Lab Guide

ServiceWatch Insight:

Integrating Events and Rules

Vinh Tran & Tony Branton

Default Login / Password: Knowledge17

admin / Knowledge17

itil / Knowledge17

employee / Knowledge17

Lab Sponsored by Amazon Web Services



This
Page
Intentionally
Left
Blank

Lab Goal

Before starting the lab, you will need to complete the following prerequisite tasks.

Lab Prerequisites

Step 1: Validate the MID Server

1. Log in to your ServiceNow lab instance.
2. Navigate to **MID Server** > **Servers** and open your MID Server shown in the list view.
3. In the **Related Links** section, click on the **Validate** link. It will take about a minute for the MID Server **Status** to change to **Down**; return to **Up** and the **Validated** field displays **Yes**.
4. Record the **IP address** for your MID Server to use in Step 5:

   MID Server IP Address: _____

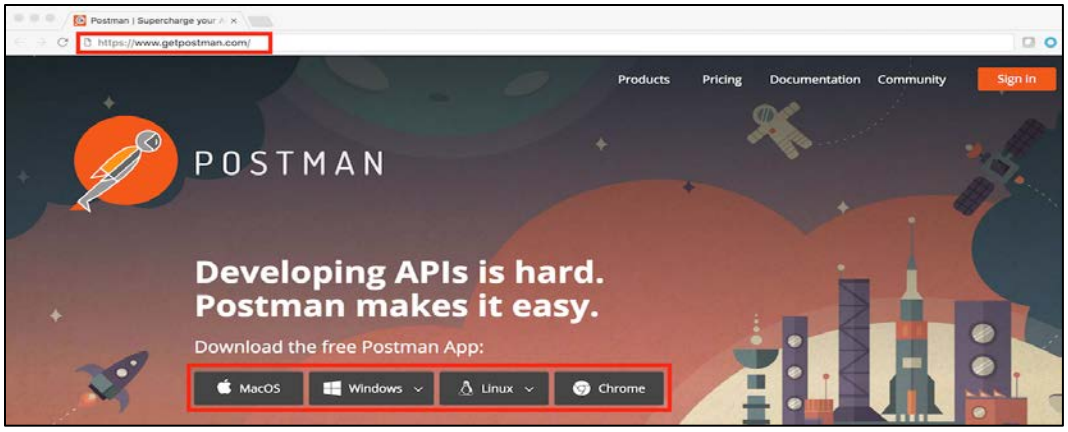
Step 2: Download MIB Files and Postman Collection

1. On your lab instance, navigate to **Knowledge > Articles > Published**.
2. Search for and open the **KB0131985** Knowledge article.
3. At the top of the page, click on **[view]** next to the **lab131985.zip** attachment. The zip file downloads to your computer.
4. On your computer, open the location the zip was downloaded to and extract the contents.

Step 3: Download and Install Postman

1. On your computer, open a new tab/window in a web browser and go to: https://www.getpostman.com.

2. Click the relevant app platform (MacOS, Windows or Linux).



3. Your download should begin automatically.

4. On your computer, go to the **Downloads** folder and open the downloaded zip file.

5. Install the **Postman** app to your preferred location.

6. Launch Postman and sign up for free (or sign in with your existing credentials).

Step 4: Set Up the Postman Collection and Environment

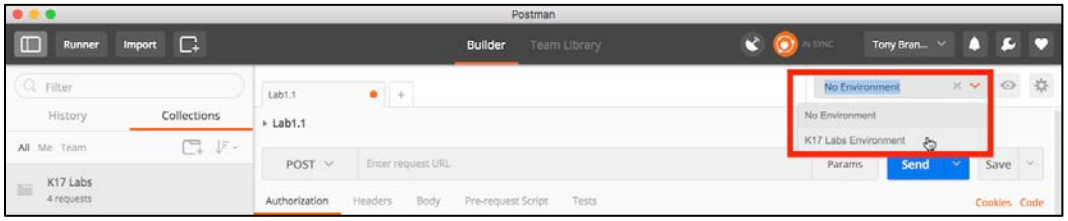
1. Click on the **Import** button in Postman.



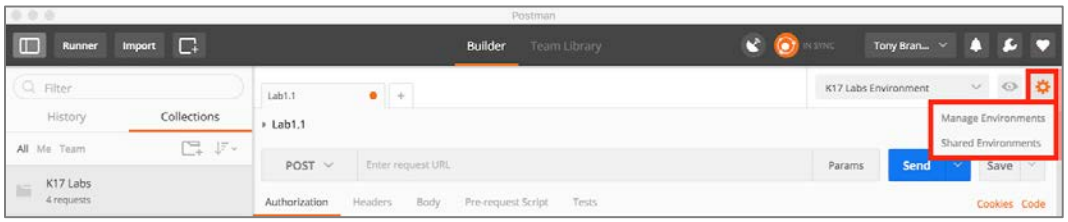
2. Click the **Choose Files** button and select the **K17 Labs.postman_collection** from the location the lab files were extracted to and import the collection.

3. Repeat the step to import the **K17 Labs Environment.postman_environment** collection.

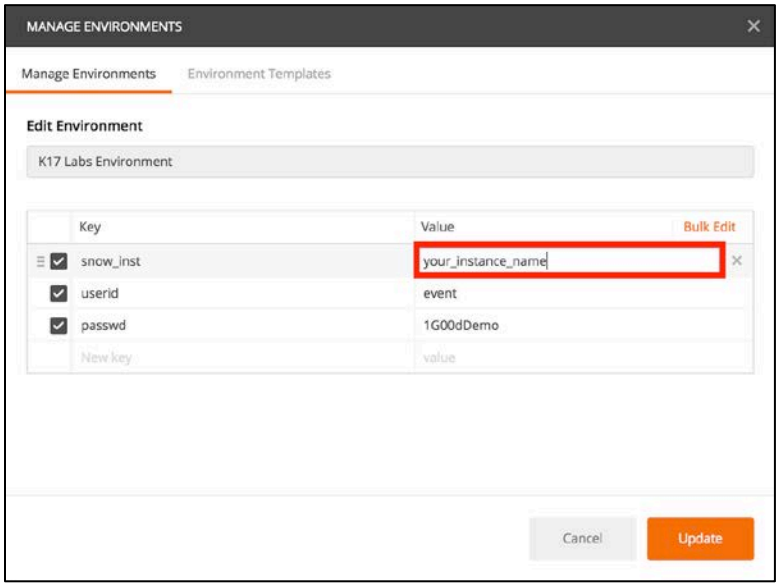
4. Choose **K17 Labs Environment** from the drop-down list at the top-right of the Postman window.



5. Click the **gear** icon at the top-right of the Postman window and select **Manage Environments** from the pop-up menu.



6. Click on the K17 Labs Environment entry in the list of environments and replace the value for the **snow_inst** Key with **your_instance_name** and click **Update**.



7. Close the Management Environments dialog box and leave Postman running.

Step 5: Download SNMP Lab Files to Your MID Server

1. Ensure you have **Remote Desktop Connection** (Windows) or **Microsoft Remote Desktop** (MacOS) on your computer.
2. Remote desktop to your MID Server using the IP address recorded in **Step 1**.
3. On your MID Server, open a browser (Google Chrome is pinned to the task bar) and login to your ServiceNow Instance.
4. Navigate to **Knowledge > Articles > Published**, search for and open the **KB0131985** knowledge article.
5. At the top of the page, click on **[view]** next to the **lab131985.zip** attachment. The zip file downloads to your computer.
6. On your computer, open the location the zip was downloaded to and extract the contents.
7. Copy the **Lab** folder to the **C:** drive on the **MID Server** (i.e., C:\Lab).

Lab Goal

This lab explains how to integrate ServiceNow with third-party monitoring tools into Event Management using REST and SNMP.

The lab contains the following sections:

- Lab 1.1 Integrate Events with REST
- Lab 1.2 Integrate Events via SNMP

Lab 1
Event
Integration

Lab 1 Goal

Lab 1.1 Integrate Events with the REST API

The ServiceNow REST API allows events to be sent and inserted as records in the [em_event] table. The REST API can be called from scripts (e.g., Python, Perl) or via utilities such as cURL and requires ServiceNow credentials to securely send events.

In this lab, you use Postman to send events to ServiceNow.

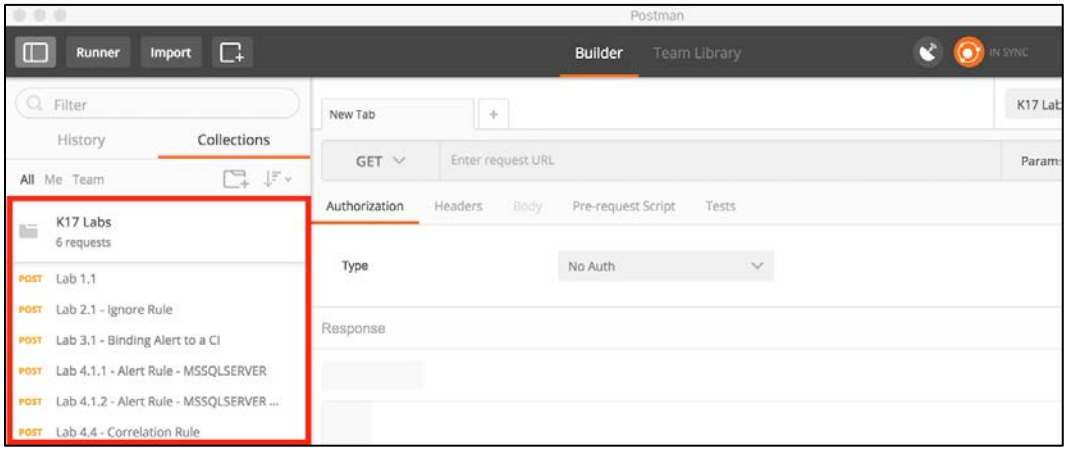
Step 1 – Create an Event Integration User

1. On your computer, launch a web browser, enter your lab instance URL, and log in as the **admin** user.
2. Navigate to **User Administration > Users** and click **New**.
3. Enter the following details:

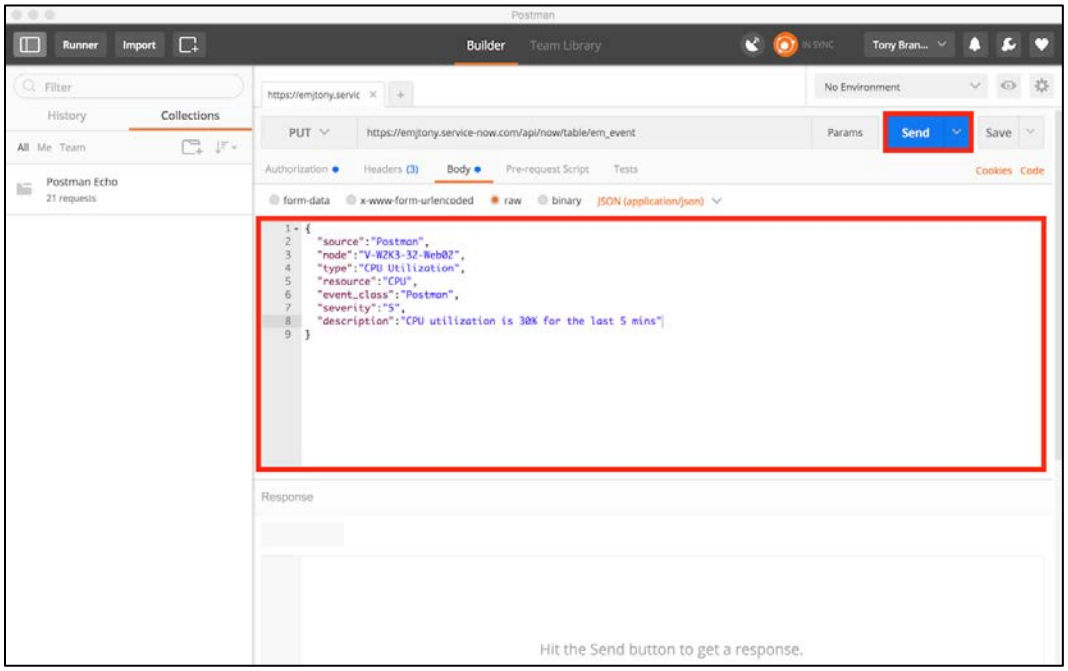
   Name: **event**
   First name: **Event**
   Last name: **Integration**
4. Enter a password of your choice.
5. Right-click on the form banner bar and select **Save**.
6. Scroll to the **Roles** section, click **Edit**, add the **evt_mgmt_integration** role. Click **Save**.

