Chronicle App for Jira Cloud

Release Documentation and User Guide

Overview

Chronicle is a global security telemetry platform for investigating incidents and hunting for threats in your enterprise network. Purpose-built on core Google infrastructure, Chronicle can ingest massive amounts of telemetry data, normalize it, index it, correlate it to known threats, and make it available for analysis in seconds. The intended application will create Jira issues based on alerts, IoCs, detections and curated detections generated by the Chronicle platform. The app also provides functionality to start, cancel and list retrohunts from the Jira platform.

Chronicle App for Jira Cloud provides functionality to periodically receive alerts, IoCs, detections, and curated detections from Google Chronicle into Jira Cloud based on specific filters and configured schedules. The app allows users to configure filters related to alerts, detections, and curated detections. The app would create issues in the configured project in Jira based on the Chronicle data. The created issues will have custom issue types, custom fields, and the Chronicle Enrichment feature. The app provides a Chronicle Enrichment manual action. Users can enrich Jira issues with any of the following that would be added as part of the Issue comment:

1. IoC Details
2. List Events Discovered
3. List Asset Impacted
4. List Asset Aliases
5. List User Aliases

Compatibility Matrix

Browser	Google Chrome, Safari
Google Chronicle REST API Version	IOCs: v1, Alerts: v1, Detections: v2, Curated Detections: v2
Jira Cloud REST API Version	v3
Forge CLI Version	v6.3.0
Development Platform	Atlassian Forge
App Hosting Type	Cloud
Supported Atlassian Products	Jira, Jira Service Management

Prerequisites

- Jira Cloud instance configured properly with Chronicle App installed.

User Permissions

- Only Jira admin users could configure the App.

Release Notes

v2.0.0

- Automated Chronicle Sync
    - Added support for Curated Detections.
    - Added certain configuration parameters like support of dynamic region, limits on tickets created per sync, and some other filter parameters.

- Manual Chronicle Enrichment
    - Added support to perform the following new enrichments:
        - List Asset Aliases
        - List User Aliases

- Manual Update Rule State
    - The app also provides the ability to activate/deactivate the Alerting Rule State and Live Rule State of a particular detection rule.
    - This action would be available in all detection Jira issues.

- Retrohunt
    - A new tab is added to the app configuration page, which allows the user to start/cancel a Retrohunt.
    - The user can provide RuleID or VersionID along with the date range while starting a new Retrohunt.

Release History

v1.1.0

- Updated ruleLabels parsing function with empty values handled.
- Updated Jira priority field mapping to handle malicious IoC severity.

v1.0.0

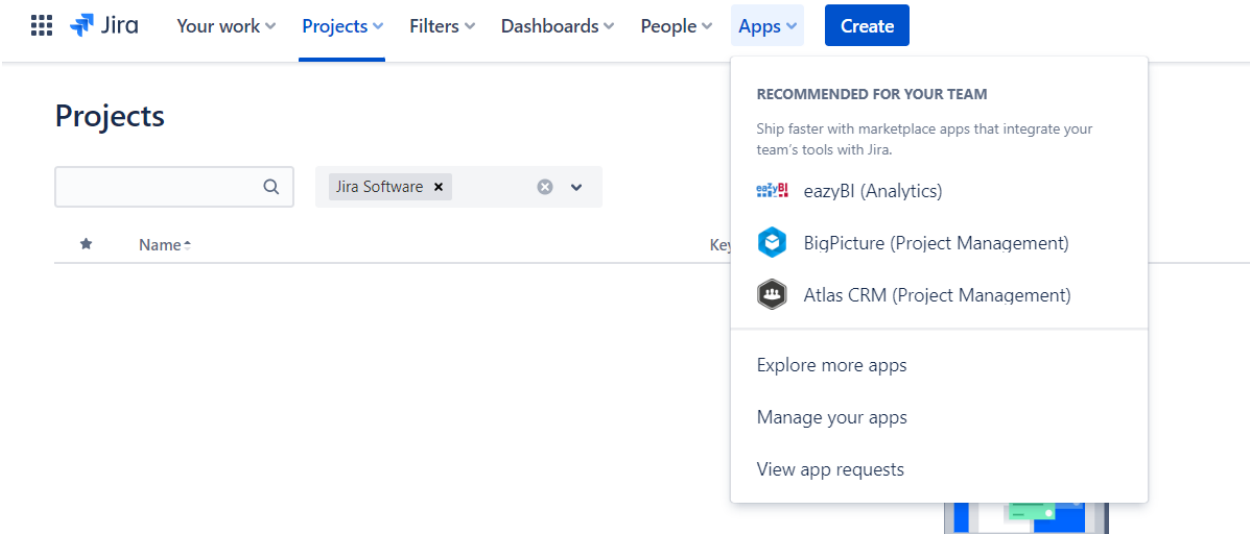
- Automated Chronicle Sync
    - The app will automatically fetch Chronicle Alerts, Detections, and Indicators of Compromise (IoCs) and create Jira issues corresponding to them
    - The syncing process occurs at the user-configured interval.
    - The app provides flexibility to the user for doing certain configurations for the scheduler, Jira projects, and filters for detections and alerts
    - The app would create custom issue types for Alerts, Detections, and IoCs in Jira and add custom fields to enrich issues with Chronicle Data

- Manual Chronicle Enrichment
    - The app provides a manual action in the issues created by the app to enrich them with the Chronicle information
    - App provides a manual action to bring data related to which Assets were impacted and which Events were discovered related to a particular Domain or IP address in the user-provided time frame and add the information in the Jira comment
    - The app allows users to perform the following enrichments:
        - IoC Details
        - List Assets Impacted
        - List Events Discovered

App Usage Instructions

Installation

1. Log in to your Atlassian Jira account. Click on the Apps tab on the top and then select Explore More Apps. Only Jira administrators have the privilege to access this.



