulnerabilities discovered via dynamic (DAST) testing, including retest status, notes and tags, and Black Duck Threat Research Center team responses to questions.

1. Select the relevant radio buttons to configure the type of vulnerability information that is visible to specific groups.



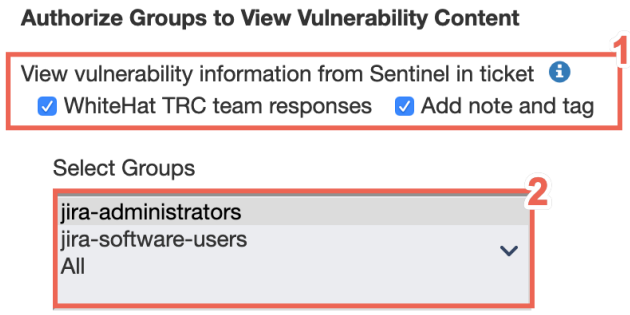
2. Select a group from the **Select Groups** table.

NOTE	This information will appear in the summary section of your tickets.

3.2.4. Set Vulnerability Viewing Authorizations for Static (SAST) Vulns

You can authorize Jira groups to view content from the Portal vulnerabilities discovered via static (SAST) testing, including notes and tags, and Black Duck Threat Research Center team responses to questions.

1. Select the relevant radio buttons to configure the type of vulnerability information that is visible to specific groups.

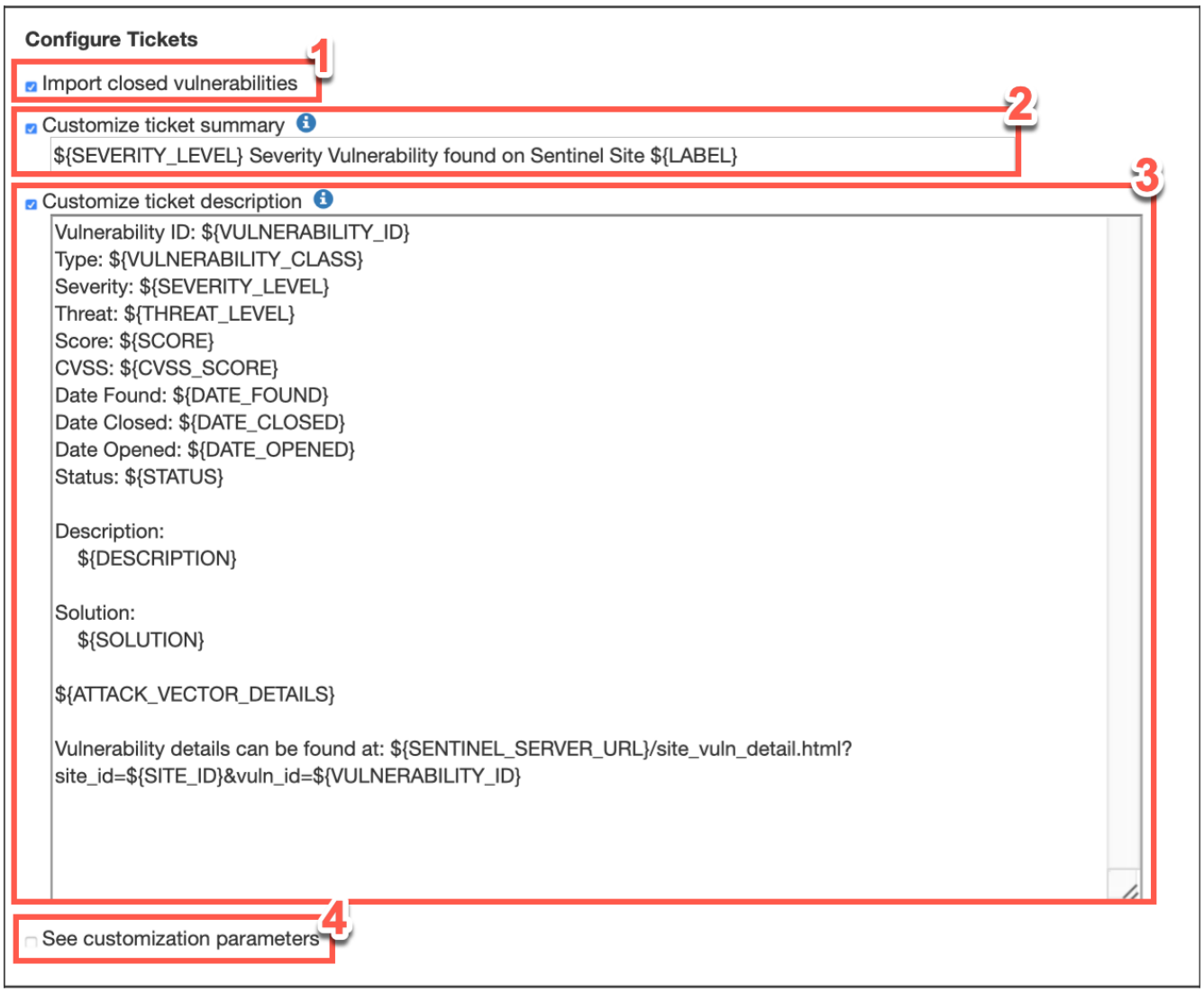


2. Select a group from the **Select Groups** table.

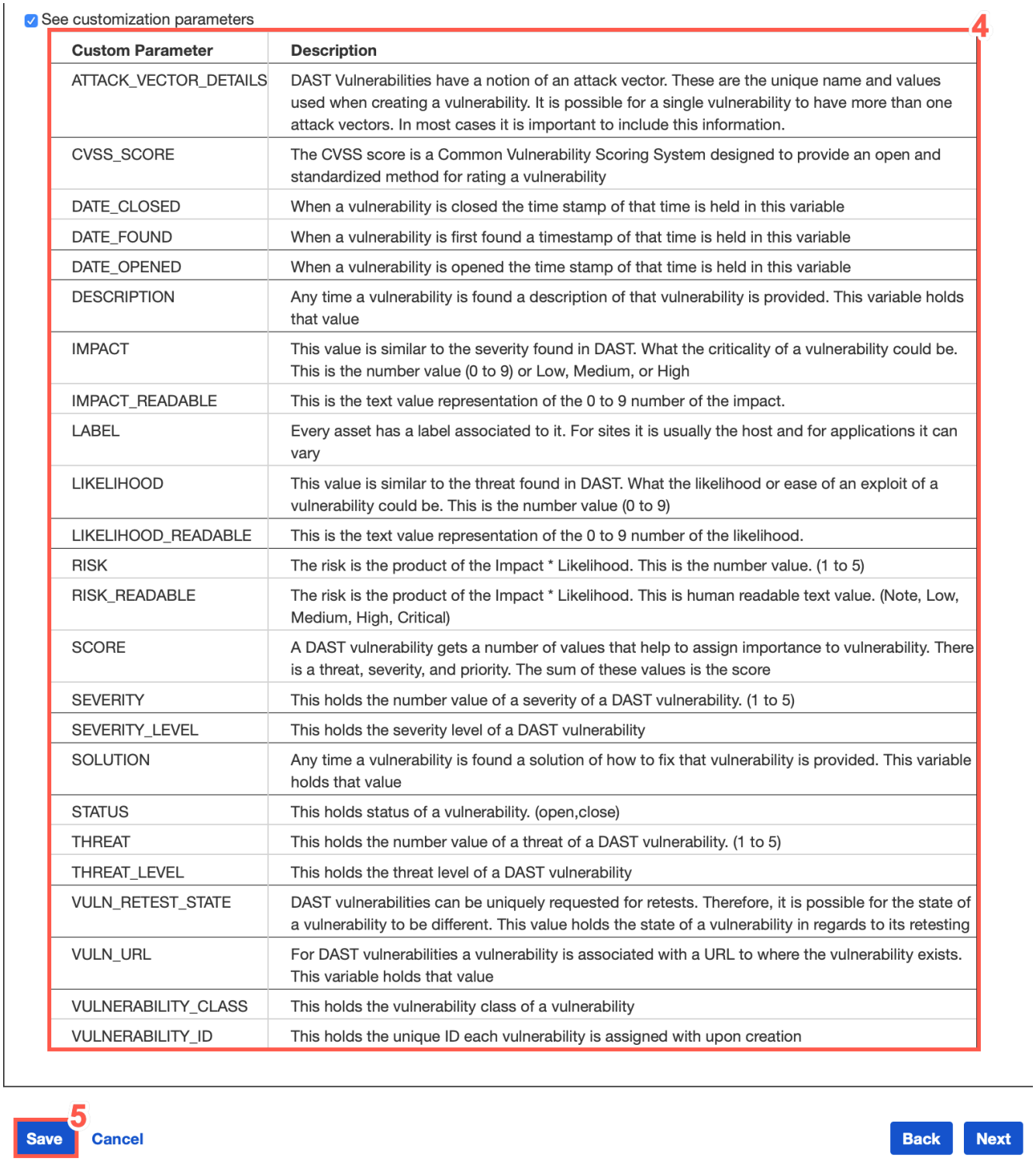
NOTE This information will appear in the summary section of your tickets.

3.2.5. Configure Jira Tickets

1. To import closed vulnerabilities select the **Import closed vulnerabilities** checkbox.



2. To customize the ticket summary select the **Customize ticket summary** checkbox.
3. To customize the ticket description select the **Customize ticket description** checkbox.
4. Optionally, to see customization parameters, check the checkbox next to **See Customization Parameters**



Custom Parameter	Description
ATTACK_VECTOR_DETAILS	DAST Vulnerabilities have a notion of an attack vector. These are the unique name and values used when creating a vulnerability. It is possible for a single vulnerability to have more than one attack vectors. In most cases it is important to include this information.
CVSS_SCORE	The CVSS score is a Common Vulnerability Scoring System designed to provide an open and standardized method for rating a vulnerability
DATE_CLOSED	When a vulnerability is closed the time stamp of that time is held in this variable
DATE_FOUND	When a vulnerability is first found a timestamp of that time is held in this variable
DATE_OPENED	When a vulnerability is opened the time stamp of that time is held in this variable
DESCRIPTION	Any time a vulnerability is found a description of that vulnerability is provided. This variable holds that value
IMPACT	This value is similar to the severity found in DAST. What the criticality of a vulnerability could be. This is the number value (0 to 9) or Low, Medium, or High
IMPACT_READABLE	This is the text value representation of the 0 to 9 number of the impact.
LABEL	Every asset has a label associated to it. For sites it is usually the host and for applications it can vary
LIKELIHOOD	This value is similar to the threat found in DAST. What the likelihood or ease of an exploit of a vulnerability could be. This is the number value (0 to 9)
LIKELIHOOD_READABLE	This is the text value representation of the 0 to 9 number of the likelihood.
RISK	The risk is the product of the Impact * Likelihood. This is the number value. (1 to 5)
RISK_READABLE	The risk is the product of the Impact * Likelihood. This is human readable text value. (Note, Low, Medium, High, Critical)
SCORE	A DAST vulnerability gets a number of values that help to assign importance to vulnerability. There is a threat, severity, and priority. The sum of these values is the score
SEVERITY	This holds the number value of a severity of a DAST vulnerability. (1 to 5)
SEVERITY_LEVEL	This holds the severity level of a DAST vulnerability
SOLUTION	Any time a vulnerability is found a solution of how to fix that vulnerability is provided. This variable holds that value
STATUS	This holds status of a vulnerability. (open,close)
THREAT	This holds the number value of a threat of a DAST vulnerability. (1 to 5)
THREAT_LEVEL	This holds the threat level of a DAST vulnerability
VULN_RETEST_STATE	DAST vulnerabilities can be uniquely requested for retests. Therefore, it is possible for the state of a vulnerability to be different. This value holds the state of a vulnerability in regards to its retesting
VULN_URL	For DAST vulnerabilities a vulnerability is associated with a URL to where the vulnerability exists. This variable holds that value
VULNERABILITY_CLASS	This holds the vulnerability class of a vulnerability
VULNERABILITY_ID	This holds the unique ID each vulnerability is assigned with upon creation

5. When you have completed configuration for **DAST** or **SAST** settings according to your preferences, click **Save**.

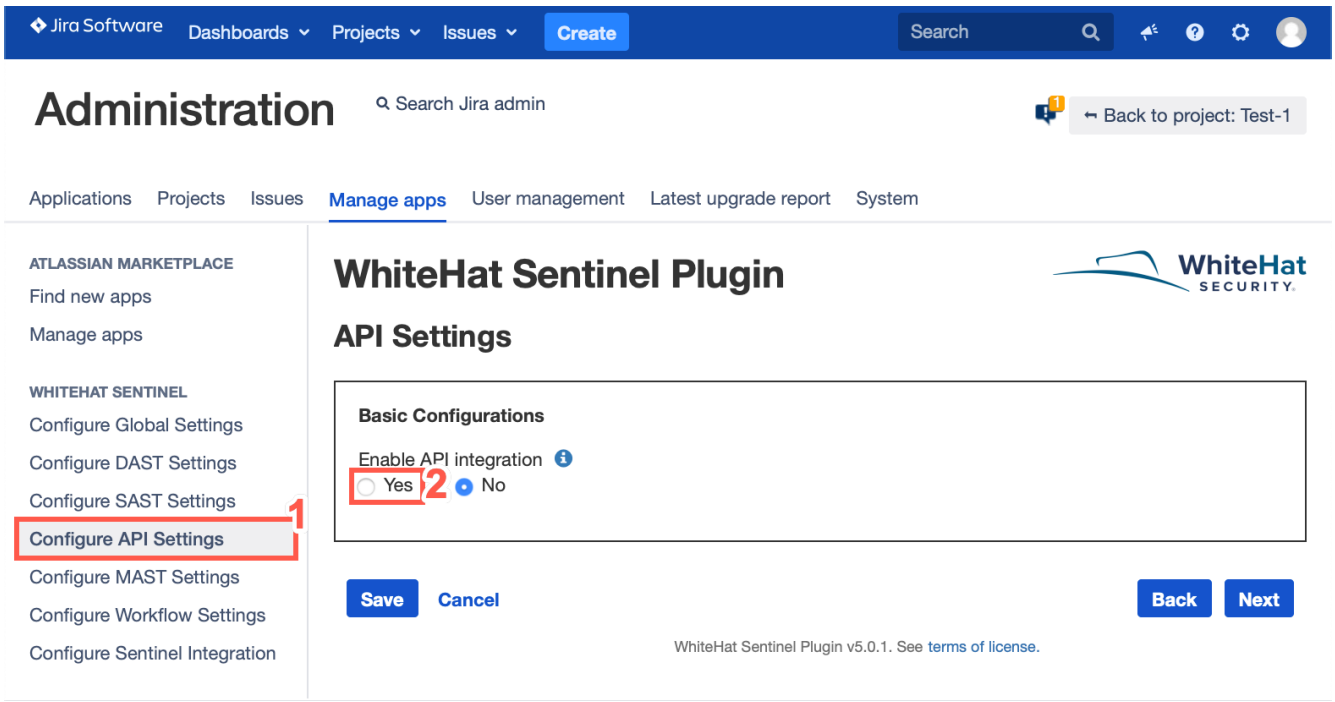
NOTE	If you've set a **Custom Asset ID** for this site or application (from the **Overview** tab in the Portal), it will appear as a field in the ticket **Details**.

3.3. Configure API Settings

Select **Configure API Settings** to set your default reporter and assignee, map assets or groups to Jira projects, and map ticket priority to Continuous Dynamic ratings.

1. To **Configure API Settings**, select **Enable API integration**.

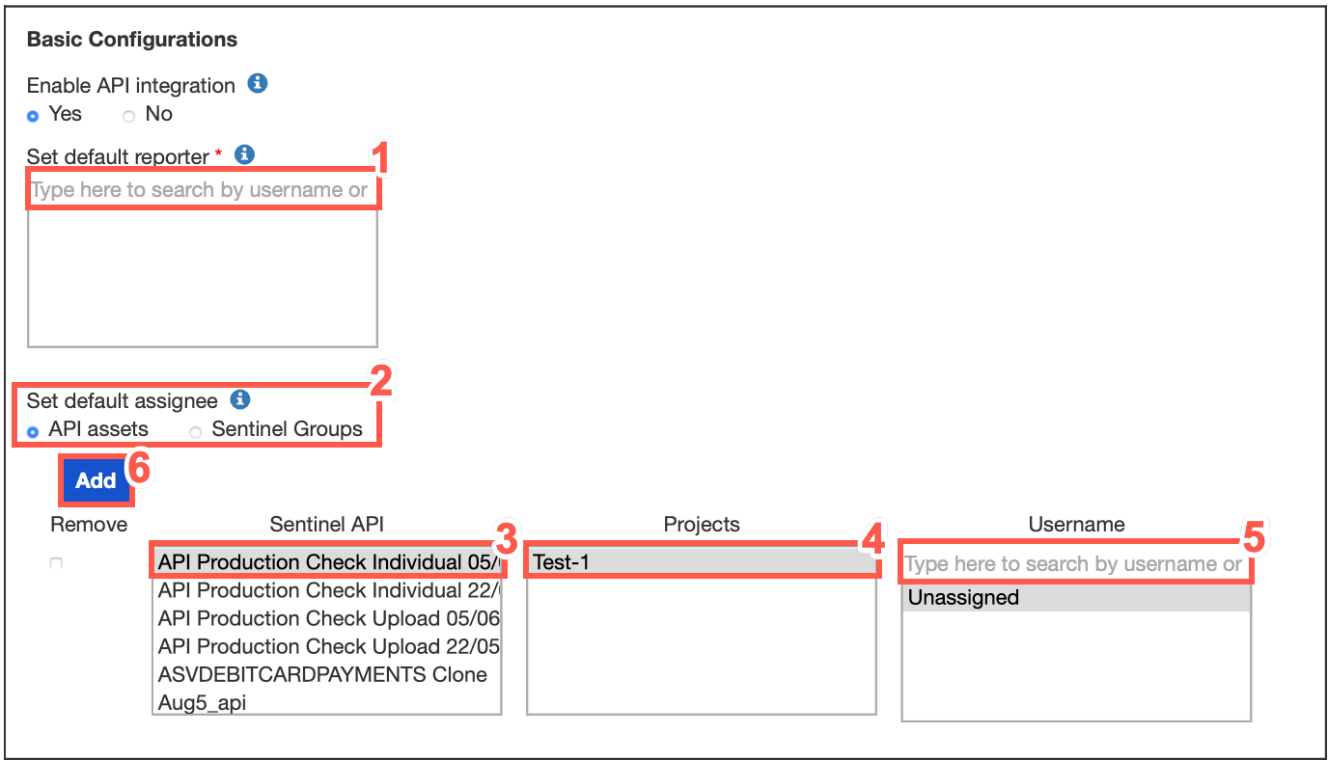
NOTE	Until the radio button is set to **Yes**, no other configuration options will be available.



2. Select the **Yes** radio button. Once you have selected **Yes**, you will see the configuration settings. This will allow you to:
   - Set the default Reporter for Jira tickets generated by the Plugin
   - Set the default Assignee for Jira tickets generated by the Plugin based on asset-and-project combinations
   - Map Vulnerability Ratings to Jira Priorities
   - Configure Jira Tickets

3.3.1. Basic Configurations

1. Enter the name or email of your default reporter in the search bar, then select your default reporter.



2. Set the default Jira assignee for a given API and associated Jira project. (This will map these assets to the Jira project(s) in question.) To set default assignees by group rather than asset, select the **Continuous Dynamic Groups** radio button. In this case, all assets in a group will be associated to the Jira project selected.
3. Select the asset from the **Continuous Dynamic API** list.
4. Select a project from the **Projects** list to assign.
5. Type the name or email of your default assignee in the search bar provided and then select them from the list.
6. To create additional default assignees and asset-to-project mappings, click **Add**.

NOTE	Only one user can be selected as the default reporter in Jira. Only one user can be set as the default assignee to any given asset-project mapping. If **Unassigned** is selected for the **Username** field, any tickets generated will show the default assignee for that project.

Reporter Permissions Required

A reporter must have the following privileges for the project:

- Assign Issue
- Close Issue
- Create Issue
- Edit Issue

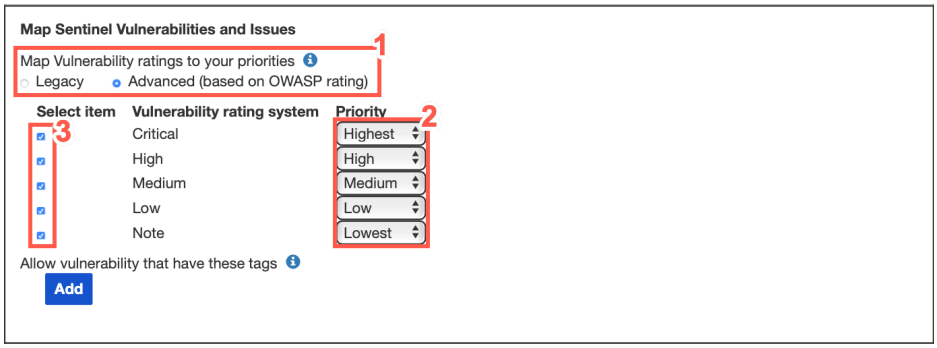
- Modify Reporter
- Resolve Issue
- Transition Issue
- Comment Issue

If you attempt to assign a user as reporter who does not have these permissions for the appropriate project, you will receive an error message.