Step 2: Use Postman to Send an Event via the REST API

1. Switch to Postman and click the **K17 Labs** collection to expand and display a list of requests.



2. Select **Lab 1.1** and click on **Body** in Postman to display the event data to be sent via REST.

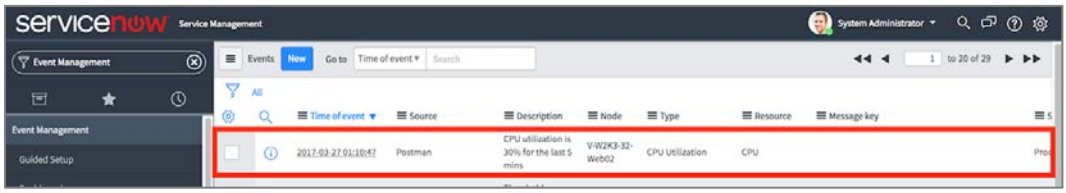


3. Click **Send**. Note the response received from the ServiceNow instance.

Check the Results

1. Switch to your ServiceNow lab instance and navigate to **Event Management > All Events**.

2. Verify that the Event you sent via Postman was received.



3. Open the Event.

4. Questions:

   What is the value in the **State** field?
   What do you see in the **Additional Information** field?
   Is there any information in the **Processing Notes** field

5. Click the **Information** (button next to the **Alert** field to open the Alert. Is the **Configuration Item** field populated?

Lab Goal

This lab explains the methods of integrating events from external event sources using SNMP Traps.

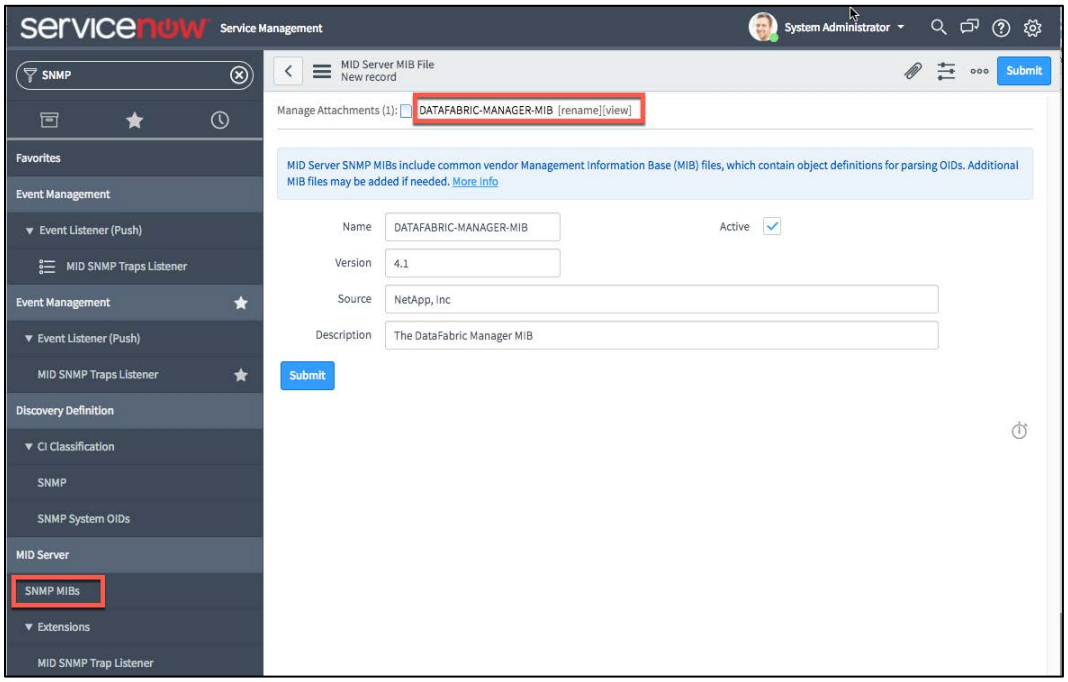
Lab 1.2 Integrate Events via SNMP

Step 1: Create SNMP MIB Records

1. Log in to your lab instance on your computer and navigate to **MID Server > SNMP MIB**.
2. Click **New** to create a new SNMP MIB record and enter the following:

   Name: **NETAPP-MIB**
   Version: **2.2.2**
   Source: **NetApp, Inc**
   Description: **The MIB module for NetApp SNMPv2 entities**
3. Attach the **NETAPP-MIB** file you downloaded in Lab Prerequisites Step 2.
4. Click **Submit** to save the new SNMP MIB record.

5. Repeat Steps 2-4 for the DATAFABRIC-MANAGER-MIB.



6. Navigate to **MID Server** > **Servers** and open the MID server record. In the UI Action, select **Restart**. Restarting the MID Server ensures the new MIBs are downloaded from the ServiceNow instance to the MID Server.

Step 2: Create an SNMP Trap Collector Extension for the MID Server

1. Navigate to **MID Server > Extensions > MID SNMP Trap Listener**.

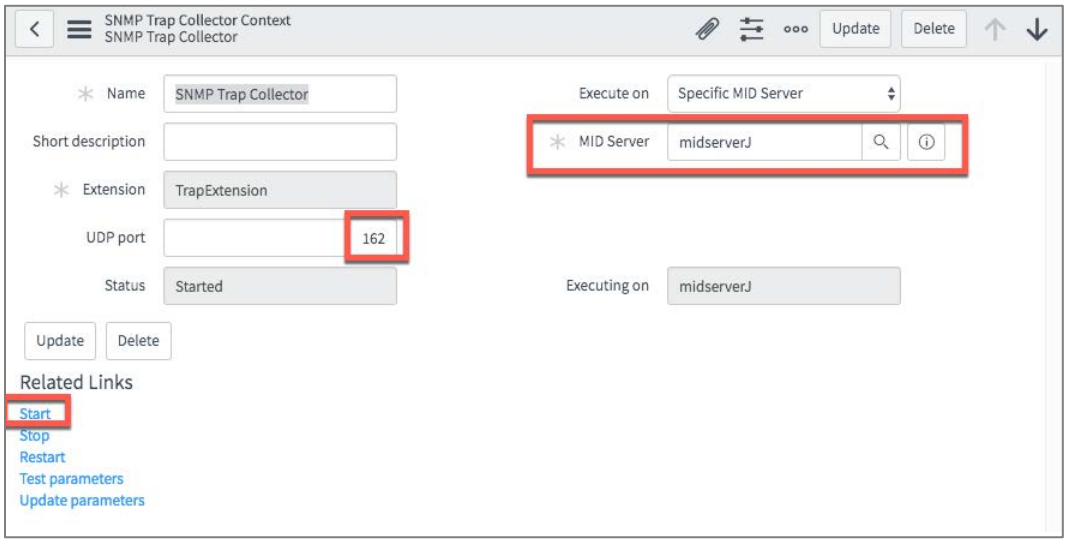
2. Click **New** and enter the following information:

   Name: **SNMP Trap Collector**
   UDP port: **162**
   Execute on: **Specific MID Server**
   MID Server: **< Your MID Server >**

3. Right-click on the top of the form (gray bar) and select **Save** from the pop-up menu.

4. Click the **Start** Related Link.

5. Reload the form a few times until you see the **Status** change to **Started**.



Step 3: Send a Test SNMP Trap

1. Remote desktop to your MID Server, open a command prompt and run the following command from the directory you downloaded the Lab files to (e.g., C:\Lab):

   `Lab1.2.bat`

Check the Result

1. Switch to your ServiceNow lab instance and navigate to **Event Management > All Events** to verify an Event was received with **Source** set to **Trap From Enterprise 789**.



2. Questions:

   What value is displayed in the **State** column for the Event?
   Was an Alert created for the Event?

Lab Goal

This lab explains how to configure rules to filter, transform and map events to produce meaningful alerts.

The lab contains the following sections:

- Lab 2.1: Filter Events
- Lab 2.2: Transform Events
- Lab 2.3: Event Field Mapping Rule
- Lab 2.4: Threshold Events

Lab 2 Event Processing Rules

Lab Goal

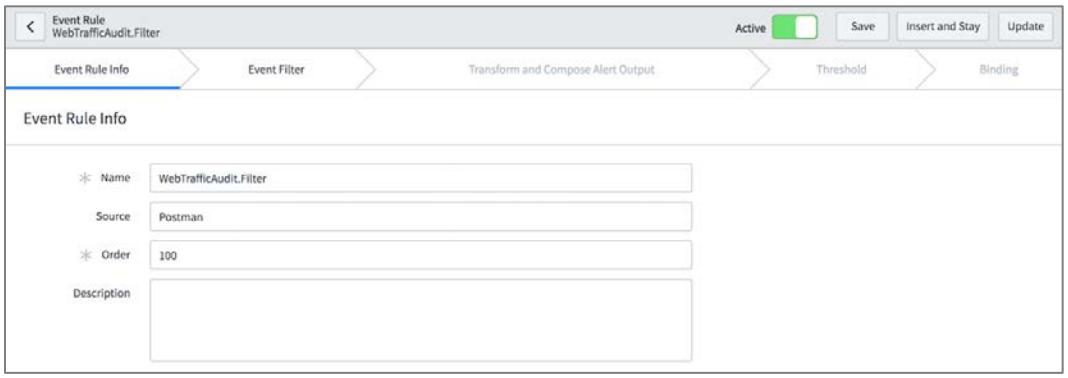
Lab 2.1
Filter Events

This lab explains how to configure an ignore rule to filter unwanted events to reduce alert noise.

1. Navigate to **Event Management > Rules > Event Rules** and click the **New** button.

2. Enter the following information in the Event Rule:

   Name: **WebTrafficAudit.Filter**
   Source: **Postman**

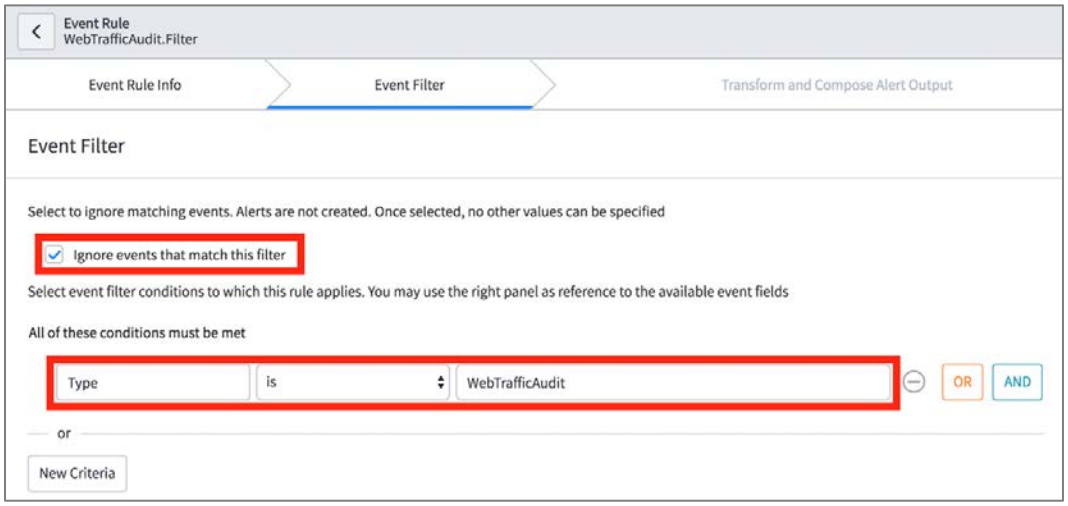


3. Click the **Event Filter** guide heading.

4. Click the **Ignore events that match this filter** check box. Note that the **Transform and Compose Alert Output** and **Threshold** guide headings are now disabled (grayed-out).