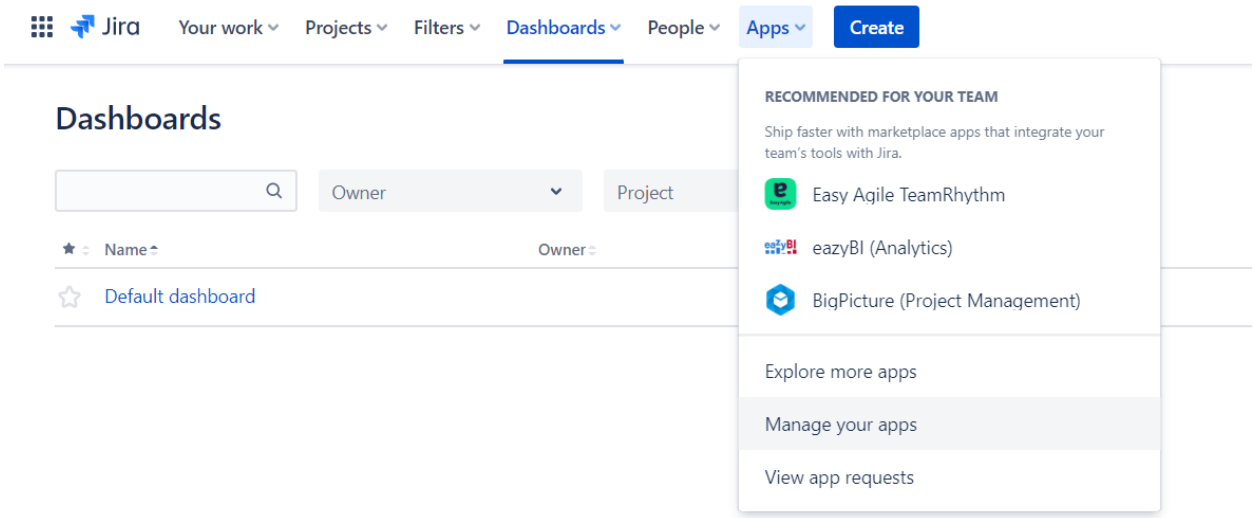
2. In the search bar, search for the Chronicle App for Jira. Click on the app and then press the *Get App* button. A pop-up would appear, then click on the *Get it now* button. Pressing that would begin the installation process. Once installed, a message would appear on the bottom left indicating that installation is successful.
3. Click on the apps tab on the top and navigate to *Manage Apps*. You can see the Chronicle App for Jira in the User-Installed Apps section.

Getting User's Service Account JSON

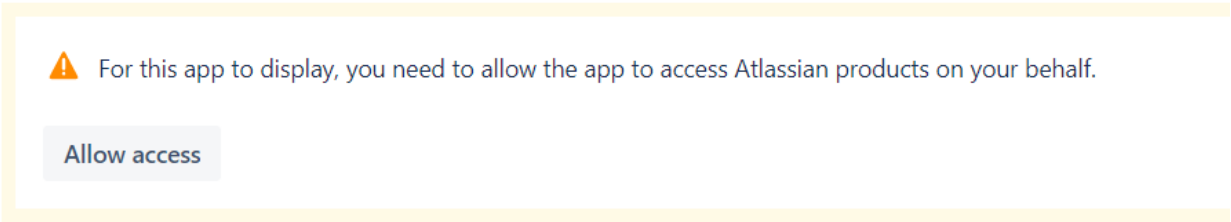
1. This app requires User's Service Account JSON from Chronicle, which is used to make API calls from Jira to Chronicle.
2. The User's Service Account JSON is required during the configuration of the app post-installation.
3. To generate the User's Service Account JSON, follow these steps.

Configuring Chronicle App

1. Post successful installation, under the Apps tab on the top, a Manage your Apps option would be visible. Clicking on it would open the Manage Apps section.

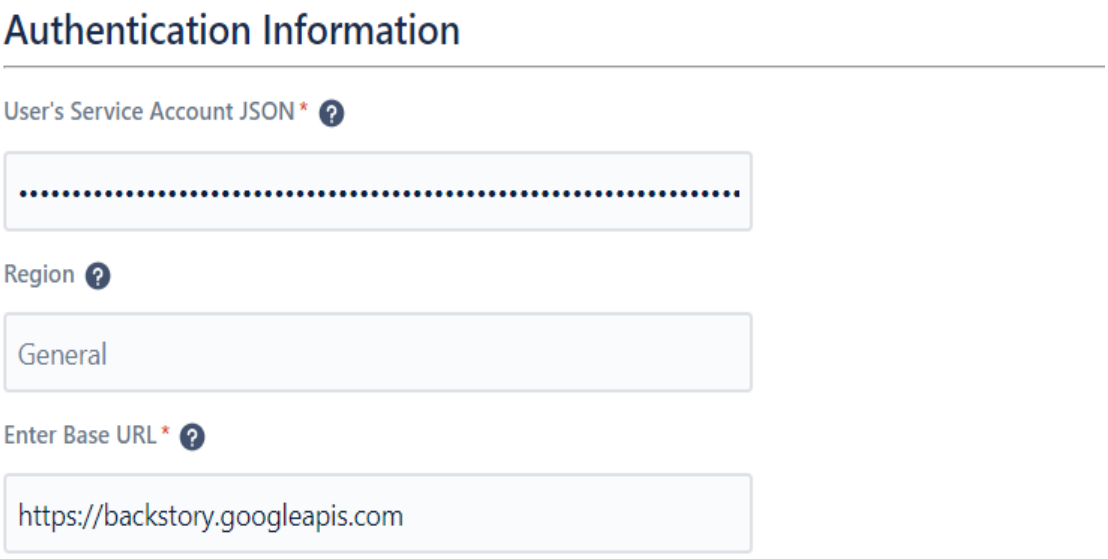


2. On the left panel, the Chronicle App for Jira Cloud under the Apps section would be visible, Clicking on it would open the configuration page for the Chronicle App for Jira Cloud.
3. For the first time, it might ask for allowing the app to access Atlassian products on your behalf. Clicking on the Allow Access button would open the authorization window. After validating the permission, the Accept button needs to be clicked.

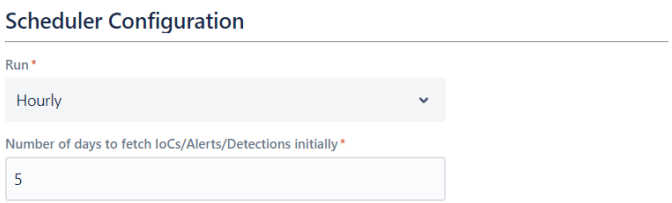


4. It would open up the Configuration page for the app.
5. The user needs to provide the *Google Service Account JSON* and *Select the region* based on the location of the chronicle backstory instance under the Authentication

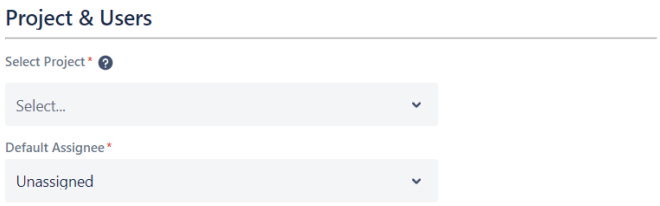
Information panel.