3.3.2. Mapping Vulnerability Ratings to Jira Priorities

Vulnerability ratings for API vulnerabilities will all automatically use the **Advanced Rating Methodology**, which is based on **OWASP** ratings. Vulnerability ratings for API vulnerabilities may use either the **Advanced Rating Methodology** or the **Legacy Methodology**.

1. Select the vulnerability rating to use for API configuration.

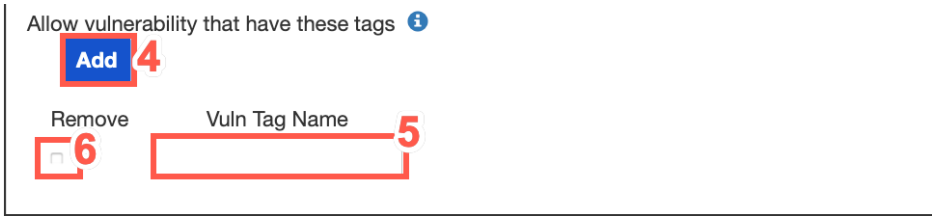


   For more information on choosing Legacy Ratings or Advanced Ratings, see Understanding the Rating Methodologies.

2. The default mapping will associate the most severe rating with the highest Jira priority. You can change this mapping using the drop-down lists.
3. Select the vulnerability ratings that should (checked) or should not (unchecked) be used to create Jira tickets.

   Jira tickets will now be created for vulnerabilities rated **Critical**, **High**, or **Medium**. **Critical** vulnerabilities will receive the **Highest** Jira priority, **High** risk vulnerabilities will receive a Jira priority of **High**, and **Medium** risk vulnerabilities will receive a Jira priority of **Medium**. Jira tickets will not be created for vulnerabilities with a rating of **Low** or **Note**. In addition it is also possible to limit vulnerabilities that will result in Jira tickets based on the Continuous Dynamic tags associated to the vulnerability.

4. Click **Add** under **Allow vulnerabilities that have these tags:**



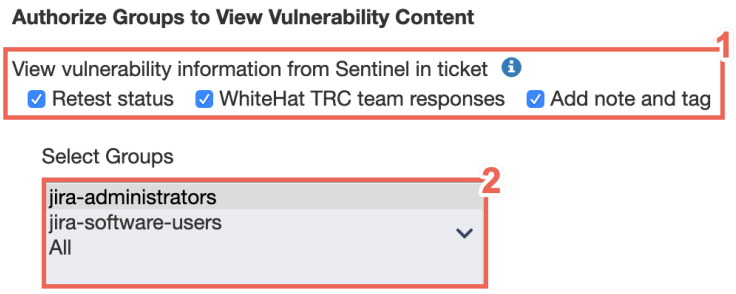
5. Type the tag name in the **Vuln Tag Name** text field.
6. To remove tags, click the checkbox next to the tag in question.

NOTE	If you select any tags to be used to create Jira tickets, only vulnerabilities that have at least one of the listed tags in the Continuous Dynamic Portal will be used to create Jira issues.

3.3.3. Set Vulnerability Viewing Authorizations for API Vulnerabilities

You can authorize Jira groups to view content from the vulnerabilities discovered via API testing, including retest status, notes and tags, and Black Duck Threat Research Center team responses to questions.

1. Select the relevant radio buttons to configure the type of vulnerability information that is visible to specific groups.

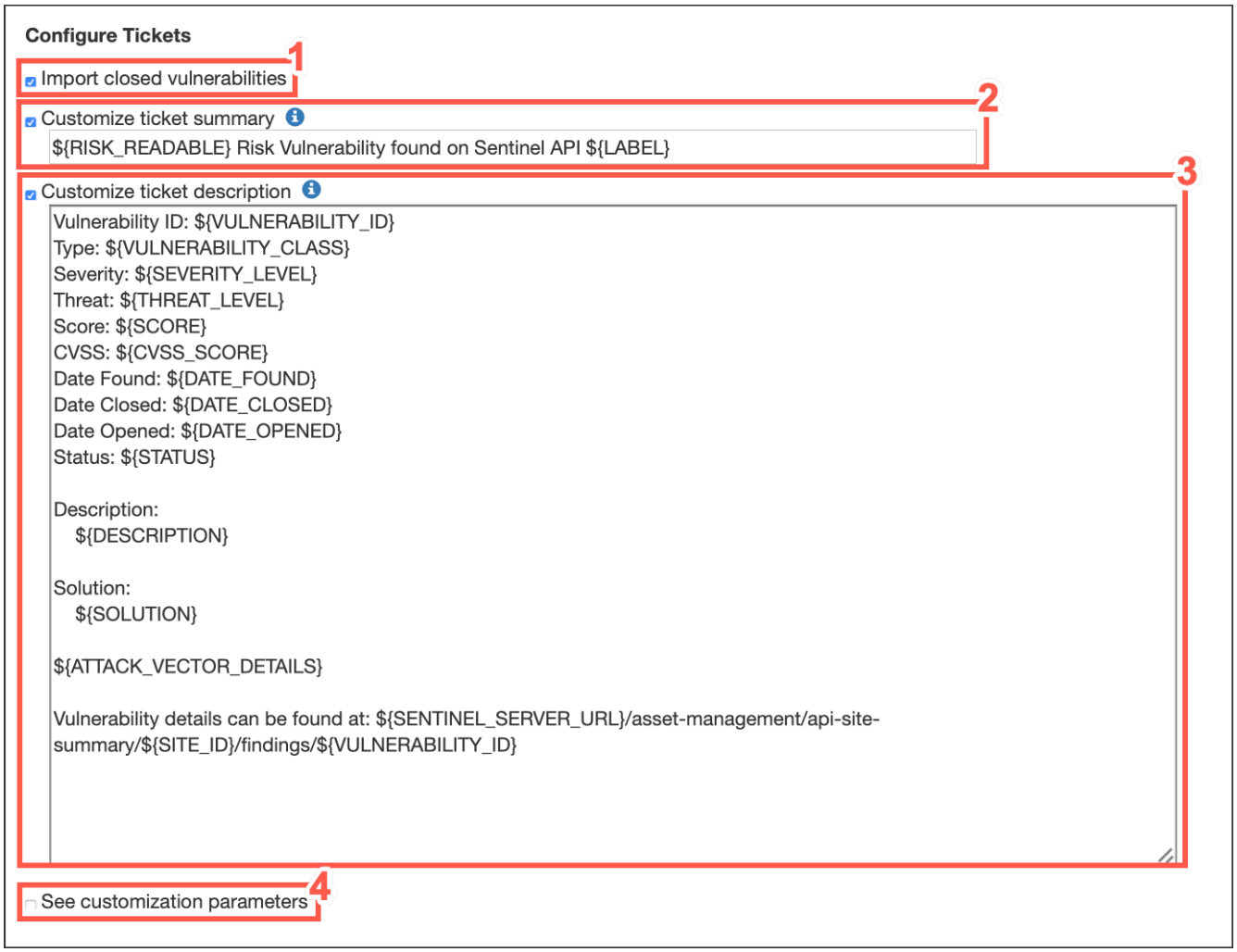


2. Select a group from the **Select Groups** table.

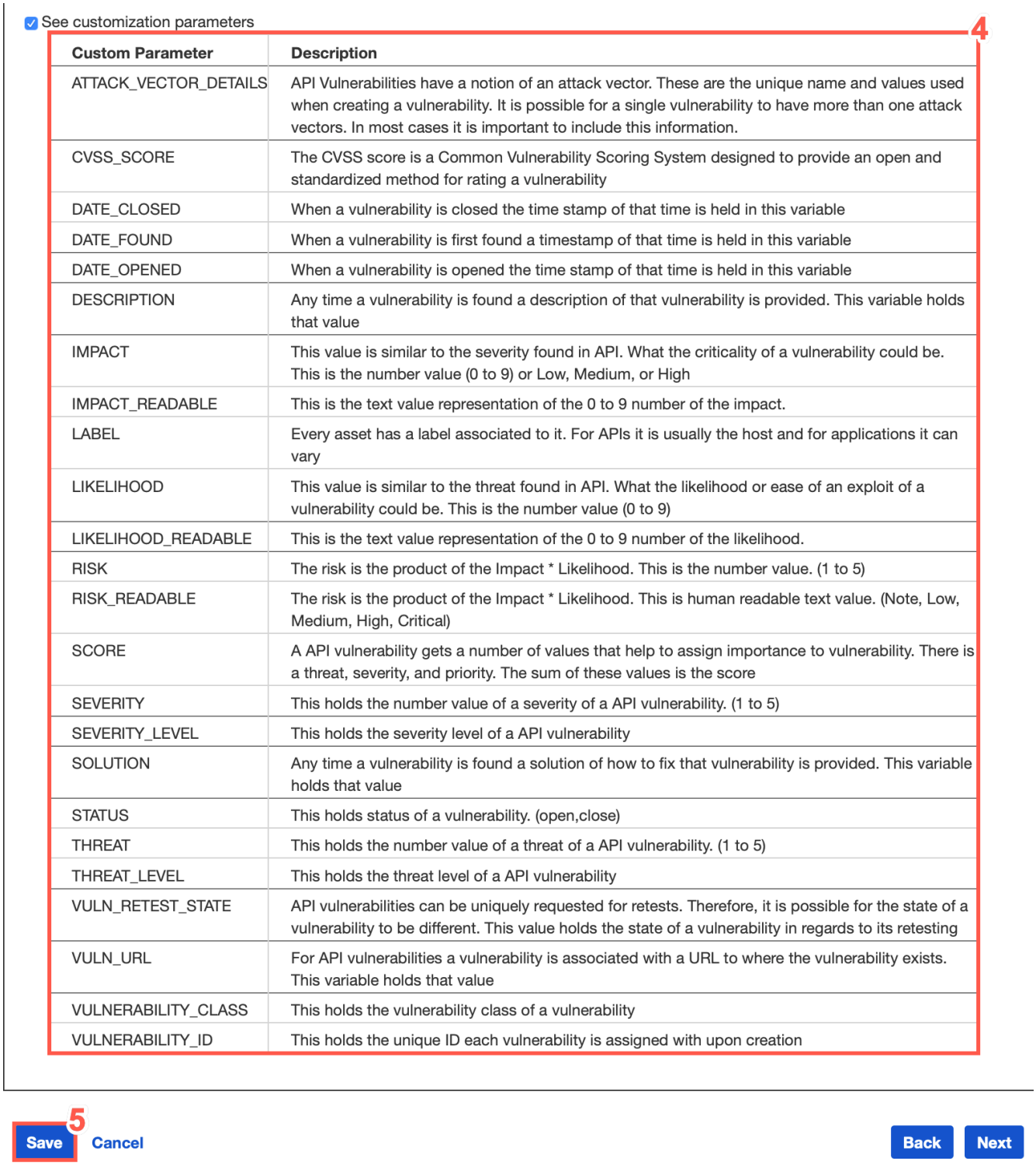
NOTE	This information will appear in the summary section of your tickets.

3.3.4. Configure Jira Tickets

1. To import closed vulnerabilities, select the **Import closed vulnerabilities** checkbox.



2. To customize the ticket summary select the **Customize ticket summary** checkbox.
3. To customize the ticket description select the **Customize ticket description** checkbox.
4. Optionally, to see customization parameters, check the checkbox next to **See Customization Parameters**



5. When you have completed configuration for **API** settings according to your preferences, click **Save**.

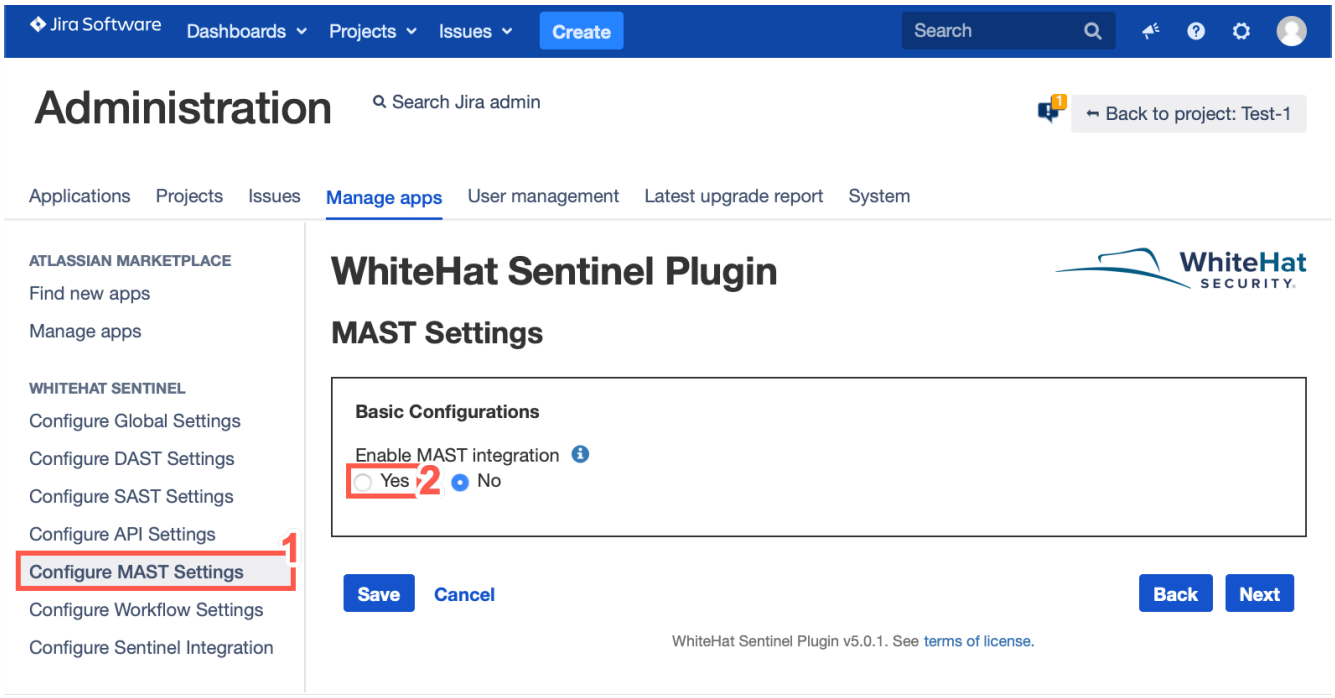
NOTE If you've set a **Custom Asset ID** for this API (from the **Overview** tab in the Portal), it will appear as a field in the ticket **Details**.

3.4. Configure MAST Settings

Select **Configure MAST Settings** to set your default reporter and assignee, map assets or groups to Jira projects, and map ticket priority to Continuous Dynamic ratings.

1. To **Configure MAST Settings**, select **Enable MAST integration**.

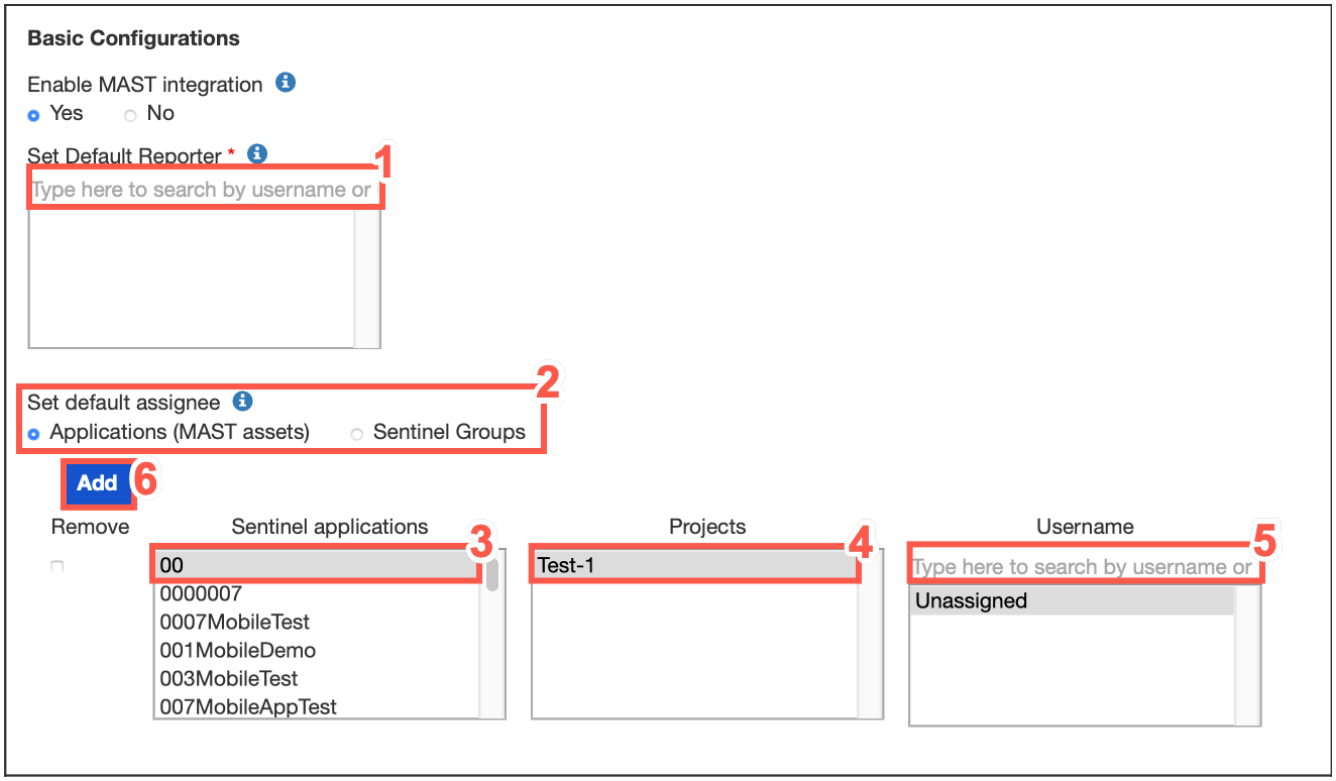
NOTE	Until the radio button is set to **Yes**, no other configuration options will be available.



2. Select the **Yes** radio button. Once you have selected **Yes**, you will see the configuration settings. This will allow you to:
   - Set the default Reporter for Jira tickets generated by the plugin
   - Set the default Assignee for Jira tickets generated by the plugin based on asset-and-project combinations
   - Map Vulnerability Ratings to Jira Priorities
   - Configure Jira Tickets

3.4.1. Basic Configurations

1. Enter the name or email of your default reporter in the search bar, then select your default reporter.



2. Set the default Jira assignee for a given application and associated Jira project. (This will map these assets to the Jira project(s) in question.) To set default assignees by group rather than asset, select the **Continuous Dynamic Groups** radio button. In this case, all assets in a group will be associated to the Jira project selected.
3. Select the asset from the list of **Sentinel applications**.
4. Select a project from the **Projects** list to assign.
5. Type the name or email of your default assignee in the search bar provided and then select them from the list.
6. To create additional default assignees and asset-to-project mappings, click **Add**.

NOTE	Only one user can be selected as the default reporter in Jira. Only one user can be set as the default assignee to any given asset-project mapping. If **Unassigned** is selected for the **Username** field, any tickets generated will show the default assignee for that project.

Reporter Permissions Required

A reporter must have the following privileges for the project:

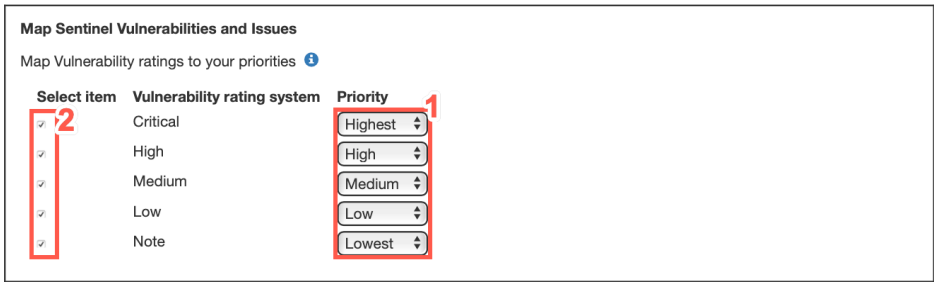
- Assign Issue
- Close Issue
- Create Issue
- Edit Issue

- Modify Reporter
- Resolve Issue
- Transition Issue
- Comment Issue

If you attempt to assign a user as reporter who does not have these permissions for the appropriate project, you will receive an error message.