5. In the **All of these conditions must be met** section, define the following condition:

   Type is WebTrafficAudit



6. Click **Submit** to save the Event Rule.

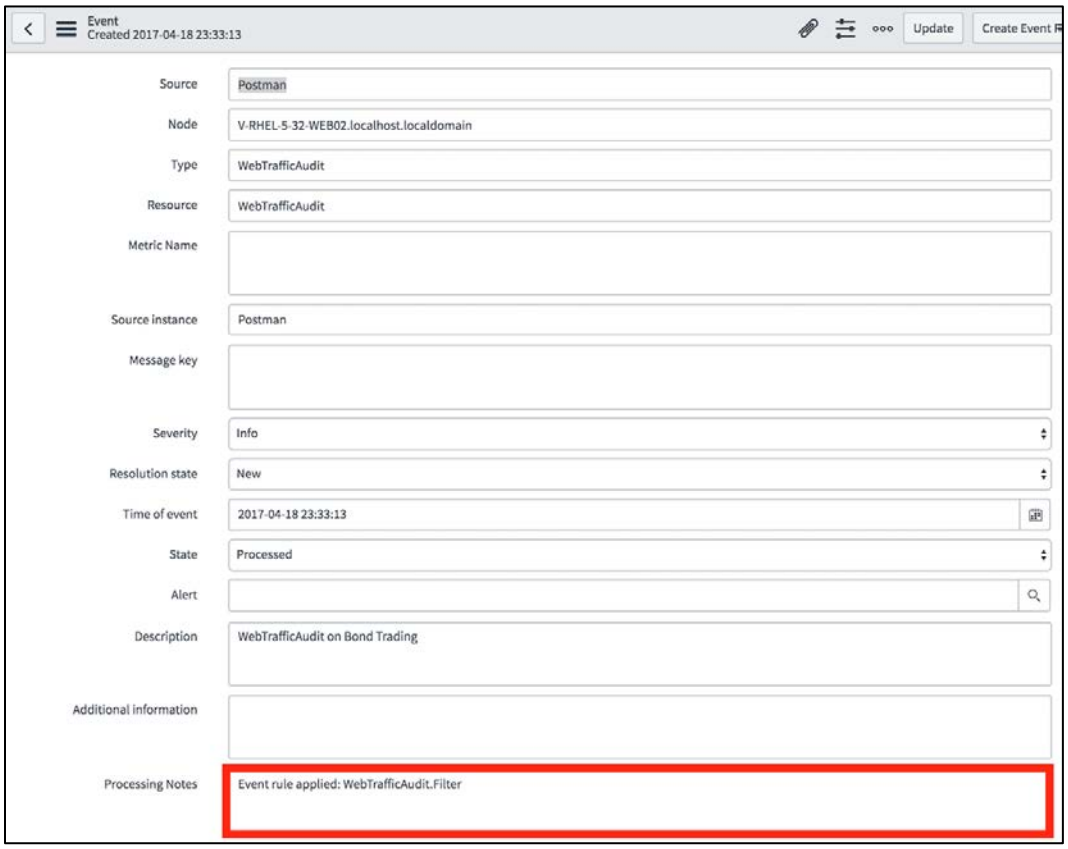
7. Switch to Postman, select **Lab 2.1 – Ignore Rule** from the K17 Labs folder and click **Send.**

Check the Result

1. Switch to your ServiceNow instance and open the **All Events** list view. You should see the WebTrafficAudit Event.

2. Open the Event and scroll to the **Processing Notes** field (if it is not displayed, add it to the form). You should see the following message:

   Event rule applied: WebTraffic.ignore.rule



3. Note that the **State** is **Processed**, not Error.

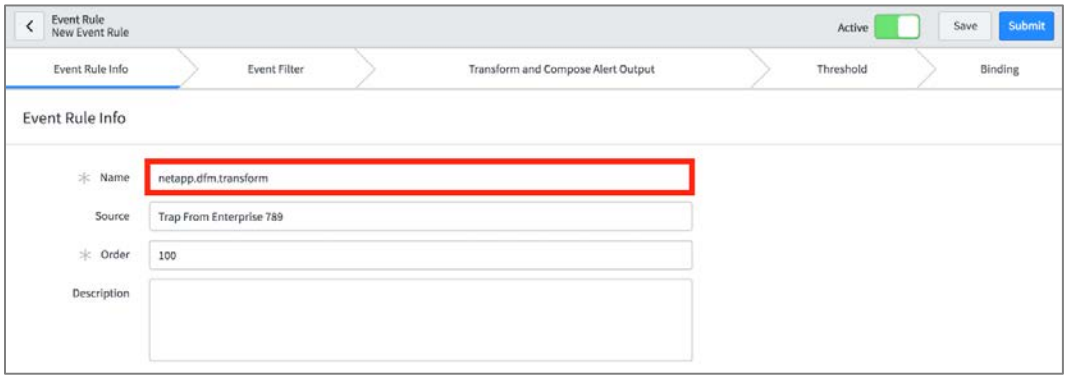
Lab Goal

Lab 2.2 Transform Events

This lab uses the SNMP trap event sent in Lab 1.2 to create an Event Rule that can be applied to transform the **Trap From Enterprise 789** class of events into alerts with a common format.

1. Navigate to **Event Management > All Events** and open the event created in Lab 1.2 whose **Source** field is set to **Trap From Enterprise 789**.
2. Click the **Create Event Rule** button at the top of the page to open the Event Rule designer.
3. In the **Event Rule Info** section, enter the following details:

   Name: **netapp.dfm.transform**

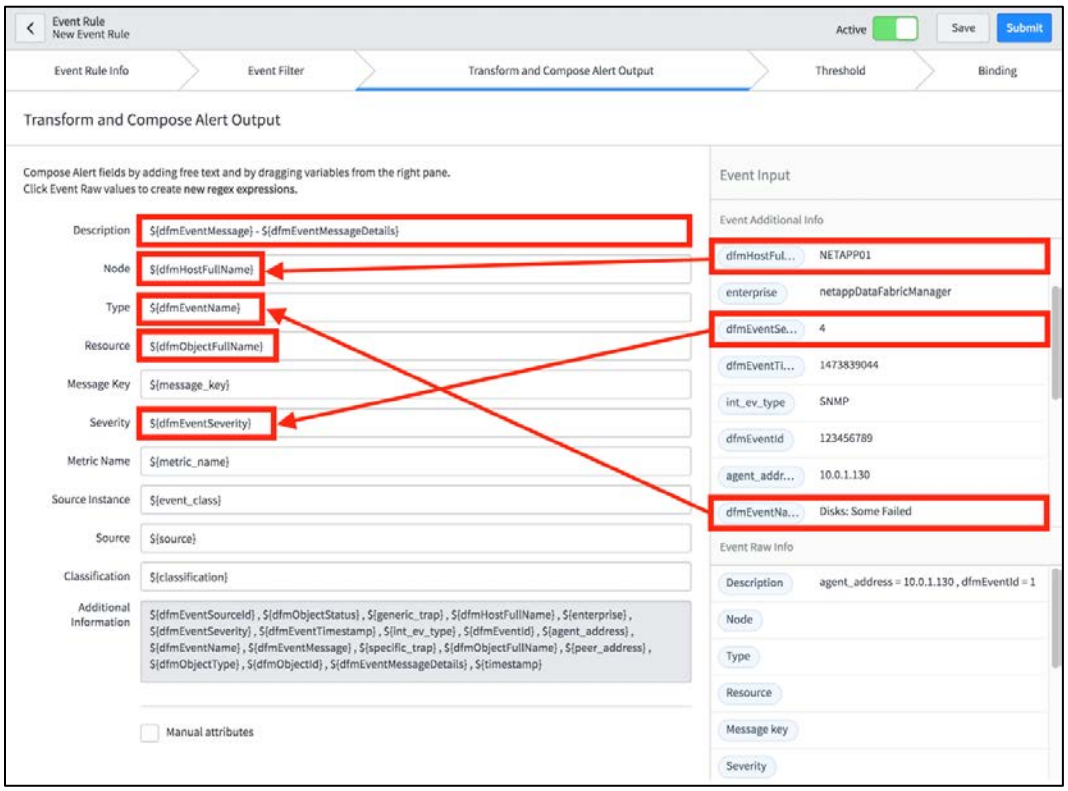


   Note that the Event Rule Designer automatically:

   Populated the Source field using data from the Event
   Set the Order to a default value of 100
   Made the Event Rule active (Active slider is set to green)

4. Click on the **Transform and Compose Alert Output** guide heading and drag the following fields from the **Event Additional Info** pane to compose the Alert fields below.

   Description: **${dfmEventMessage} - ${dfmEventMessageDetails}**
   Node: **${dfmHostFullName}**
   Type: **${dfmEventName}**
   Resource: **${dfmObjectFullName}**
   Severity: **${dfmEventSeverity}**



   Note that Event Rule Designer:
   Automatically parses the Event data into individual fields for you.
   Allows you to drag and drop both OOTB Event fields and data from the **Additional Info** field to compose Alert fields.

5. Click **Save**.

Lab Goal

Lab 2.3 Event Field Mapping Rule

Before you attempt to resend the SNMP trap ensure the Event severity value in the SNMP trap is mapped correctly to the list of values used in an Alert (i.e., a major severity in the trap matches a major severity in an Alert). In this scenario:

- NetApp (dfmEventSeverity) severity field values are: normal(1), information(2), warning(3), error(4), critical(5), emergency(6)
- Alert severity field values in ServiceNow are: 0 = Clear, 1 = Critical, 2=Major, 3 = Minor 4=Warning, 5=Informational, 0=Clear.

Use an Event Field Mapping rule to map the NetApp values to those used in an Alert.

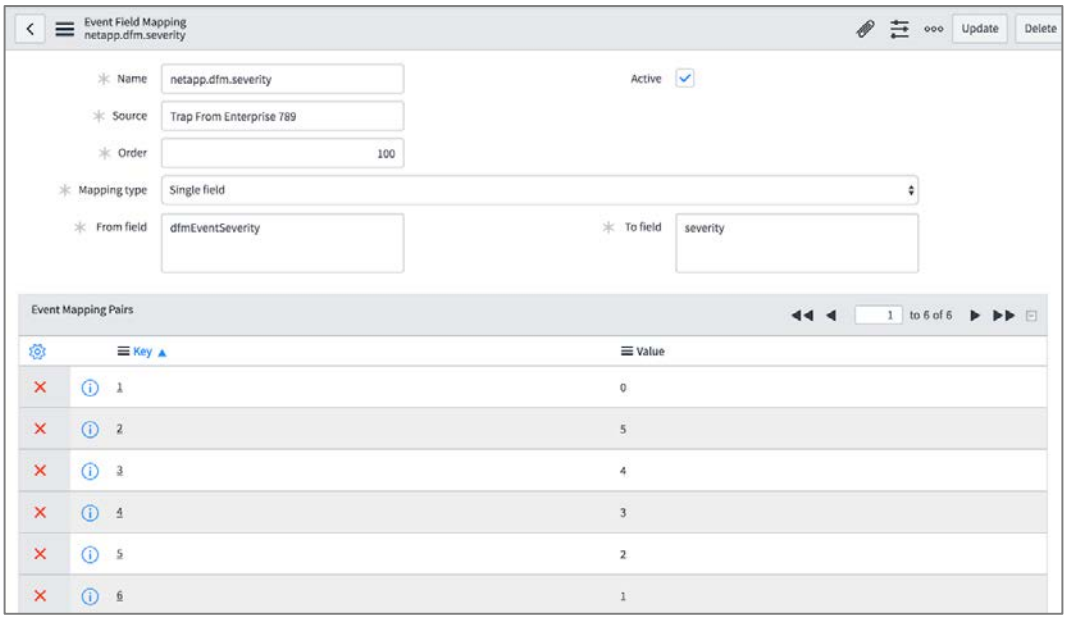
1. Navigate to **Event Management > Rules > Event Field Mapping** and click **New**.