6. In Scheduler Configurations, a user needs to select the interval at which the syncing process should occur between Chronicle and Jira. The provided options are Hourly, Daily and Weekly. The app also allows fetching historical data. In the field *Number of days, to fetch IoCs/Alerts/Detections/CuratedDetections initially* users can add the number of days from which they want to collect data from the Chronicle and create Jira issues.



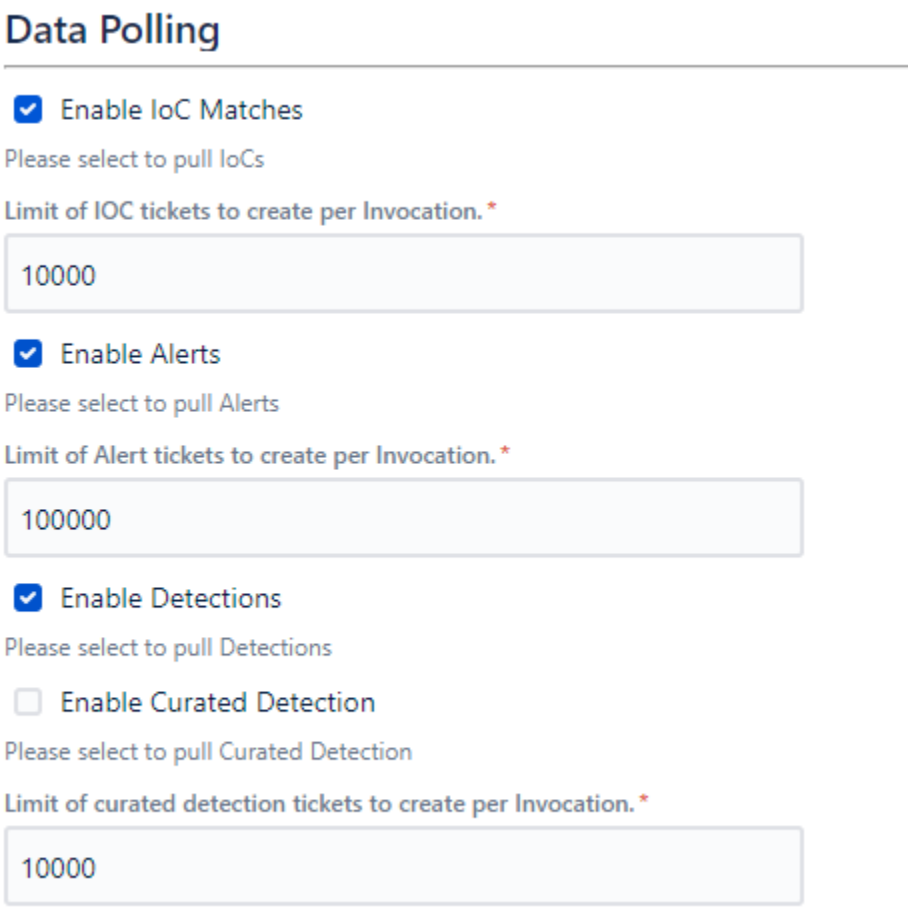
7. The user needs to select the project in which Jira issues will be created. If the required project is not visible in the dropdown, type the project name in the project selection field and if the project with the entered name is present it will show up for selection. The project selection field restricts team-managed projects. Users also need to select the *default Assignee*, the default will be unassigned.



8. The user needs to select the data polling checkbox for which the data needs to be fetched from the Chronicle and Jira issues need to be created. At least one checkbox selection is required. The user needs to specify the number of the ticket that needs to be

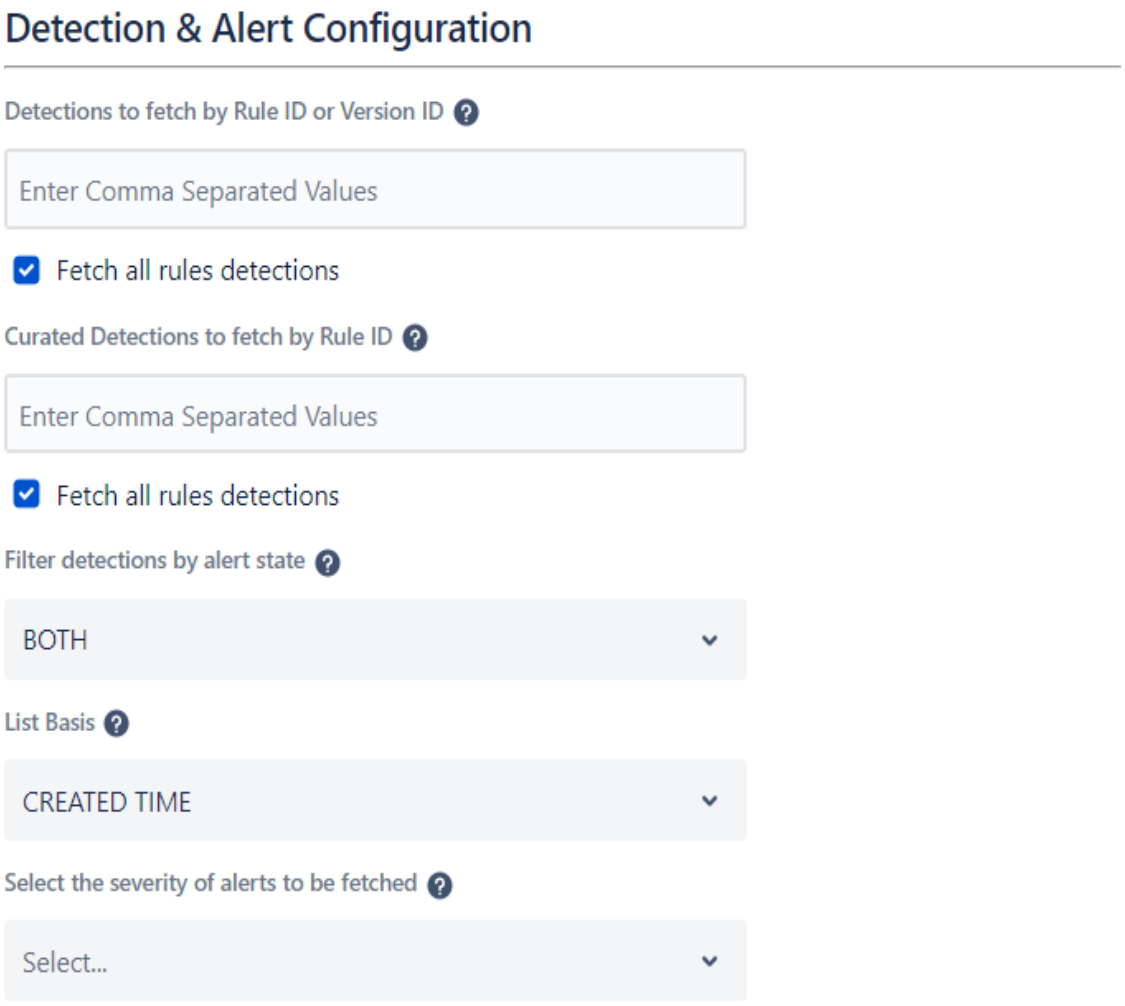
created per invocation. The default limit for IoC and Curated Detection is 10000 and for Alerts is 100000.