3.4.2. Mapping Vulnerability Ratings to Jira Priorities

1. The default mapping will associate the most severe rating with the highest Jira priority. You can change this mapping using the drop-down lists.



   For more information on choosing Legacy Ratings or Advanced Ratings, see Understanding the Rating Methodologies.

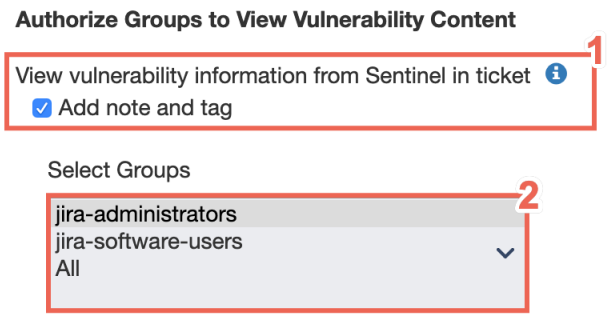
2. Select the vulnerability ratings that should (checked) or should not (unchecked) be used to create Jira tickets.

   Jira tickets will now be created for vulnerabilities rated **Critical**, **High**, or **Medium**. **Critical** vulnerabilities will receive the **Highest** Jira priority, **High** risk vulnerabilities will receive a Jira priority of **High**, and **Medium** risk vulnerabilities will receive a Jira priority of **Medium**. Jira tickets will not be created for vulnerabilities with a rating of **Low** or **Note**. In addition it is also possible to limit vulnerabilities that will result in Jira tickets based on the Continuous Dynamic tags associated to the vulnerability.

3.4.3. Set Vulnerability Viewing Authorizations for MAST Vulnerabilities

You can authorize Jira groups to view content from the vulnerabilities discovered via MAST testing, including vulnerability notes and tags.

1. Select the relevant radio buttons to configure the type of vulnerability information that is visible to specific groups.

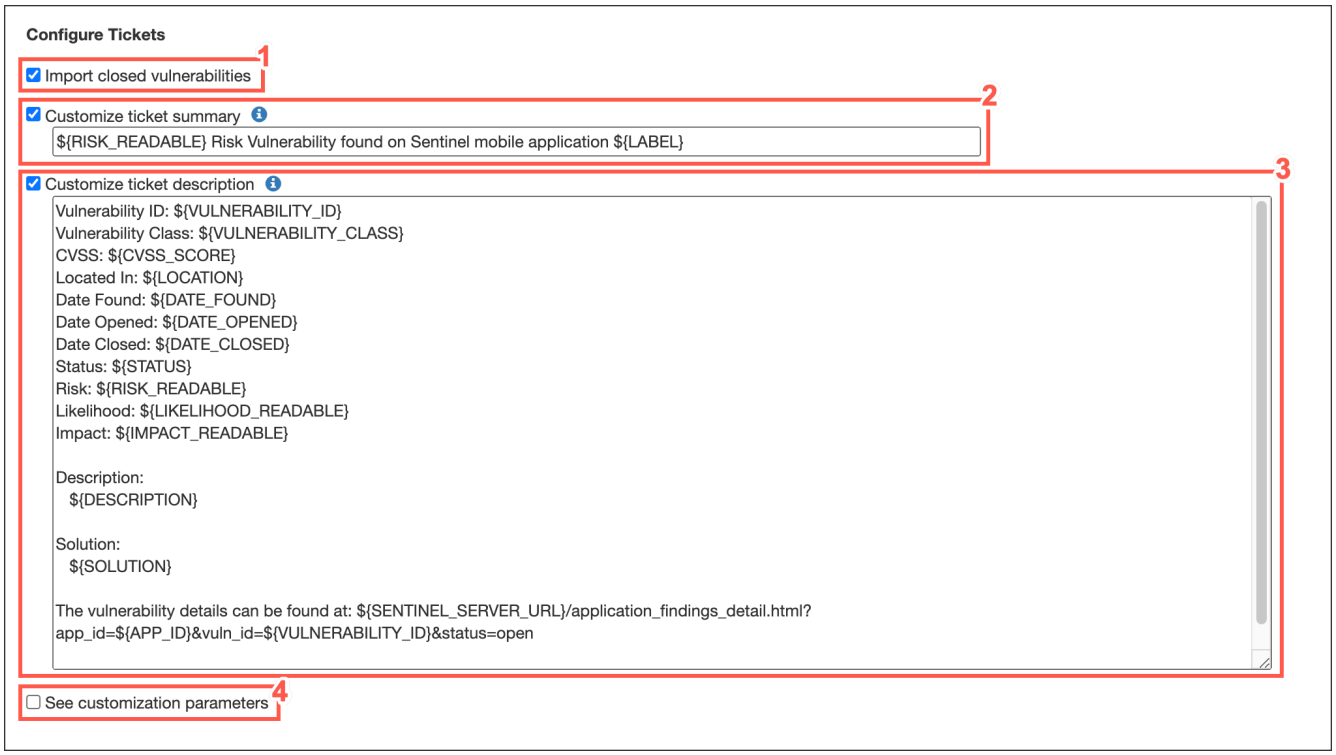


2. Select a group from the **Select Groups** table.

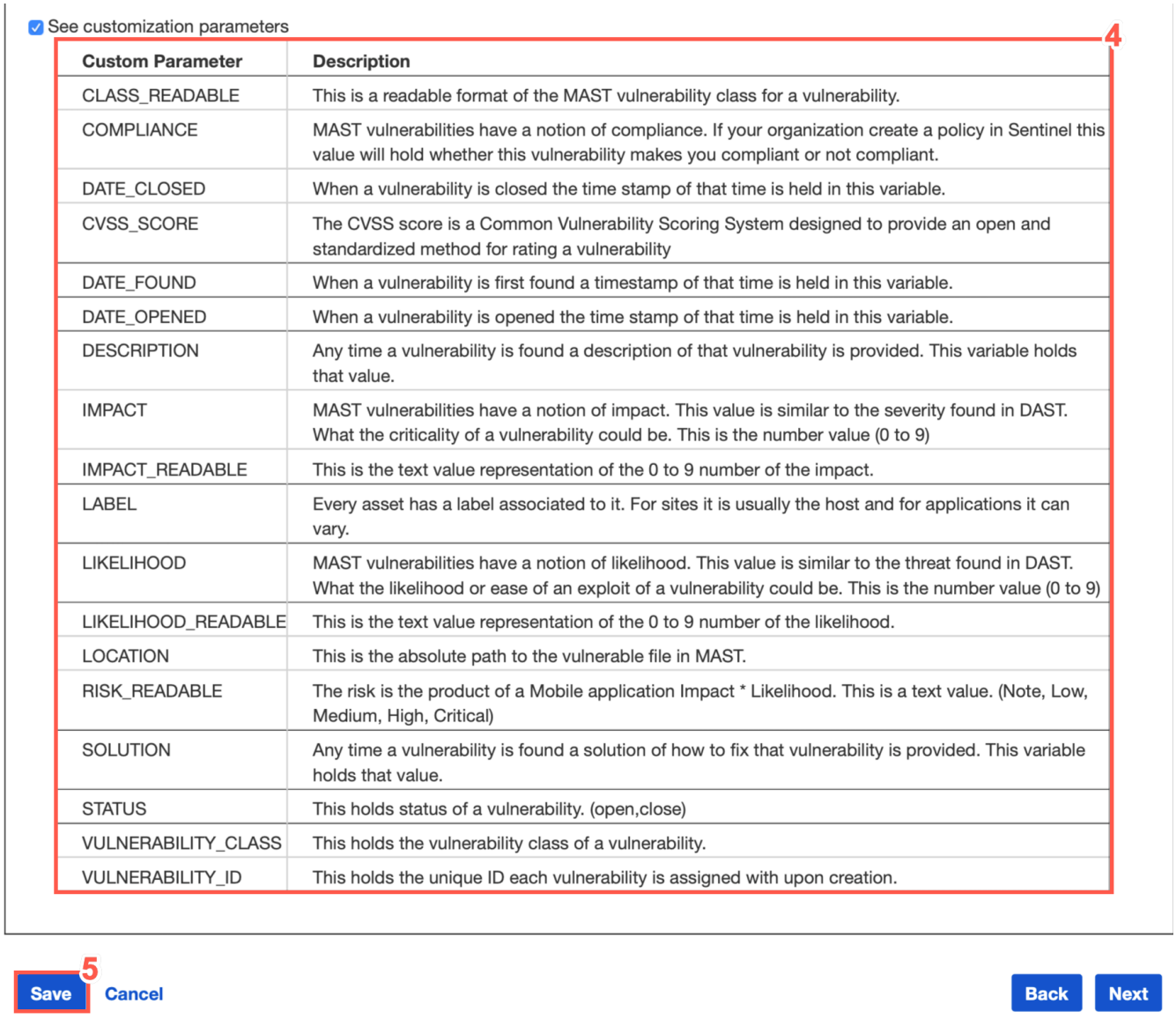
NOTE	This information will appear in the summary section of your tickets.

3.4.4. Configure Jira Tickets

1. To import closed vulnerabilities select the **Import closed vulnerabilities** checkbox.



2. To customize the ticket summary select the **Customize ticket summary** checkbox.
3. To customize the ticket description select the **Customize ticket description** checkbox.
4. Optionally, to see customization parameters, check the checkbox next to **See Customization Parameters**



5. When you have completed configuration for **MAST** settings according to your preferences, click **Save**.

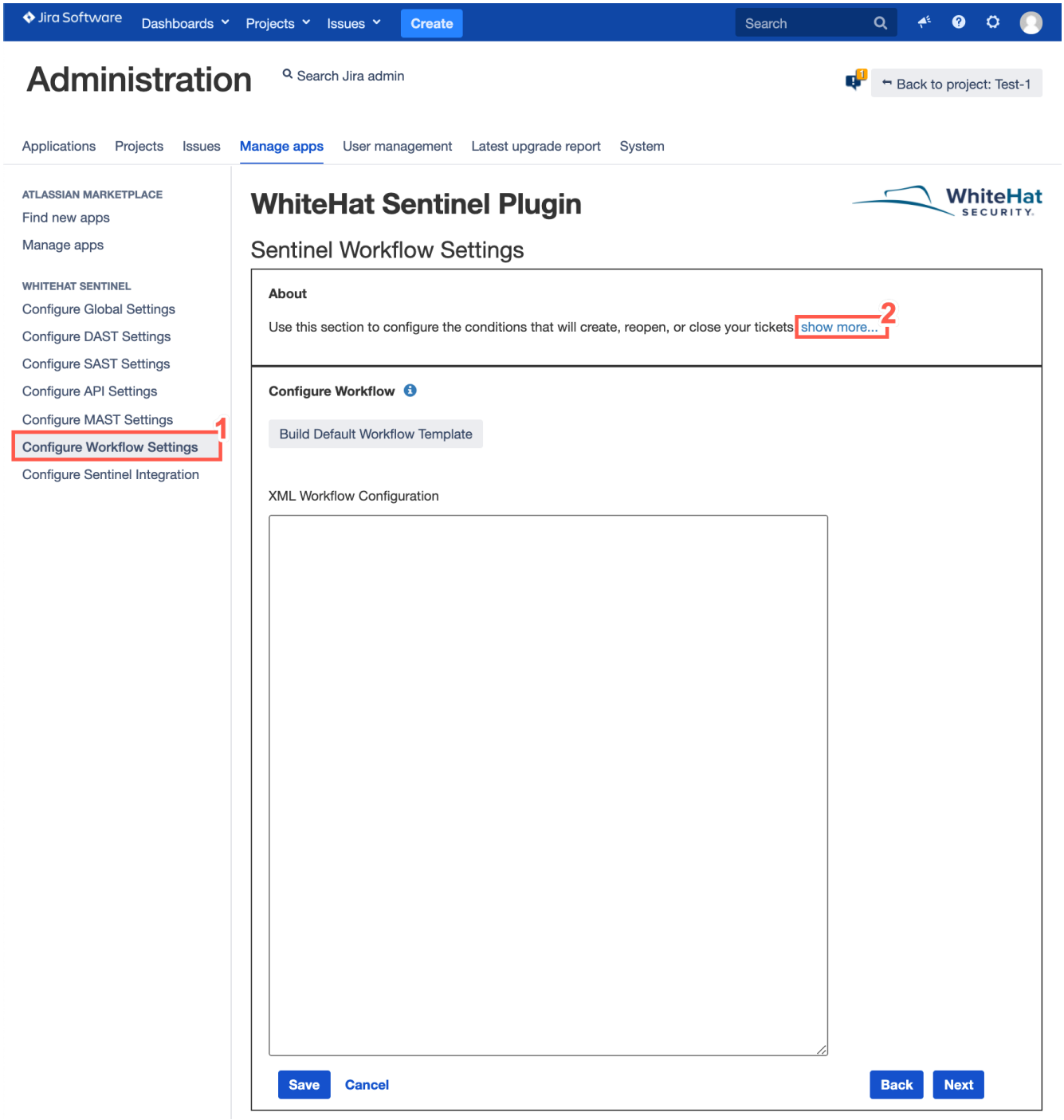
NOTE	If you've set a **Custom Asset ID** for this mobile application (from the Overview tab in the Continuous Dynamic Portal), it will appear as a field in the ticket **Details**.

3.5. Continuous Dynamic Workflow Settings

This section will define how issues will be created, reopened and closed. The workflow scheme in Jira is highly configurable, and it can be difficult to account for all possible statuses and transitions created by an organization. This section allows you to define what each status should do based on the action of creating, closing, or reopening an issue in Jira.

You must use the Jira workflow that is associated to your projects (as defined in **DAST**, **SAST**, **API** and **MAST** configurations) and define the pertinent information. You must be using a single workflow, and that workflow must be associated to all your projects.

1. Select **Configure Workflow Settings**.



2. Optionally, select **show more...** to view the following information regarding creating a custom workflow and configuration parameters.

3.5.1. About

Use this section to configure the conditions that will create, reopen, or close your tickets. Using your custom workflow, you can define the actions that a status can perform whenever a ticket is created, closed, or reopened.

1. Select the workflow associated with your projects (defined in DAST and/or SAST configurations).
2. Create an XML file and add required configuration parameters. See the table below to learn about these parameters.

NOTE	The workflow scheme in is an advanced configuration and therefore you must be aware of all the possible statuses and transitions between these statuses before you configure your custom workflow.

See configuration parameters:

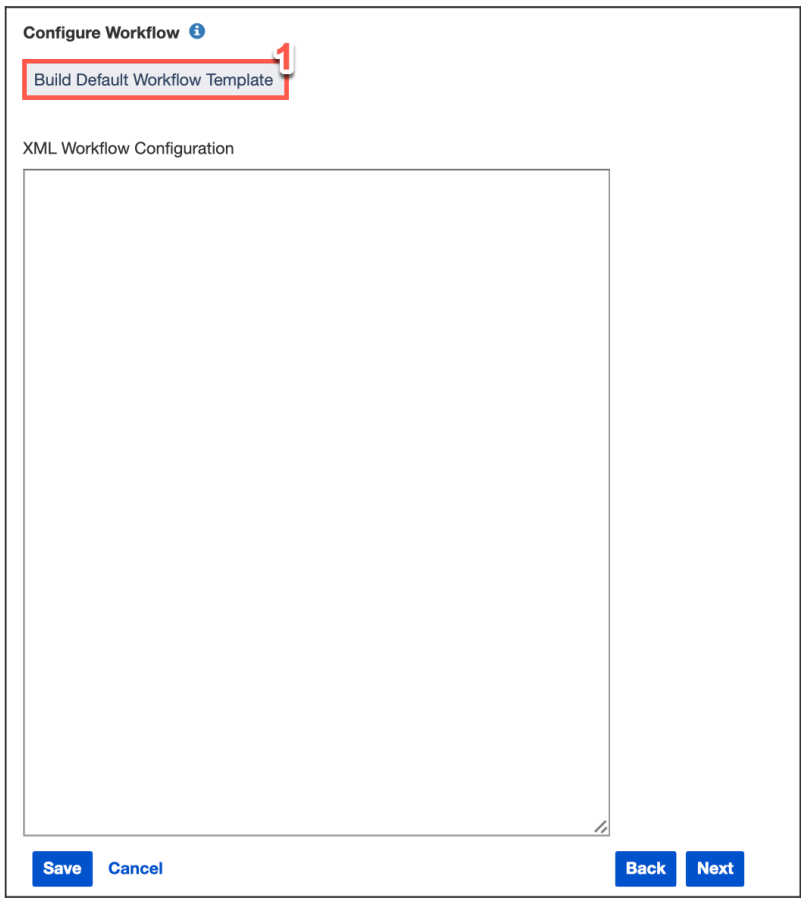
Parameter	Description
<**workflow**>	Define the workflow so that Continuous Dynamic add-on can create tickets.
<**transition_action id**="">	In order to transition into another status you must define the transition ID. The workflow provides you a list of all transitions for a given status when you use the workflow editor in text view.
<**transition type**="">	Define the type of transition for ticket. For example, you can define whether you want to close, reopen, or create the issue on a given transition.
<**status_map**>	Use this tag to define the state of each status within your workflow. The state can be either Opened or Closed. This allows the Continuous Dynamic add-on to make appropriate transition based on the ticket's status.
<**status name**="" **state**="">	Define the name of the status and the state your ticket to be considered Opened or Closed.
<**projects**>	Define each project that you want to associate within a workflow. This tag must be defined inside the tag.
<**project name**="" **key**="">	Define the name and key for the project.
<**issue_type id**="">	Define the ID to use whenever new tickets are created by the Continuous Dynamic add-on. To find this ID, go to Administration → Issues → Issue Types. Hover over the Related Schemes column for that issue. This should show you the scheme ID for that issue. This is the ID that you will use for the issue_type tag.
<**field name**="" **value**="">	In some cases certain fields are required to be filled in order to transition a ticket. Any required fields can be hard-coded values that will be filled in on making the transition defined by the parent tags.

3.5.2. Configure Workflow

To configure your workflow perform the following steps:

1. Click **Build Default Workflow Template** button to generate a generic template workflow XML file. This will generate a default XML workflow template that you can customize to match your chosen

workflow. The XML will be generated under the XML Workflow Configuration section, and will be a starting point for the workflow configuration.