2. Enter the following information for the Event Field Mapping rule:

   Name: **netapp.dfm.severity**
   Source: **Trap From Enterprise 789**
   Active: **True**
   Mapping type: **Single field**
   From field: **dfmEventSeverity**
   To field: **severity**

3. Complete the **Event Mapping Pairs** table as shown below.

Key	Value
1	0
2	5
3	4
4	3
5	2
6	1



4. Click **Submit** to save the Event Field Mapping Rule.

5. Test the rules by returning to your MID Server and from a command prompt running the following command from the lab files directory (e.g., C:\Lab):

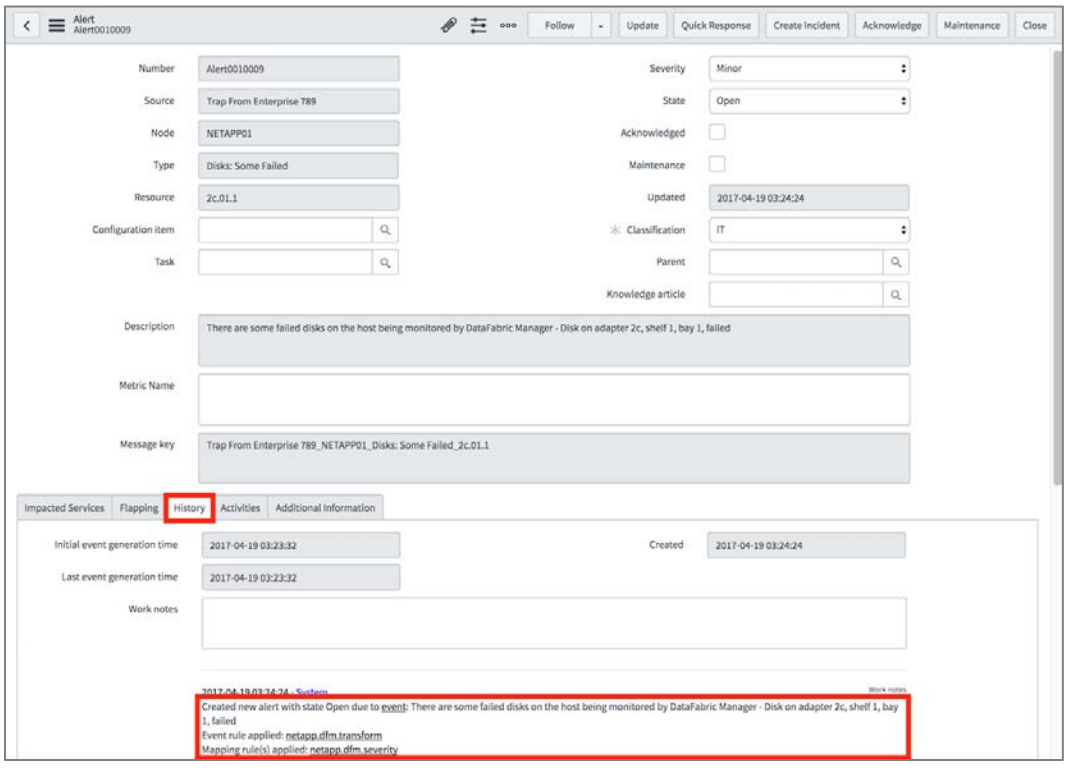
   `Lab1.2.bat`

Check the Result

1. Navigate to **Event Management > All Events** list view to verify that the Event has been successfully processed and an Alert has been created.



2. Open the Alert and click the **History** tab to see the rules applied to the Event.



Lab 2.4: Threshold Events

Lab 2.4 Threshold Events

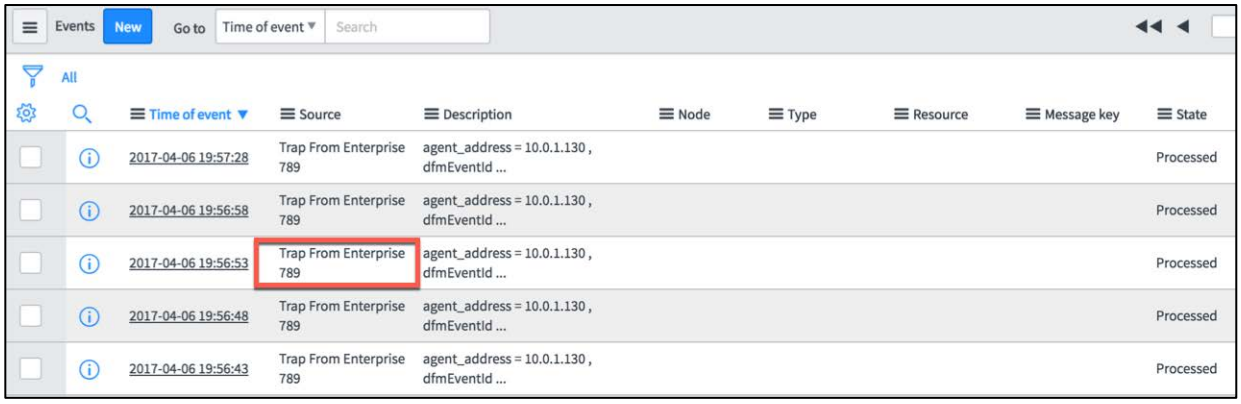
This lab shows how to create an alert when certain events breach a threshold. The lab explains how to create an alert for DFM events if it meets the following conditions:

- If the DFM event name is **Inodes Nearly Full**.
- More than 85% of inodes are used.
- At least two events should occur within a 2-minute period.

1. Return to the MID Server and from a command prompt run the following command from the lab files directory (e.g., C:\Lab).

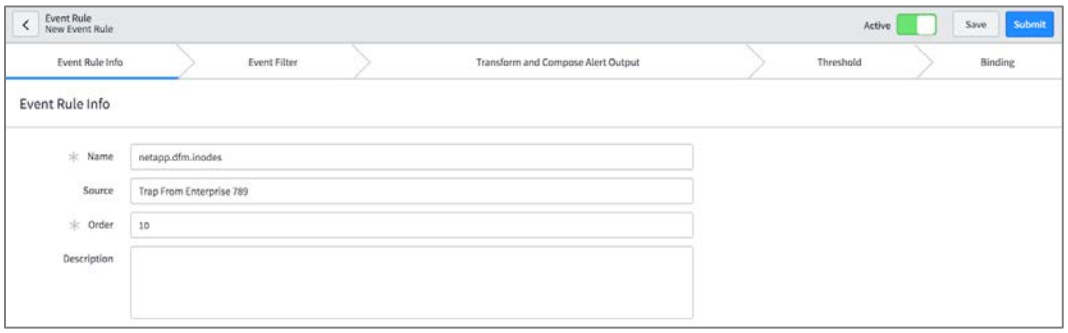
   lab2.4.bat

2. Navigate to **All Event > Open one of the SNMP trap 789 event**. Click the **New** button.



3. Fill in the following details in the **Event Rule Info** section.

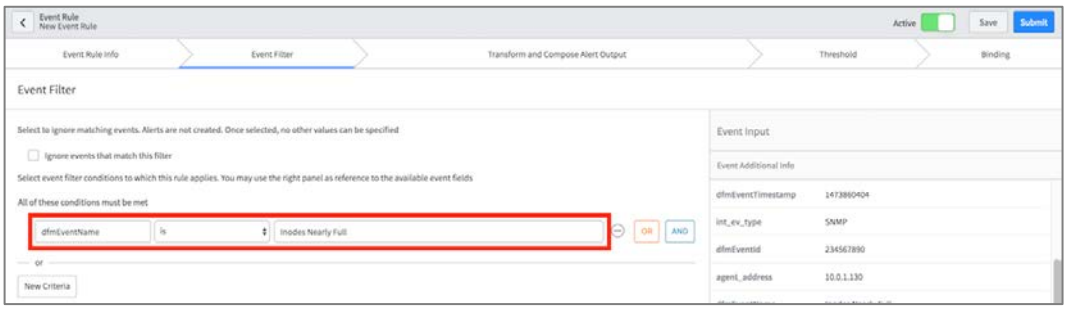
   Name: **netapp.dfm.inodes**
   Order: **10**



4. Click the **Event Filter** guide heading.

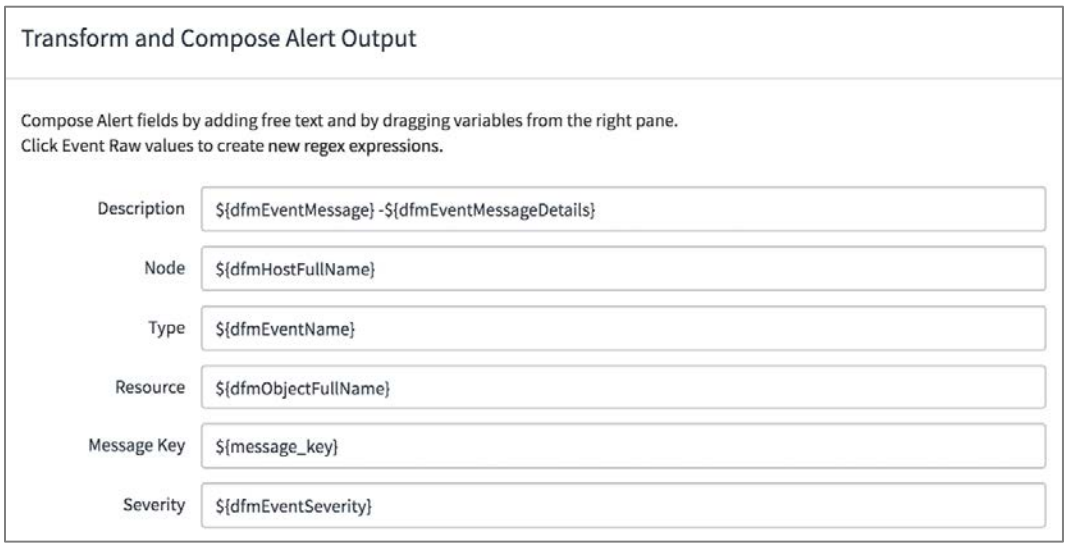
5. Under **All of these conditions must be met**, type **dfmEventName** in the first text box (it should appear in a list as you type). Set the condition to:

   dfmEventName → is → Inodes Nearly Full

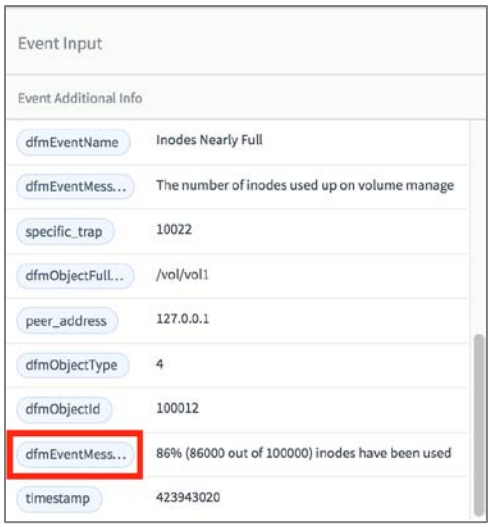


Click the **Transform and Compose Alert Output** guide heading and drag the following fields from the Event Input pane to populate the following Alert fields:

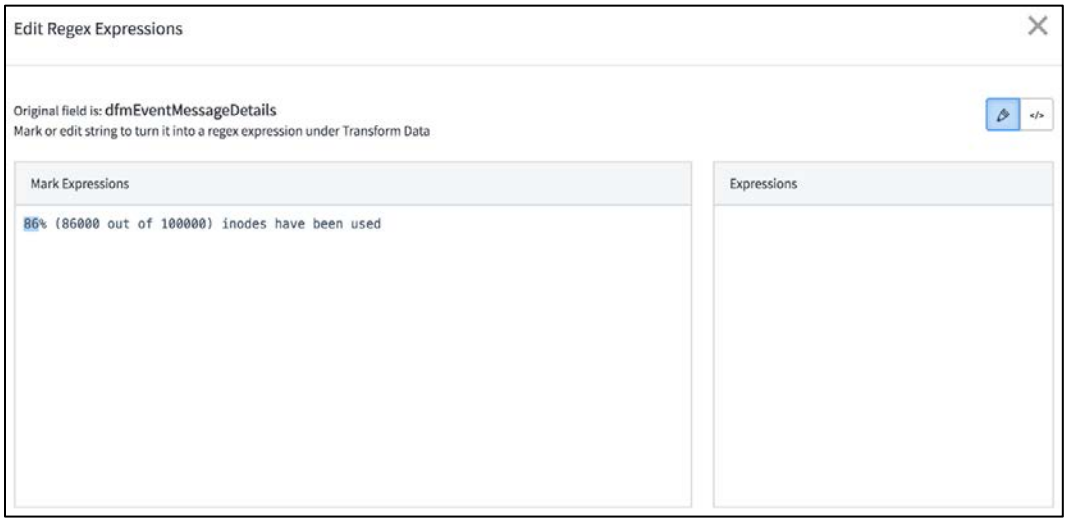
Description: **${dfmEventMessage} - ${dfmEventMessageDetails}**
Node: **${dfmHostFullName}**
Type: **${dfmEventName}**
Resource: **${dfmObjectFullName}**
Severity: **${dfmEventSeverity}**