8.1 - Two types of detections can be fetched from the Chronicle, one with alert_state as ALERTING and another one with NOT_ALERTING. By default, the app retrieves both detections. However, this can be changed in the filters panel, as explained in the 11th point below.

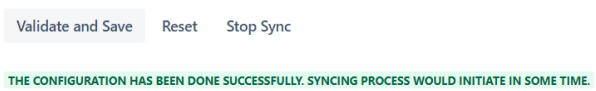


9. In the *Detections to fetch by Rule ID or Version ID* field user can provide comma-separated ruleID and versionID and if it is not provided, by default detection of all ruleIDs and versionIDs will be collected.
10. In the *Curated Detections to fetch by Rule ID* field user can provide comma-separated ruleID and if it is not provided, by default detection of all ruleIDs and versionIDs will be collected.
11. By default the Curated Detections would be disabled. The user needs to enable it to start getting Curated Detections.
12. In the *Filter detections by alert state* field, the user can filter detections based on alert state. By default, it will be filtered by both states.

13. In the *List Basis*, the user can select which sort type the detections will be fetched. By default, it will be Created Time.
14. In the *Select the severity of alerts to be fetched* field user can select which severity of alerts issue will be created. Multiple severities can be provided. By default, issues will be created for all alert severities.

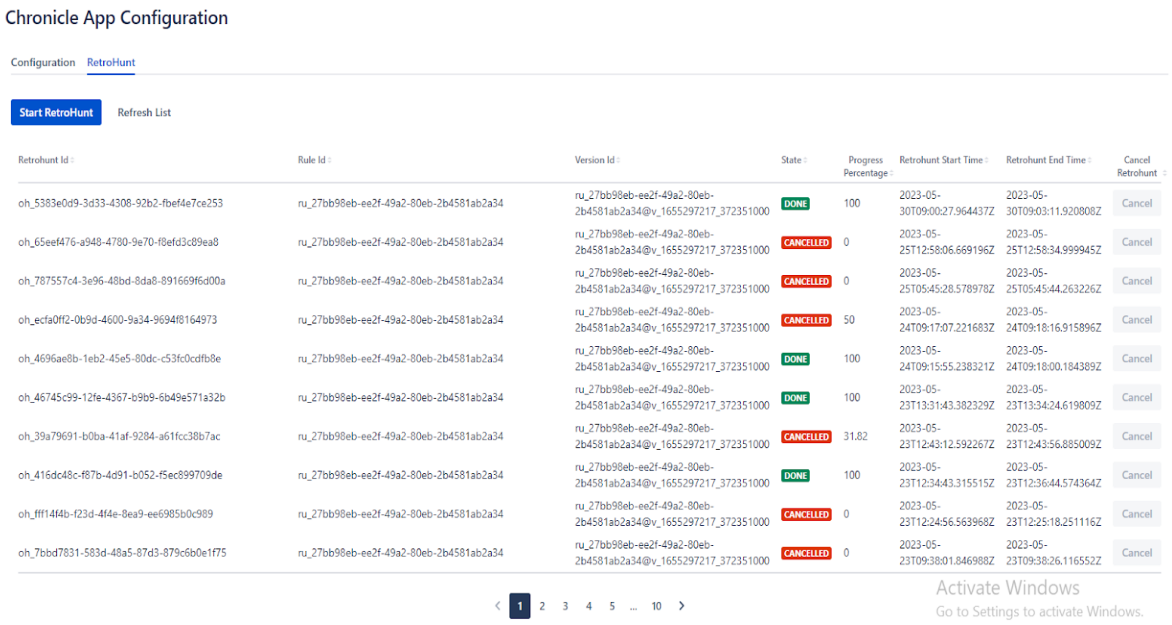


15. After configuring all these fields, the user needs to click the *Validate and Save* button to save the configuration. On successful authentication, it would show a message as seen in the below image.