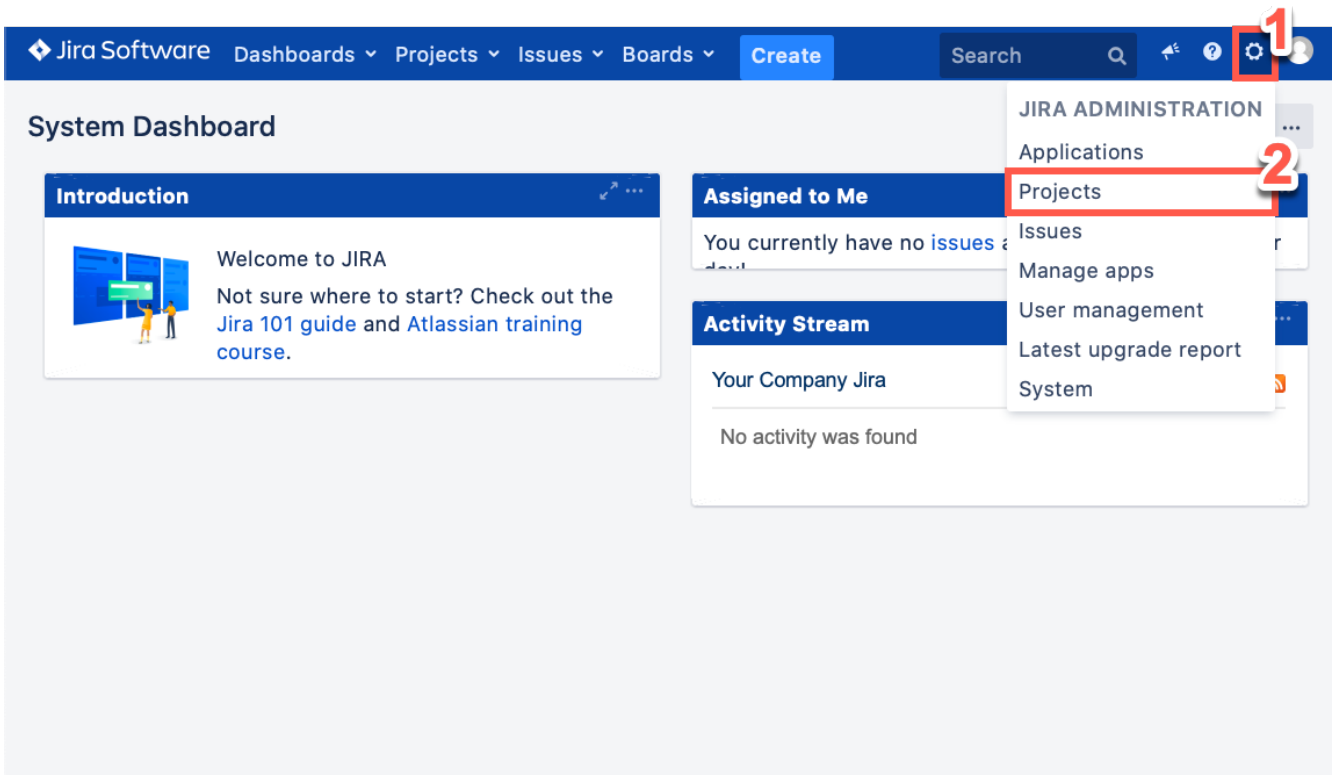
2. In the **XML Workflow Configuration**, provide a **Workflow Name**, **Project Name**, and **Key values**. This will be the name of the Jira workflow you are using and the name and key of the Jira project where the issues will be created. An example of this is shown below:

```
<workflows>
  <workflow name="DEMO:Simple Issue Tracking Workflow">
    <projects>
      <project name= "DEMO" key= "DEMO"></project>
```

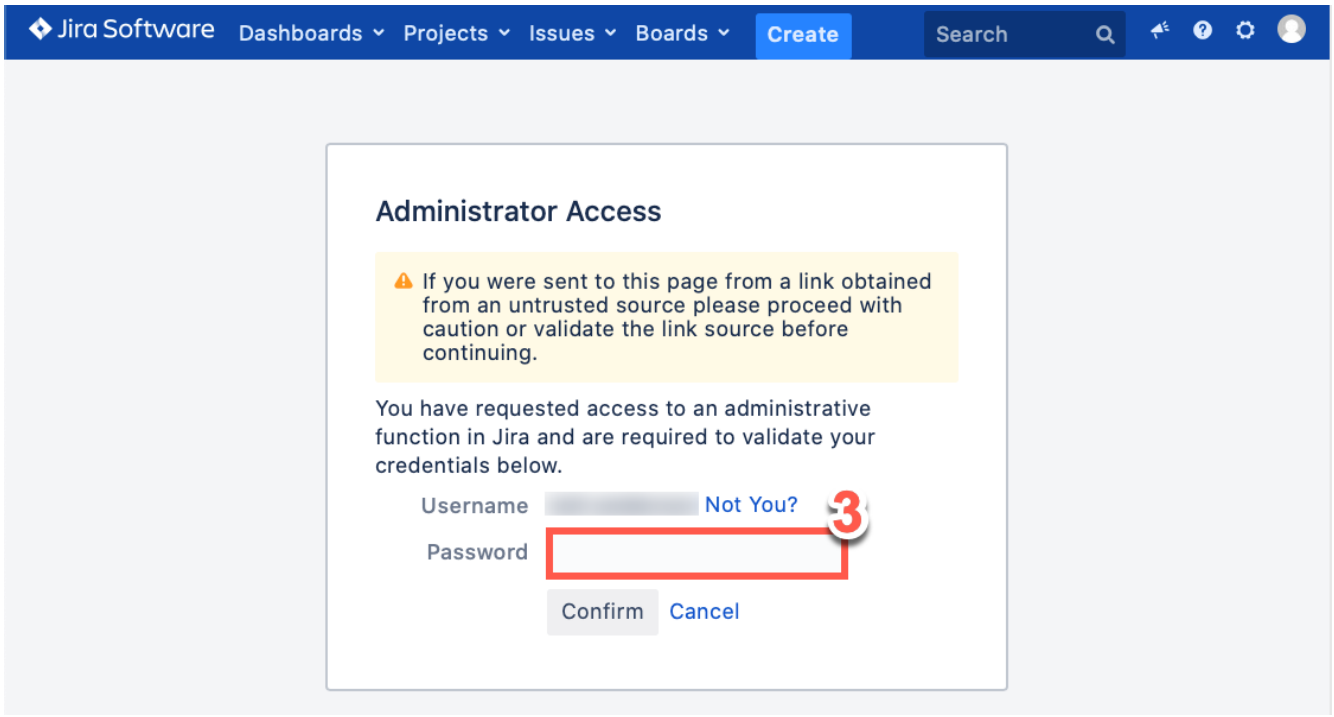
```
    </projects>
```

Locating your Project Name and Key in Jira

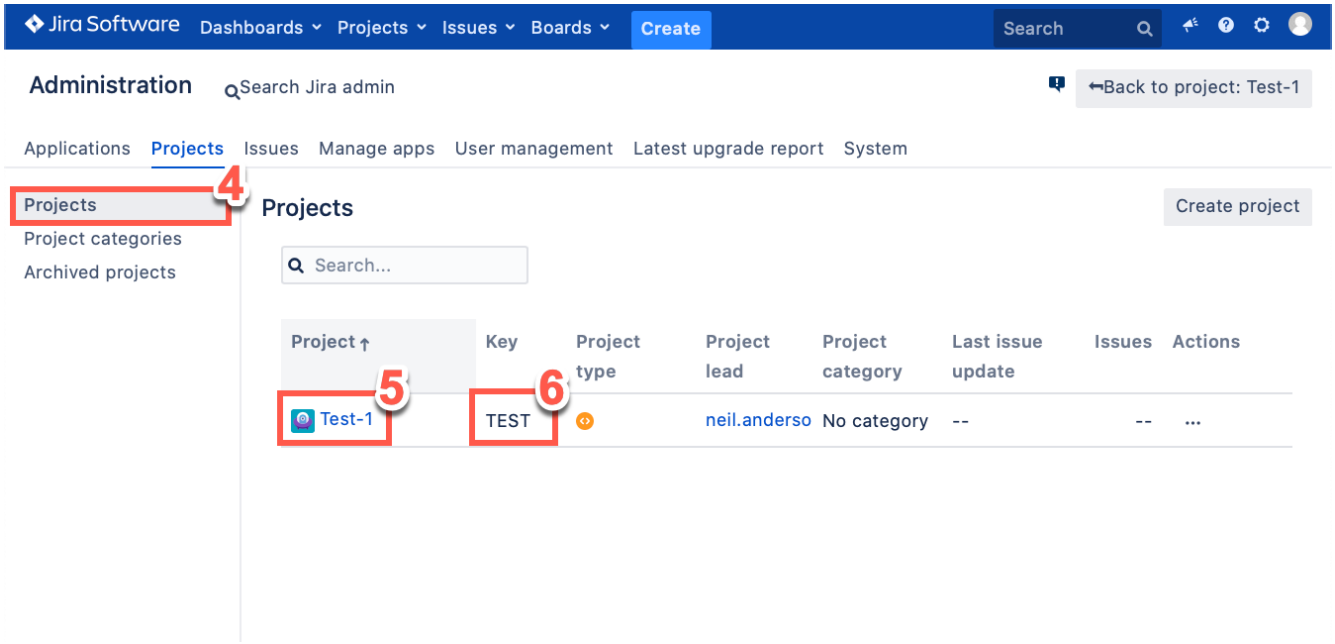
1. From the Jira **System Dashboard** click the gear icon.



2. Click **Projects**.
3. Log into an Admin account to manage the projects on Jira.



4. Click **Projects**.



5. The **Project Name** is shown here.
6. The **Key** is shown here.

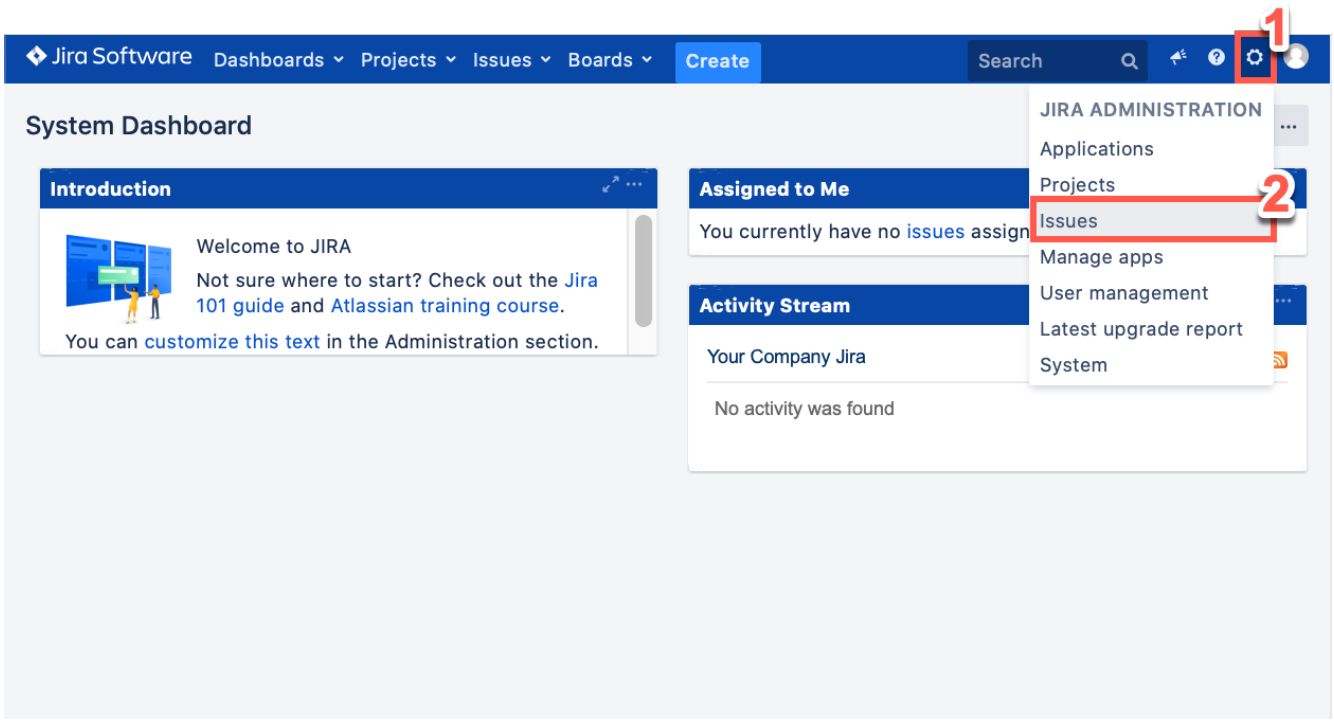
3.5.3. Issue Type ID

Define the issue type IDs to be used for Jira tickets, based on Continuous Dynamic Portal vulnerability data. The Issue Type is available in Jira.

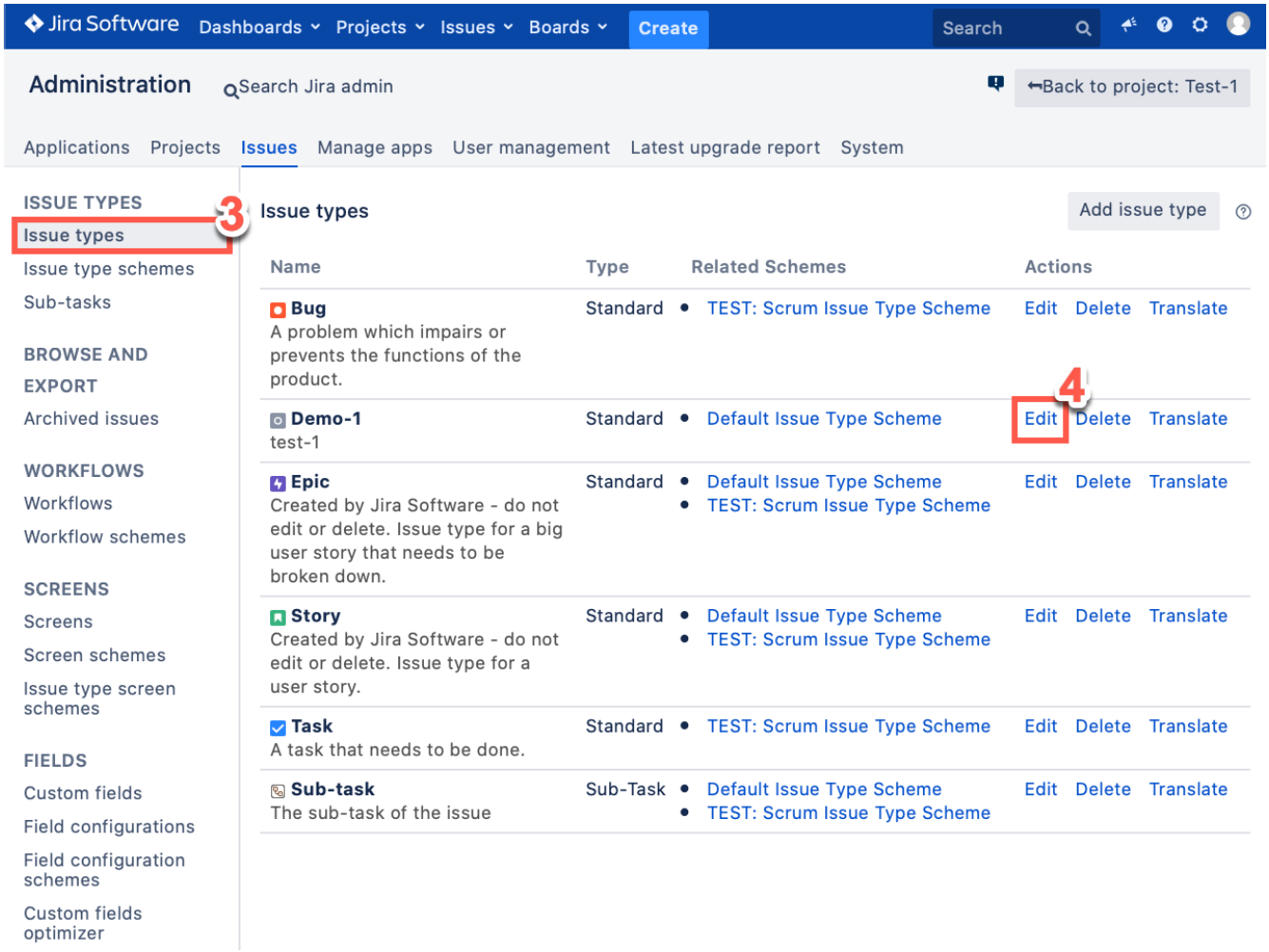
```
<issue_type id="5"></issue_type>
```

The issue type ID in Jira can be located by performing the following steps:

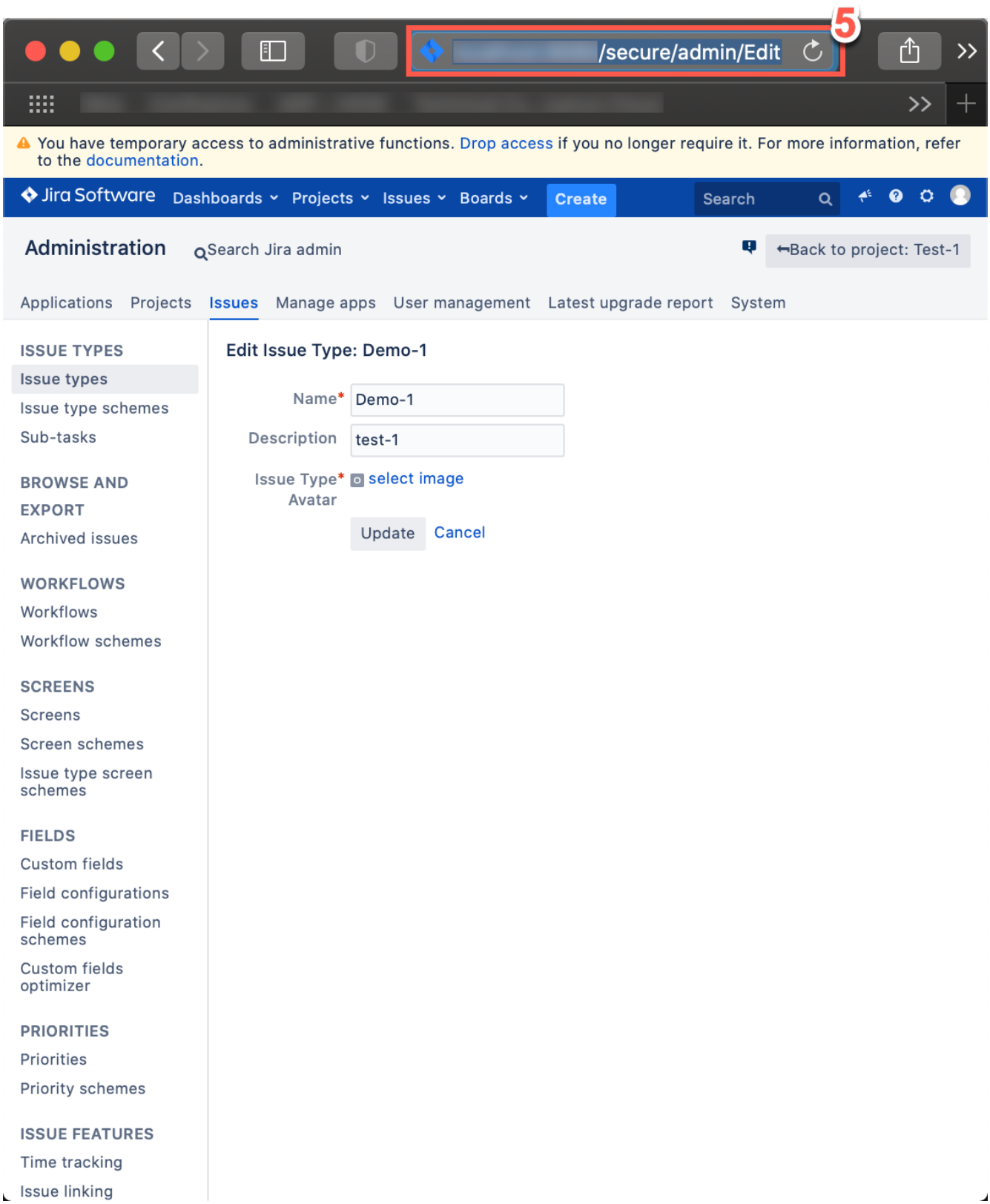
1. From the Jira **System Dashboard** click the gear icon.