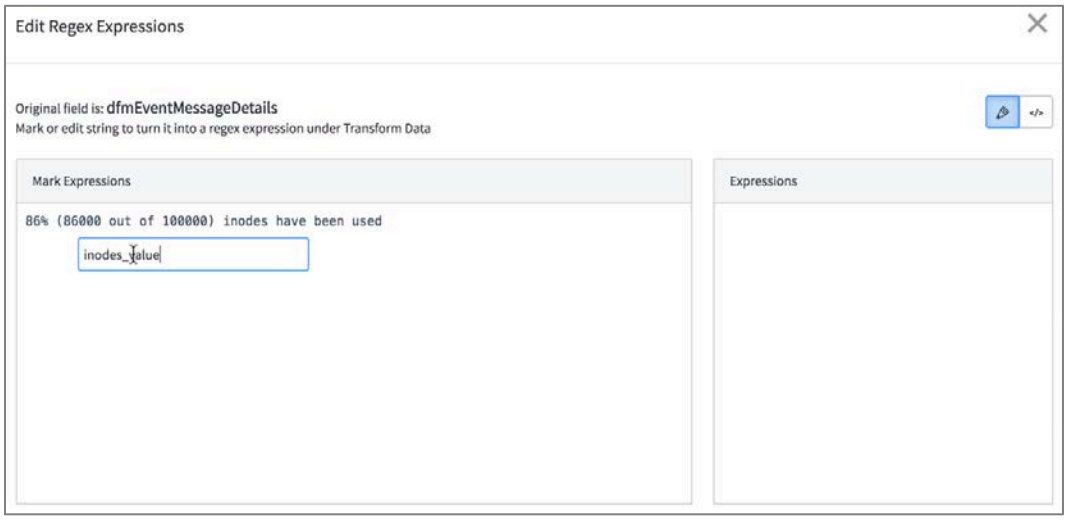
6. The percentage of inodes used is compared against a threshold. To do, this extract a value and store it in a variable for comparison with a set threshold. This SNMP trap includes a **dfmEventMessageDetails** field that you use to extract a value to use as a metric. In the **Event Input** pane, click the **dfmEventMessageDetails** field that shows the text **86% (86000 out of 100000) inodes have been used**.



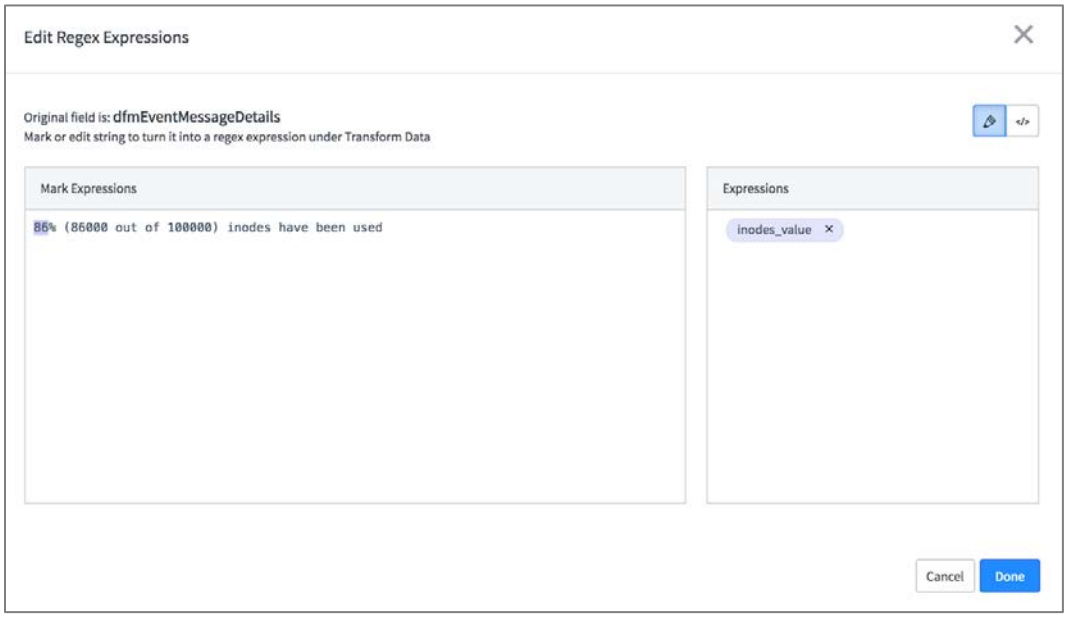
7. In the **Edit Regex Expression** dialog box, highlight the value **86.**



8. Click the highlighted text and enter **inodes_value** in the pop-up list. Press the **Enter** key.



9. An **inodes_value** field appears in the **Expressions** pane as shown below. Click the **Done** button to continue.



10. Click the **Threshold** guide heading and configure the threshold condition as follows:

    Create Alert Operator: **>**
    Field Name: **inodes_value**
    Threshold Value: **85**
    Occurs: **2**
    Over (seconds): **120**



11. Click **Save**.

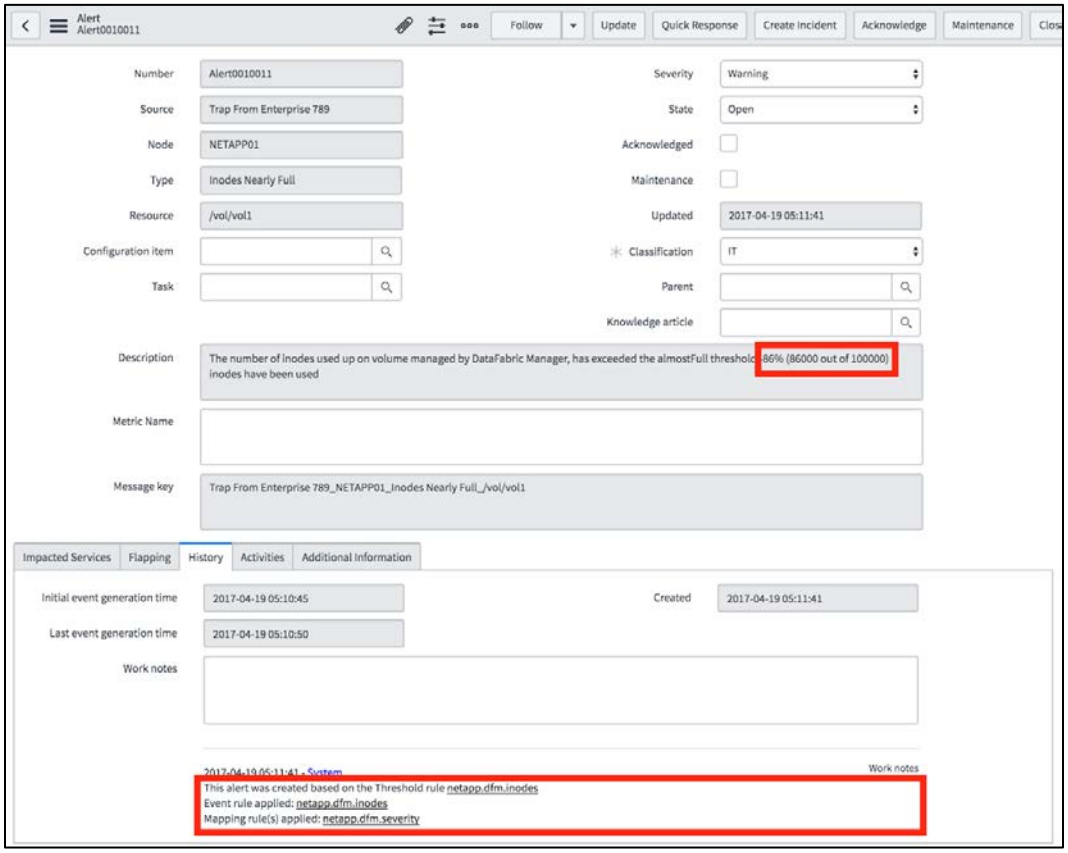
12. On the MID Server, open a command prompt and run the following command from the directory containing the lab files (e.g. C:\Lab).

    lab2.4.bat

Check the Result

1. Navigate to **Event Management > All Alerts** in your ServiceNow lab instance and open the new Alert.

2. Note the value shown in the **Description** field (86) which exceeds the threshold value of 85. Click the **History** tab to see the Event Rule that was applied.



Lab Goal

This lab explains how to use an Event Rule to bind (or associate) an alert with a CI in the CMDB, enabling the availability of business services to be monitored.

It also explains how an event rule can be configured to match information in an event with a CI in the CMDB.

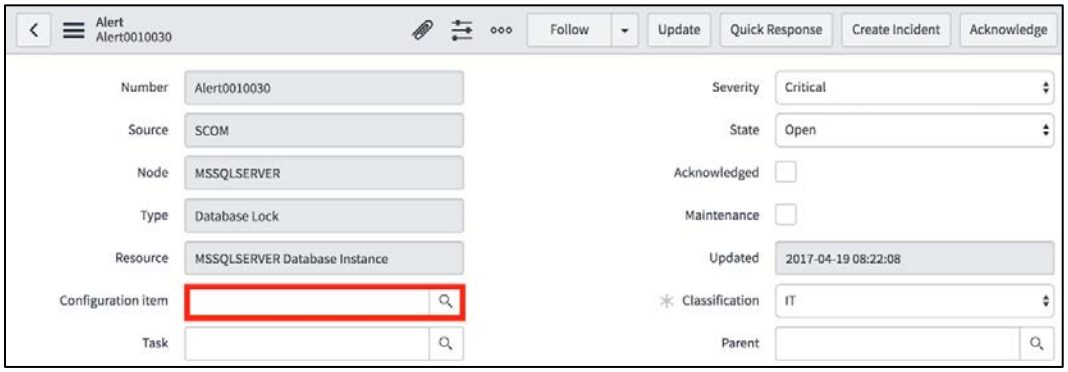
Lab 3
Binding Alerts to CIs

Step 1: Send an Event Not Bound to a CI

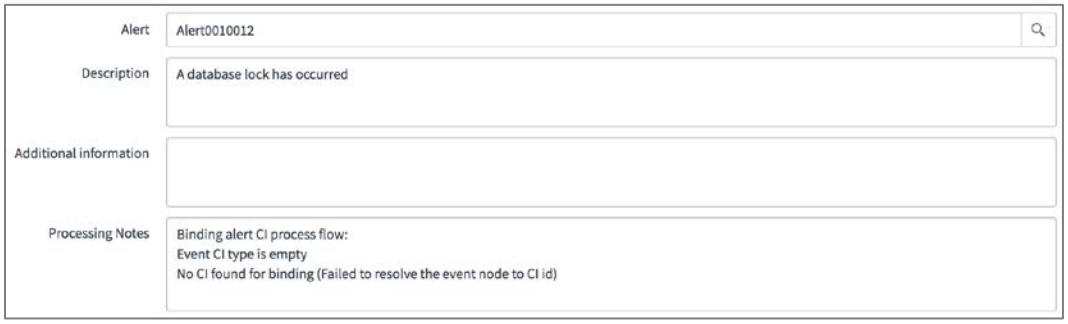
1. Switch to Postman and select **Lab 3.1 – Binding Alert to a CI** from the K17 Labs collection and click **Send**.

