

1. Navigate to **Settings > Notifications > Event Log. Integration: Slack** is listed in the **Actions** column for the event. If the notification was successful, the integration label is green.
2. Hover over the integration to verify that an event was sent.

Additional resources

- » For more information about behavior groups, see [Configuring Hybrid Cloud Console notification behavior groups](#).

CHAPTER 4. INTEGRATING EVENT-DRIVEN ANSIBLE WITH THE RED HAT HYBRID CLOUD CONSOLE

You can use Event-Driven Ansible to take advantage of Hybrid Cloud Console capabilities such as Red Hat Insights to continuously analyze your inventory for potential issues and recommendations. Event-Driven Ansible connects sources of events with corresponding actions through rules.

The integration with the Hybrid Cloud Console notifications service uses a webhook to trigger actions. Each account configures how and who can receive these events, with the ability to perform actions depending on the event type.

Contacting support

If you have any issues with the Hybrid Cloud Console integration with Event-Driven Ansible, contact Red Hat for support. You can open a Red Hat support case directly from the Hybrid Cloud Console by clicking Help (? icon) > **Open a support case**, or view more options from ? > **Support options**.

4.1. CONFIGURING EVENT-DRIVEN ANSIBLE FOR INTEGRATION WITH THE RED HAT HYBRID CLOUD CONSOLE

You can configure Event-Driven Ansible to receive event notifications from the Red Hat Hybrid Cloud Console or a third-party application.

Prerequisites

- » The **ansible-rulebook** CLI is installed. See [Ansible rulebook installation](#) for instructions.
- » You have Organization Administrator or Notifications administrator permissions for the Hybrid Cloud Console.

Procedure

1. To install the Red Hat Insights collection for Event-Driven Ansible, enter the following commands:

```
$ ansible-galaxy collection install redhat.insights_eda servicenow.itsm
$ pip3 install -r
~/.ansible/collections/ansible_collections/redhat/insights_eda/requirements.txt
```

2. Enter the following command and verify that **redhat.insights_eda** is included in the output:

```
$ ansible-galaxy collection list
```

Sample output for this command:

```
# /home/username/.ansible/collections/ansible_collections
Collection          Version
-----
ansible.eda         1.3.8
community.general   7.0.1
redhat.insights_eda 1.0.0
servicenow.itsm     2.2.0
```

3. Create a simple Ansible rulebook using the Red Hat Insights collection. The rulebook exposes an endpoint (on port 5000 by default) that is used to receive events and communicate with the Hybrid Cloud Console. See the Event-Driven Ansible for Red Hat Insights documentation in [Ansible Automation Hub](#) for examples for the Red Hat Insights Advisor, compliance, and vulnerability services.
4. Create an associated playbook based on the examples in the [Ansible Automation Hub](#) on the Hybrid Cloud Console.
5. In the Hybrid Cloud Console, navigate to **Settings > Integrations**.
6. Select the **Reporting & Automation** tab.
7. Click **Add integration**.
8. Select **Event-Driven Ansible** as the integration type, and click **Next**.
9. In the **Integration name** field, enter a name for your integration (for example, *ansible*).
10. Paste the URL that you copied from your Ansible rulebook into the **Endpoint URL** field. This is the endpoint URL that points to the controller running the Event-Driven Ansible receiver.

Note

The **Endpoint URL** must include **http://** or **https://**, for example https://<eda_hostname>:5000/endpoint.

11. Optional: Provide a **Secret token** if one is configured.

Note

A secret token is essential for protecting the data sent to the integration endpoint and should always be used when integrating the Hybrid Cloud Console with third-party applications.

12. Click **Next**.

13. Review the integration details, and click **Submit** to enable the integration.

Additional resources

- » For more Event-Driven Ansible configuration information, see the documentation in [Event-Driven Ansible for Red Hat Insights](#) on the Red Hat Hybrid Cloud Console.
- » For more information about setting up Notifications administrator permissions, see [Configure user access](#) in *Configuring notifications on the Red Hat Hybrid Cloud Console*.

4.2. CREATING THE BEHAVIOR GROUP FOR THE EVENT-DRIVEN ANSIBLE INTEGRATION

A behavior group defines which notifications will be sent to external services such as Event-Driven Ansible when a specific event is received by the notifications service. You can link events from any Red Hat Hybrid Cloud Console service to your behavior group.

Prerequisites

- » You are logged in to the Hybrid Cloud Console as an Organization Administrator or as a user with Notifications administrator permissions.
- » The Event-Driven Ansible integration with the Hybrid Cloud Console is configured. See [Section 4.1, "Configuring Event-Driven Ansible for integration with the Red Hat Hybrid Cloud Console"](#) for information about configuring Event-Driven Ansible integration.

Procedure

1. In the Hybrid Cloud Console, navigate to **Settings > Notifications**.
2. Under **Notifications**, select **Configure Events**.
3. Select the application bundle tab you want to configure event notification behavior for: **Red Hat Enterprise Linux, Console**, or **OpenShift**.
4. Click the **Behavior Groups** tab.
5. Click **Create new group** to open the **Create behavior group** wizard.
6. Type a name for the behavior group, and click **Next**.
7. In the **Actions and Recipients** step, select **Integration: Event-Driven Ansible** from the **Actions** drop-down list.
8. From the **Recipient** drop-down list, select the name of the integration you created (for example, *console-teams*) and click **Next**.
9. In the **Associate event types** step, select one or more events for which you want to send notifications (for example, **Advisor: New recommendation**), and click **Next**.
10. Review your behavior group settings, and click **Finish**. The new behavior group appears on the **Notifications** page.

Verification

- 1. Create an event that will trigger a Hybrid Cloud Console notification. See the [Ansible Automation Hub page](#) for an example.
- 2. To test that the Hybrid Cloud Console integration was successful, in the Hybrid Cloud Console, go to **Settings > Notifications > Event Log** and check for an event that shows the label **Integration: Event-Driven Ansible**.

Event	Application	Actions ?	Date ... ↓
Detected Malware	Malware - Red Hat Enterprise Linux	<div><div>✓ Integration: Event-Driven Ansible</div><div>✓ Integration: ServiceNow</div><div>✓ Integration: Splunk</div><div>✓ Email</div><div>✓ Integration: Webhook</div><div>Show Less</div></div>	3 minutes ago

- » If the label is green, the notification succeeded.
- » If the label is red, verify that the webhook has been properly created and exposed in Event-Driven Ansible, and that the correct webhook URL is configured in the Hybrid Cloud Console integration configuration.

Note

See [Troubleshooting notification failures with the event log and integration settings](#) in the notifications documentation for more details.

4.3. ADDITIONAL RESOURCES

- » For more information about Event Driven Ansible, see the [Getting Started with Event-Driven Ansible Guide](#).
- » For information about how to configure and use the Event-Driven Ansible controller, see [Event-Driven Ansible controller user guide](#).
- » For more information about behavior groups, see [Configuring Hybrid Cloud Console notification behavior groups](#).
- » For configuration examples, see [Red Hat Insights Collection for Event-Driven Ansible](#) and [Using Red Hat Insights as a source of events for Event-Driven Ansible automation](#) on the Red Hat Ansible Automation Platform website.

CHAPTER 5. INTEGRATING MICROSOFT TEAMS WITH THE HYBRID CLOUD CONSOLE

You can configure the Red Hat Hybrid Cloud Console to send event notifications to all users on a new or existing channel in Microsoft Teams. The Microsoft Teams integration supports events from all services in the Hybrid Cloud Console. The Microsoft Teams integration uses incoming webhooks to receive event data.

Contacting support