2. Click **Issues**.
3. Click **Issue types**.



4. Click **Edit** on any of the **Issue types**.
5. To view the issue type ID click the URL.



6. The issue type ID is shown at the end of the URL.

3.5.4. Status Names & Transition ID's

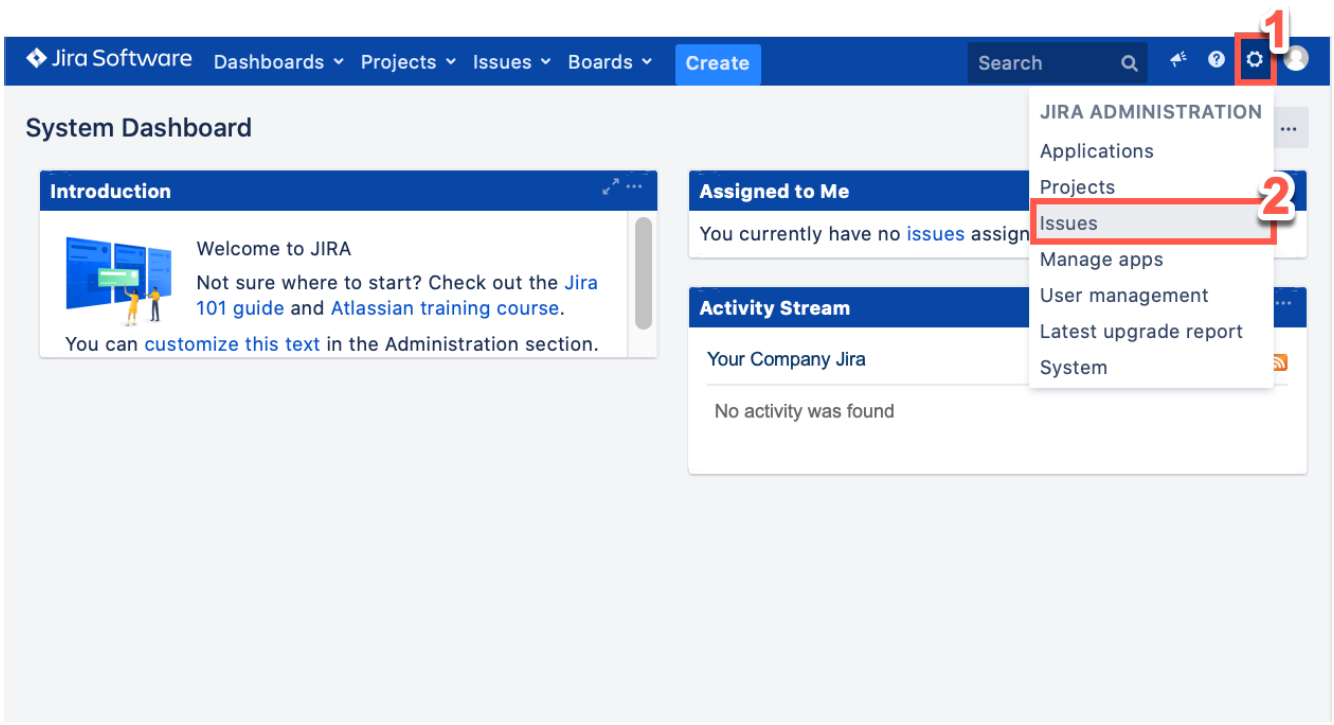
Define the statuses that an issue can belong to, every status state must be defined in the workflow. The status names can be found in Jira.

```
<status_map>
<status name="To Do" state="Open"></status>
<status name="In Progress" state="Open"></status>
<status name="Done" state="Closed"></status>
</status_map>
```

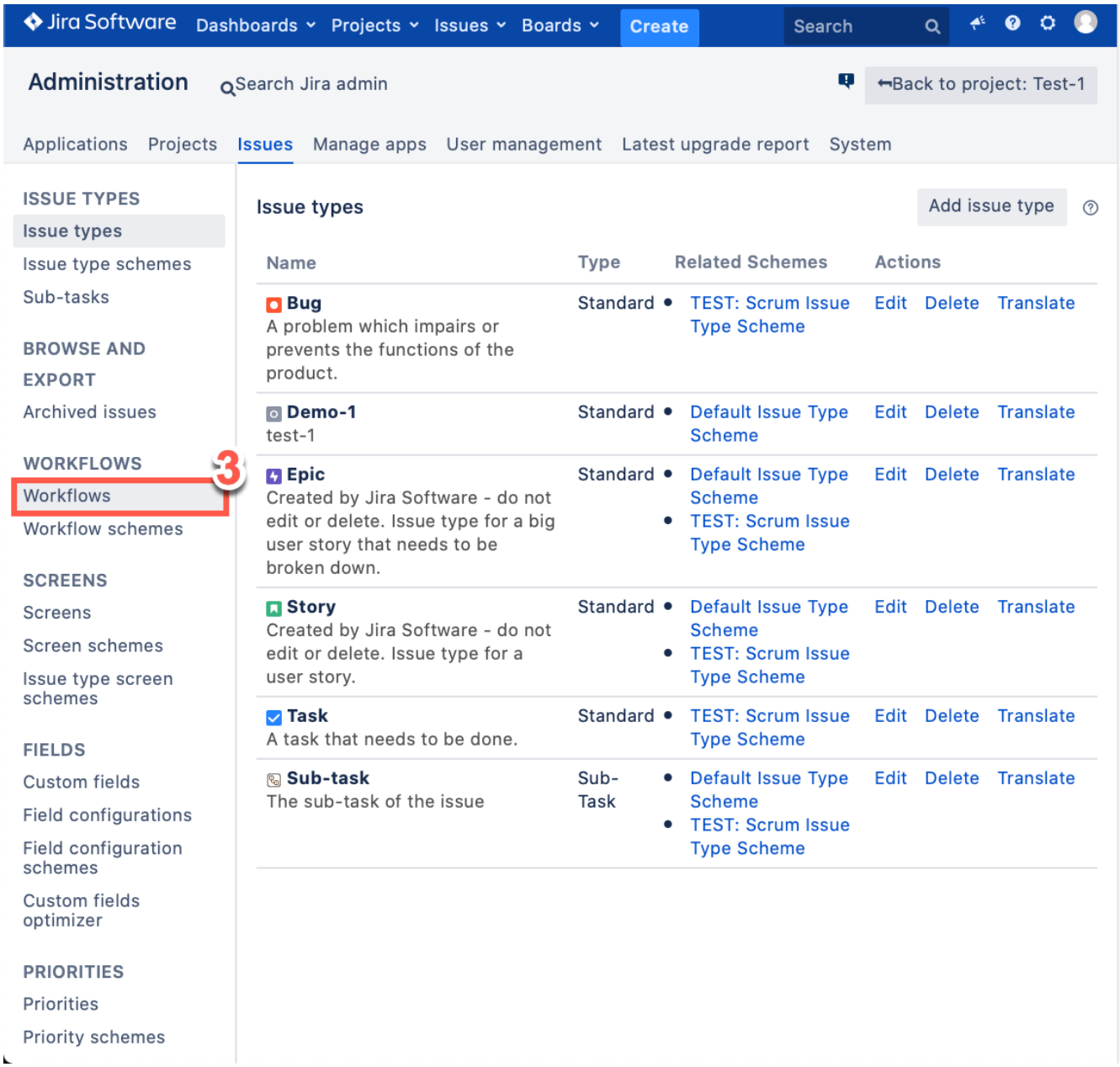
NOTE	The status names given here are case sensitive and must exactly match the status name given in Jira.

The status names in Jira can be located by performing the following steps:

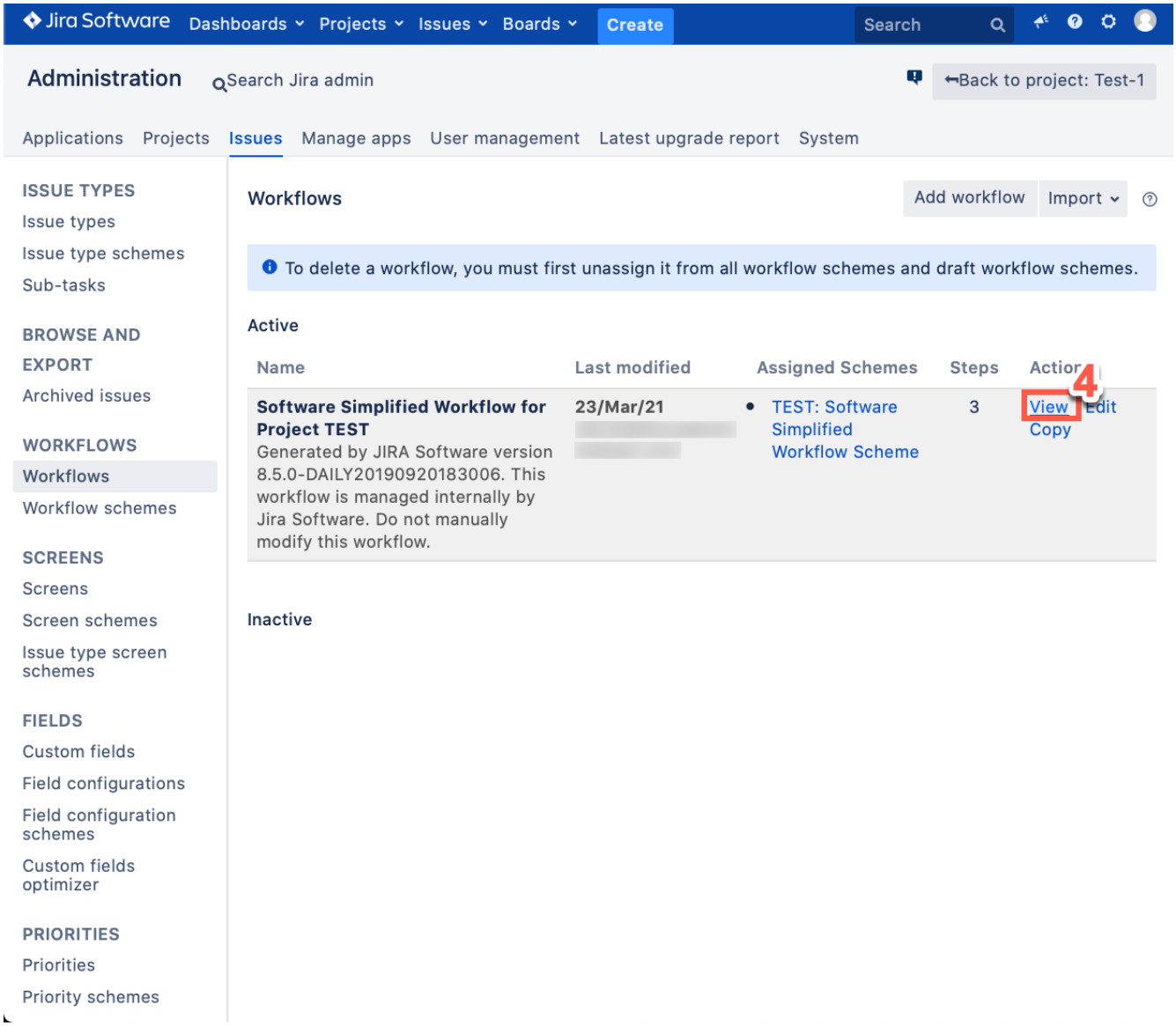
1. From the Jira **System Dashboard** click the gear icon.



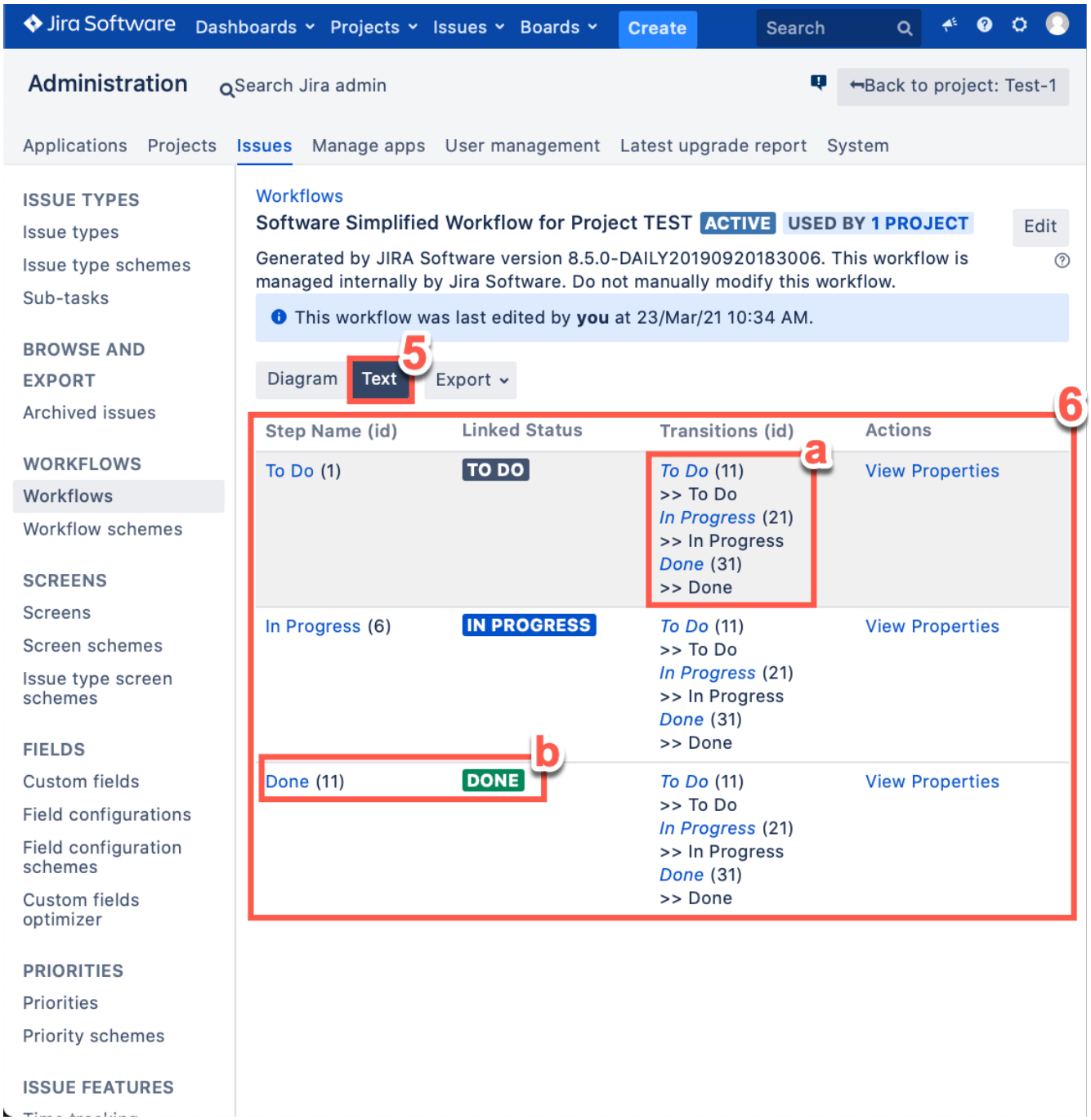
2. Click **Issues**.
3. Click **Workflows**.



4. Click the **View** link on any workflow.



5. Click **Text** to see the list of statuses contained in the workflow.



6. The **Jira Workflow** has a list of all transitions that are possible for a given Status.
   a. **Transitions (id)** shows each possible transition out of that state with the transition ID in parentheses. The transition ID for moving from a status of **To Do** to a status of **Done** is **31**.
   b. **Step Name (id)** and **Linked Status** show each status and its status ID in parentheses.
7. Define the `transition_action id` for each relevant status change in the workflow. Every status will include a `transition type` and `transition_action id`, any required fields must be included.

   For example, to move a Jira ticket from a state of **To Do** (as shown above) to a state of **Done**, your XML may look like this:

```
<status name="To Do">
  <transition type="Close">
    <transition_action id="31">
    </transition_action>
  </transition>
</status>
```

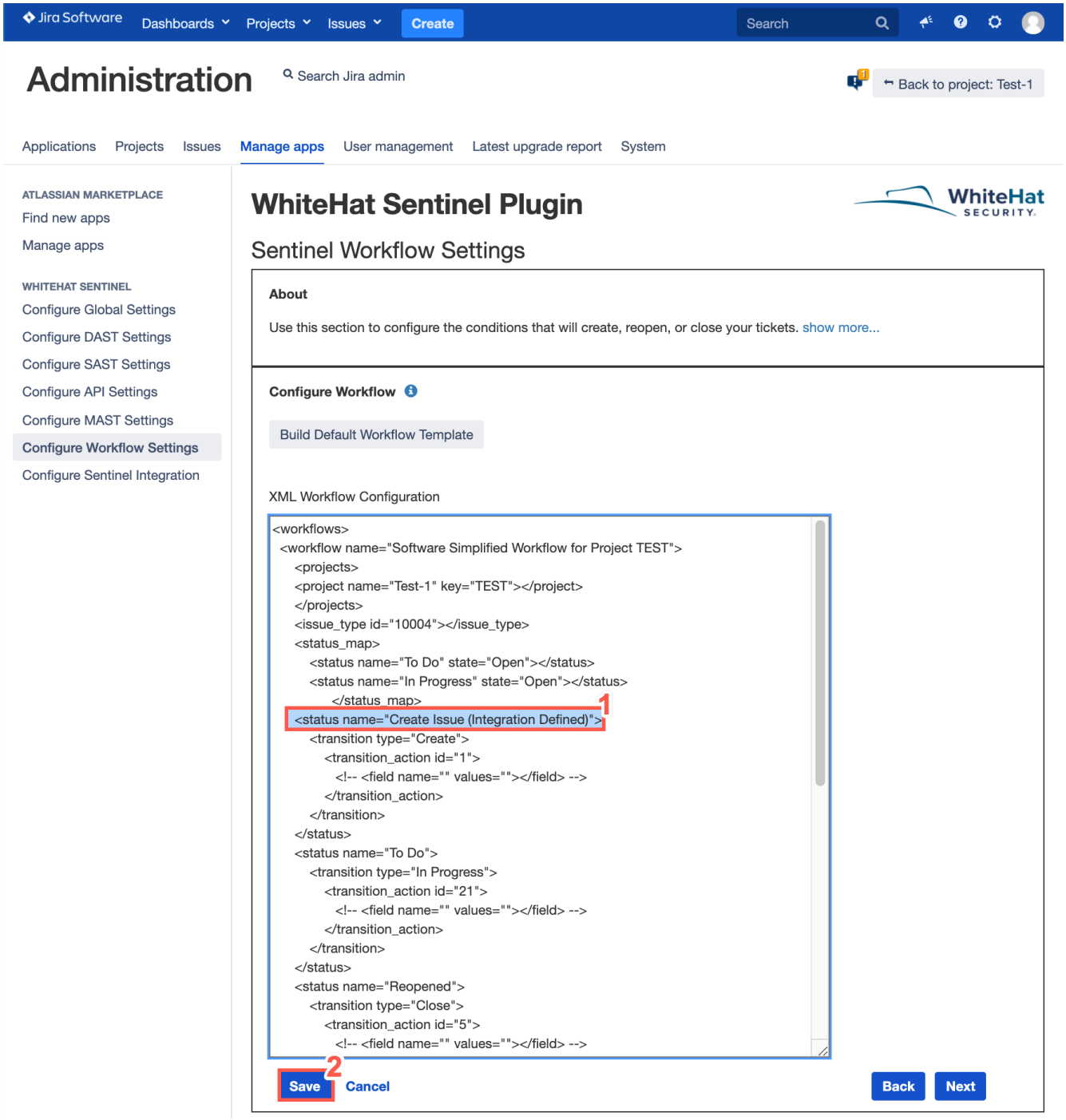
If any transition has a required field, that field must be included in the XML for the transition. If the transition above had a required field called `Resolved`, and you want to configure the field value to be `Yes`, the XML above would become:

```
<status name="To Do">
  <transition type="Close">
    <transition_action id="31">
     <field name="Resolved" values="Yes"></field>
    </transition_action>
  </transition>
</status>
```

3.5.5. The Create Issue (Integration Defined) Status

The default XML Workflow includes the status `Create Issue (Integration Defined)` which is a custom status for the Continuous Dynamic to Jira integration.

1. If any custom fields are required to create an issue, include them here. Otherwise please do not edit the `Create Issue (Integration Defined)` status.



2. When you have defined the XML workflow configuration, click **Save**.

For an example of a completed XML Workflow configuration, please see Sample Workflow.

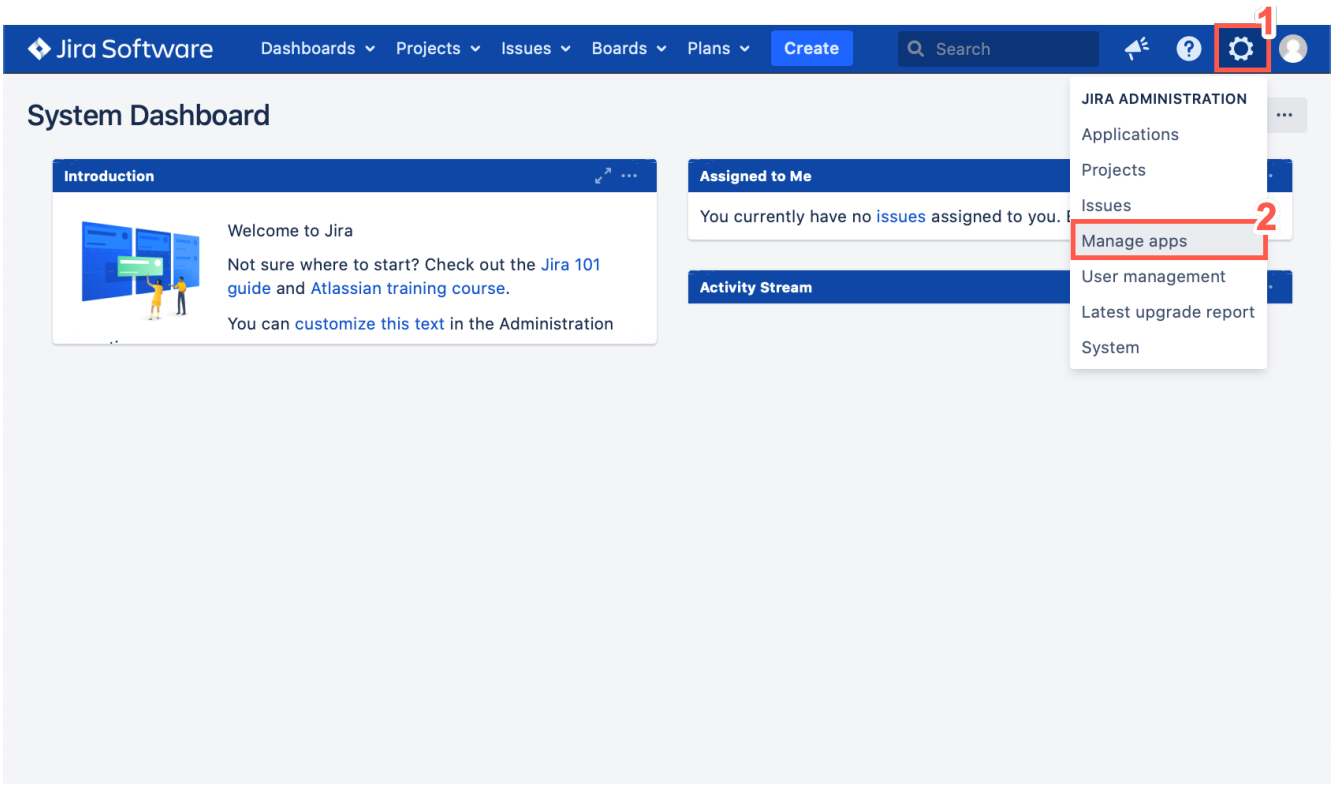
3.6. Continuous Dynamic Integration

This page allows an admin user to start and stop the Continuous Dynamic Plugin for Jira Data Center integration or view the plugin log files. Once all fields are populated correctly and the previous settings are saved, integration can begin.