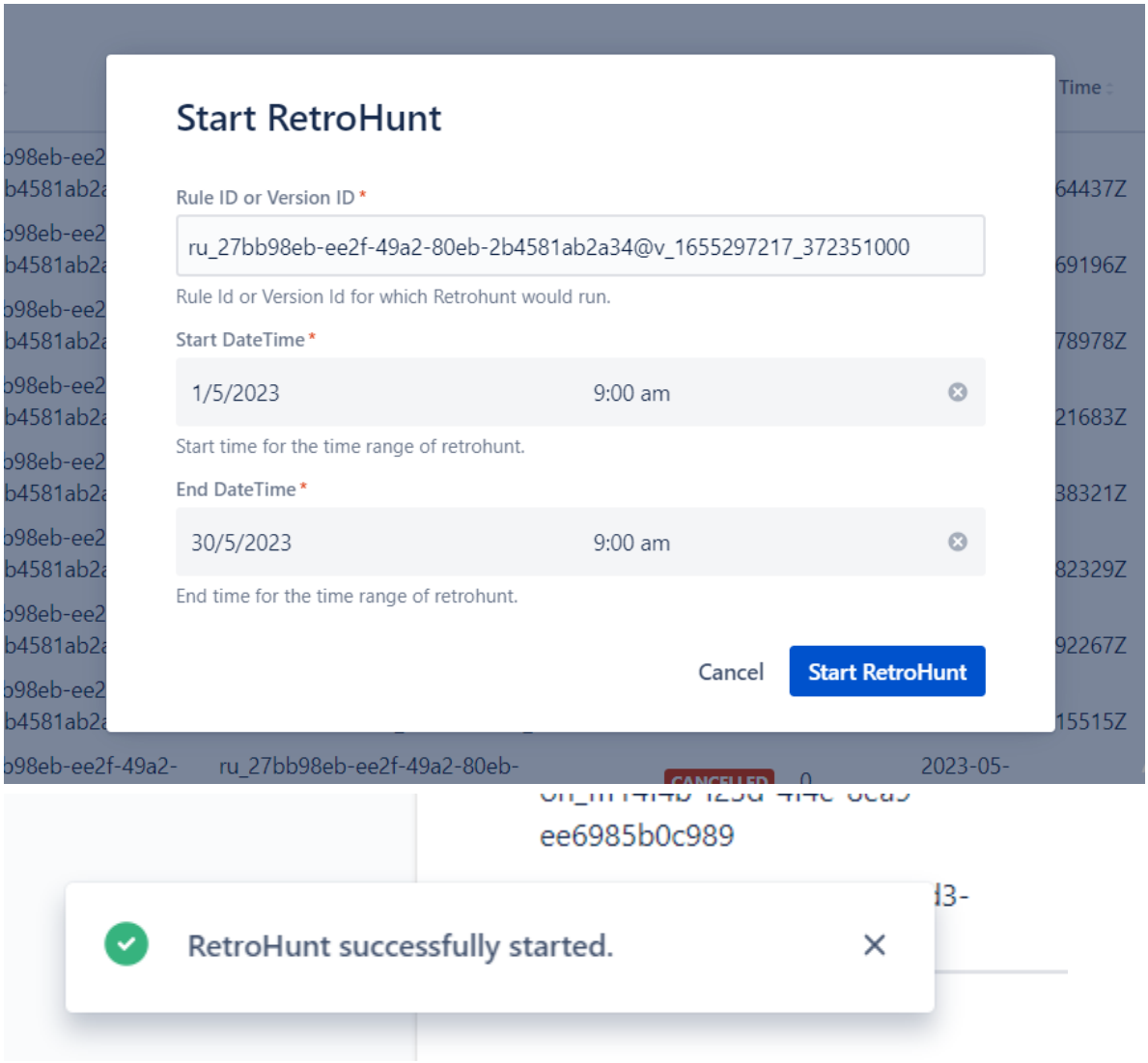


16. In case of a failed authentication, it would show a message as seen in the below image.

17. Reset Button: When the *Reset* button is clicked it will clear all the previously saved configuration parameters and checkpoints.
18. Stop Sync: When the *Stop Sync* button is clicked, further scheduler runs will be stopped.
19. In the Retrohunt tab, all previously started Retrohunts will be listed. There would be a cancel button for each Retrohunt to cancel that particular Retrohunt. The cancel button would be clickable only for Retrohunt those who are in the “RUNNING” state.



20. By clicking on Start Retrohunt, a dialogue box will open and the User needs to provide the RuleID/VersionID, StartTime, and EndTime details. After that User needs to click on Start RetroHunt. On the successful start of Retrohunt, the success message would be visible.



21. The Users can click on the Refresh List button to see the latest List of the Retrohunts.

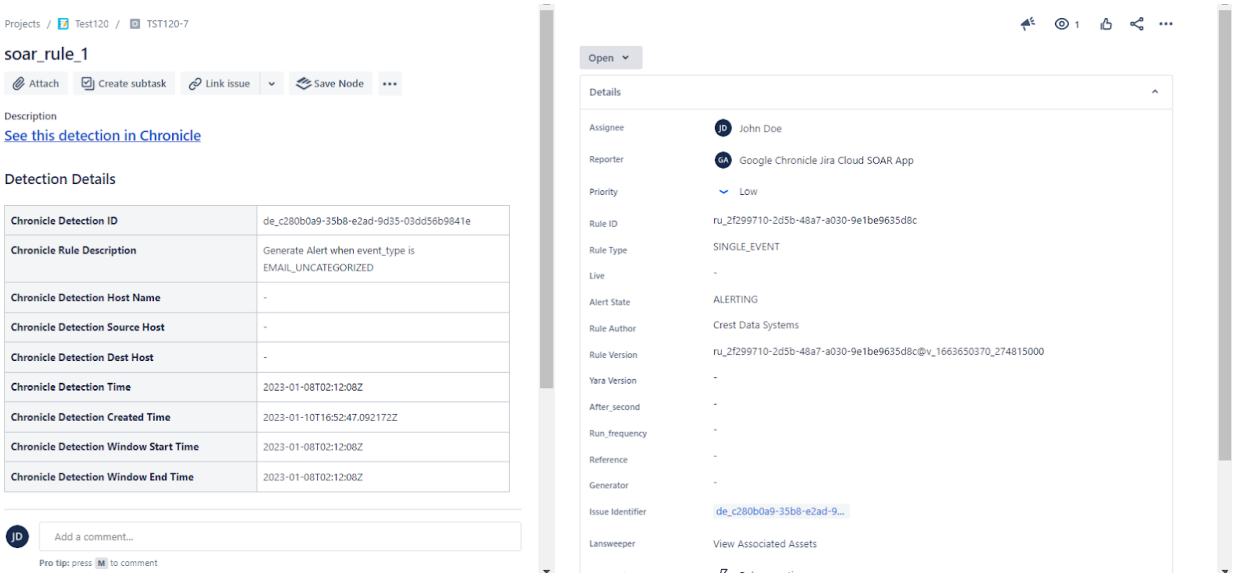
Configuration Page Field Information

Field Name	Description
User's Service Account JSON	Enter service account JSON file contents. Steps to get service account JSON file.
Select Region	Select the region based on the location of the chronicle backstory instance. Options: General, Europe, Europe West, Asia
Run	Select the interval at which the scheduler should run. Options: Hourly, Daily, Weekly
Number of days to fetch IoCs/Alerts/Detections initially	For how many days of historical data to be fetched. Max: 31 days
Select Project	Select the project in which Jira issues will be created. This field restricts Team-managed projects.
Default Assignee	Select to whom the created issues will be assigned by default. Options: unassigned, administrator, automatically
Enable IoC Matches	Whether to poll IoC Matches data or not
Enable Alerts	Whether to poll Alerts data or not
Enable Detections	Whether to poll Detections data or not
Detections to fetch by Rule ID or Version ID	Fetches detection by either Rule ID (format: ru_{UUID}) or Version ID (format: {ruleId}@v_{int64}_{int64}. Enter in comma-separated format to add multiple. Entered rules have precedence over the 'Fetch all rules detections' checkbox.
Fetch all rules detections	The detections of all rules and versions will be fetched.
Filter detections by alert state	Select the alert state to filter the detections to be fetched using fetch incidents. Available options are 'ALERTING' and 'NOT_ALERTING'. By default 'BOTH' option will be selected.
List Basis	Sort detections by 'DETECTION_TIME' or by 'CREATED_TIME'. If not specified, it defaults to 'CREATED_TIME'. This configuration is applicable to 'Detection alerts' only.