2. In your ServiceNow lab instance, navigate to **Event Management > All Alerts** and locate and open the Alert with **Source** set to **SCOM** and **Node** set to **MSSQLSERVER**.

3. Note that the **Configuration item** field is blank.



4. Scroll in the Alert to the Events related list. Click the **Event** in the list and scroll to the **Processing notes** field.



Notice that a CI could **not** be automatically found to bind the Alert with. Automated matching relies on the Event Node field containing a hostname, FQHN or IP address for a CI whose parent CMDB class is cmdb_ci_hardware.

In this scenario, the CI is a Microsoft SQL Server database instance whose CI class is cmdb_ci_db_mssql_instance.

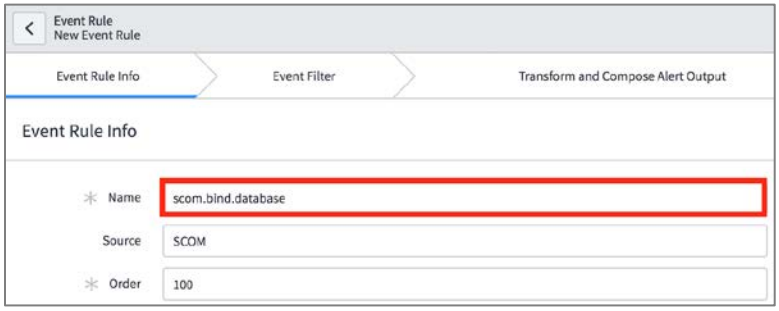
Fortunately, an Event Rule can be used to assist with binding Alerts to CIs.

Step 2: Configure an Event Rule to Bind an Alert to a CI

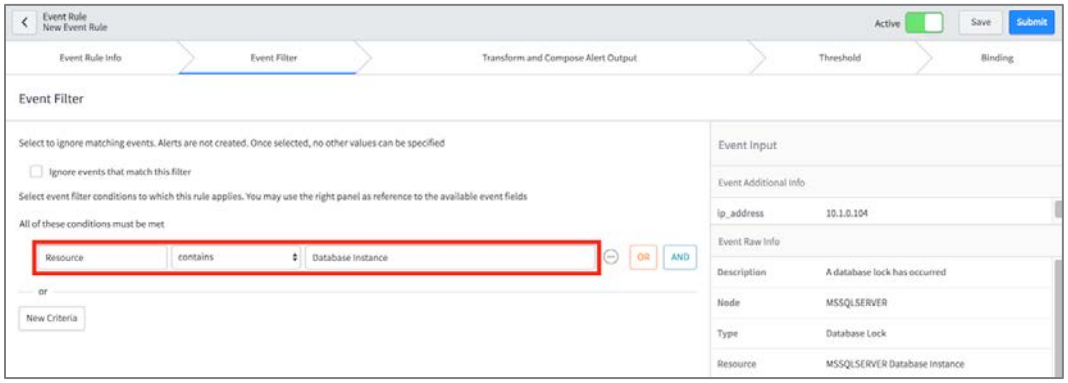
1. Navigate to **Event Management > All Events** and locate and open the event with Source = **SCOM** and Node = **mssqlserver**. Click the **Create Event Rule** button.

2. Enter the following information in the Event Rule:

   Name: **scom.bind.database**

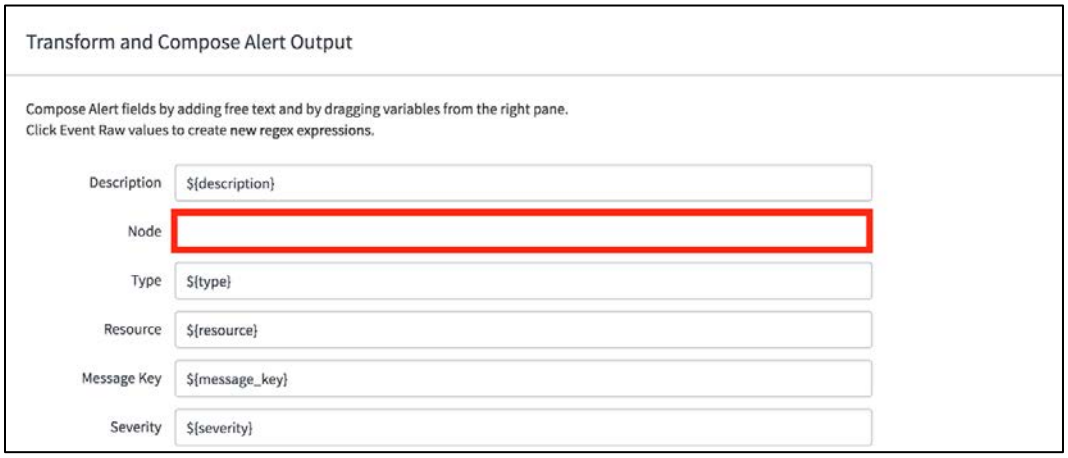


3. Click the **Event Filter** guide heading and configure a **Filter** condition where: **Resource→contains→Database Instance**



4. Click the **Transform and Compose Alert Output** guide heading.

5. Clear the **Node** field so the text box is blank.



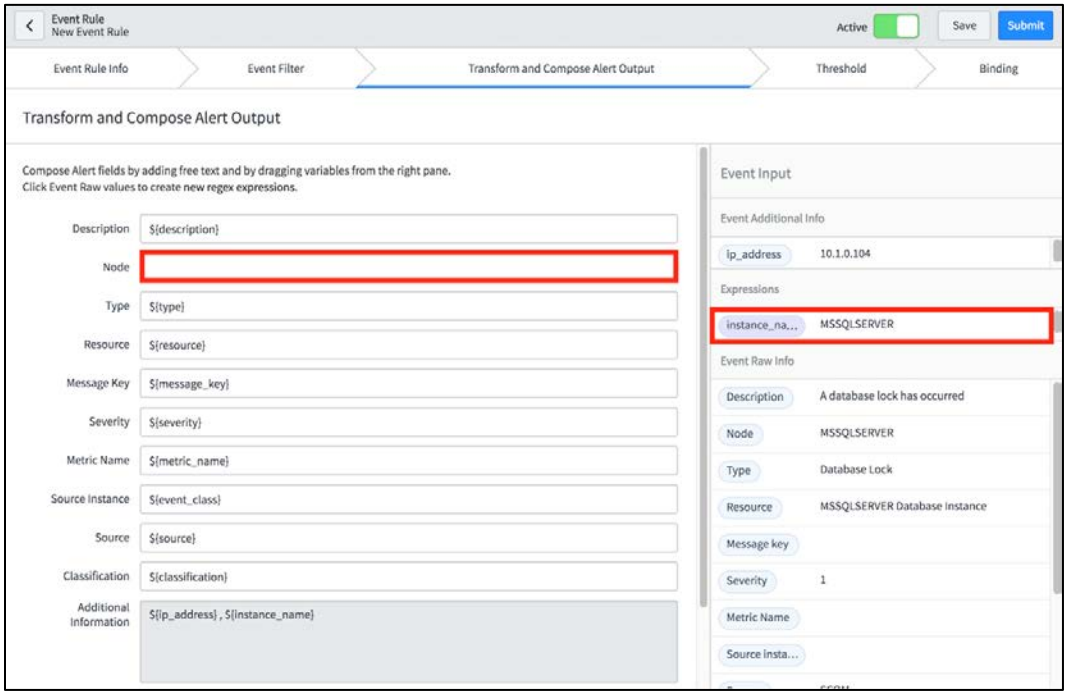
6. On the **Event Raw Info** section, find and click the **Resource** field to open the **Edit Regex Expressions** dialog box.

7. Highlight the word **MSSQLSERVER** and give it a field named **instance_name**.



8. Click **Done** to continue.

9. The **Transform and Compose Alert Output** section should look like the example.



10. Click the **Binding** guide heading and enable (i.e., tick) the **Override default binding field**.

11. Set **Binding Type** to **CI field matching** and set **CI Type** to **MSFT SQL Instance**.



Using these settings, fields contained within the **Additional Info** field are used to match any CIs whose CMDB CI class is cmdb_ci_db_mssql_instance. In this case, **Additional Info** contains the CI fields **instance_name** and **ip_address**.

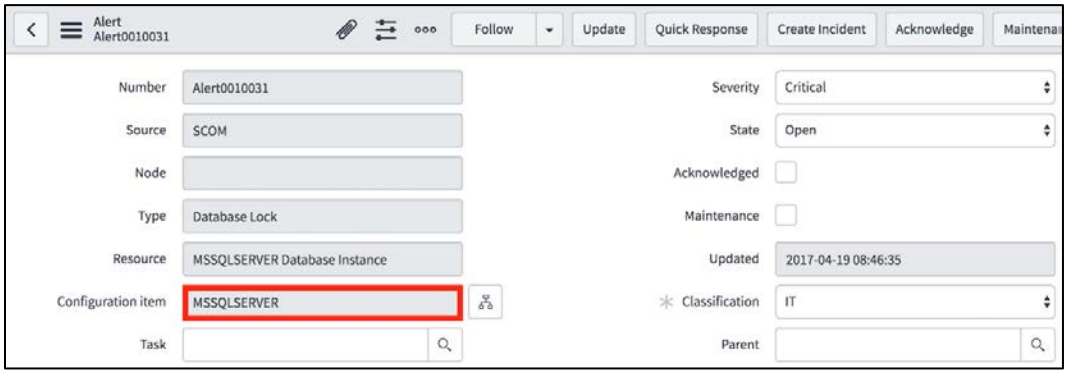
12. Click **Save**.

13. Switch to Postman, select **Lab 3.1 – Binding Alert to a CI** from the K17 Labs collection and click **Send**.

Check the Result

1. In your lab instance, navigate to **Event Management > All Alerts**. Locate and open the Alert with Source = **SCOM** and Node = **MSSQLSERVER**.

2. Note that the **Configuration item** field is now set to **MSSQLSERVER**.



Lab Goal

Lab 4
Alert
Automation

This lab explains how to process Alerts with Alert Rules and Correlation Rules. It also explains how Incident Templates can be used to pre-fill fields in incidents automatically created for Alerts meeting specified criteria.

The lab contains the following sections:

- Lab 4.1: Create an Incident Template
- Lab 4.2: Create Alert Action Rule
- Lab 4.3: Create Alert Correlation Rule
- Lab 4.4: Create Alert Action Rule for Correlated Alerts

Lab Goal

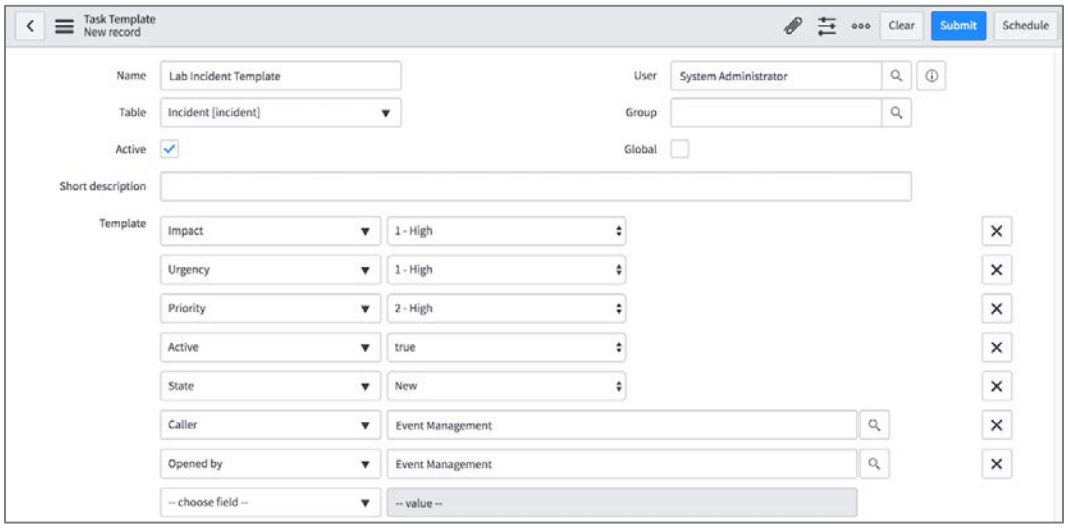
Incident templates can be used to automatically prefill fields in Incidents. This lab explains how to configure an incident template to be used in an Alert rule that you create in the next lab.