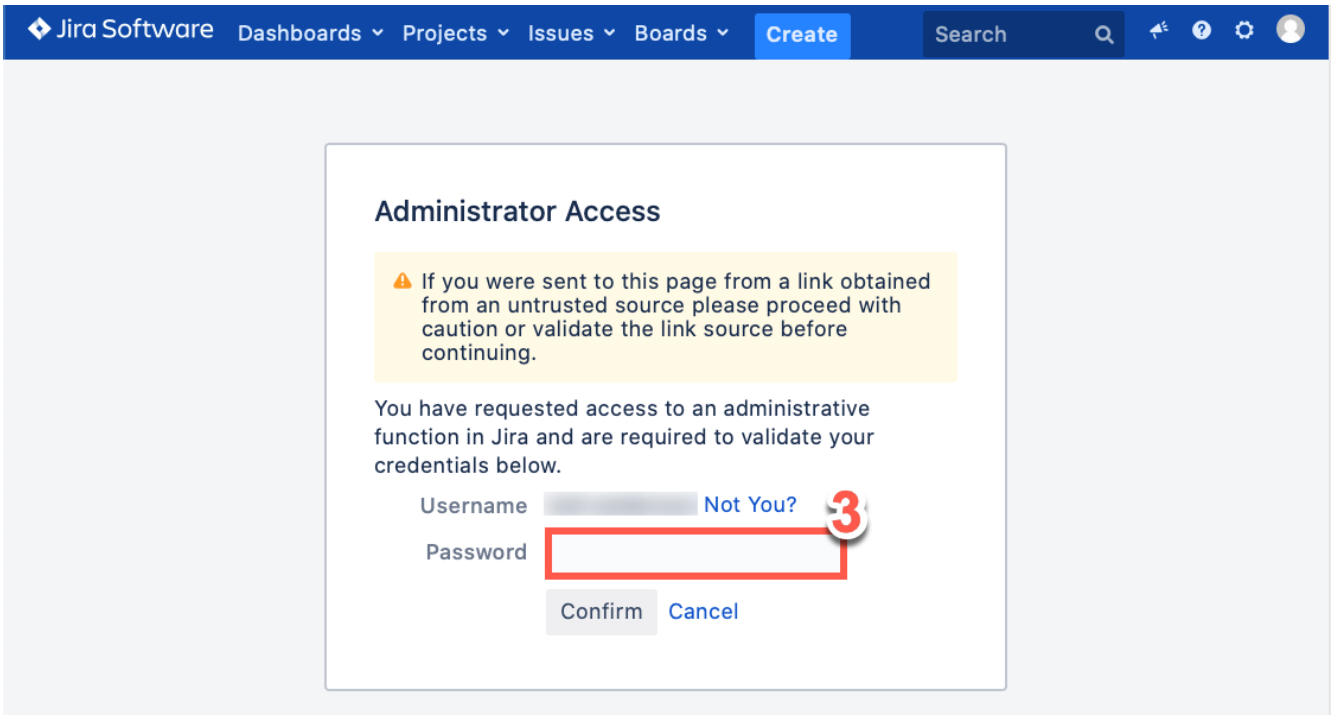
NOTE	If any changes to the settings of the Plugin are made while integration is running, you will need to stop the integration and re-start it to reflect the changed values.

To begin the initial integration process, perform the following steps:

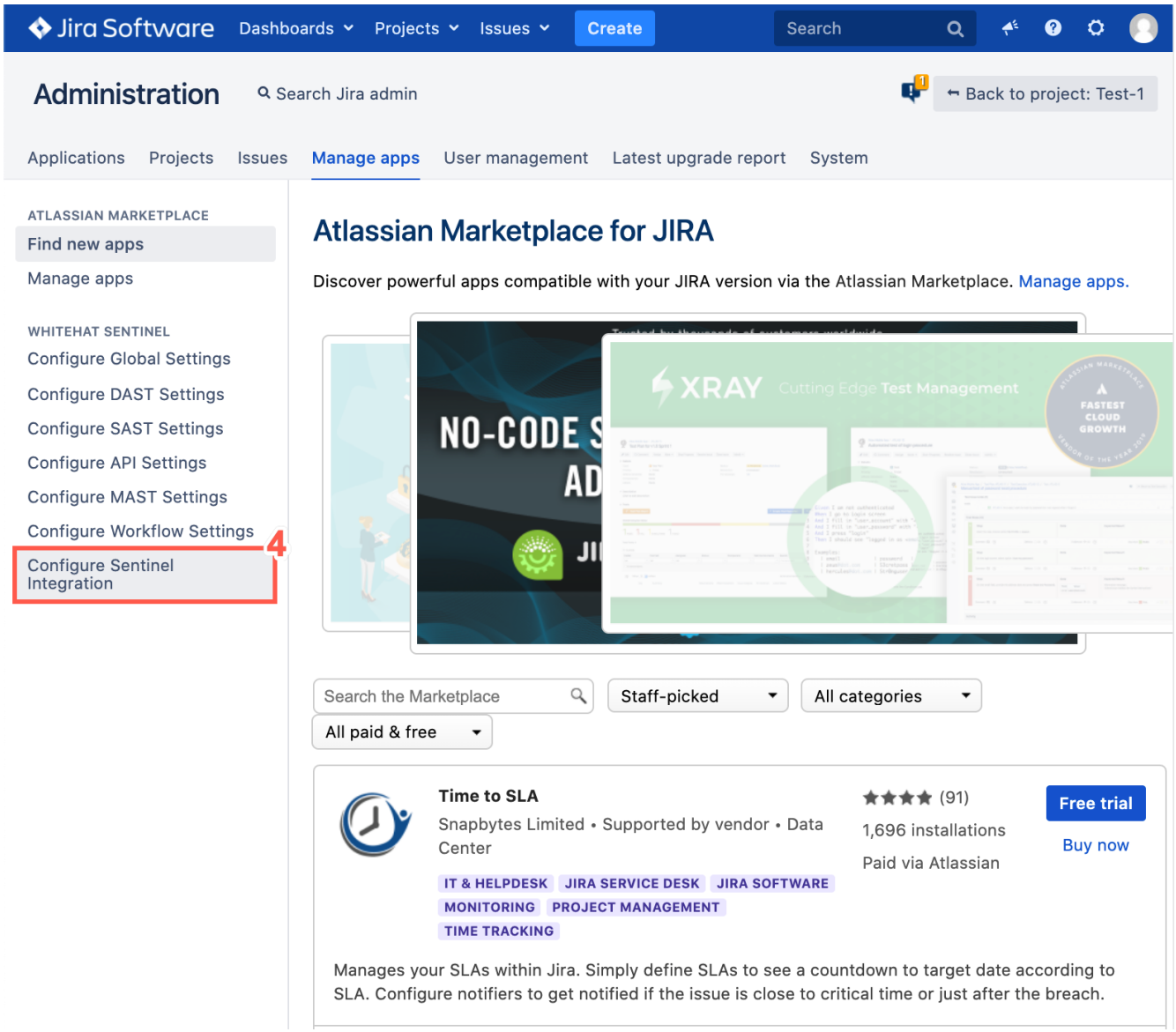
1. From the Jira **System Dashboard** click the gear icon.



2. Click **Manage apps**
3. Log into an Admin account to manage the apps installed on Jira.

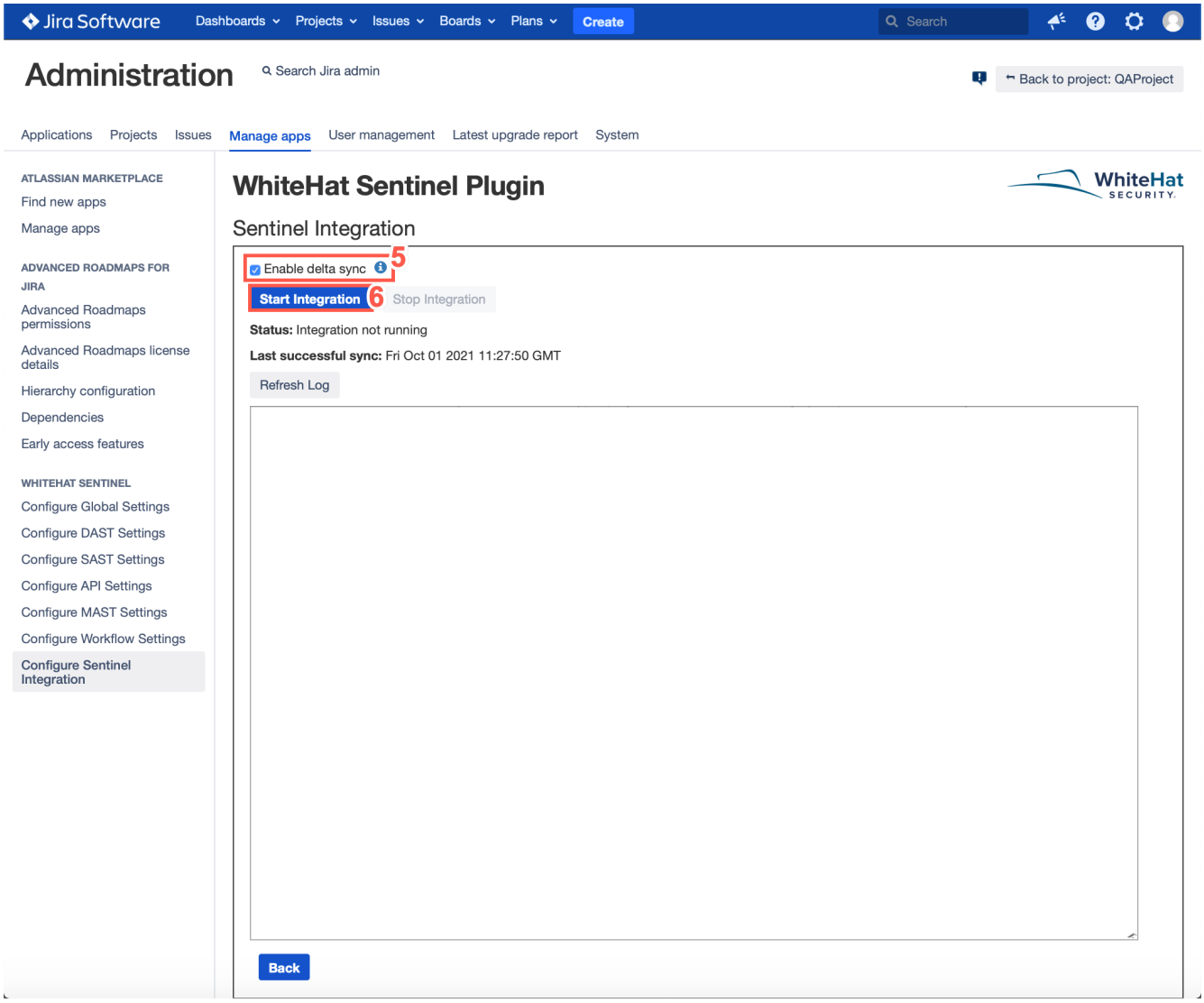


4. Click **Configure Continuous Dynamic Integration**.



5. Ensure the **Delta Sync** box is **unchecked**.

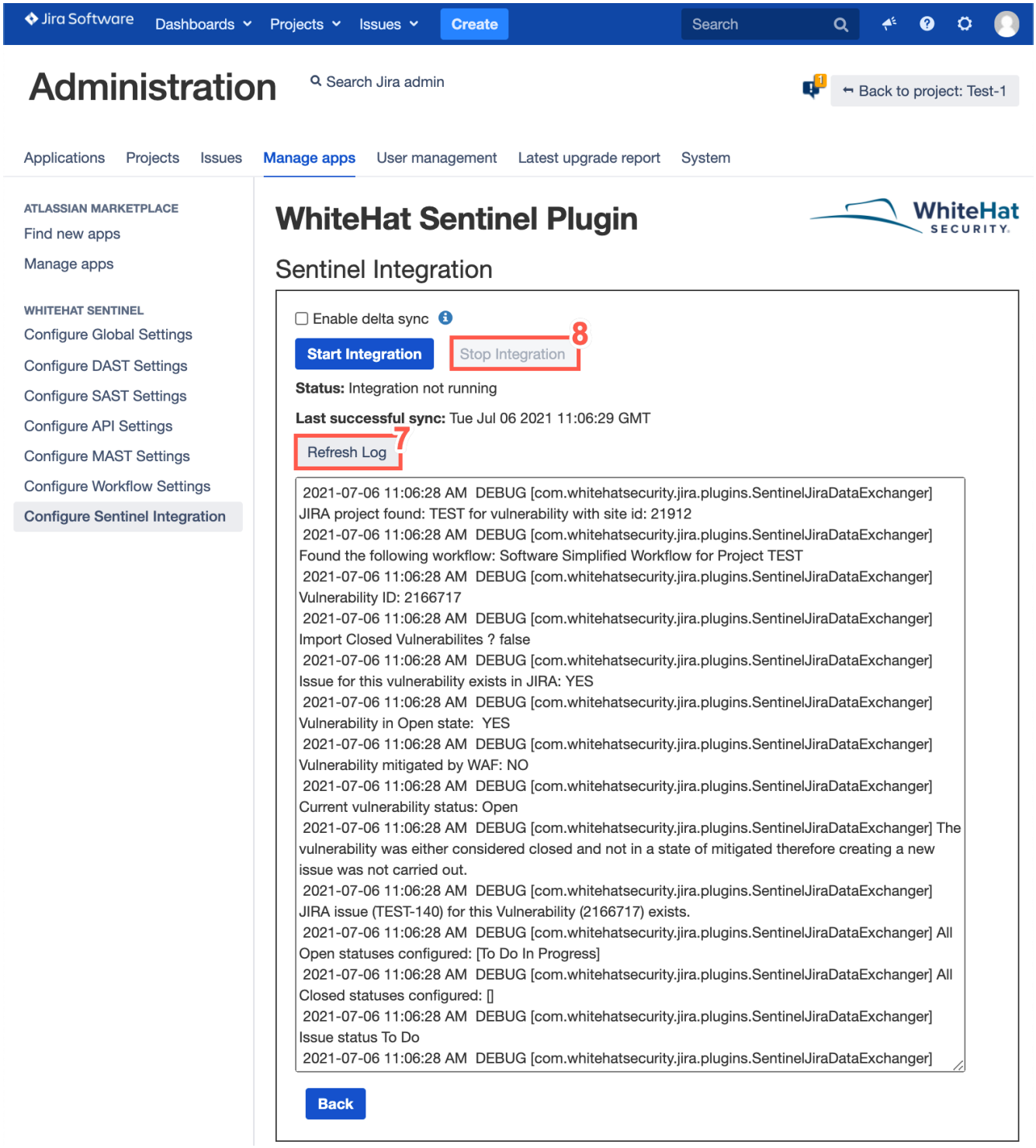
NOTE	Enabling this option only integrates new changes to the existing Plugin configuration, since the last successful integration.



6. Click **Start Integration** to begin Continuous Dynamic Portal data syncing with Jira. This runs on the interval specified by the user.

NOTE	Once the initial integration is complete, optionally Delta Sync may be turned on. This will enable the synchronization to be run only for items that have changed since the last successful synchronization. To run a full integration again, Delta Sync will need to be disabled.

7. To view the log files, click **Refresh Log**. This will display the latest one thousand lines of the log, and will automatically take you to the most recent portion of the log.



8. If necessary, use the **Stop Integration** button to stop your integration. Any vulnerabilities that have been added to Jira will be available to you in Jira, but no further vulnerabilities will be added unless and until you start a new integration process.

3.7. Sample Workflow

The following is a sample Plugin XML Workflow Configuration designed to work with the Jira **DEMO: Simple Issue Tracking Workflow**.

3.7.1. Sample Plugin XML Workflow Configuration

```
<workflows>
<workflow name="DEMO: Simple Issue Tracking Workflow">
  <projects>
  <project name="DEMO" key="DEMO"></project>
  </projects>
  <issue_type id="3"></issue_type>
  <status_map>
  <status name="To Do" state="Open"></status>
  <status name="In Progress" state="Open"> </status>
  <status name="Done" state="Closed"></status>
  </status_map>
  <status name="Create Issue (Integration Defined)">
  <transition type="Create">
    <transition_action id="1">
    </transition_action>
  </transition>
  </status>
  <status name="To Do">
  <transition type="Done">
    <transition_action id="21">
    <field name="Resolved" values="Yes"></field>
    </transition_action>
  </transition>
  </status>
  <status name="In Progress">
  <transition type="Done">
    <transition_action id="41">
    <!-- <field name="" values=""></field> -->
    </transition_action>
  </transition>
  </status>
  <status name="Done">
  <transition type="To Do">
    <transition_action id="51">
    </transition_action>
  </transition>
  </status>
</workflow>
</workflows>
```

3.7.2. Jira DEMO: Simple Issue Tracking Workflow

This table shows the following issue types IDs:

Type	Label	ID
Step	**To Do**	1
Step	**In Progress**	2
Step	**Done**	3
Transition	From **To Do** to **Done**	21
Transition	From **In Progress** to **Done**	31
Transition	From **Done** to **To Do**	51

The XML must reflect the terminology in Jira exactly, including letter case and spacing if any.

Chapter 4. The Issues Page with the Continuous Dynamic Plugin for Jira Data Center

The **Issues** page of every vulnerability created by the Continuous Dynamic Plugin for Jira Data Center has now integrated key Continuous Dynamic features, allowing users to get the information they need to remediate vulnerabilities without leaving Jira.

Continuous Dynamic's vulnerability data is presented in the description section of the Issue. At the bottom of every Issue you will see the **Continuous Dynamic - Vulnerability Management** section.

NOTE	In order to see this information, you must have configured your Global Settings appropriately. For more information, see Configure Global Settings / Configure Vulnerability Content. For troubleshooting help, see Troubleshooting.

4.1. Enable Vulnerability Management

To enable **Vulnerability Management**, perform the following:

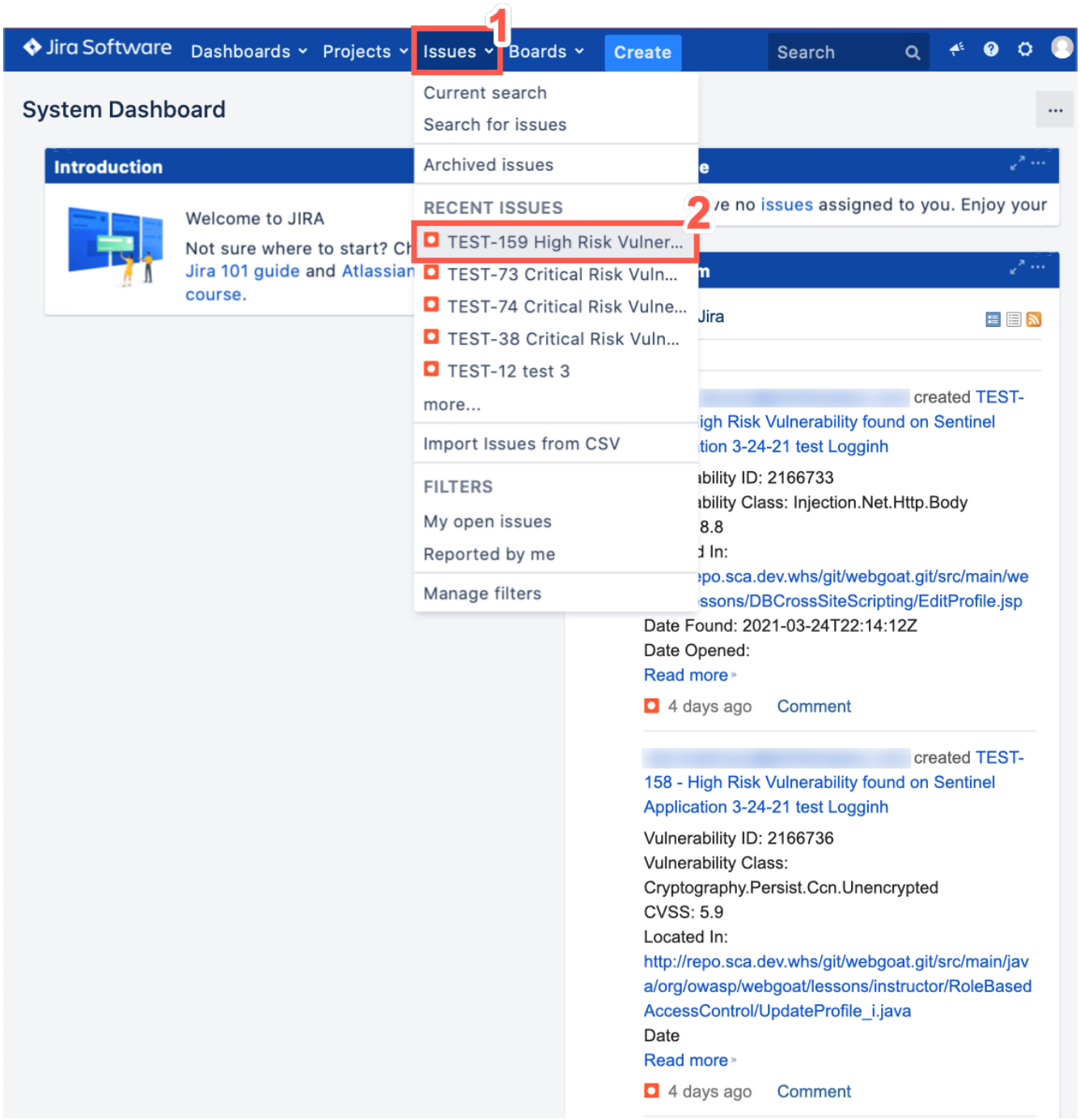
1. When completing the **Configure Global Settings**, select the **Yes** radio button.