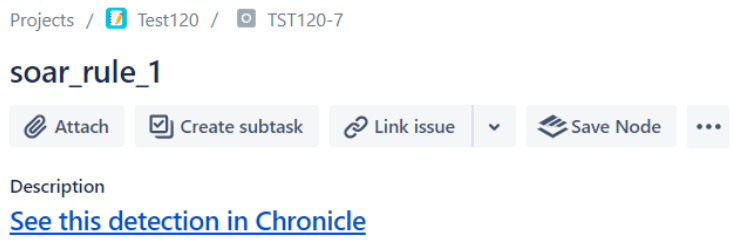
Select the severity of alerts to be fetched	Select the severity of alerts to be filtered for Fetch Incidents. Available options are 'High', 'Medium', 'Low', and 'Unspecified' (If not selected, fetches all alerts).
Validate and Save	This button will authenticate and validate all the configurations entered by the user. On successful authentication and validation, the configuration will be stored in the Forge storage.
Reset	This button will reset all the configurations and remove any old checkpoints stored.
Stop Sync	This button will stop any further scheduler runs.

Issue Creation

1. Once the Configurations are validated and saved successfully, the issue creation process will be initiated.
2. The issues will get created in the configured project and syncing will occur at user-configured intervals.

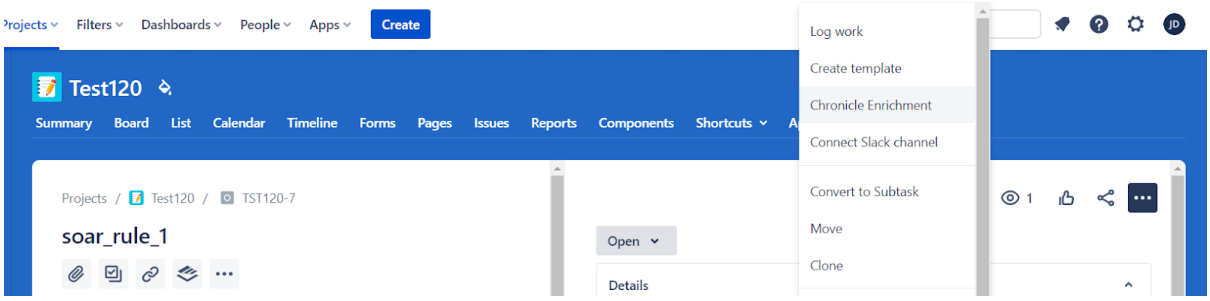


3. All the details related to the corresponding Detection/Alert/IOC will be added in the description along with the Event Discovered.
4. In the Details panel, the user would be able to see the custom fields which have the Chronicle information.
5. In the Description, a link is provided to see the Detection/Alert/Ioc/CuratedDetections in Chronicle. When the user clicks on the provided link, it navigates to the Chronicle platform where details of the detection can be seen.

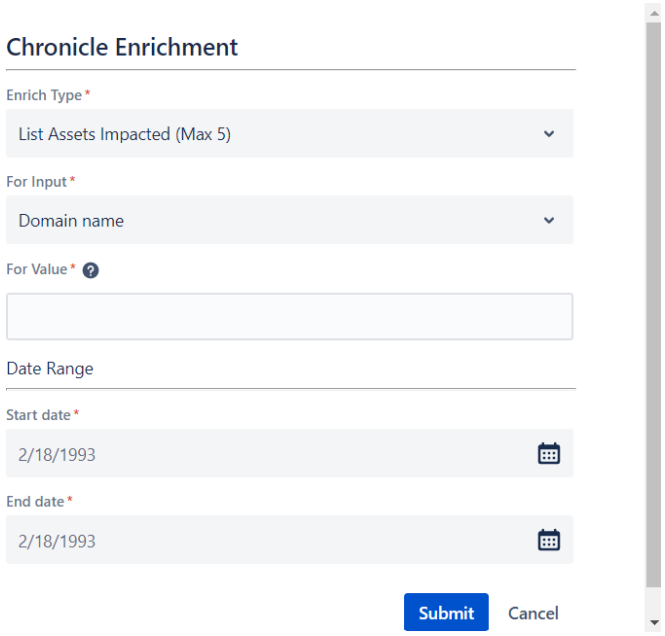


Issue Enrichment

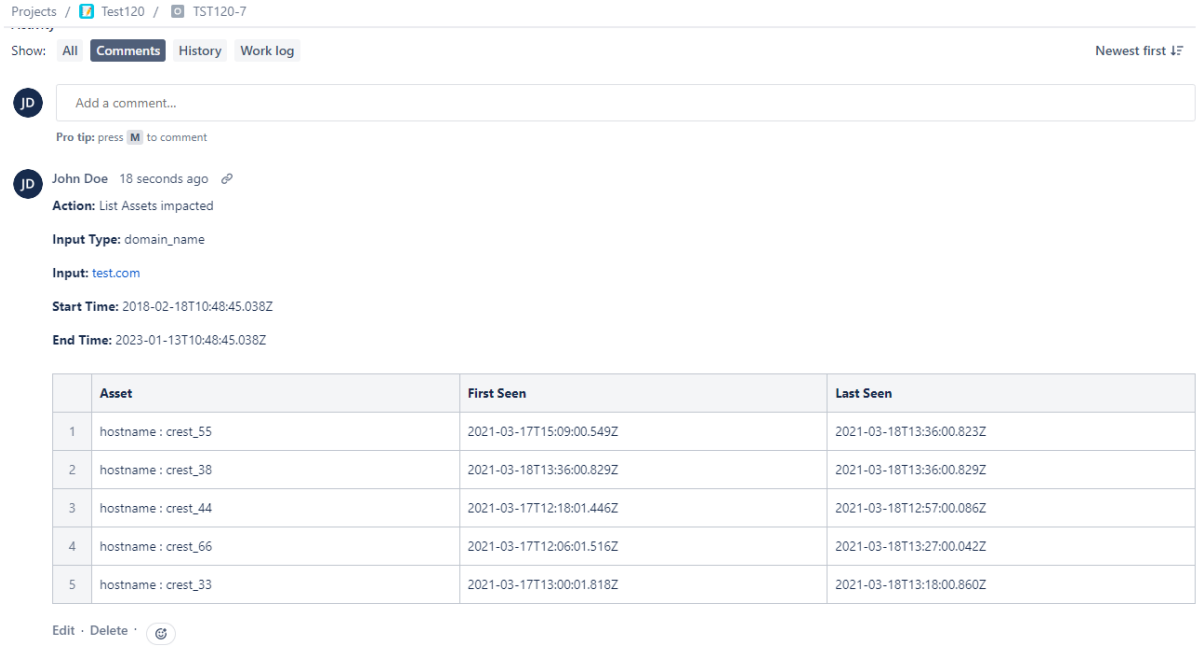
1. If the user opens any of the issues created by the app and navigates to the top right corner, the user will be able to see a Chronicle Enrichment option.