Lab 4.1 Create an Incident Template

1. Navigate to **Event Management > Settings > Task Templates** and click **New**.
2. Set the **Name** to **Lab Incident Template**.
3. Select **Incident [Incident]** for **Table**.
4. In the **Template** section, set the following fields and values:

Field	Value
Impact	1 – High
Urgency	1 – High
Priority	1 – Critical
Active	True
State	New
Caller	Event Management
Opened by	Event Management

5. The **Task Template** should look like the example.



6. Click **Submit** to save the Task template.

Lab Goal

Lab 4.2
Create Alert Action Rule

1. Navigate to **Event Management > Rules > Alert Action Rules** and click **New**.
2. Set the following details:

   Name: **Critical Severity Alert Incident**
   Active: **True**
   Order: **50**

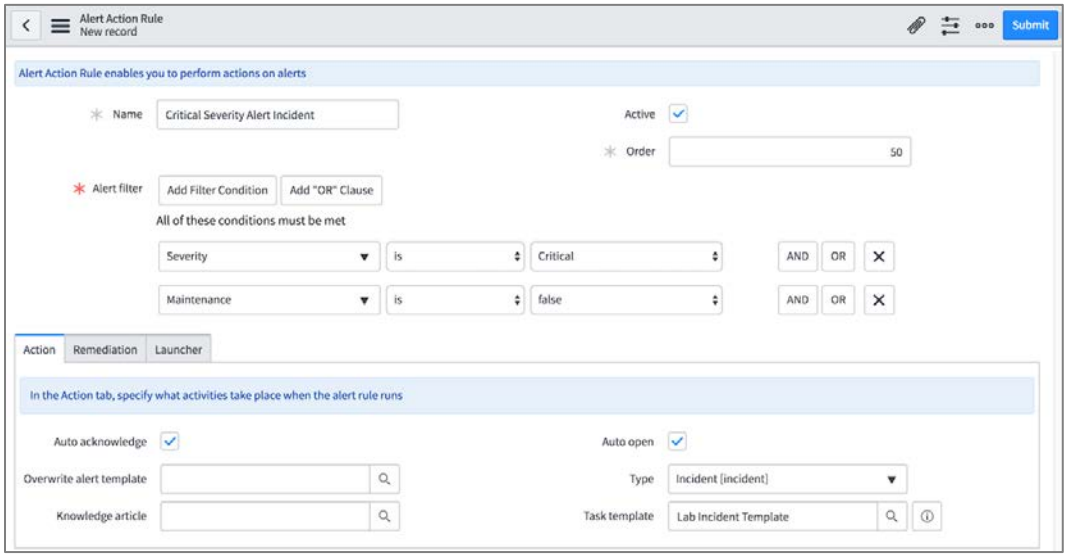
3. In the Alert filter section configure a condition as shown below:

   Severity → is → Critical
   Maintenance → is → false



4. In the Action tab section, check the **Auto acknowledge** and **Auto-open** check boxes.
5. Ensure **Incident [incident]** is selected in the Type drop-down box.

6. Select the **Lab Incident Template** from the Task template reference field (click the magnifying glass to open the selection dialog box). The Alert Action Rule should look like the example.



7. Click **Submit** to save the Template.

8. Switch to Postman, select **Lab 4.1.1 – Alert Rule – MSSQLSERVER** from the K17 Labs collection and click **Send**.

Check the Result

1. In your lab instance, navigate to **Event Management > All Alerts** and locate and open the Alert with **Critical** Severity for Node **MSSQLSERVER**.

2. Note that the **Task** field now displays an incident number.

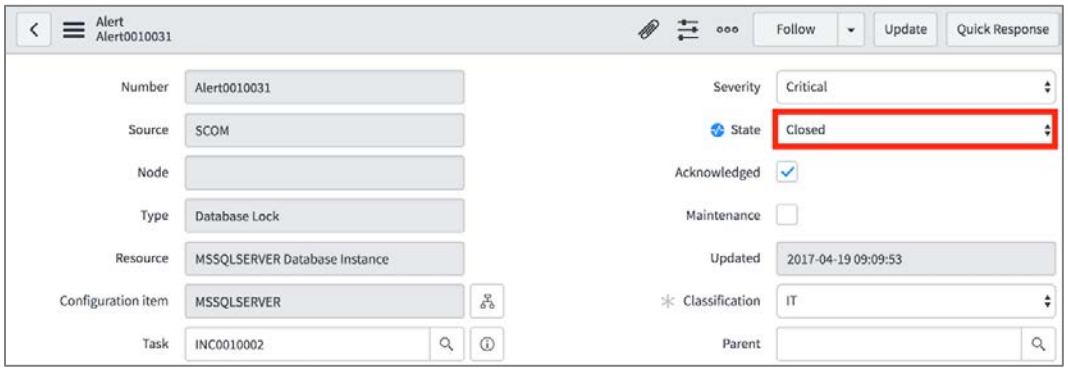
3. Hover the mouse pointer over the information icon next to the **Task** field to see a quick view of the Incident and its field settings.

4. Scroll down the Alert and click the **History** tabbed section. Note the audit trail capturing the application of Event *and* Alert Rules.

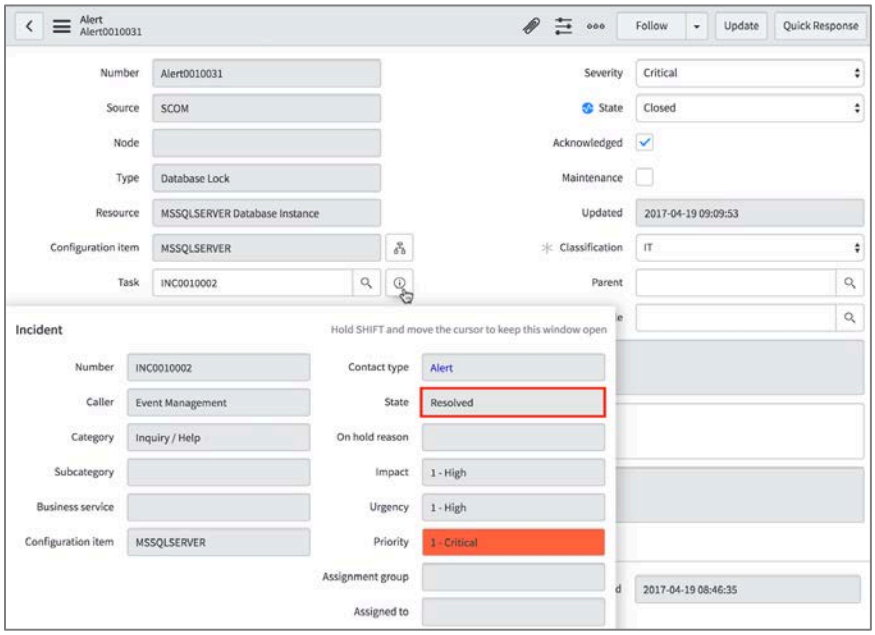
5. Now see what effect a clearing Event has on the Alert and Incident.

6. Switch to Postman again, select **Lab 4.1.2 - Alert Rule – MSSQLSERVER** (clear event) from the K17 Labs collection, and click **Send**.

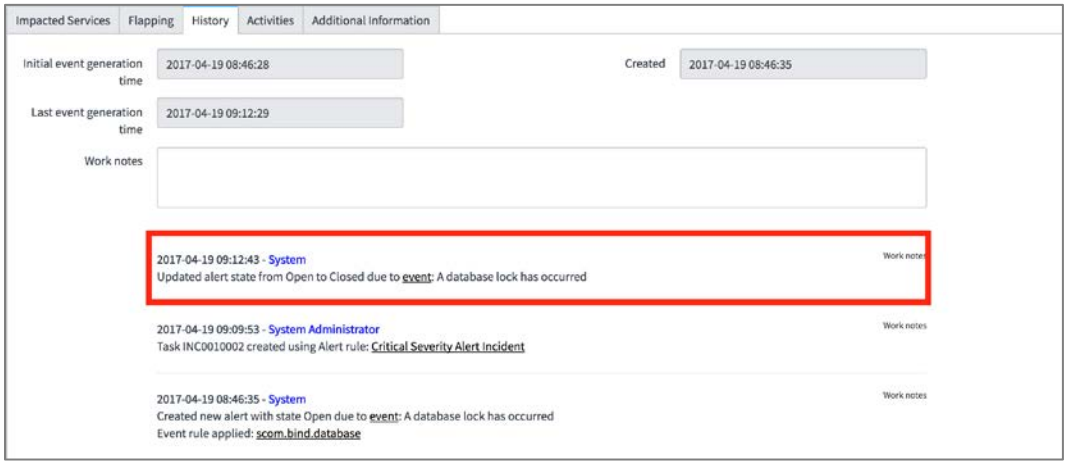
7. Switch to your ServiceNow instance and watch the Alert State field change to **Closed**.



8. Hover the mouse pointer over the **information** icon next to the **Task** field to see a quick view of the incident and its field settings. Note that the incident **State** field is now set to **Resolved**.



9. Scroll down the alert form and click the **History** tabbed section. Note the audit trail has captured the closure of the alert after receiving a clearing event.



Lab Goal

This lab explains how to create a Correlation Rule to correlate and classify a high CPU alert (root cause alert) and a database query slow alert (symptomatic alert) as Primary and Secondary alerts respectively. Classifying related alerts enables you to create an incident for only the Primary (i.e., root cause) alert.

Lab 4.3 Create a Correlation Rule

1. In your lab instance navigate to **Event Management > Rules > Alert Correlation Rules** and click **New**.

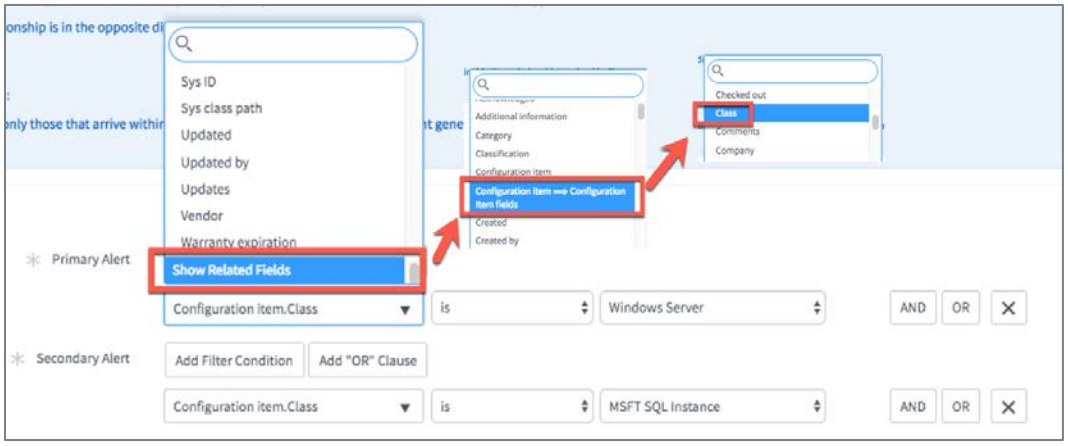
2. Enter the following details:

   Name: **cpu and db correlation**
   Order: **100**
   Active: **True**

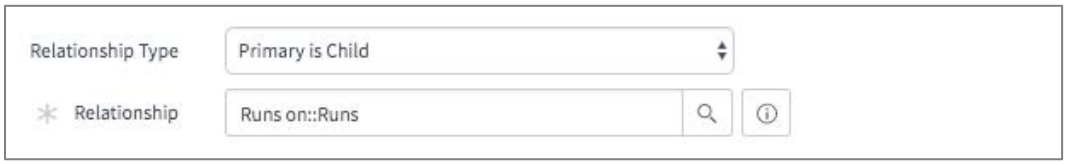
3. Configure the Primary Alert and Secondary Alert filter conditions as shown below:

   Primary: **Configuration item.Class → is → Windows Server**
   Secondary: **Configuration item.Class → is → MSFT SQL Instance**

   Note that you need to use the alert Related Fields to select **Configuration item.Class**. (**Show related fields > Configuration Item => Configuration Item fields > Class**)



4. Configure the **Relationship Type** and related details as shown in the example.

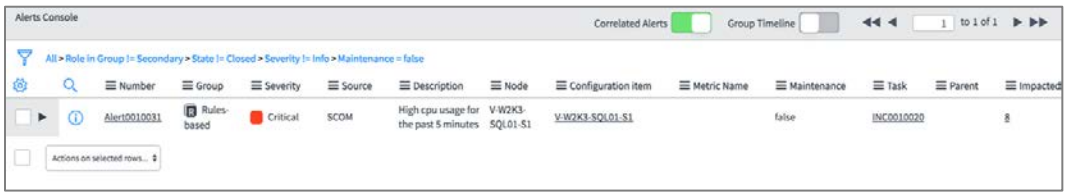


5. Click **Submit** to save the Alert Correlation Rule.

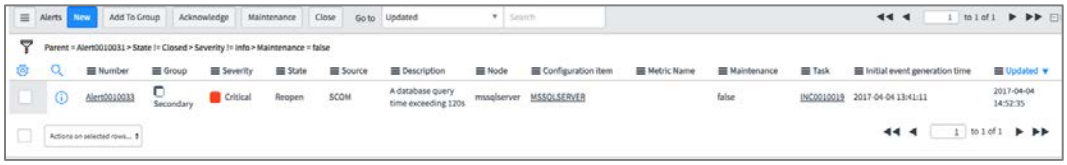
6. Switch to Postman, select **Lab 4.4 – Correlation Rule** from the K17 Labs collection and click **Send**.