2. Select **Save**.

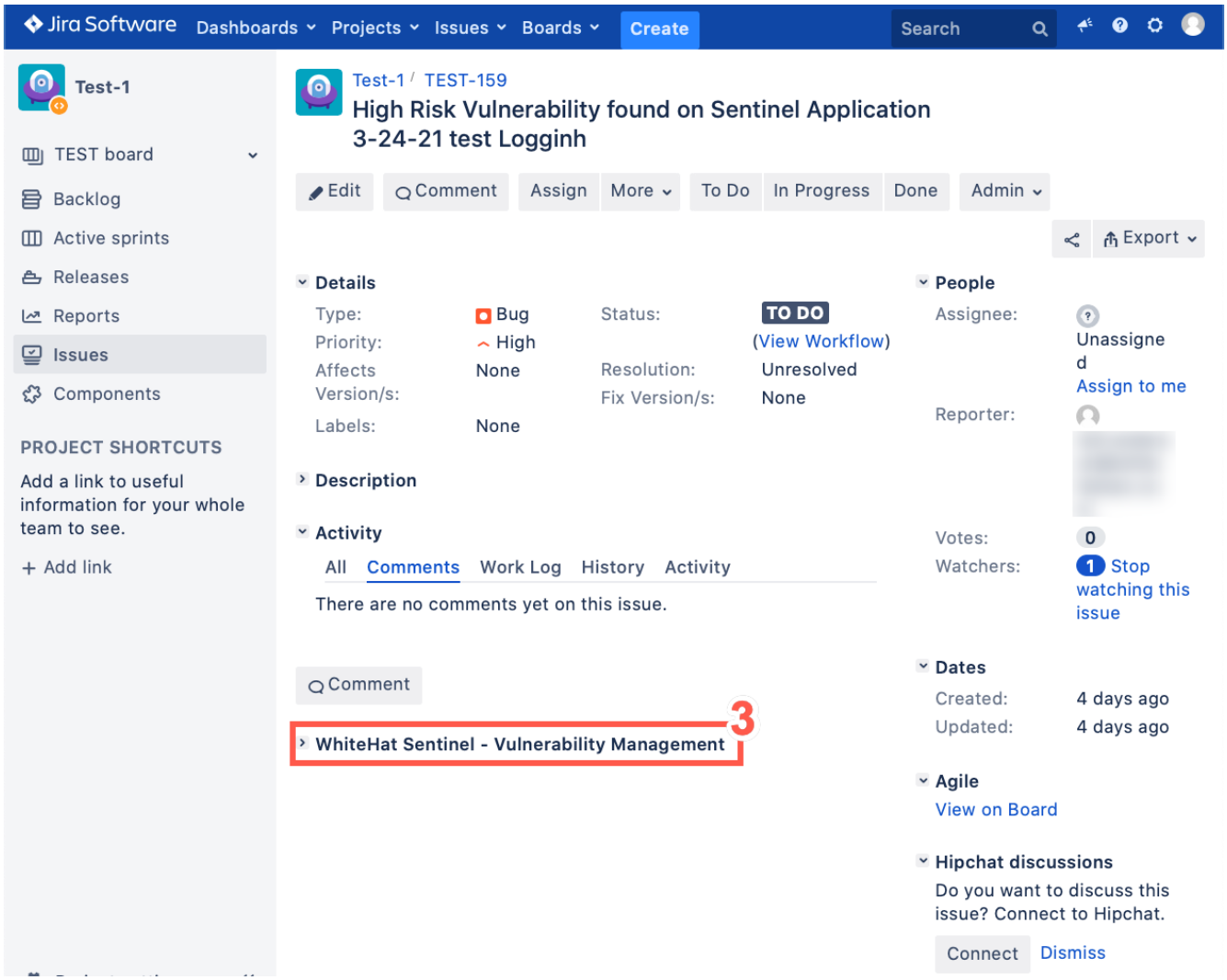
4.2. Adding a Note or Tag

To add a **Note** or **Tag** using the **Vulnerability Management** functions perform the following steps:

1. From the Jira system Dashboard, select **Issues**.



2. Select an Issue.
3. Click **Continuous Dynamic - Vulnerability Management** to expand the section.



a. Optionally, to retest this Issue select **Vulnerability Retest**.

b. Select from the drop down menu which update type you want to add.

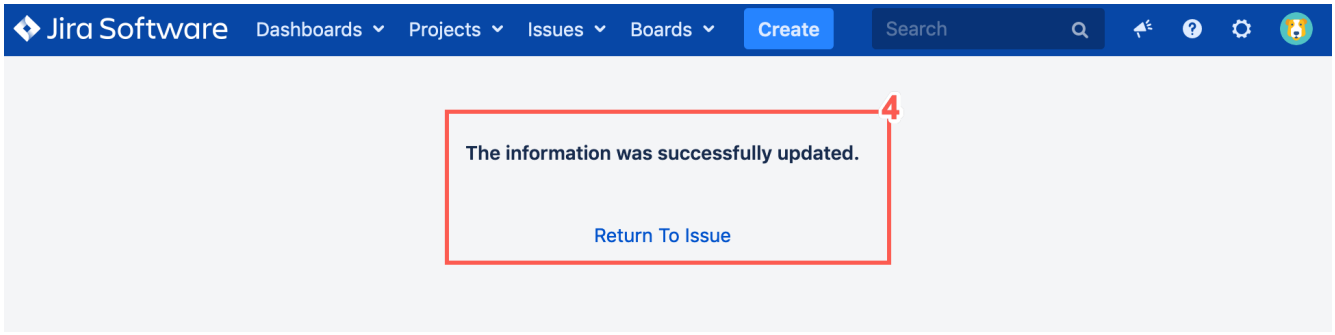
c. Select from the drop down menu which attack vector the note or tag will be added to.

> **NOTE** For **SAST** and **MAST** assets, the option to select an **Attack vector** is unavailable.

d. Type the **Note** or **Tag** in the text field.

e. Click **Submit**.

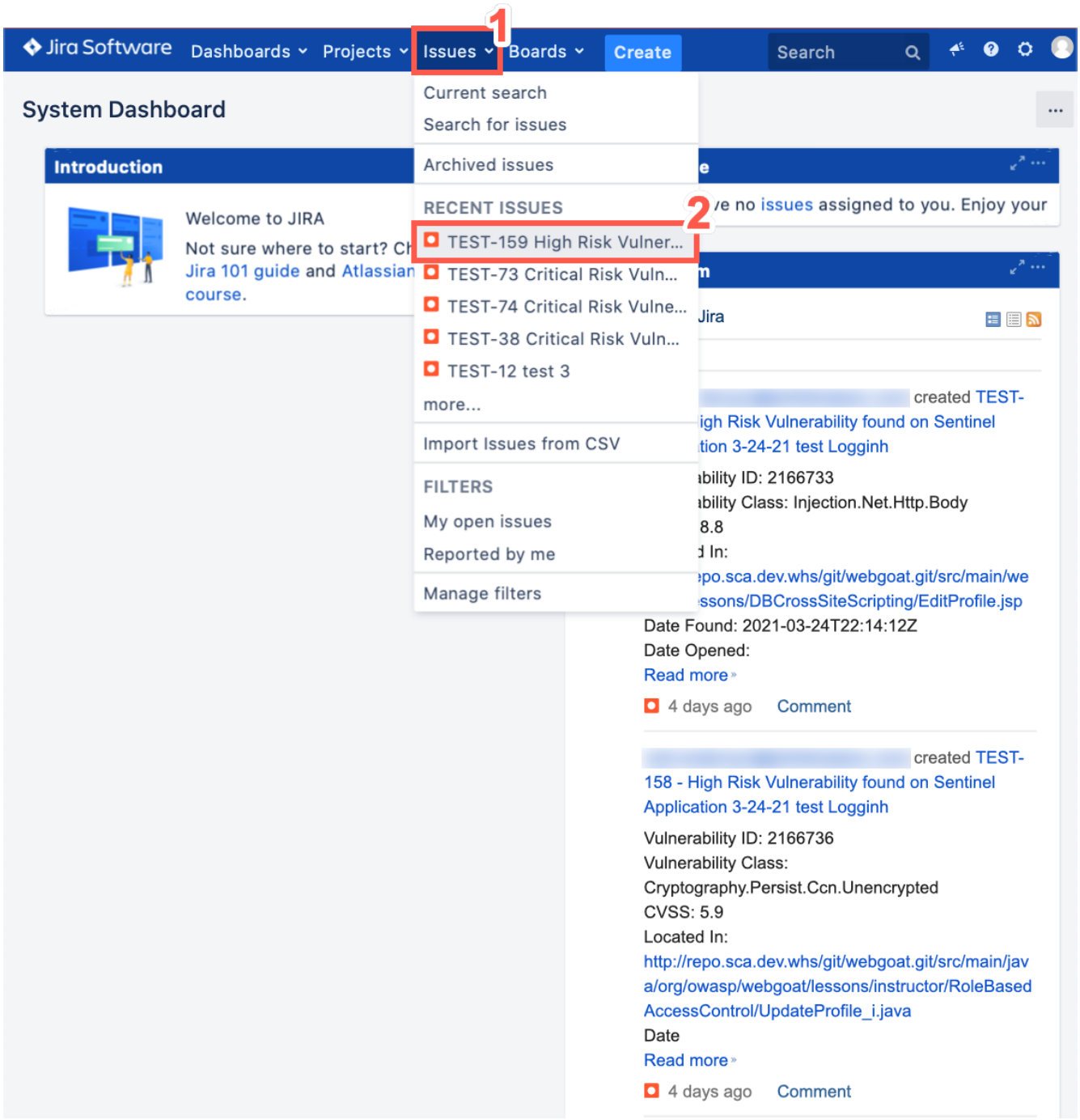
4. If the **Note** or **Tag** is successfully added, the following message is displayed.



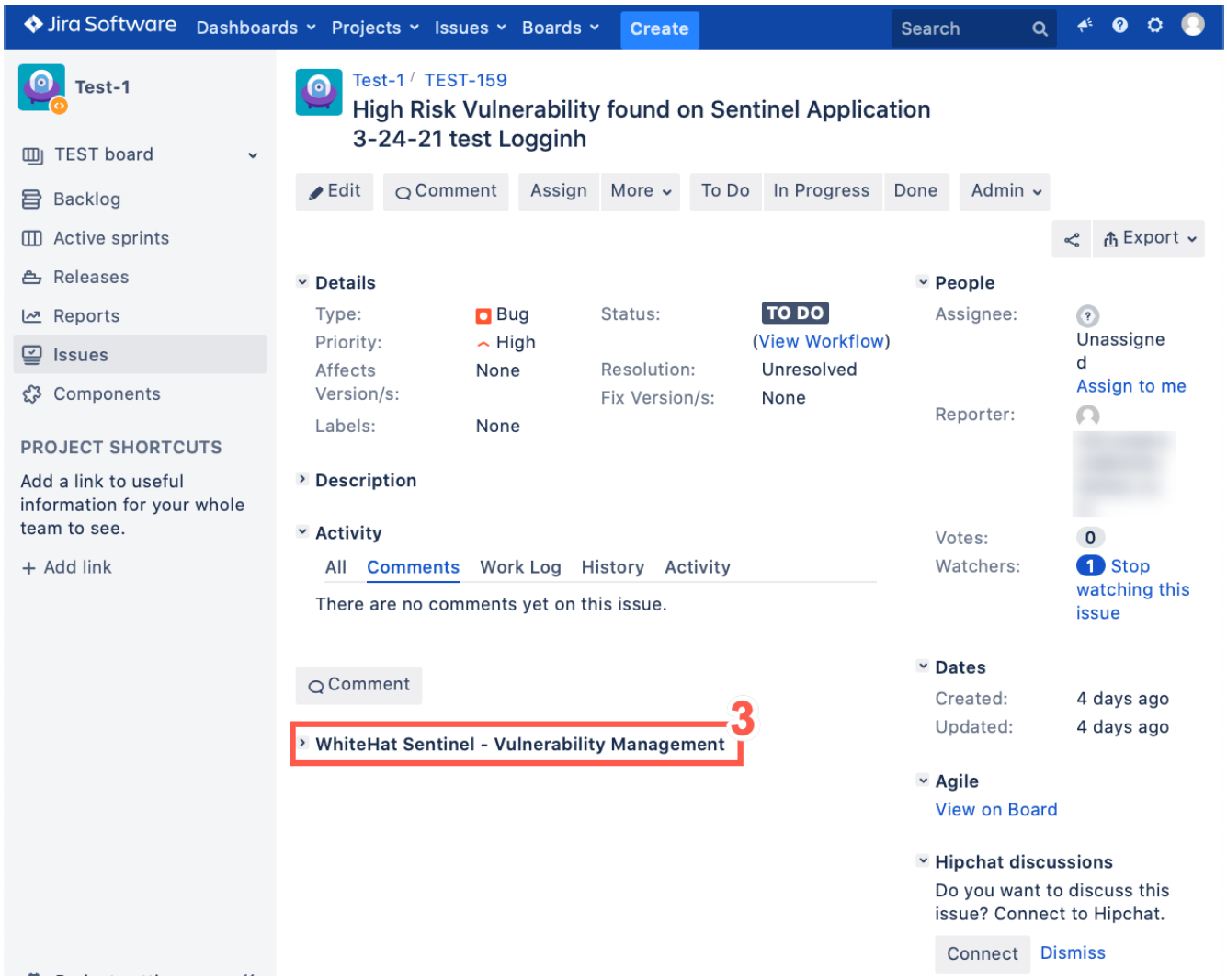
4.3. Ask a Question

To ask a question using the **Vulnerability Management** functions, perform the following steps:

1. From the Jira system Dashboard, select **Issues**.



2. Select an Issue.
3. Click **Continuous Dynamic - Vulnerability Management** to expand the section.

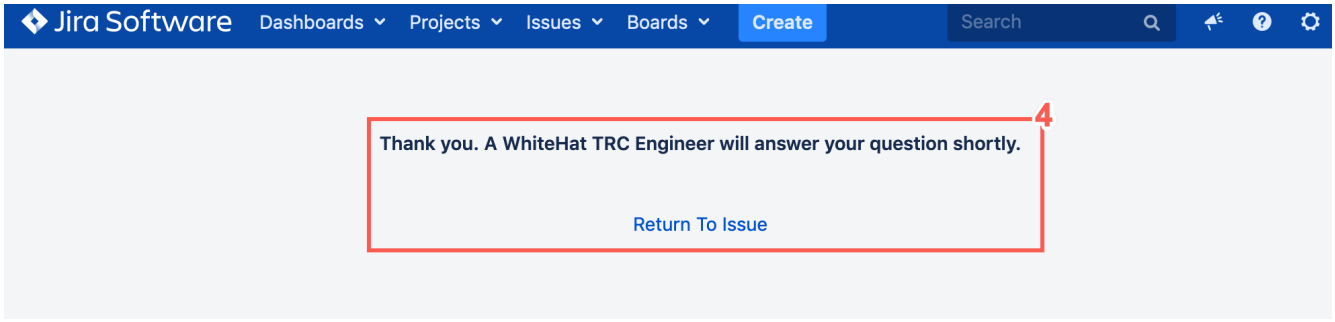


a. From the drop down menu, select the **Category** that your question belongs to.

b. Type your question in the text field.

c. Select **Submit** to send your question to the Black Duck support team.

4. If the question is successfully submitted, the following message is displayed.



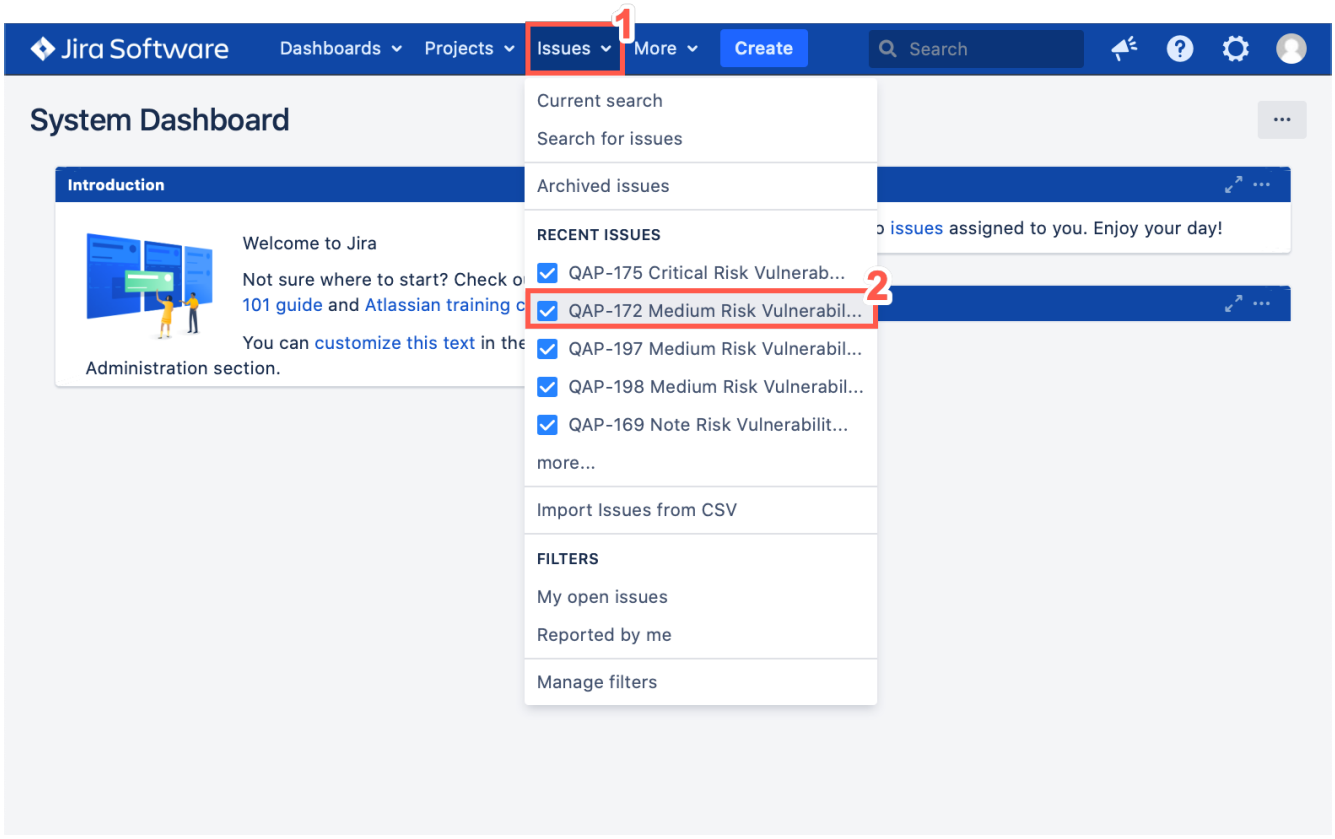
5. Previous questions and responses can be reviewed in the **Ask a Question** section of the **Vulnerability Management** functions.

NOTE	The **Retest** and **Add Tag or Note Vulnerability Management** functions are only available for **DAST** Issues.

4.4. View Vulnerability Trace

To view vulnerability traces for Continuous Dynamic issues with a description and solution, perform the following steps:

1. From the Jira system Dashboard, select **Issues**.



2. Select an Issue.

3. At the bottom of the **Description**, click the link for **More information**. This includes the vulnerability traces, description and solution.