2. On clicking the Chronicle Enrichment option, it would open a panel where the user needs to configure the data with which they want to enrich the Jira issue

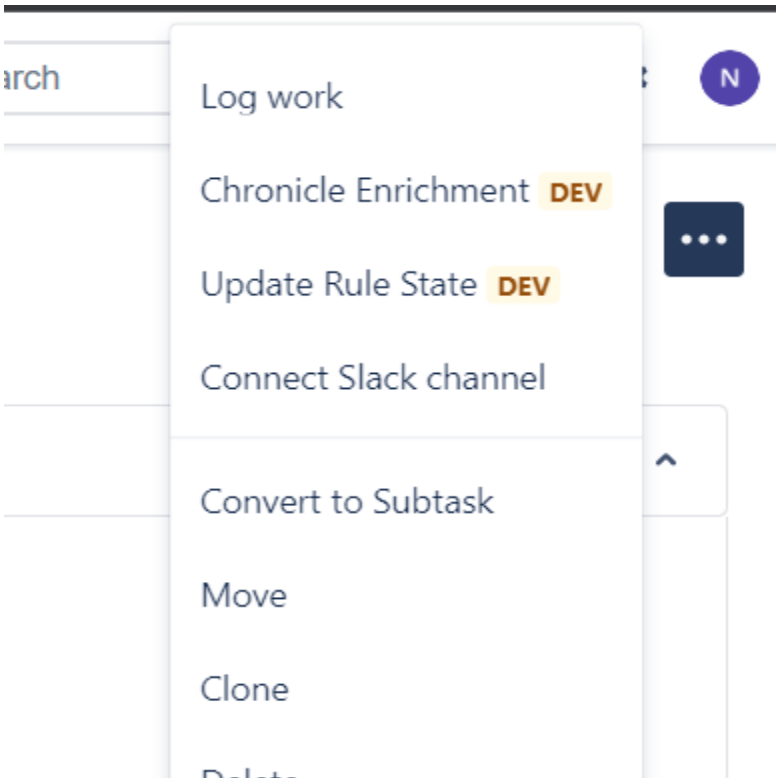


3. Once the details are added and clicked on the submit button, it would add the details in the comment section of the issue.

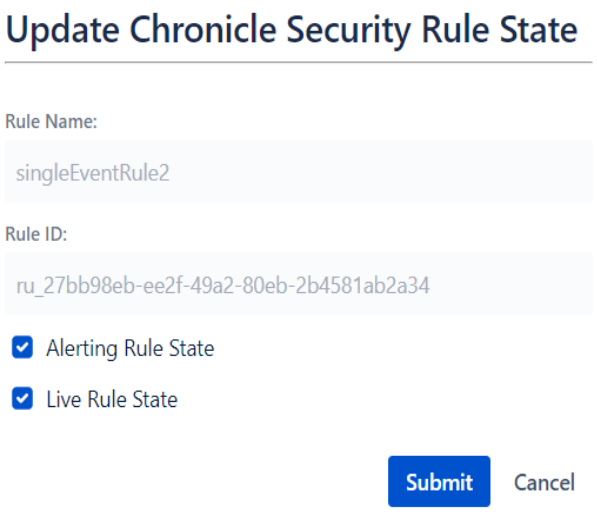


	Asset	First Seen	Last Seen
1	hostname : crest_55	2021-03-17T15:09:00.549Z	2021-03-18T13:36:00.823Z
2	hostname : crest_38	2021-03-18T13:36:00.829Z	2021-03-18T13:36:00.829Z
3	hostname : crest_44	2021-03-17T12:18:01.446Z	2021-03-18T12:57:00.086Z
4	hostname : crest_66	2021-03-17T12:06:01.516Z	2021-03-18T13:27:00.042Z
5	hostname : crest_33	2021-03-17T13:00:01.818Z	2021-03-18T13:18:00.860Z

4. For Update Rule State User needs to click on the 3 dots shown in the top right of the issue details screen.
5. Select the Update Rule State option from the menu.



6. A model will open containing the fields to be entered by the user.



7. User can enable or disable the Alerting Rule State and Live Rule State by checking and unchecking the checkbox. On submission, the App would update the Rule state on the chronicle side and the message would be populated in the comment section of the Jira issue for failure or success.

Chronicle Enrichment Field Information

Field Name	**Description**
Enrich Type	Select which type of data to be fetched. Options: List Assets Impacted (Max 5), List Events Discovered (Max 5), IoC Details
For Input	The options of this field are dependent on Enrich type: List Assets Impacted ● Domain Name ● IP Address ● HASH MD5 ● HASH SHA1 ● HASH SHA256 List Events Discovered ● Host Name ● IP Address ● MAC Address ● Product ID IoC Details ● Domain Name ● IP Address
For Value	Enter value based on selected For Input option. Multiple values are not supported.
Date Range	Select the start date and end date. This is only applicable to List Assets Impacted and List Events Impacted. Future dates are not allowed.

Third-Party Libraries Used

Library	Version	Github/Bitbucket	License
eslint	7.32.0	https://github.com/eslint/eslint	https://github.com/eslint/eslint/blob/v7.32.0/LICENSE
eslint-plugin-react-hooks	4.2.0	https://github.com/facebook/react	https://github.com/facebook/react/blob/main/LICENSE
husky	8.0.3	https://github.com/typicode/husky	https://github.com/typicode/husky/blob/main/LICENSE
atlaskit/button	16.4.0	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/LICENSE
atlaskit/css-reset	6.3.19	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/LICENSE
atlaskit/datetime-picker	12.3.0	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/LICENSE
atlaskit/form	8.7.0	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/LICENSE
atlaskit/icon-object	6.2.7	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/LICENSE
atlaskit/select	15.7.5	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/LICENSE
atlaskit/textfield	5.3.1	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/LICENSE
atlaskit/tooltip	17.6.1	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/LICENSE
atlaskit/checkbox	12.4.0	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/LICENSE

atlaskit/lozenge	11.3.1	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/LICENSE
atlaskit/modal-dialog	12.4.1	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/	https://bitbucket.org/atlassian/atlassian-frontend-mirror/src/master/LICENSE
react	16.13.1	github.com/facebook/react	https://github.com/facebook/react/blob/main/LICENSE
react-dom	16.13.1	github.com/facebook/react	https://github.com/facebook/react/blob/main/LICENSE