Check the Result

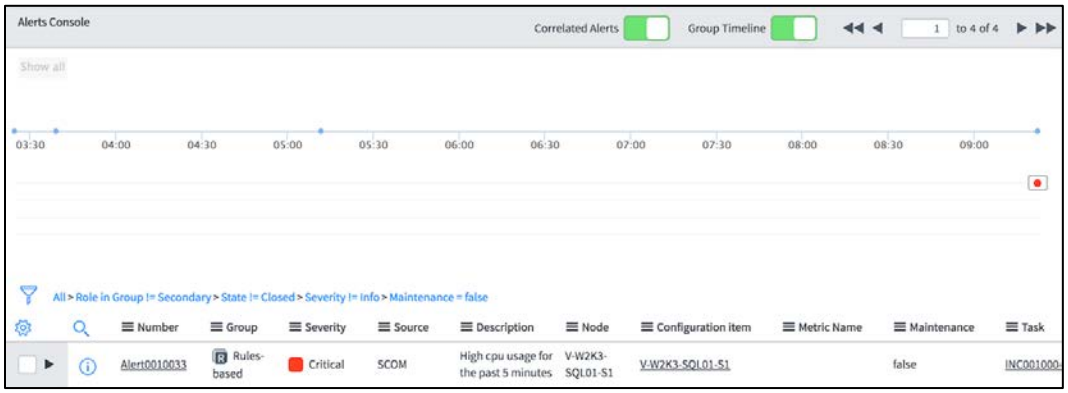
1. Navigate to **Event Management > Alert Console** in your ServiceNow lab instance and enable **Correlated Alerts** (slide switch to green).



2. Open the Alert, note the value of the Group field, and scroll down the page to see the list of correlated alerts (see group = secondary).

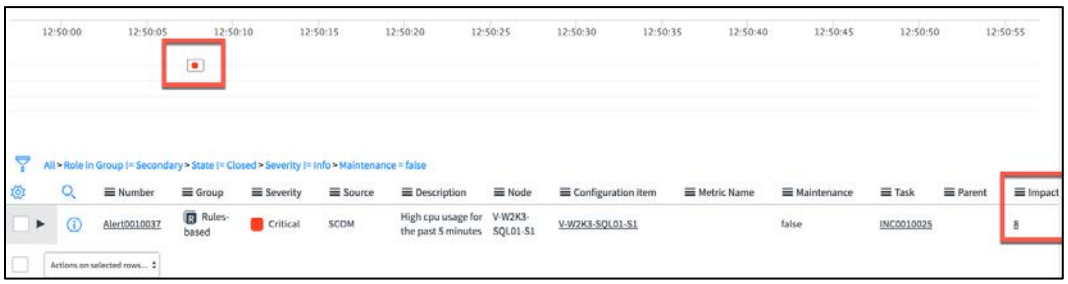


3. Exit the Alert and click the **Group Timeline** button in the Alert Console.

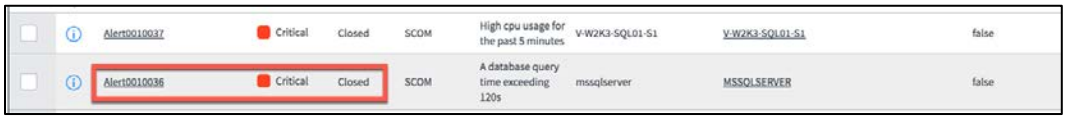


The alerts correlated by the Correlation Rule are displayed as a group at the right of the timeline.

4. Click the **Alert** group to drill down to the alerts. Click the **Impacted Services** button to display the business service impacted by the alert group.



5. Return to the **Alert Console**, open the Alert for **High CPU Load**. Click **Close** to close the Alert. Now go to the **All Alerts** list view and examine the secondary alerts.

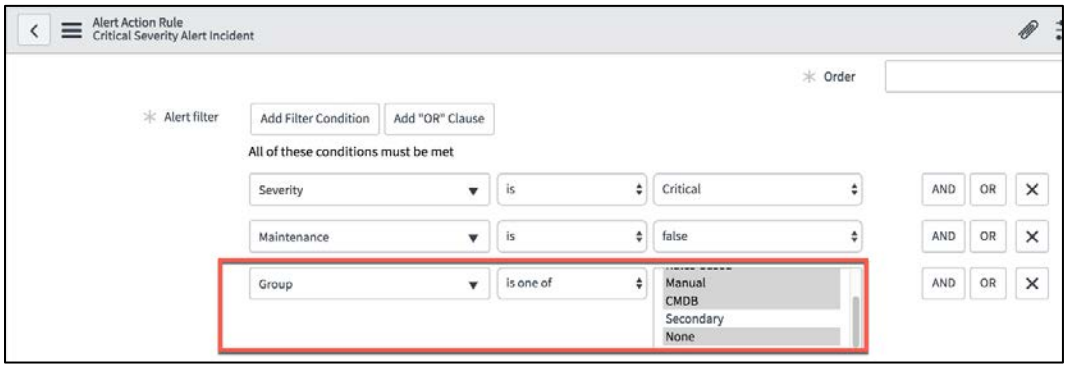


Lab Goal

Lab 4.4 Create Alert Rule for Correlated Alerts

Although the Correlation Rule correlated the alerts, Incidents have been created for all Alerts, rather than just the Primary Alert. Fortunately, the Group field can be used in an Alert Action Rule to ensure that Incidents are not created for secondary alerts.

1. Navigate to **Event Management > Rules > Alert Action Rules**.
2. Open the **Critical Severity Alert Incident** Alert Action Rule you created earlier.
3. Add **[Group] [is one of]** [select everything except **Secondary**] to the **Alert filter** and click **Update** to save the change. (e.g., select all and just deselect Secondary).



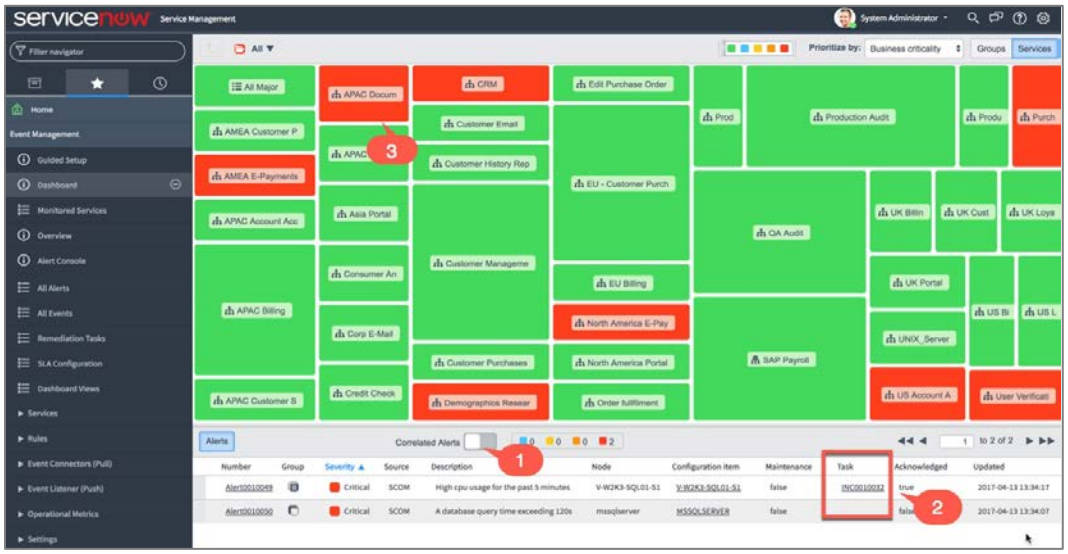
4. Go to the **All Alerts** list view and **close** and **delete** the primary and secondary correlated alerts.
5. Switch to Postman, select **Lab 4.4 – Correlation Rule** from the K17 Labs collection and click **Send**.

Check the Result

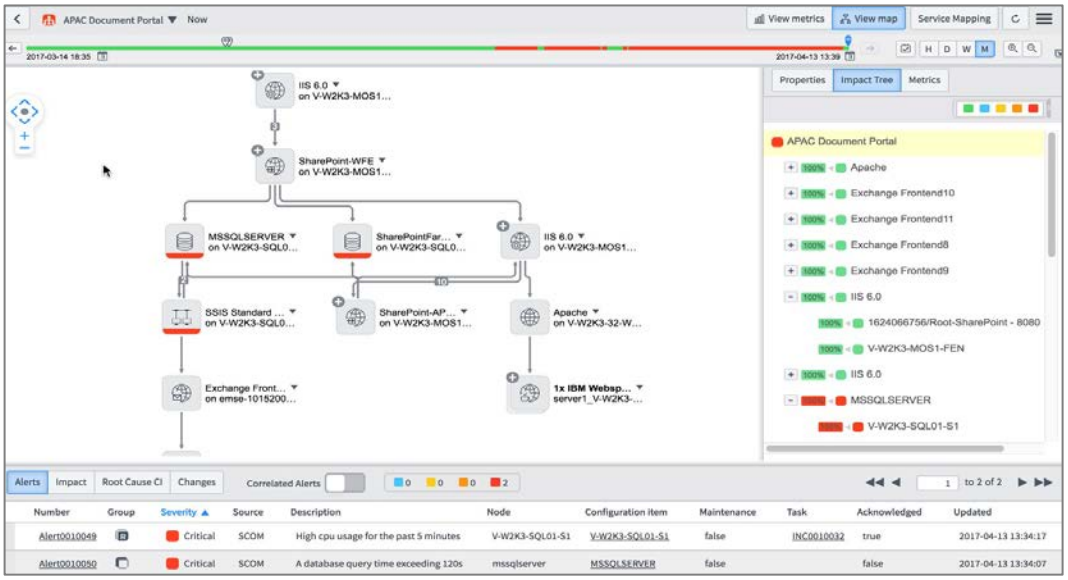
Navigate to **Event Management > Dashboard** in your ServiceNow lab instance. The dashboard shows all business services impacted by the alerts.

1. Click on the Correlated Alerts to turn off the display so it shows both alerts (none correlated view).

2. Verify that only the primary alert has an incident.



3. Double-click the APAC Document Portal business service tile to open the service health map view.



4. Note how V-W2K3-SQL01-S1 and MSSQLSERVER CIs are highlighted in the service health map.

5. Right-click the **Primary Alert** in the list below the service health map and select **Close**.



6. Wait a few moments and then navigate to **Event Management > Dashboard** to see the effect of closing the Alert.

Congratulations, you have successfully completed the lab exercises to demonstrate how events can be processed with Event Rules to produce meaningful alerts to help monitor the availability of business services, and responses can be automated using Alert Rules.