Jira Software
Dashboards
Projects
Issues
Boards
Plans
Create
Search
QAProject / QAP-172
Medium Risk Vulnerability found on Sentinel Application 3-6 Sast EE Cloud upload
Edit
Add comment
Assign
More
To Do
In Progress
Workflow
Admin
Export
Details
Type: Task
Priority: Medium
Labels: None
Status: TO DO (View Workflow)
Resolution: Unresolved
People
Assignee: Unassigned
Assign to me
Reporter: QA
Votes: 0
Watchers: 1 Stop watching this issue
Dates
Created: 13/Sep/21 12:22 AM
Updated: 13/Sep/21 12:22 AM
Agile
View on Board
Description
Vulnerability ID: 1405043
Vulnerability Class: Error.Handler.Global
CVSS: 5.3
Located In: sample.war/WEB-INF/web.xml
Date Found: 2020-03-06T20:43:05Z
Date Opened: 2020-03-06T20:43:11Z
Date Closed:
Status: Open
Risk: Medium
Likelihood: Medium
Impact: Medium
Description:
The application fails to make sufficient use of a global error handling mechanism. Analysis of the deployment descriptor either noted a lack of or an incorrect use of the error-page directive. Failure to leverage a global error handling mechanism increases the risk that verbose implementation details will be revealed to attackers through a stack trace. Developers not using this mechanism are required to implement their own global error handling mechanism.
Solution:
Ensure the application deployment descriptor declares an error-page declaration that catches all uncaught exceptions thrown by the application. Such practice will significantly minimize the risk of disclosing sensitive implementation details through error messages.
In Java:
The web.xml should include error pages for 403, 404, 500, java.lang.Throwable, and a "catch-all":
<error-page>
<error-code>403</error-code>
<location>/error.jsp</location>
</error-page>
<error-page>
<error-code>404</error-code>
<location>/error.jsp</location>
</error-page>
<error-page>
<error-code>500</error-code>
<location>/error.jsp</location>
</error-page>
<error-page>
<exception-type>java.lang.Throwable</exception-type>
<location>/error.jsp</location>
</error-page>
<error-page>
<location>/error.jsp</location>
</error-page>
In C#:
The web.config should set the custom error mode to "on", define a defaultRedirect page, and include redirect pages for 403, 404, and 500:
<customErrors mode="On" defaultRedirect="error.html">
<error statusCode="500" redirect="err500.aspx"/>
<error statusCode="404" redirect="notHere.aspx"/>
<error statusCode="403" redirect="notAuthz.aspx"/>
</customErrors>
The vulnerability details can be found at:
https://source.whitehatsec.com/application_findings_detail.html?app_id=17446&vuln_id=1405043&status=open
More information can be found at: http://docker-host06.qa.whs:8080/user/viewSASTVulnDetails.jspa?vulnID=1405043
3
Attachments
Activity
Add comment
WhiteHat Sentinel - Vulnerability Management

4. This table displays the **Vector ID**, **Line Number** and code **Snippet** relating to this vulnerability.

WhiteHat Sentinel Plugin

SAST Vuln View

Vulnerability Class
Application Misconfiguration: Global Error Handling Disabled

Vulnerability Id
1405043

Traces

4

Vector ID	Line Number	Snippet
96445318-1		
FileReference	1	**Path:** sample.war/WEB-INF/web.xml

```
1  <?xml version="1.0" encoding="ISO-8859-1"?> /*FileReference*/
2  <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4
   xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
5      version="2.4">
6
7     <display-name>Hello, World Application</display-name>
8     <description>
9     This is a simple web application with a source code organization
10     based on the recommendations of the Application Developer's Guide.
11     </description>
12
13     <servlet>
14        <servlet-name>HelloServlet</servlet-name>
15        <servlet-class>mypackage.Hello</servlet-class>
16     </servlet>
17
```

Description

5

The application fails to make sufficient use of a global error handling mechanism. Analysis of the deployment descriptor either noted a lack of or an incorrect use of the error-page directive. Failure to leverage a global error handling mechanism increases the risk that verbose implementation details will be revealed to attackers through a stack trace. Developers not using this mechanism are required to implement their own global error handling mechanism.

Solution

6

Ensure the application deployment descriptor declares an error-page declaration that catches all uncaught exceptions thrown by the application. Such practice will significantly minimize the risk of disclosing sensitive implementation details through error messages.

In Java:

The web.xml should include error pages for 403, 404, 500, java.lang.Throwable, and a "catch-all":

```
<error-page>
  <error-code>403</error-code>
  <location>/error.jsp</location>
</error-page>
<error-page>
  <error-code>404</error-code>
  <location>/error.jsp</location>
</error-page>
<error-page>
  <error-code>500</error-code>
  <location>/error.jsp</location>
</error-page>
<error-page>
  <exception-type>java.lang.Throwable</exception-type>
  <location>/error.jsp</location>
</error-page>
<error-page>
  <location>/error.jsp</location>
</error-page>
```

In C#:

The web.config should set the custom error mode to "on", define a defaultRedirect page, and include redirect pages for 403, 404, and 500:

```
<customErrors mode="On" defaultRedirect="error.html">
    <error statusCode="500" redirect="err500.aspx"/>
    <error statusCode="404" redirect="notHere.aspx"/>
    <error statusCode="403" redirect="notAuthz.aspx"/>
</customErrors>
```

5. The **Description** tab offers a description of the vulnerability class in question.
6. The **Solution** tab will offer an approach to resolving the vulnerability in question.

For troubleshooting help, please see [^top].

Chapter 5. Troubleshooting the Continuous Dynamic Plugin for Jira Data Center

The most common issue encountered is when, after starting the integration, no **Issues** are generated.

In most cases, this is caused by the selected **Assignee** not having adequate privileges to create that issue. To verify this, you can manually create an example issue in the Jira project.

For easy troubleshooting of other issues, the Plugin has been set up to generate verbose debugging. This can be viewed by going to the Jira log folder and pulling data from the `atlassian-jira.log` file. The verbosity of the log files generated can be configured in the Jira log configurations.

Chapter 6. Updating the Continuous Dynamic Plugin for Jira Data Center

6.1. Download the new Plugin

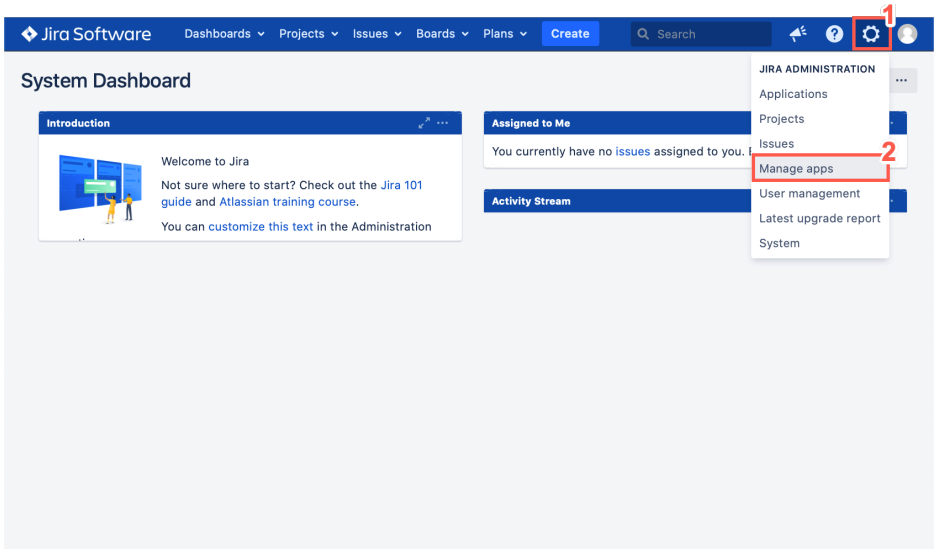
The latest version of the Continuous Dynamic Plugin for Jira Data Center is available to download from:

- The Atlassian Marketplace.

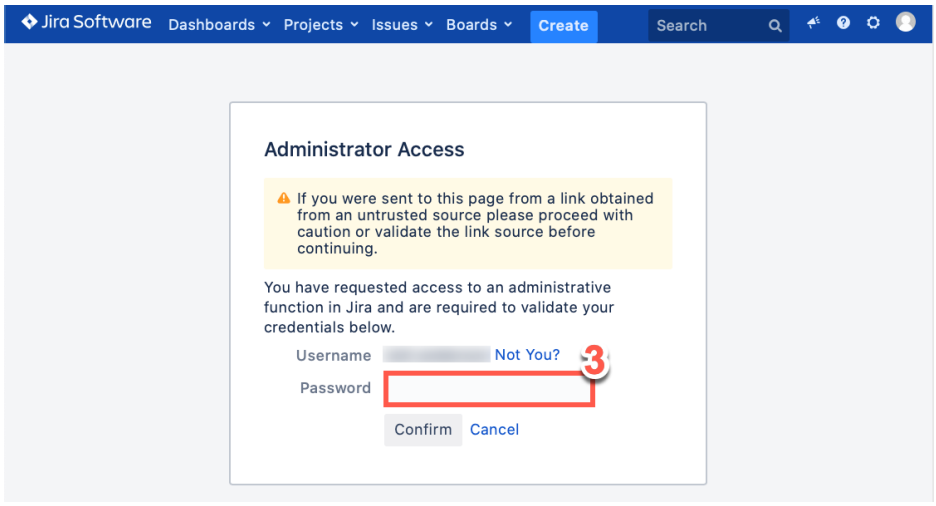
6.2. Update Process

To update the Continuous Dynamic Plugin for Jira Data Center from v4.0.1 to the latest version, perform the following steps:

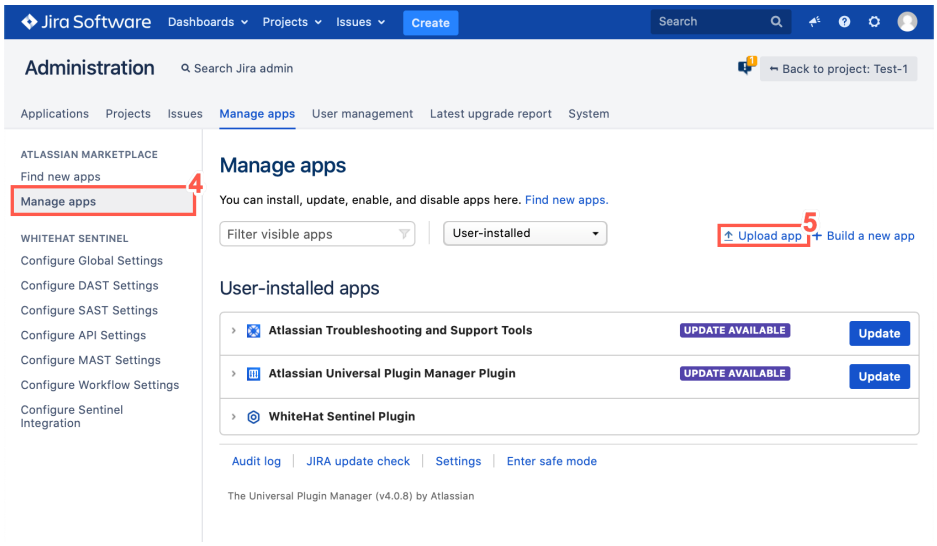
1. From the Jira **System Dashboard** click the gear icon.



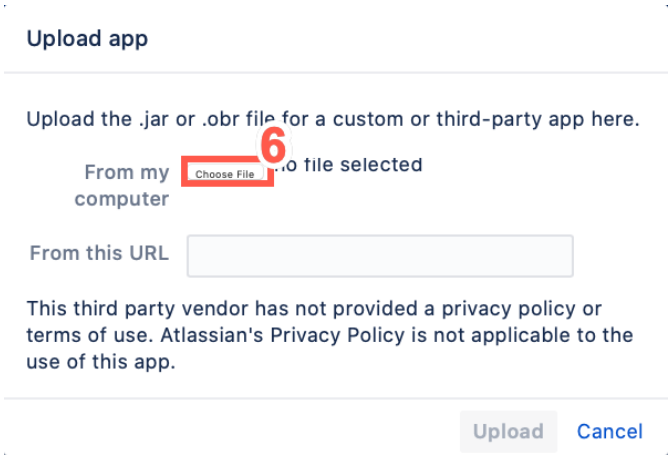
2. Click **Manage apps**.
3. Log into an Admin account to manage the apps installed on Jira.



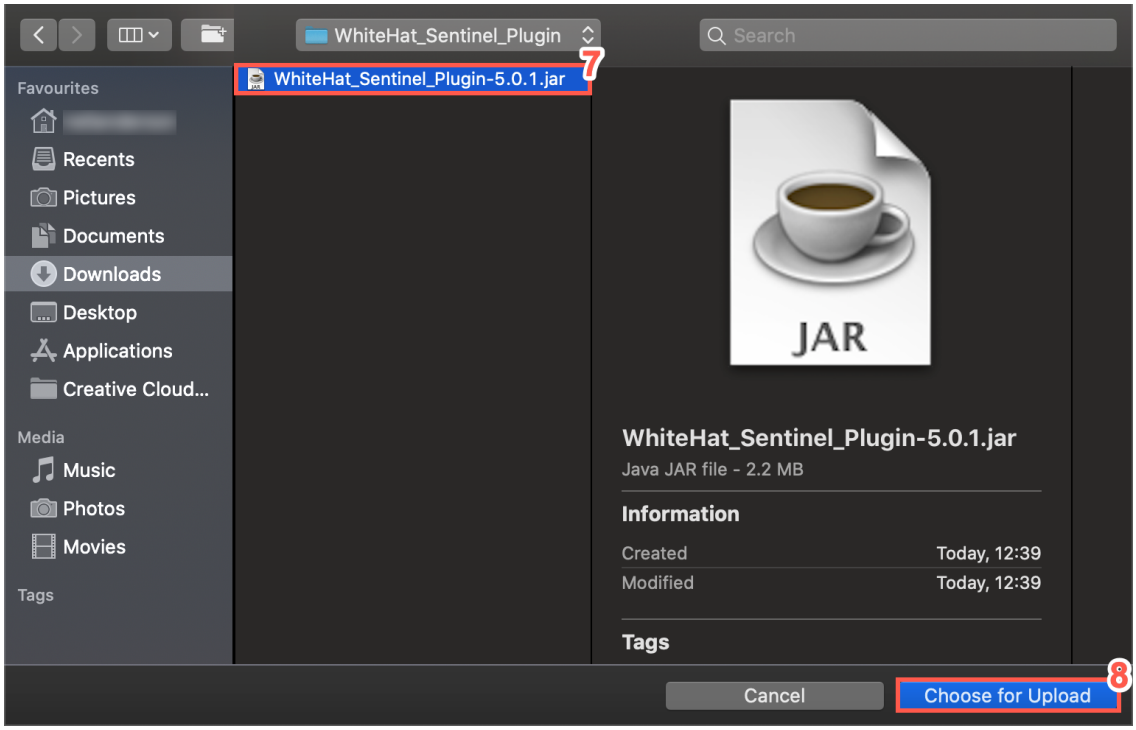
4. Click **Manage apps**.



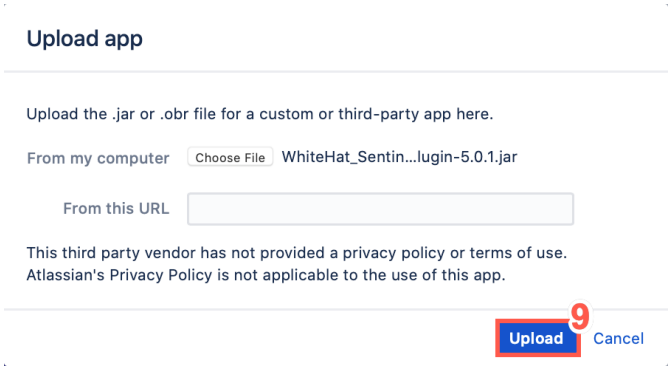
5. Click **Upload app**.
6. Click the **Choose File** option.



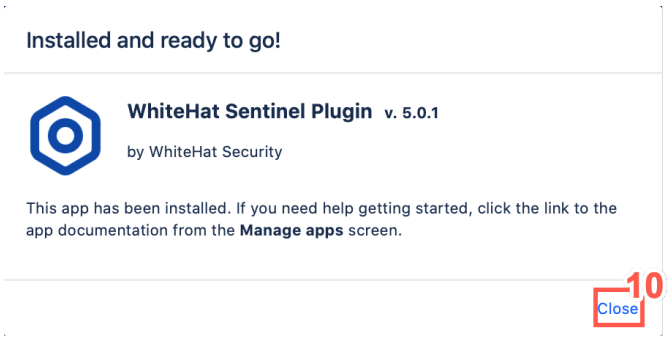
7. Select the latest **WHS JIRA Plugin .jar** file.



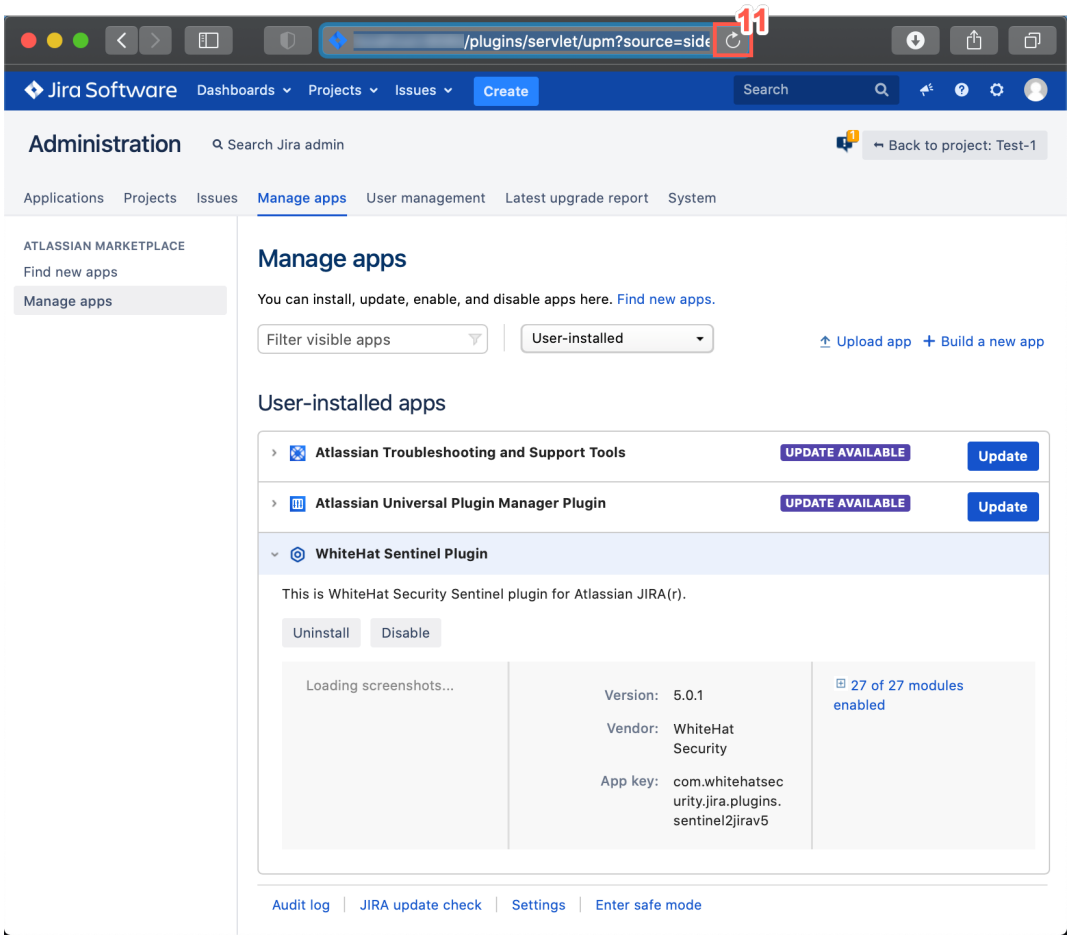
8. Click **Choose for Upload**.
9. Click **Upload**.



10. Click **Close**.



11. If necessary, refresh the **Manage apps** page to display the updated Plugin.



12. Once this process is complete, the Plugin has been successfully updated.

6.3. Plugin Update FAQ

1. **How do I access the Customer Support Portal ?**
   Log in to the Black Duck Community.
2. **How do I install the new Plugin ?**
   Follow the install process detailed in Installing the Continuous Dynamic Plugin for Jira Data Center for more information.
3. **Can I download the Plugin from the Atlassian Marketplace ?**
   Yes, currently the Plugin is available to download from The Atlassian Marketplace.
4. **Do I need to reconfigure the Plugin ?**
   When you update from 4.0.1 or earlier versions to 5.0.1 and above, you don't need to reconfigure the Plugin.
5. **What will happen to Jira issues already created ?**
   Issues created by the prior Plugin will be recognized and handled appropriately by the updated Plugin.
6. **Previously, I manually created tickets for each vulnerability in the Continuous Dynamic Portal. Will the Plugin create duplicates when syncing ?**

Unfortunately the Plugin will not be able to recognize and properly synchronize or transition Jira issues not created by our Plugin.

7. **How do I contact the support team?**
   You can log in to the Community Portal and contact Customer Support by creating or responding to a case. If you do not have Community Portal access you can email support@whitehatsec.com.