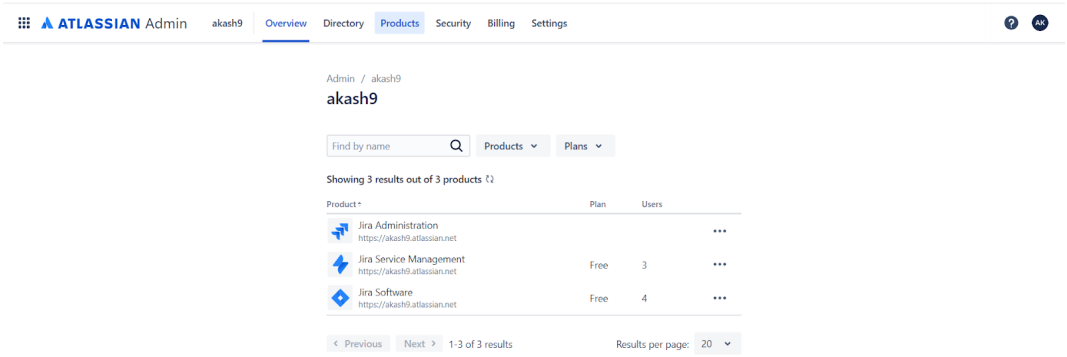
Known Behavior

1. After the successful configuration, the Atlassian Forge platform may take some time to initiate the scheduler.
2. In case the user resets the app configuration then all the data of the configured interval will be fetched again and respective issues will be created. This will result in the duplication of previously created issues.
3. In case during the scheduler run, a platform unexpected error is faced then the scheduler execution will be stopped and will continue from the next scheduler run based on the checkpoint saved before the error occurrence.
4. In case an error occurred during the issue creation process, the app will retry the issue creation process of the failed records a maximum of 4 times. If after 4 times still the failure occurs then proper logs will be added for those failures.

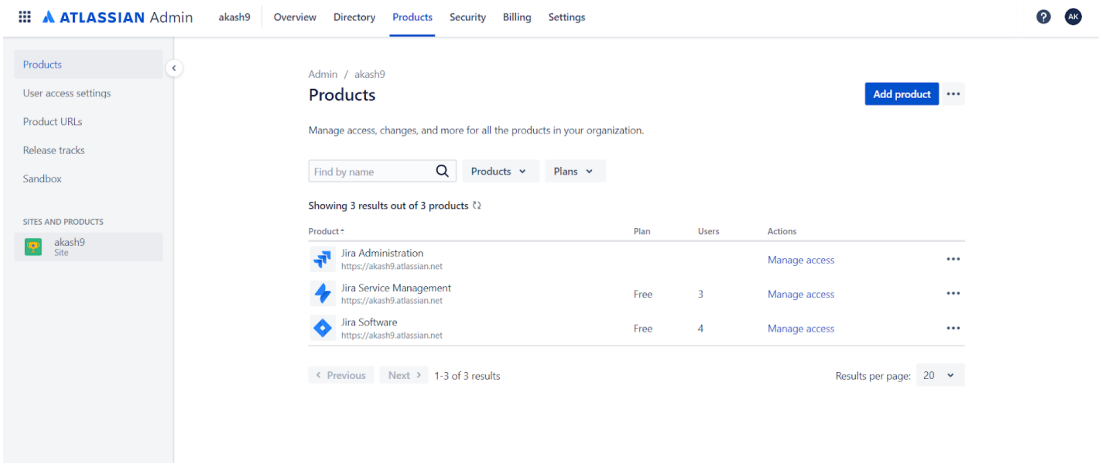
Troubleshooting

1. To see the application logs, follow the below steps. It would require the role of a system administrator.

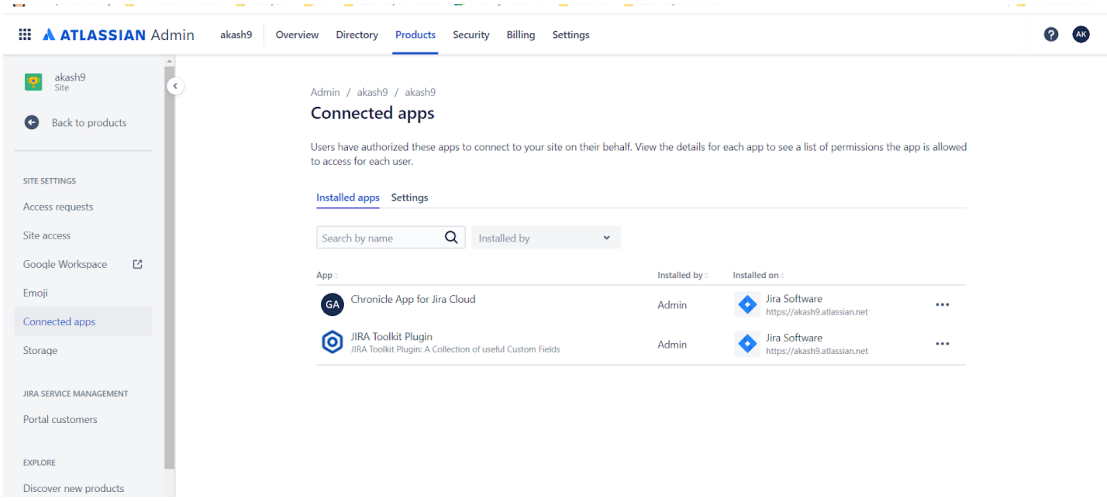
    a. Go to https://admin.atlassian.com/.
    b. Click on Products.



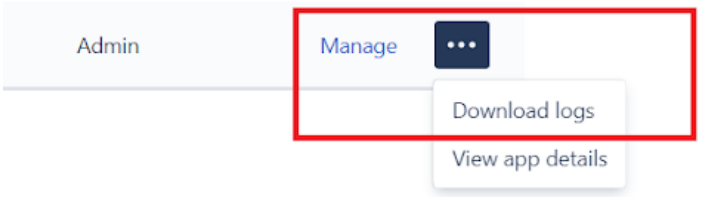
c. Click on SITES AND PRODUCTS.



d. Navigate to “Connected Apps”.



e. Click on 3 dots and click on Download logs.



2. Manage users, groups, permissions, and roles in Jira Cloud
    a. To manage users, groups, permissions, and roles in Jira Cloud review the following link and execute the steps

https://support.atlassian.com/jira-cloud-administration/docs/manage-users-groups-permissions-and-roles-in-jira-cloud/

3. Unable to install/activate the app on Jira Cloud
    a. If any issue is faced during installation/activation of the app on the Jira Cloud, review the following link and execute the steps. https://confluence.atlassian.com/upm/installing-marketplace-apps-273875715.html

4. Issue encountered in Jira Issue creation
    a. If an error occurs during authentication, in the creation of issues based on the user's configured details, or in the creation of new issues after resetting/updating the configured fields, the user can check the error message and reset the data on the Configuration page. Make sure to reset cautiously, as it might result in duplication of issues.